CELONCEL: Effective Design Technique for 3-D Monolithic Integration targeting High Performance Integrated Circuits

Shashikanth Bobba[||1], Ashutosh Chakraborty[||2], Olivier Thomas[||3], Perrine Batude[||3], Thomas Ernst[||3], Olivier Faynot[||3], David Z. Pan[||2], and Giovanni De Micheli[||1]

||1 Integrated Systems Laboratory (LSI), EPFL, Switzerland
||2 Department of Electrical & Computer Engineering, University of Texas Austin, USA
||3 CEA-LETI/MINATEC, 17 rue de Martyrs, 38000 Grenoble, France

Abstract

3-D monolithic integration (3DMI), also termed as sequential integration, is a potential technology for future gigascale circuits. Since the device layers are processed in sequential order, the size of the vertical contacts is similar to traditional contacts unlike in the case of parallel 3-D integration with through silicon vias (TSVs). Given the advantage of such small contacts, 3DMI enables manufacturing multiple active layers very close to each other. In this work we propose two different strategies of stacking standard cells in 3-D without breaking the regularity of the conventional design flow: a) Vertical stacking of diffusion areas (Intra-Cell stacking) that supports complete reuse of 2-D physical design tools and b) vertical stacking of cells over others (Cell-on-Cell stacking). A placement tool (CELONCEL-placer) targeting the Cell-on-Cell placement problem is proposed to allow high quality 3-D layout generation. Our experiments demonstrate the effectiveness of CELONCEL technique, fetching us an area gain of 37.5%, 15.51% reduction in wirelength, and 13.49% improvement in overall delay, compared with a 2-D case when benchmarked across an interconnect dominated low-density-parity-check (LDPC) decoder at 45nm technology node.

Keywords

3-D monolithic Integration, Standard cell, Placement, Partitioning, Optimization.

1. Introduction

3-D integration provides an effective platform for realizing future gigascale circuits by integrating multiple layers of active devices on a single 3-D chip [Banerjee 01, Pavlidis 09]. 3-D fabrication technologies can be broadly classified into two groups according to preferred integration scheme: a) 3-D parallel integration (or TSV based technology) in which each active layer, along with their respective interconnect metal layers, is fabricated separately and is subsequently stacked via TSVs [Koester 08, Sillon 08], and b) 3-D monolithic integration (3DMI), in which the stacked transistors are processed sequentially on the same wafer. Developing TSV manufacturing technologies is expensive in terms of cost, yield and area. For example, the TSV pitches are usually around (5 μm -10 μm) [Mit 08] compared to 100 nm contact dimensions offered by 3DMI technology [Batude 09a].
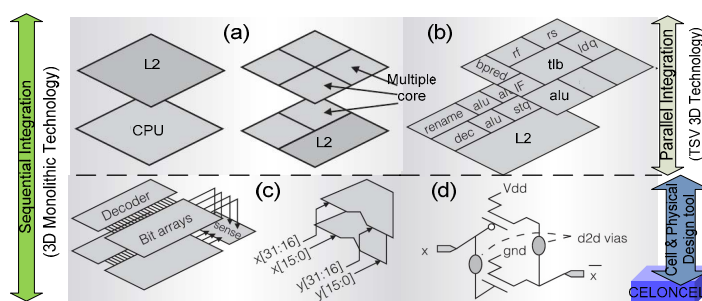

Figure 1. 3-D design approaches, depending on the granularity of the units to be stacked and the process technology [Loh07] (a) Entire core, (b) Functional unit blocks, (c) Logic gates and (d) Transistors.

The performance of ICs in advanced technology nodes is dominated by the interconnect delays [Havemann 2001]. In 3-D integration, the benefits in terms of wire-length, latency and power depend on the granularity level at which the circuit is partitioned [Loh07]. Figure 1 illustrates the circuit partitioning at various granularities. At a coarse-grain level we can have cache on top of cores, or cores on top of cores, as presented in Figure 1a. At finer level of granularity we can realize functional blocks on top of each other (Figure 1b). Going at even finer level, we can perform 3-D stacking at the gate and standard cell level, as illustrated in Figure 1(c,d). In the case of TSV technology, due to low precision of the alignment capability of the equipment and the relatively large size of TSVs, circuit integration at transistor/gate level cannot be done. Consequently, 3DMI is an ideal choice for ultra-high density 3-D circuits. In this work we focus on new design techniques for realizing fine-stacking at the transistor/gate level. At this level, redesign effort is high and hence we need new CAD tools to bridge the time gap for designers [Das 03].

In recent years, there has been extensive work done in developing new physical design tools for 3-D IC design. However, all these tools are mainly linked to 3-D TSV technology [Cong 10, Zhou 06], as their main objective is to minimize the number of TSVs while reducing the average

wirelength of the routed circuit. 3-D monolithic integration has seen substantially less research effort at CAD level. With this work we take the first step towards providing a complete design flow for 3-D monolithic technology. `CELONCEL` tool, comprising of `CELONCEL-placer` and `CELONCEL-lib`, can be integrated into the traditional 2-D design flow, for designing 3DMI circuits.

2-D placement has been extensively studied over decades, resulting in very efficient placement engines [Chan 05, Jiang 06]. Many 3-D placement tools [Das 03, Deng 01, Cong 09] employ the existing 2-D placement engine at each level after partitioning the 3-D circuit into many 2-D regions. In this work we present a `CELONCEL-placer`, which acts as a wrapper around any commercial physical design engine, for placing standard cells in 3D. Taking into account the fine granularity from 3DMI technology we envisage stacking of standard cells in 3D. We present a novel library, `CELONCEL-lib`, which gives the flexibility of placing cells on top of each other without any pin conflicts. Since we are dealing at a very fine granularity, we keep the bound on active layers to be two, though technology permits more silicon layers. To the best of our knowledge we are the first to address the cell and physical design issues at a fine granularity for standard cell based ASIC design for 3DMI technology.

To summarize, the main contribution of this paper are:
1. We present the first detailed study on the cell and physical design techniques for 3-D monolithic integration. We explore solutions for stacking standard cells in two active layers while keeping the regularity of standard cell approach.
2. We present a new 3-D placement tool, `CELONCEL-placer`, which places cells in two active layers for improved area, wirelength and delay.
3. The ability of `CELONCEL` tool kit (comprising a standard cell library `CELONCEL-lib` and `CELONCEL-Place`) is demonstrated through the 3-D physical design on a set of open-source benchmarks [Opencores]. We also show how `CELONCEL` fits into conventional 2-D tool-chain while building the cores in 3D.

The remainder of this paper is organized as follows. In Section 2, we give a quick insight on the technology background. In Section 3, we explain the standard cell optimization specific to 3-D monolithic integration. Our design flow is presented in section 4. Section 5 deals with the description of our new 3-D placement tool. The simulation framework and the related experiment results are presented in Section 6 to demonstrate the effectiveness of our proposed algorithm. Finally, Section 7 concludes the paper shedding some light on future work.

2. 3-D Monolithic Integration

In 3-D monolithic integration, top transistor layers are processed sequentially on the lower transistor layers. As the alignment of top transistor-lithography-levels occurs after bonding of new top active layer, its precision is only linked to the performance of the stepper (for example, $3\sigma=10\text{nm}$ for 45 nm node equipment [ITRS]). Currently, 3-D contact

dimensions of ~100 nm have been demonstrated [Jung 07]. On the other hand, in 3-D parallel integration (or 3D-TSV integration), two wafers are stacked after they are individually processed, thereby demanding high alignment precision ($3\sigma=1\mu\text{m}$ [Mit 08]).

Achieving small 3-D contact pitch in monolithic integration is definitely an asset. However, it faces the challenge of making high quality top FET at low temperature in order to preserve bottom FET and metal interconnections from any degradation. To achieve reasonable performance for top transistor, a minimum value of 600°C-650°C for overall thermal budget is needed. Recently, Batude et al. have demonstrated the top and bottom transistors with similar characteristics [Batude 09a]. Advancements in this technology have already been demonstrated both for memory [Jung 07] and logic applications [Batude 09b].
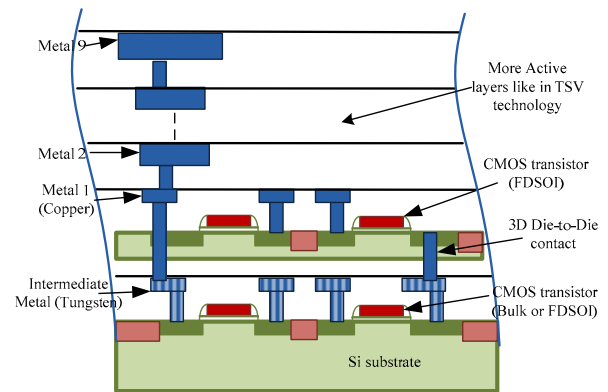

Figure 2. Cross-section view of the 3-D monolithic die with 2 active layers separated by one Intermediate metal layer.

Figure 2 illustrates the cross-section view of the 3-D monolithic stack with two active layers separated by an intermediate metal layer. Copper cannot be used for the intermediate layer as the top transistors demand a very high thermal budget. We consider tungsten [Jung 07] instead, as it can withstand high temperatures without degrading. In our work we assume only one intermediate metal layer, as we employ it for realizing intra-cell routing of a standard cell realized in the bottom active layer.

3. 3-D Monolithic Library Design

In this section, we discuss two methods of modifying a traditional 2-D standard cell library to enable fine grained 3-D integration. Using the first methods, the layout of the cell is folded in multiple active layers. The second method modifies the layout of the standard cells in a way that they can be stacked on top of each layer. These methods are respectively referred to as intra-cell stacking and cell-on-cell stacking.

3.1. Intra-Cell Stacking Transformation

Standard cells implement a pre-defined logic function (for example, AND gates, OR gates and flip-flops) and have fixed

height but varying widths. The structure of a typical standard cell laid in 2-D is shown in Figure 3a. 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 diffusion gap region, where we place the input pins. Since 3DMI technology offers multiple active layers adjacent to each other, the layout of the standard cell can be folded in multiple layers [Batude 09a]. For instance, as illustrated in Figure 3b, p-type devices (forming the PUN) are realized on the top active layer and n-type devices (forming the PDN) at the bottom active layer. Since the PUN is typically larger than the PDN, the active region height for a 3-D cell (H_{ACT3D}) is limited by the height of the P-diffusion (H_{Pdiff}). The active region height of a 3-D library is given by the following equation, when mapped directly from a 2-D library:

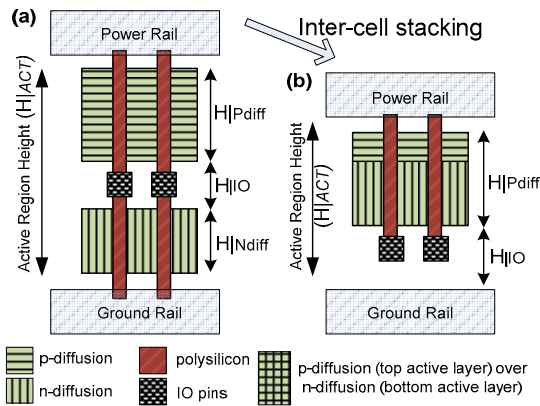$$H_{ACT3-D} = H_{ACT2-D} - H_{Ndiff} + H_{IO} \qquad (1)$$

Figure 3. (a) Typical cell in 2-D (planar) 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.

In the above transformation, it can be observed that the reduction in the height of a 3-D cell is due to the N-diffusion region. Moreover, there can be a slight increase in the space needed for *input-output* (I/O) pins in the 3-D layout, as the design rules should be obeyed, considering the close proximity of wide power rails.

3.2. Cell-on-Cell Stacking Transformation

To allow 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 allowed to be planar (i.e. in one active layer) but such cells can be placed on top of each other. One of the main challenges for this approach is to get the *input-output* (I/O) pins of the bottom cell to the top metal layers (for instance metal 2) without any short-circuit with I/O pins of the standard cell on top active layer. Though in many cases it may be possible to shift the cell in the top active layer laterally to

access pins of the cell lying at 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 4 shows example of *Cell-on-Cell* stacking of two standard cells on top of each other.
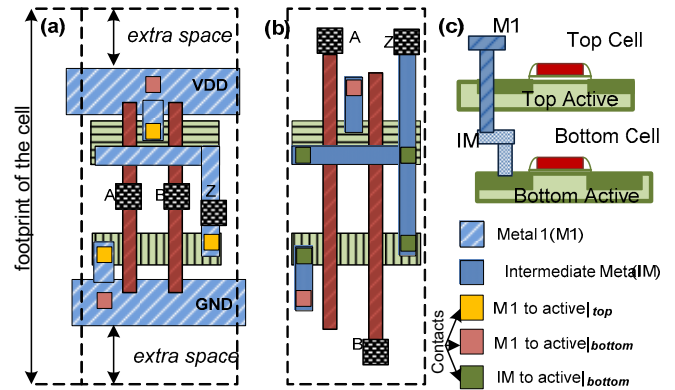

Figure 4. CELONCEL NAND2 layout (a) Cell realized in the top active layer (b) Corresponding cell in the bottom active layer.

Figure 4a and 4b shows 2-input NAND gate, realized in the top active layer and 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 both the active layers. Tungsten is used for ICR of the bottom cell, whereas copper is used for the top cell. Note that the resistance of tungsten is roughly 3 times higher than copper. Nevertheless we did not observe any delay degradation of the bottom cell when compared to the similar cell realized in the top layer. In order to get the I/O pins of the bottom cell to the top metal layer, we need to allocate some space at 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 the figure below). Whereas the I/O pins of the bottom cell are placed beyond the rails. Hence the cell height (or footprint) has to take into account the space for the I/O pins coming from the bottom cell and also the respective design rule for avoiding short circuits with the I/O pins of the neighboring cell.

3.3. Quantifying Planar-to-3-D Library Mapping

Until now we have explained the two cell transformation methods. In this section, we focus on the implementation details of these transformations. Figure 5 shows the two approaches to realize a 3-D standard cell library from a 2-D library. Corresponding arrows in green, red and brown quantifies the normalized standard cell height in all cases. Table 1 compares the standard cell height of existing 2-D standard cell libraries before and after the cell transformation. We have benchmarked across three important cell libraries at 45 nm and 65 nm technology node.
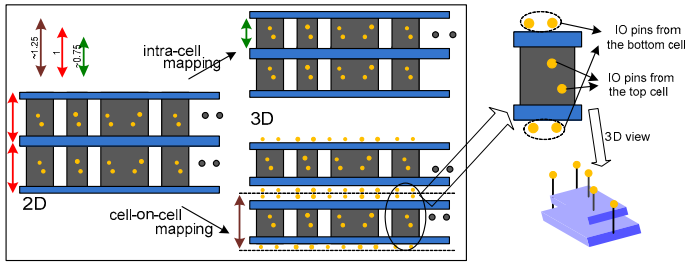
Figure 5. 2-D to 3-D Cell Transformation

Few key observations from planar to 3-D cell transformation:

1. By intra-cell stacking, all the cells are spread across two active layers (3-D cell) fetching 29% gain in the standard cell height. One of the primary advantages of this transformation is the ease in integration with the conventional design flow, as the design effort consists of developing only the 3-D library. A Cadence Virtuoso snapshot of a 3-D flipflop at 65 nm technology node is depicted in Figure 6. Hence to realize circuits with these new 3-D cells, the existing physical design tools can be used without any change.
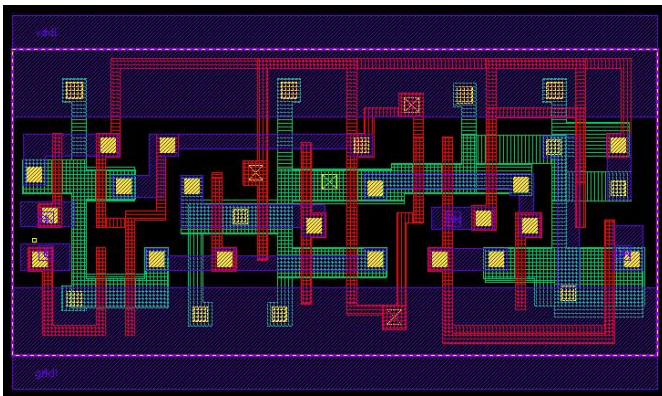

Figure 6. Virtuoso snap-shot of a D-flipflop built in 3-D at 65 nm technology node

2. On the other hand, cell-on-cell stacking leads to 25% increase in the cell height. However, in this case all the cells are planar, i.e. they occupy only one active layer and therefore one cell can be placed on top of the other. The design effort for cell-on-cell stacking is higher as we need twice the number of cells, for the top and bottom layers. Moreover, a new physical design tool is needed to place the cells spread across multiple layers.

Table 1. Normalized height of existing standard cell libraries before and after cell transformation

Cell Height	45nm Nangate Library	45nm commercial library	65nm commercial library
Planar (2D)	100 %	100 %	100 %
Intra-cell (3D)	71.43 %	71.61 %	69.05 %
Cell-on-cell	125.71 %	125.93 %	125.00 %

4. Physical Design Flow for Cell-on-Cell Stacked Layouts

In this section, we will describe the physical design technique used for 3-D layout generation using Cell-on-Cell stacking. The design flow of our CELONCEL tool is presented in Figure 7. CELONCEL-lib is the new standard library with cells designed by cell-on-cell stacking (section 3.2). CELONCEL-placer has 4 main steps in the flow. Each of these steps is described in the following section. In the first step, we transform the standard cell library using procedure DEFLATE. At this stage any commercial placement tool can be used to generate a virtual seed placement without any overlap among the transformed cells. The seed placement result then undergoes the step INFLATE. 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 perturbation legalization is done to remove rest of the overlaps in the step LEGALIZE thus solving the placement.
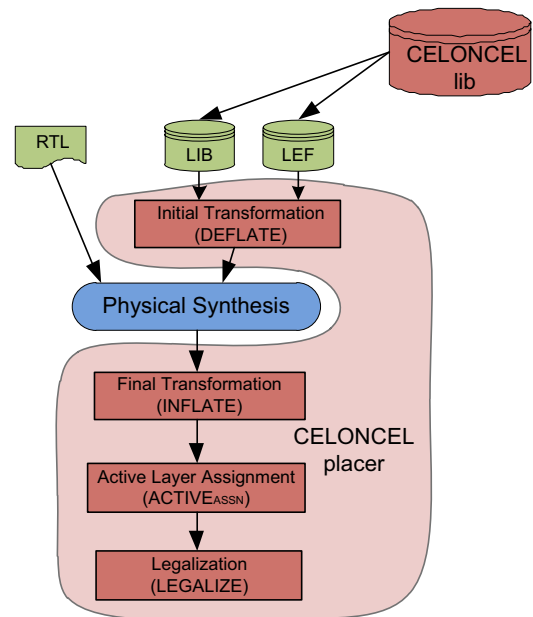

Figure 7. Our Design Flow

5. Physical Design Tool: CELONCEL-Placer

The key observations we take forward for developing CELONCEL placement tool is that with the Cell-on-Cell stacking, the footprint and delay of a cell is independent of the choice of active layer it is manufactured on. Based on this, we conjecture that during physical synthesis the choice of active layer of 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 can be made by a dedicated step. One of the critical benefit of isolating layer assignment and placement is that other 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 transform, generates a virtual cell library from a given real cell library such that cell dimension and pin location are modified. Consider a cell whose layout is 2-D in nature: To stack such cell on top of the other, the placer should effectively consider the area contribution of each cell to be half of its actual value. Thus, we shrink the width of each cell by half. Note that, to maintain placement sanity, we also need to scale down the x-coordinates of the pin geometry defined for such a cell. Previous works such as [Yang 03, Chakraborty 09] have used the concept of cell expansion/deflation for congestion alleviation and transforming placement with blockages to contiguous placement respectively. Figure 8 shows an example of a 2-D cell undergoing initial transformation. At this stage, we can run any 2-D placement engine to generate legalized placement consisting of transformed cells.
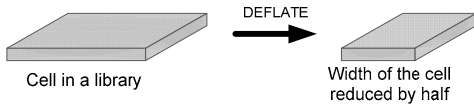

Figure 8. DEFLATE transformation applied to all the library cells.

5.2. Final Transformation (INFLATE):

The INFLATE transform takes the placement information from the solution of commercial placer on the virtual library and applies an inverse transform such that the width of the cells are expanded back to their original size. While doing this expansion, 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 of circuit row. INFLATE also snaps such cells back to be inside the placement area. Once all the 2-D cell's width is doubled, the placement now has huge overlap among different cells because all the cells are now placed in only one active layer oblivious of the availability of another active layer. Figure 9 shows an example of two neighboring cells undergoing INFLATE transform.
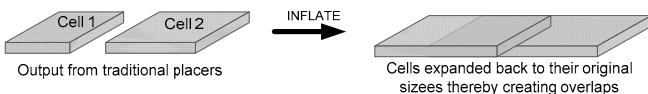

Figure 9. INFLATE applied to two neighboring cells after placement.

5.3. Active Layer Assignment (ACTIVE$_{ASSN}$):

This step assigns the active layer of each cell with the objective of minimizing the overlap with the neighboring. 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. Lower remaining overlap directly means less movement during legalization thus reduces moving cells away from their optimal location determined by the placer. Let us denote the set of cells lying in a circuit row by *CLS*. Further, let *OV(a, b)* denote the 2-D overlap between two cells *a* and *b* in the row. For each cell *a*, let *Xa* be the binary variable whose value determines the active layer in which the cell *a* will reside in the 3-D layout, and *Wa* be the width of the cell *a*. With this terminology, the ZOLP can be formulated as:

$$\text{Minimize} : \sum OV(c_1, c_2) \, (X_{c_1} \text{ XNOR } X_{c_2}) \; \forall \; c_1, c_2 \in C$$

$$\text{Subject To:} \sum X_c \times W_c \leq \text{Width}$$

$$\sum (1 - X_c) \times W_c \leq \text{Width}$$

$$X_c \text{ is binary } \forall \, c \in C$$

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 the same active layer, the corresponding overlap value adds to the cost function. The pair of constraints is to bind the width of cells in a row at each layer, equal to be less than width of the row. Note that XNOR implies multiplication of two variables thus it may seem that 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. $X_A X_B$) appears in the cost function expression. Introduce a new binary variable X_{AB} such that:

$$X_A + X_B \leq 1 + X_{AB}$$
$$(1 - X_A) + (1 - X_B) \leq 2 - 2 \times X_{AB}$$

By replacing $X_A X_B$ by X_{AB}, and adding the above constraints to the ILP, the new problem is equivalent but without any multiplication of binary variables.

ZOLP Speed Up:

The number of binary variables in the ZOLP above 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 components as follows: We scan the layout of a row from left to right. Any time a whitespace is encountered; this means the ZOLP problem of the cells on left of the whitespace is independent of the ZOLP problem of the cells on the right. This is because during the active layer assignment cells cannot move in the 2-D plane thus the cells on both sides of a whitespace cannot generate new overlaps between them and can be treated independently.

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 total displacement of all the 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 CLS0 and active layer 1 as CLS1. Further, let the original and post-legalization 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:

Minimize : $\sum |X(a) - XO(a)|$ $\forall$ a in {CLS0} $\cup$ {CLS1}

Subject To:

$X(a) + W(a) \leq X(RT(a))$ $\forall$ a in {CLS0}

$X(L0) \geq START$

$X(R0) + W(R0) \leq END$

$X(a) + W(a) \leq X(RT(a))$ $\forall$ a in {CLS1}

$X(L1) \geq START$

$X(R1) + W(R1) \leq END$

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

$$X(a) - XO(a) \leq MOVE_a$$
$$X(a) - XO(a) \geq -MOVE_a$$

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

6. Experimental Setup and Results

The core components shown in Figure 7 were implemented or integrated using C++ language. We used open source MILP solver Gurobi [Gurobi] as our ZOLP and LP solver engine. Synopsys Design Compiler (A-2007.12-SP4) [DC] was used for mapping the RTL of the benchmarks onto target standard cell library. Cadence SOC Encounter (v8.1) [SOCE] was used as the physical synthesis engine to generate the virtual seed placement in timing driven mode. Timing analysis was performed with Synopsys PrimeTime (D-2009.12-SP2) using cap table of the standard cell library. In this study we have mapped the opensource 45nm Nangate [Nan] (v1.3) library to different 3-D libraries by changing only the physical attributes of the cell. `INTRACEL-lib` has cells, built in 3-D by intra-cell stacking transformation (sec. 3.1), with 30% less height, whereas `CELONCEL-lib` (sec. 3.2) has cells which span 25% more in height.

Table II shows the benchmark circuits used to quantize the benefit of using our placement engine. The *Dmin* of the circuit indicates the minimum possible delay achievable if no changes in the circuit netlist are allowed during placement. Note that *Dmin* sets the starting seed value for timing optimization. We performed optimization in three configurations: 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 optimization such as buffer insertion, gate sizing, cell replication, etc.

Experimental results are summarized in Table II. In this table we report total wirelength, total power and critical path delay of different benchmarks after placement is performed using the three cases mentioned above. 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.

341

By applying INTRACEL and CELONCEL transformation, we can obtain a reduction in the die area by 29% and 37.5% respectively. In the wirelength driven placement mode, we observe consistent wirelength improvement when comparing CELONCEL and INTRACEL stacking to 2-D placement. In general, the results for CELONCEL are better than INTRACEL in this mode. The average improvement in the wirelength using CELONCEL technique is approximately 15%. The wirelength reduction using INTRACEL technique is nearly 10%.

In the timing driven placement mode, the placer is allowed to move the cells to reduce timing without changing the netlist in any manner. Due to smaller die sizes when CELONCEL or INTRACEL technique is used, we conjecture that critical path delay should also reduce accordingly. Averaged over all benchmarks, the critical path delay of the circuit using CELONCEL technique is 6.1% smaller than 2-D planar case. However, using INTRACEL technique does not show any consistent trend compared to the 2-D case with the average improvement in the critical path delay improving by less than

1%. For this set of experiments, the timing constraint for each benchmark was set to be equal to the theoretical maximum performance it can achieve. This number was obtained by setting interconnect resistance and capacitance equal to zero and running timing analysis.

In the timing driven placement with in-place optimization mode, the placer has flexibility to apply any synthesis or timing optimization transforms to the netlist on the fly to improve the timing. For these set of experiments, we set the timing constraint corresponding to an unachievable number (10 GHz). In this manner, we can test the best performance that each of the technique can ever give. Compared to the 2-D case, use of CELONCEL can reduce the critical path delay even further by 2.75%. Similarly, by using INTRACEL technique, the critical path delay can be reduced by approximately 2.7%. Note that this improvement in critical path delay is over and above the best solution obtained using 2-D planar case, thus hard to obtain.

In summary, the CELONCEL integration proposed by us

Table II. Area, Wirelength, and Circuit delay information of various benchmarks employing CELONCEL design technique

Circuit	Objective	Wirelength Driven			Timing Driven			Timing driven + In-Place Opt.		
		Standard (2D)	INTRACEL (3D)	CELONCEL	Standard (2D)	INTRACEL (3D)	CELONCEL	Standard (2D)	INTRACEL (3D)	CELONCEL
LDPC #Nets = 48K #Cells = 44K #Pins =4100 Dmin=6.904	Wirelength	1.54E+06	1.38E+06	1.37E+06	1.67E+06	1.48E+06	1.42E+06	1.83E+06	1.60E+06	1.54E+06
	Circuit delay (ns)	8.503	7.312	6.904	6.877	6.904	6.904	2.461	2.421	2.129
	Power (mW)	1201	1105	1025	1147	1064	1018	1554	1461	1470
Wb_conmax #Nets = 29K #Cells = 27K #Pins =2546 Dmin=2.382	Wirelength	3.70E+05	3.53E+05	3.24E+05	3.77 E+05	3.38E+05	3.33E+05	3.76E+05	3.64E+05	3.33E+05
	Circuit delay (ns)	4.553	4.845	4.834	4.661	4.628	4.449	1.039	1.041	1.083
	Power (mW)	100.2	100.2	99.97	100.4	99.93	99.64	70.63	71.14	72.05
B19 #Nets = 99K #Cells = 87K #Pins = 77 Dmin=4.305	Wirelength	8.29E+05	7.24 E+05	6.89 E+05	8.60 E+05	7.57 E+05	7.04 E+05	7.93 E+05	7.00 E+05	6.49 E+05
	Circuit delay (ns)	4.822	4.774	4.681	4.723	4.691	4.691	4.224	4.219	4.185
	Power (mW)	434.6	428.5	425.9	434.5	429.7	425.2	337	312.7	314.7
Ethernet #Nets = 43K #Cells = 42K #Pins = 210 Dmin=14.74	Wirelength	4.21 E+05	3.72 E+05	3.43 E+05	4.30 E+05	3.87 E+05	3.57 E+05	4.94 E+05	4.38 E+05	3.97 E+05
	Circuit delay (ns)	30.238	28.97	27.527	30.598	29.063	26.427	1.252	1.281	1.336
	Power (mW)	176.2	171.9	166.8	175.3	170.6	165.6	133.2	132.7	130.7
Des #Nets = 59K #Cells = 56K #Pins = 298 Dmin=2.532	Wirelength	5.84 E+05	5.08 E+05	4.54 E+05	6.06E+05	5.19E+05	4.66E+05	6.71E+05	5.81E+05	5.45E+05
	Circuit delay (ns)	3.316	3.518	4.006	3.854	3.944	3.39	1.132	0.971	1.016
	Power (mW)	536.5	526.8	517.7	535.8	525.3	517.5	620.2	608.2	580.5
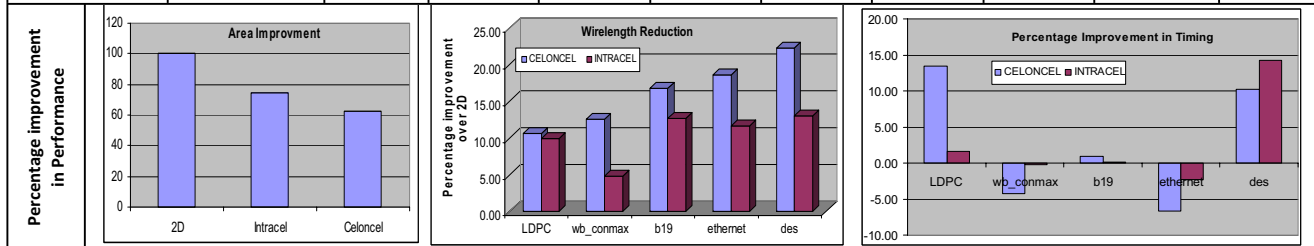
provide improvement, compared to planar and INTRACEL, in several figures of merits of a design (such as wirelength, die area, critical path delay) in different modes of placement. In LDPC decoders, interconnects play a very dominant role as half of the total wires traverse the chip from one end to the other; thereby consuming substantial power as well as creating congestion. When subjected to timing and in-place optimization mode, we could achieve 15.51% gain in wirelength and 13.49% gain in delay when compared to the planar case.

7. Conclusion

3DMI technology, offering 3-D contacts with sizes in the order of ~100 nm, is an effective vehicle for future gigascale circuits. In this work, we focused on new layout stacking techniques leveraging the key benefits of 3-D monolithic integration which allows very small inter-layer vias. INTRACEL stacking folds the diffusion areas within the cells helping in reducing the cell height and thus the die area. On an average the wirelength, critical path delay and the die area were observed to be improved by 10.45%, 1% and 29% respectively. CELONCEL stacking on the other hand allows cells to be placed on top of each other taking into account the pin access issues. As compared to traditional 2-D physical synthesis flow, CELONCEL methods can reduce the wirelength, critical path delay and the die area by 15%, 6.1%, and 37.5% respectively. A placement algorithm was proposed that transforms the monolithic 3-D placement problem into a virtual 2-D problem solved using any commercial 2-D placement tools. A highly parallelizable zero-one linear program (ZOLP) formulation is used for layer assignment followed by linear program (LP) based minimum perturbation for high quality 3-D layout.

Major areas for future research include: Placement techniques taking into account the routing congestion of 3DMI circuits; and layout techniques taking into account multiple (more than two) active layers.

Acknowledgments

References

[Banerjee 01] Banerjee, K.; *et al.*, "3-D ICs: a novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration," *Proceedings of the IEEE*, vol.89, no.5, pp.602-633, May 2001

[Pavlidis 09] V. Pavlidis and E. Friedman, *Three-Dimensional Integrated Circuit Design*. Morgan Kaufmann, 2009.

[Koester 08]: Koester, S. J.; *et al.*, "Wafer-level 3-D integration technology," *IBM Journal of Research and Development*, vol.52, no.6, pp.583-597, Nov. 2008

[Sillon 08]: Sillon, N.; *et al.*, "Enabling technologies for 3-D integration: From packaging miniaturization to advanced stacked ICs," *Proc. IEDM*, pp.1-4, 2008

[ITRS]: www.itrs.net/Links/2009ITRS/2009Chapters_2009Tables

[Jung 07]: S-M Jung; *et al.*, "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," *Proc. VLSI Tech.,*pp.82-83, 2007

[Opencores] www.opencores.org

[Chan 05] Chan, T. F., *et al.*, "mPL6: enhanced multilevel mixed-size placement," *Proc. ISPD, 2006*

[Deng 01] "Deng, Y. and Maly, W. P. 2001, "Interconnect characteristics of 2.5-D system integration scheme," *Proc. ISPD, 2001*

[Loh 07]: Loh, Gabriel H.; Xie, Yuan; Black, Bryan; , "Processor Design in 3-D Die-Stacking Technologies," *Micro, IEEE*, vol.27, no.3, pp.31-48, 2007

[Batude 09a] P. Batude., *et al.*, "GeOI and SOI 3-D monolithic cell integrations for high density applications," *Proc. VLSI Tech.,* pp.166-167, 2009

[Batude 09b] P. Batude., *et al.*, "Advances in 3-D CMOS sequential integration," *Proc. IEDM,* pp.1-4, 2009

[Cong 09] Jason Cong; Guojie Luo; Jie Wei; Yan Zhang;, "Thermal-Aware 3-D IC Placement Via Transformation," Proc. ASP-DAC, pp.780-785, 2007

[Cong 10] J. Cong and G. Luo, "Advances and Challenges in 3-D Physical Design," *IPSJ Trans. on System LSI Design Methodology*, pp 2-18, 2010

[Zhou 06] L. Zhou, C. Wakayama and C.-J. Richard Shi, "CASCADE: A Standard Super-Cell Design Methodology with Congestion-Driven Placement for Three-Dimensional Interconnect-heavy very Large Scale Integrated Circuits", *IEEE Trans. CAD*, 2006.

[Das 03] Das, S.; Chandrakasan, A.; Reif, R.;, "Design tools for 3-D integrated circuits," *Proc. ASP-DAC*, pp. 53- 56, 2003

[Mit 08] MIT 3-D Desisgn Kits, version 3DEM.

[NCSU 45] NCSU FreePDK45 Design Kit.

[Jiang 06] Z.-W. Jiang, *et al.*, "Ntuplace2: a hybrid placer using partitioning and analytical techniques," *proc. ISPD,* pp. 215–217, 2006.

[Kahng 05] A. B. Kahng, S. Reda, and Q. Wang, "Aplace: a general analytic placement framework," *Proc. ISPD,* pp. 233–235, ACM, 2005

[Yang 03] X. Yang, *et al.*, "Congestion reduction during placement with provably good approximation bound," *ACM Trans. Des. Autom. Electron. Syst.*, pp. 316–333, 2003.

[Chakraborty 09] A. Chakraborty, A. Kumar, and D. Z. Pan, "Regplace: a high quality open-source placement framework for structured asics," *Proc. DAC,* pp. 442–447, 2009

[Gurobi] "Gurobi Optimization." http://www.gurobi.com/.

[Nan] "Nangate 45nm Library." http://www.nangate.com/.

[DC] "Synopsys Design Compiler."

[SOCE] "SOC Encounter tool."

[Havemann 2001] Havemann, R.H.; Hutchby, J.A.;, "High-performance interconnects: an integration overview," *Proceedings of the IEEE*, vol.89, no.5, pp.586-601, May 2001.