# Six-track Standard Cell Libraries with Fin Depopulation, Contact over Active Gate, and Narrower Diffusion Break in 7nm Technology

Tzu-Hsuan Wang, Chih-Chun Hsu, Li Kao, Bing-Yu Li, Tung-Chun Wu,
Tsao-Hsuan Peng, Rung-Bin Lin
Yuan Ze University

## Abstract

In this article we present three 6-track standard cell libraries based on ASAP7 PDK which is extended to include three technologies, contacts over active gates (COAG), fin depopulation, and a diffusion break taking a space of one contacted poly pitch (CPP). All these three technologies are invented to reduce standard cell area and thus chip area. Experimental results show that fin depopulation solely can achieve 8.3% area saving, COAG brings about another 9.3%, and a diffusion break of 1 CPP adds another 2.5% more. These three technologies all together bring about 20% area saving when compared with that obtained by employing a 7.5-track cell library without excising these three technologies.

## Keywords

Standard cell, library, FinFET, contact, fin depopulation, diffusion break

## 1. Introduction

The development of semiconductor technology is characterized by numerous innovations and inventions, simply naming a few, from two metal layers to more than 10 metal layers, from polysilicon gate to metal gate, from 2-D transistor to 3-D FinFET transistor [1], from aluminum to copper and then cobalt interconnects [2-4], strained silicon, buried power rails gotten access from back-side [5], low-k dielectric, air gaps, etc. Every piece of the work showcases the incredible engineering talents born to advance semiconductor manufacturing technology. As the technology approaches the physical geometry limit, the momentum of innovation and invention remains unexpectedly strong. Numerous manufacturing technologies and design methods are still relentlessly invented to purport Moore's law. Among these are contacts over active gates (COAG) [2,6-9], fin depopulation [10,11], and narrower diffusion break [12-14]. COAG allows a contact originally located in the isolation region being dropped onto a poly wire in an active region. This greatly eases establishing a connection from an M1 pin to a poly gate, indirectly improving signal routing in a standard cell, and thus provides a chance of reducing cell height further. Intel has observed a 10% reduction in chip area by this technology [2,4]. Besides, COAG can be leveraged to reduce the parasitic resistance of a metal gate to the channel for high-fin count transistors employed for 5G applications [15]. Fabricating COAG typically requires placing an etch-stop layer above a ploy gate and employs self-aligned patterning lithography [7,8].

Fin depopulation is to reduce the number of active fins per ploy gate in a standard cell. This technology can be employed to reduce standard cell height. A cell height of 7 or 7.5 tracks typically can provide three active fins per poly gate for P(N) transistor in 7nm process technology. A cell height of 6 tracks might have to reduce it to two fins. We are very likely to see only one fin per ploy gate for 5-track cells. Nevertheless, fin depopulation cannot come without any cost if drive current delivered per transistor must be maintained. The cost is to increase fin height, which is certainly a challenging task [16]. Intel's 10nm technology employs a fin height of 46nm, a fin width of 7nm, and a gate length of 18nm [3].

A narrower diffusion break for isolating two active regions can further reduce standard cell area. Such a break is needed not only at the both sides of a standard cell but also inside a cell where transistor abutment cannot be done by diffusion sharing. ASAP7 PDK [17-19] specifies a diffusion break of two contacted ploy pitches (CPP) whereas Intel's 10nm technology takes a space of only one CPP [2].

ASAP7 PDK features a 7nm FinFET technology [17-19]. Although it is only a simulated process technology, it has essential features found in a corresponding commercial process technology. It is now an important vehicle for academia to get access to advanced process technology. However, it does not provide the three advanced technologies discussed above. Therefore, this work makes the first attempt to include them into ASAP7 PDK. Our work has the following contributions.

- Including COAG, narrower diffusion break, and fin depopulation technologies into ASAP7 PDK.
- Three 6-track cell libraries that explore either one or two, or all of the three technologies.
- Experimental results show that fin depopulation solely can achieve 8.3% area saving, COAG brings about another 9.3%, and a narrower diffusion break adds another 2.5%. These three technologies all together bring about 20% area saving. We also find that the merit of fin depopulation can be better realized only when COAG is excised.
- It is observed that the three technologies are beneficial much less to a circuit with a large Rent's exponent which implies existence of highly congested spots. In other words, routing congestion partially reduces the merit of these three technologies. Degradation of pin accessibility reduces their merit further.

The rest of the article is organized as follows. Section 2 gives an overview of ASAP7 PDK. Section 3 presents our methodology. Section 4 provides some experimental results. The last section gives a conclusion.

## 2. ASAP7 PDK and ASAP7L

ASAP7 PDK [17-20] depicts a 7nm FinFET process technology whose key features are derived from technology

978-1-7281-7641-3/21/$31.00 ©2021 IEEE

data found in the open literature. Table 1 presents the dimensions of the major features used to form FinFETs and interconnects [20]. Fig. 1 shows a 3D view of the major layers for forming a transistor [21]. LIG and LISD are local interconnects under M1. LISD wires can bring out source/drain signals from a transistor. LIG wires can send signals from M1 input pins down to poly gates. Here, *an LIG wire serves as a gate contact and cannot locate inside an active region*. Such a limitation forces gate contacts to crowd in the isolation region between P and N active regions. As a result, there may not have enough space for deploying input pins of a high-pin count cell. It may also restrict the use of M1 for routing within a standard cell. This is the reason why COAG is a key technology for creating denser circuits.

Table 1: Dimensions of major features in ASAP7 PDK

| | | | | | |
|---|---|---|---|---|---|
| Fin height* | 32 | Fin thickness | 7 | Fin pitch | 27 |
| Poly gate length | 20 | Contacted poly pitch (CPP) | 54 | M1, M2, M3 pitch | 36 |
| M4, M5 pitch | 48 | M6, M7 pitch | 64 | M8, M9 pitch | 80 |

*unit in *nm*; metal width is 1/2 pitch of the underlying layer.

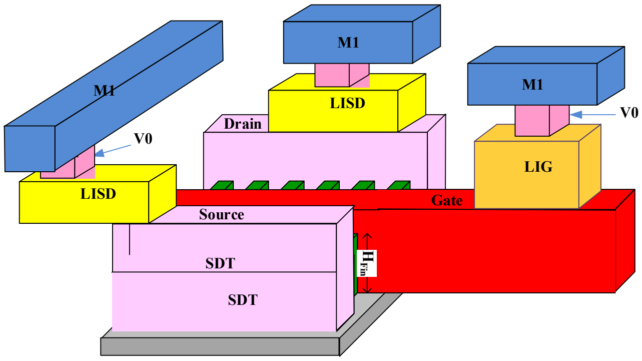

Fig. 1. A 3-D view of FinFET and related layers defined in ASAP7 PDK

Accompanied with ASAP7 PDK is a standard cell library. This cell library simply named **ASAP7L** has 197 cells [13]. ASAP7L adopts a cell height of 7.5 M1 pitches (i.e., a cell height of 270nm). This height allows deploying 10 fins in a standard cell. Among them, up to three fins can locate inside a P active region and an N active region, respectively. Gate contacts (LIG wires) must locate in the isolation region between a P and an N active region. The two sides of a standard cell should have respectively a dummy ploy gate for accommodating a diffusion break. Hence, when two standard cells abut, a diffusion break of 2 CPPs is formed.

## 3. Methodology

This section first shows how the layout design rules can be made to enable COAG, narrower diffusion breaks, and fin depopulation in ASAP7 PDK. It then presents three standard cell libraries designed with these technologies.

### 3.1. COAG

To include COAG technology into ASAP7 PDK, we need modify the layout design rules to allow an LIG wire being placed inside an active region. Fig. 2 shows the modified layout design rules. The contact on the middle poly gate is on the active region while the one on the leftmost poly gate is not. The minimal spacing (Rules 5 and 6) between LIG and LISD (also SDT) is reduced from 14nm to 4nm. A spacing of 4nm is determined on the premise that other related rules except Rule 8 are kept unchanged while a minimum-sized LIG wire of 22nm wide is laid over an active gate. Inspecting Fig. 2 closely, we can easily come up with this number. Besides Rules 5 and 6, the spacing specified in Rule 8 should be reduced from 15nm to 5nm. The reduction amount is the same as that found in Rules 5 and 6. Since contacts can now reside over active poly gates, Rule 9 should be deleted. One should refer to [18] for the details of layout design rules. Here, we assume that self-aligned patterning technology can be employed to realize such a geometrical structure [13,14].

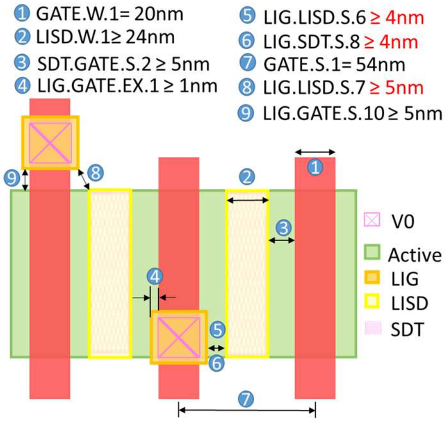

Fig. 2. Rules 5, 6, and 8 modified and rule 9 deleted.

### 3.2. Narrower diffusion break

The width of a diffusion break specified in ASAP7 PDK is 2 CPPs. In our work, we reduce it to 1 CPP. Similarly, we need modify the spacing rules between two active regions to allow a narrower diffusion break. The modification is based on the premise that the minimum-sized inverter (INVX1) has a width of 2 CPPs and the minimum-sized tap cell has a width of only 1 CPP as shown in Fig. 3. In this example, the total width of these two cells is reduced by 2 CPPs if a diffusion break of 1 CPP is employed. Note that cell width can be reduced further if there is a diffusion break inside a cell. One may notice that the width of a 2-CPP diffusion break is actually smaller than 2 CPPs. However, it is usually called "double diffusion breaks" [12,13]. Calling double diffusion breaks is misleading because it merely means a wider diffusion break rather than two diffusion breaks. So what a 2-CPPs really means here is a space shown in Fig. 3(a). Note that a tap cell can connect P-substrate to GND and N-well to VDD. Since a tap cell now has a minimum width of 1 CPP, minimum NSELECT (PSELECT) width should be reduced from 2 CPPs to 1 CPP (i.e., 54nm). Accordingly, Rule 1 through Rule 5 should be modified as shown in Fig. 4. Note that if a rule for NSELECT need be modified, its counterpart for PSELECT should be also modified. Here, we assume the state-of-the-art lithography technology is capable of realizing the so-specified geometry patterns [12,13].

Given a cell of N poly gates, such a technology can reduce cell area by at least 200/N percent. However, a side effect is the reduction of wiring resources for signal routing inside a cell. This side effect may be carried over to chip-level routing and thus reduce the merit delivered by this technology.

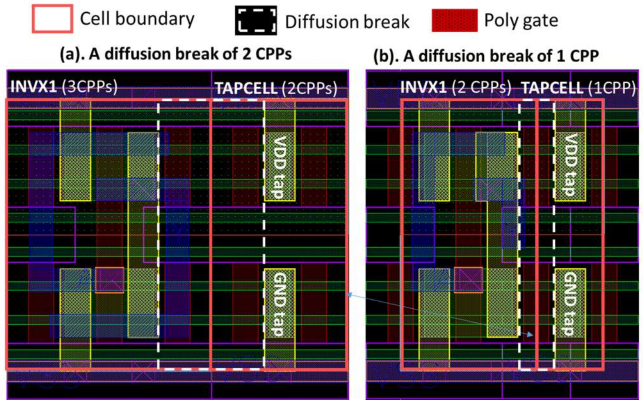

Fig. 3. Narrowing a diffusion break.

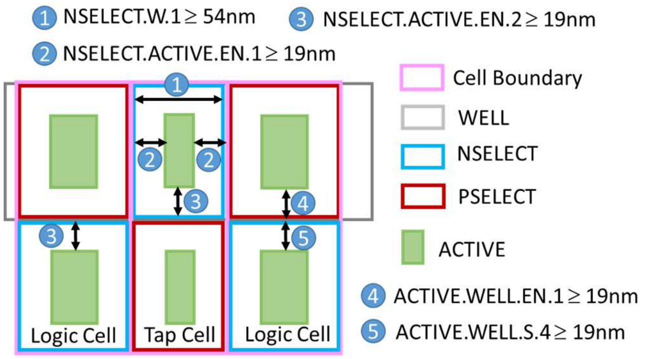

Fig. 4. Rules modified for enabling a narrower diffusion break.

### 3.3. Fin depopulation

To include fin depopulation technology into ASAP7 PDK, we do it on the premise that the drive current of three FinFETs should be close to that of two FintFETs after fin depopulation. In other words, a transistor of three FinFETs will be replaced by a transistor of two FinFETs with taller fins. In this work, fin height is increased from 32nm to 49nm. It is close to 46nm which is the fin height for Intel's 10nm technology.

Basically, we can solve equation (1) to find out the new fin height $h' = 1.5h + 0.25w$.

$$3(2h + w) = 2(2h' + w) \tag{1}$$

where $h$ is the original fin height and $w$ is the original fin width. Here, we assume fin width is not changed. Given $h = 32nm$ and $w = 6.5nm$ specified in the SPICE device model in ASAP7 PDK, $h' = 50nm$ (rounded to $nm$). The fin height should be increased from $32nm$ to $50nm$. Although such a simple calculation offers us a clue about fin height for fin depopulation, we need check further by simulation to see whether a transistor of two fins with a fin height of $50nm$ will deliver an amount of drive current close to that delivered by a transistor of three fins with a fin height of $32nm$. Hence, we re-characterize the delays of some selected cells in ASAP7L with different fin heights. We then choose the fin height that results in the delays of the selected cells closest to the delays of these cells with a fin height of 32nm. We totally select from ASAP7L 37 cells that perform INV, NAND, AND, NOR and OR functions with different number of inputs and multiple drive classes per logic function. Before re-characterization, the SPICE netlists of the selected cells are modified to reflect the scaling of fin numbers by 2/3. We round the fin number to the nearest integer. For example, if a transistor originally employs 5 fins, it becomes 3 fins after fin depopulation. Nevertheless, a transistor should contain at least one fin after fin depopulation. Once we complete modification of SPICE netlists, we handcraft the layouts of the selected cells. If layouts pass DRC and LVS checks, we extract their SPICE netlists along with parasitic resistance and capacitance. We then perform timing characterization for each cell respectively with a fin height from 45nm to 55nm. We adopt the fin height that results in the smallest average discrepancy in rise transition delay, fall transition delay, rise delay, and fall delay with respect to the delays presented in the ASAP7L timing library. We thus determine finally that a fin height of 49nm is employed for fin depopulation in our work.

### 3.4. Six-track standard cell libraries

Given the three technologies added to ASAP7 DPK, we are curious about to what extent chip area saving can be achieved. Hence, we take advantage of COAG and fin depopulation to create a cell library whose cells employ a cell height of 6 tracks. A 6-track cell uses 20% less area than the same cell designed with a 7.5-track cell height. Given the M1 track pitch of 36nm, 6-track cell height is 216nm. With this cell height and a fin pitch of 27nm, exactly 8 fins from top to bottom can be laid out in a cell. Among them, up to two fins are in P and N active regions, respectively. This library is called **CPS6L**. The first letter **C** means that the cells are designed with COAG technology. The second letter **P** means that the cells are designed with fin depopulation. The third letter **S** means that a diffusion break of 1 CPP is employed.

For the purpose of comparison, we also create a 6-track cell library that employs a diffusion break of 2 CPPs. This library is called **CPD6L**. The third letter **D** means that a diffusion break of 2 CPPs is employed. Note that the cells in this library allow COAG and fin depopulation. We further create yet another 6-track cell library without COAG. This cell library is called **XPD6L**. The first letter **X** means that the related technology COAG is not excised for cell design. Besides, we also obtain the cell library presented in [20]. This cell library is here called **XXD7.5L** for simplicity.

All the cell layouts in the three 6-track cell libraries are handcrafted. Routing layers are up to M2. I/O pins of a cell are on M1 or M2. Note that 6-track cells may employ more M2 wires. With a cell height of 6 tracks, the layout task for high-pin count cells or large-sized cells such as D flip flops, full adder, etc. is challenging. CPS6L, CPD6L, and XXD7.5L each have 208 logic cells (without including filler and tap cells). XPD6L has 207 cells due to excluding some non-routable D flip-flops. After layout design is completed, post-layout netlist and parasitic R/C are extracted. A LEF file for each cell library is also generated. *Liberate* from Cadence is used to perform timing characterization and create a timing library. Non-linear delay model is employed in the library. Note that the cells in these four libraries have similar timing characteristics.

Table 2 summarizes the technologies employed for cell library design. Fig. 5 shows the layouts of MAJxp5 respectively from the four libraries. The cell from XXD7.5L has the largest area whereas the one from CPS6L has the smallest area. The cell from CPD6L is of the same size as the cell from XPD6L.

Table 2: Technologies employed in the four cell libraries.

| | Cell height (tracks) | COAG | Fin depopulation | Diffusion break width (CPP) |
|---|---|---|---|---|
| **CPS6L** | 6 | yes | yes | 1 |
| **CPD6L** | 6 | yes | yes | 2 |
| **XPD6L** | 6 | no | yes | 2 |
| **XXD7.5L** | 7.5 | no | no | 2 |

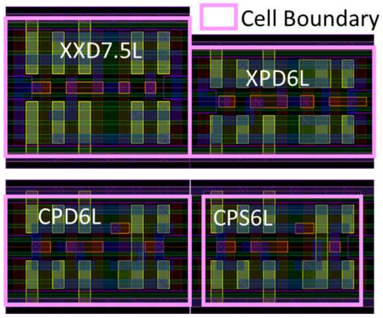

Fig. 5. Layouts of MAJxp5 from different cell libraries.

If a chip is designed with a cell library containing smaller cells, not only are routing resources fewer, but also pin access points are fewer. Routing of the chip will be more difficult. Table 3 shows the number of access points per pin for the cells in these four libraries. The column *onTrk* gives the average number of on-track access points. *offTrk* denotes otherwise. Here, an on-track access point is the one having a chance of locating on an M2 track [20]. Table 3 shows that CPS6L has the smallest number of access points per pin because its cells have the smallest area. Although the cells in CPD6L and XPD6L have similar areas, CPD6L with COAG has a larger number of access points per pin than XPD6L does. Note that output pin access is not an issue because the average number of access points per output pin is about twice the average number of access points per input pin. Also note that most of pin access points are on track.

Table 3: Average number of access points per pin.

| Library | CPS6L | | CPD6L | | XPD6L | | XXD7.5L | |
|---|---|---|---|---|---|---|---|---|
| | onTrk + offTrk | onTrk | onTrk + offTrk | onTrk | onTrk + offTrk | onTrk | onTrk + offTrk | onTrk |
| **per pin** | 5.75 | 5.74 | 6.96 | 6.95 | 6.10 | 5.90 | 7.62 | 7.15 |
| **per input pin** | 4.90 | 4.89 | 5.89 | 5.89 | 4.63 | 4.42 | 5.88 | 5.47 |
| **per output pin** | 8.90 | 8.90 | 10.88 | 10.88 | 11.50 | 11.30 | 14.05 | 13.31 |
| **STDEV per input pin** | 3.49 | 3.5 | 4.09 | 4.09 | 4.29 | 4.21 | 4.03 | 3.62 |
| **STDEV per output pin** | 8.4 | 8.4 | 8.36 | 8.36 | 9.11 | 9.21 | 9.16 | 9.08 |

## 4. Experimental results

The experiments are performed to see how much area saving can be achieved by the three 6-track cell libraries, CPS6L, CPD6L, and XPD6L. Before doing so, we first look into the benchmark circuits for our experiments and then show the experimental results.

### 4.1. Benchmark circuits

The five large benchmark circuits from [20] are used for our experiments. They are resynthesized by Synopsys's Design Compiler with different clock periods. Table 4 shows some statistics of these circuits synthesized with XXD7.5L. The statistics for the circuits synthesized with other three 6-track cell libraries are similar to that presented in Table 4. Note that *Neural Network* has much more primary I/O pins. The clock periods given in the table are the targeted performance indices for synthesis and place&route tools. We here deliberately set a looser timing performance index for *Neural Network* to diversify chip-design scenarios.

We further look into the Rent's exponents of these circuits [27] as shown in Table 5. The same circuit is synthesized with each individual library and then *hMETIS* [28] is used for partitioning circuits to obtain Rent's exponent. *RISC-V* and *AES* have a higher Rent's exponent which indicates that routing of these two circuits will be more difficult. Although *Neural Network* has a smaller Rent's exponent, it has 1602 I/O pins which could also make routing more challenging. Note that there is a noticeable difference in Rent's exponents of the same circuit synthesized with different cell library.

Table 4: Statistics of benchmark circuits.

| Circuits | # of cells | # of nets | # of FFs | Clk pd | # of I/Os |
|---|---|---|---|---|---|
| **b19 [22]** | 54118 | 54141 | 5519 | 1.8ns | 77 |
| **Neural Network [23]** | 111207 | 112489 | 4671 | 1.33ns | 1602 |
| **GPU [24]** | 190907 | 190992 | 48247 | 2.4ns | 269 |
| **RISC-V[25]** | 18529 | 18677 | 2347 | 2.4ns | 267 |
| **AES [26]** | 159298 | 159683 | 11696 | 0.71ns | 513 |

Table 5: Rent's exponents of benchmark circuits.

| Circuits | CPS6L | CPD6L | XPD6L | XXD7.5L |
|---|---|---|---|---|
| **b19** | 0.619 | 0.628 | 0.618 | 0.628 |
| **Neural Network** | 0.518 | 0.511 | 0.514 | 0.498 |
| **GPU** | 0.596 | 0.633 | 0.605 | 0.627 |
| **RISC-V** | 0.679 | 0.677 | 0.692 | 0.674 |
| **AES** | 0.745 | 0.748 | 0.743 | 0.748 |

### 4.2. Experimental setup

Our experiments are performed in the following manner. First, we try to find a core utilization percentage that will provide enough routing resource for completing chip routing without any DRC violations (DRC clean for short) using just M1 (M2) through M5. We start this from a core utilization of 90% and decreasing it gradually. Next, we do the same task but also use M6 for routing if routing of a circuit using only M1(M2) though M5 with a core utilization of 90% is completed with DRC violations. The above tasks are repeated for the following four settings

**M1&mayVia:** Also use M1 for routing but may use a via for pin access or an M1 (M2) wire directly connected to M1 pins (M2 pins).

**noM1&mayVia:** Do not use M1 for routing but may use a via for pin access or an M2 wire directly connected to M2 pins.

**M1&mustVia:** Also use M1 for routing and must use a via for pin access.

**noM1&mustVia:** Do not use M1 for routing but must use a via for pin access.

Note that if there are no M2 pins, *noM1&mustVia* and *noM1&mayVia* are the same. Also, one can expect that *M1&mustVia* will be similar to *noM1&mustVia* if M1 routing resources are scarce. These four settings are available in Cadence's Innovus. They can be used to evaluate pin accessibility of a chip designed with different cell libraries. Hence, we use Cadence's Innovus to perform place&route.

Power stripes are deployed on M3. Clock tree synthesis is done before routing.

### 4.3. Results

Tables 6 through 9 give area saving percentage, worst case negative slack (WNS), and total negative slack (TNS) achieved by the four cell libraries respectively with the four settings presented in Subsection 4.2. Area saving is computed with respect to the area obtained using XXD7.5L. Each table has two parts. One part contains data obtained when the top most routing layer is M5. In this part, column "Core ut" gives the largest core utilization percentage when a DRC-clean design is obtained. The other part contains data obtained when the top most routing layer is M6. Logically, given one more layer for routing, the largest core utilization percentage for obtaining a DRC-clean design should be larger for some circuits. Taking b19 under *M1&mayVia* setting in Table 6 for example, with CPS6L we obtain a DRC-clean design at core utilization 75% using M1~M5 for routing whereas it is 76% if M1~M6 are used. Here we find that CPS6L has the smallest core utilization for obtaining a DRC-clean design. Nevertheless, in some situation we still cannot obtain a DRC-clean design even though a core utilization below 60% is used. This situation often occurs to the designs that employ XPD6L. Looking into XPD6L further, we find from Table 3 that it has the least number of access points per input pin. Especially, a few cells in XPD6L contain some pin having only one access point. Once a circuit employs these cells, the place&route tools may not obtain a DRC-clean design even though a very small core utilization percentage is used. In this situation, our exploration stops at a core utilization percentage that will first make the DRC errors of a routed design fewer than 10. This is noted by labeling a * after core utilization percentage in Tables 7 and 9. The reason for doing so is that we believe a DRC-clean design can be obtained with such a core utilization percentage if these cells are re-designed with a larger footprint to improve their pin accessibility. In general, CPS6L achieves largest area saving, then CPD6L, and then XPD6L. The saving can be up to 27.56% for *GPU* with CPS6L. Also, with CPS6L, the circuits typically achieve best slack. One may notice that CPS6L does not achieve good area saving for *Neural Network* and *AES* with *noM1&mustVia* and *noM1&mustVia* settings. Also, all the three 6-track libraries achieve relatively a smaller area saving for *RISC-V* and *AES* due to routing congestion as dictated by their larger Rent's exponents.

Table 10 summarizes the average area saving percentage over the four settings. CPS6L achieves on average 20% area saving. Fin depopulation solely can achieve 8.3% area saving due to cell height reduction. As shown in the last column of Table 10, area saving can be up to 19.5% (calculated using total cell area of designs before place&route). This indicates that the merit of fin depopulation is not fully realized due to poor pin accessibility of the cells in XPD6L. COAG brings about another 9.3% area saving. This is close to 10% area saving reported by Intel [2,4]. Note that COAG provides more accessible pins that help realize most of the merit of fin depopulation. A diffusion break of 1 CPP yet brings about 2.5% more. Clearly, the merit of a narrower diffusion break employed by CPS6L is not fully realized.

Table 6: Area saving percentages for M1&mayVia.

| M1&mayVia | | M1~M5 for routing | | | | | M1~M6 | |
|---|---|---|---|---|---|---|---|---|
| | | Core ut % | Core area (µm^2) | Area saving | WNS | TNS | Core ut % | Area saving |
| b19 | XXD7.5L | 90 | 101296 | 0 | -0.09 | -3.3 | 90 | 0 |
| | XPD6L | 88 | 83480 | 17.59 | -0.08 | -8.05 | 88 | 17.59 |
| | CPD6L | 90 | 81825 | 19.22 | -0.08 | -2.87 | 90 | 19.22 |
| | CPS6L | 75 | 79251 | 21.76 | -0.07 | -1.3 | 76 | 22.8 |
| Neural Network | XXD7.5L | 90 | 167685 | 0 | 0.28 | 0 | 90 | 0 |
| | XPD6L | 84 | 144134 | 14.04 | 0.07 | 0 | 84 | 14.04 |
| | CPD6L | 90 | 132251 | 21.13 | 0 | -0.15 | 90 | 21.13 |
| | CPS6L | 69 | 136739 | 18.45 | 0.3 | 0 | 69 | 18.45 |
| GPU | XXD7.5L | 90 | 477046 | 0 | 0 | 0 | 90 | 0 |
| | XPD6L | 85 | 429179 | 10.03 | -0.1 | -1.29 | 85 | 10.03 |
| | CPD6L | 90 | 382418 | 19.84 | -0.04 | -0.07 | 90 | 19.84 |
| | CPS6L | 82 | 354024 | 25.79 | 0.03 | 0 | 84 | 27.56 |
| RISC-V | XXD7.5L | 90 | 38519 | 0 | -0.06 | -0.76 | 90 | 0 |
| | XPD6L | 80 | 35638 | 7.48 | -0.07 | -0.51 | 82 | 9.73 |
| | CPD6L | 75 | 37370 | 2.98 | 0.06 | 0 | 85 | 14.4 |
| | CPS6L | 70 | 33024 | 14.27 | 0.12 | 0 | 73 | 17.77 |
| AES | XXD7.5L | 90 | 248558 | 0 | -0.02 | -0.15 | 90 | 0 |
| | XPD6L | 80 | 227863 | 8.33 | -0.17 | -10.33 | 80 | 8.33 |
| | CPD6L | 80 | 218211 | 12.21 | -0.02 | -0.04 | 82 | 14.34 |
| | CPS6L | 70 | 200488 | 19.34 | 0.03 | 0 | 70 | 19.34 |
| Average area saving (%) | XXD7.5L | 0 | | | | | 0 | |
| | XPD6L | 11.49 | | | | | 11.94 | |
| | CPD6L | 15.08 | | | | | 17.79 | |
| | CPS6L | 19.92 | | | | | 21.18 | |

Table 7: Area saving percentages for noM1&mayVia.

| noM1&mayVia | | M2~M5 for routing | | | | | M2~M6 | |
|---|---|---|---|---|---|---|---|---|
| | | Core ut % | Core area (µm^2) | Area saving | WNS | TNS | Core ut % | Area saving |
| b19 | XXD7.5L | 90 | 101296 | 0 | -0.09 | -3.63 | 90 | 0 |
| | XPD6L | 68 | 109642 | -8.24 | -0.06 | -4.56 | 68 | -8.24 |
| | CPD6L | 90 | 81825 | 19.22 | -0.09 | -3.03 | 90 | 19.22 |
| | CPS6L | 75 | 79251 | 21.76 | -0.07 | -1.41 | 75 | 21.76 |
| Neural Network | XXD7.5L | 90 | 167685 | 0 | 0.28 | 0 | 90 | 0 |
| | XPD6L | 81* | 149486 | 10.85 | 0.09 | 0 | 81* | 10.85 |
| | CPD6L | 90 | 132251 | 21.13 | 0.28 | 0 | 90 | 21.13 |
| | CPS6L | 68 | 138720 | 17.27 | 0.28 | 0 | 68 | 17.27 |
| GPU | XXD7.5L | 90 | 477046 | 0 | 0 | 0 | 90 | 0 |
| | XPD6L | 72* | 506630 | -6.2 | -0.07 | -1 | 80 | 4.41 |
| | CPD6L | 90 | 382418 | 19.84 | -0.03 | -0.06 | 90 | 19.84 |
| | CPS6L | 82 | 354024 | 25.79 | 0.03 | 0 | 84 | 27.56 |
| RISC-V | XXD7.5L | 90 | 38519 | 0 | -0.06 | -0.7 | 90 | 0 |
| | XPD6L | 74* | 38522 | -0.01 | -0.01 | -0.04 | 74* | -0.01 |
| | CPD6L | 80 | 35015 | 9.1 | -0.03 | -0.11 | 83 | 12.36 |
| | CPS6L | 70 | 33024 | 14.27 | 0.12 | 0 | 73 | 17.77 |
| AES | XXD7.5L | 90 | 248558 | 0 | -0.02 | -0.12 | 90 | 0 |
| | XPD6L | 78 | 233743 | 5.96 | -0.13 | -11.11 | 78 | 5.96 |
| | CPD6L | 77 | 226767 | 8.77 | -0.03 | -0.03 | 82 | 14.34 |
| | CPS6L | 62 | 226357 | 8.93 | 0.02 | 0 | 68 | 16.96 |
| Average area saving (%) | XXD7.5L | 0 | | | | | 0 | |
| | XPD6L | 0.47 | | | | | 2.6 | |
| | CPD6L | 15.61 | | | | | 17.38 | |
| | CPS6L | 17.6 | | | | | 20.26 | |

We have found that pin access problem is very difficult to get around although our layout designs are constantly reviewed and modified for better pin accessibility. Pin access problem typically occurs when a pin having only one or two access points locates right under an M3 power stripe. We believe it is better to insert some dummy poly gates into high-pin count cells rather than lowering core utilization percentage to address pin access problem so that the merit of fin depopulation and narrower diffusion breaks can be better realized. We also guess that a narrower diffusion break may be more useful with a 7.5-track cell library.

Table 8: Area saving percentages for M1&mustVia.

| M1&mustVia | | M1~M5 for routing | | | | | M1~M6 | |
|---|---|---|---|---|---|---|---|---|
| | | Core ut % | Core area (µm^2) | Area saving | WNS | TNS | Core ut % | Area saving |
| b19 | XXD7.5L | 90 | 101296 | 0 | -0.1 | -3.8 | 90 | 0 |
| | XPD6L | 77 | 95402 | 5.82 | -0.08 | -8.64 | 78 | 7.04 |
| | CPD6L | 90 | 81825 | 19.22 | -0.1 | -3.42 | 90 | 19.22 |
| | CPS6L | 75 | 79251 | 21.76 | -0.07 | -1.04 | 76 | 22.80 |
| Neural Network | XXD7.5L | 90 | 167685 | 0 | 0.26 | 0 | 90 | 0 |
| | XPD6L | 84 | 144134 | 14.04 | 0.05 | 0 | 84 | 14.04 |
| | CPD6L | 90 | 132251 | 21.13 | 0.28 | 0 | 90 | 21.13 |
| | CPS6L | 69 | 136739 | 18.45 | 0.3 | 0 | 69 | 18.45 |
| GPU | XXD7.5L | 90 | 477046 | 0 | 0 | 0 | 90 | 0 |
| | XPD6L | 80 | 455998 | 4.41 | -0.13 | -2.74 | 80 | 4.41 |
| | CPD6L | 90 | 382418 | 19.84 | -0.04 | -0.17 | 90 | 19.84 |
| | CPS6L | 82 | 354024 | 25.79 | 0 | 0 | 84 | 27.56 |
| RISC-V | XXD7.5L | 90 | 38519 | 0 | -0.07 | -0.56 | 90 | 0 |
| | XPD6L | 80 | 35638 | 7.48 | -0.07 | -0.57 | 81 | 8.6 |
| | CPD6L | 75 | 37370 | 2.98 | 0.05 | 0 | 85 | 14.4 |
| | CPS6L | 68 | 33999 | 11.74 | 0.1 | 0 | 75 | 19.94 |
| AES | XXD7.5L | 90 | 248558 | 0 | -0.02 | -0.18 | 90 | 0 |
| | XPD6L | 75 | 243094 | 2.2 | -0.07 | -11.31 | 75 | 2.2 |
| | CPD6L | 80 | 218211 | 12.21 | -0.02 | -0.04 | 82 | 14.34 |
| | CPS6L | 70 | 200488 | 19.34 | 0.03 | 0 | 70 | 19.34 |
| Average area saving (%) | XXD7.5L | 0 | | | | | 0 | |
| | XPD6L | 6.79 | | | | | 7.26 | |
| | CPD6L | 15.08 | | | | | 17.79 | |
| | CPS6L | 19.42 | | | | | 21.62 | |

Table 9: Area saving percentages for noM1&mustVia.

| noM1&mustVia | | M2~M5 for routing | | | | | M2~M6 | |
|---|---|---|---|---|---|---|---|---|
| | | Core ut % | Core area (µm^2) | Area saving | WNS | TNS | Core ut % | Area saving |
| b19 | XXD7.5L | 90 | 101296 | 0 | -0.1 | -3.62 | 90 | 0 |
| | XPD6L | 70* | 104927 | -3.58 | -0.07 | -7.33 | 72* | 25.23 |
| | CPD6L | 90 | 81825 | 19.22 | -0.1 | -3.33 | 90 | 19.22 |
| | CPS6L | 75 | 79251 | 21.76 | -0.07 | -1.13 | 75 | 21.76 |
| Neural Network | XXD7.5L | 90 | 167685 | 0 | 0.28 | 0 | 90 | 0 |
| | XPD6L | 84 | 144134 | 14.04 | 0.05 | 0 | 84 | 14.04 |
| | CPD6L | 90 | 132251 | 21.13 | 0.27 | 0 | 90 | 21.13 |
| | CPS6L | 57 | 165480 | 1.31 | -0.03 | -0.03 | 57 | 1.31 |
| GPU | XXD7.5L | 90 | 477046 | 0 | 0 | 0 | 90 | 0 |
| | XPD6L | 64* | 569995 | -19.48 | -0.07 | -1.15 | 75* | -1.95 |
| | CPD6L | 90 | 382418 | 19.84 | -0.04 | -0.23 | 90 | 19.84 |
| | CPS6L | 80 | 362901 | 23.93 | -0.01 | -0.01 | 83 | 26.68 |
| RISC-V | XXD7.5L | 88 | 39400 | 0 | 0.04 | 0 | 88 | 0 |
| | XPD6L | 80 | 35638 | 9.55 | -0.07 | -0.59 | 80 | 9.55 |
| | CPD6L | 79 | 35473 | 9.97 | 0 | 0 | 84 | 15.32 |
| | CPS6L | 68 | 33999 | 13.71 | 0.09 | 0 | 73 | 19.61 |
| AES | XXD7.5L | 90 | 248558 | 0 | -0.02 | -0.18 | 90 | 0 |
| | XPD6L | 80* | 227863 | 8.33 | -0.08 | -9.89 | 82* | 10.55 |
| | CPD6L | 78 | 223830 | 9.95 | -0.03 | -0.08 | 80 | 12.21 |
| | CPS6L | 58 | 241961 | 2.65 | 0.03 | 0 | 68 | 16.96 |
| Average area saving (%) | XXD7.5L | 0 | | | | | 0 | |
| | XPD6L | 1.77 | | | | | 11.48 | |
| | CPD6L | 16.02 | | | | | 17.54 | |
| | CPS6L | 12.67 | | | | | 17.27 | |

Table 10: Average area saving over the four settings.

| | Up to M5 (%) | Up to M6 (%) | Upper Bound (%) |
|---|---|---|---|
| XPD6L | 5.1 | 8.3 | 19.5 |
| CPD6L | 15.4 | 17.6 | 21.6 |
| CPS6L | 17.4 | 20.1 | 35.7 |

## 5. Conclusion

In this work we extend ASAP7 PDK to include three technologies, contacts over active gates (COAG), a diffusion break of 1 CPP, and fin depopulation. We have designed three 6-track standard cell libraries to assess the merit of these technologies. We find that fin depopulation solely can achieve 8.3% area saving. COAG brings about another 9.3%, and a narrower diffusion break adds 2.5% more. These three technologies all together bring about 20% area saving.

## 6. References

[1] C. Auth, C. Allen, A. Blattner, D. Bergstrom, M. Brazier, M. Bost, et al., "A 22nm High Performance and Low-power CMOS Technology Featuring Fully-depleted Tri-gate Transistors, Self-aligned Contacts and High Density MIM Capacitors," Symposium on VLSI Technology, pp. 131-132, 2012.

[2] C. Auth, A. Aliyarukunju, M. Asoro, D. Bergstrom, V. Bhagwat, et al., "A 10nm High Performance and Low-Power CMOS Technology Featuring 3rd Generation FinFET Transistors, Self-Aligned Quad Patterning, Contact over Active Gate and Cobalt Local Interconnects," IEDM, 2017, pp. 29.1.1-29.1.4.

[3] A. Yeoh, A. Madhavan, N. Kybert, S. Anand, J. Shin, M. Asoro et al., "Interconnect Stack Using Self-aligned Quad and Double Patterning for 10nm High Volume Manufacturing," IITC, 2018, pp. 144-147.

[4] IEDM 2017 + ISSCC 2018: Intel's 10nm, Switching to Cobalt Interconnects, https://fuse.wikichip.org/news/525/iedm-2017-isscc-2018-intels-10nm-switching-to-cobalt-interconnects/.

[5] D. Prasad, S. S. Teja Nibhanupudi, S. Das, O. Zografos, B. Chehab, S. Sarkar, et al., "Buried Power Rails and Back-side Power Grids: Arm ® CPU Power Delivery Network Design Beyond 5nm," IEDM, 2019, pp. 19.1.1~19.1.4.

[6] D. James, "Update: TSMC's 5nm CMOS Technology Platform," Semiconductor Digest, Feb. 2, 2020.

[7] A. J. Pethe, T. Ghani, M. Bohr, C. Webb, H. Gomez, and A. Cappellani, "Gate Contact Structure over Active Gate and Method to Fabricate Same," US Patent, US9461143B2, Oct. 4, 2016.

[8] Y. Shusterman, M. Sachan, S. S. Roy, R. Freed, and S. Natarajan, "Self-aligned Contact and Contact Over Active Gate Structures," US Patent Application 20200279773, Sept. 3, 2020.

[9] K. Cheng, "Contact Over Active Gate Employing a Stacked Spacer," US Patent Application 20200066866, Feb. 27, 2020.

[10] S. C. Song, J. Xu, N. N. Mojumder, K. Kim, D. Yang, et al., "Holistic Technology Optimization and Key Enablers for 7nm Mobile SoC," Symposium on VLSI Circuits, 2015, pp. 145-146.

[11] B. Chehab, P. Weckx Sr., J. Ryckaert Sr., D. Jang Sr., D. Verkest, and A. Spessot, "Standard Cell Architectures for N2 Node: Transition from FinFET to Nanosheet and to Forksheet Device," Vol. 11328, Design-Process-Technology Co-optimization for Manufacturability XIV; SPIE Advanced Lithography, 2020.

[12] K. Miyaguchi, F. M. Bufler, T. Chiarella, P. Matagne, N. Horiguchi, A. D. Keersgieter, et al., "Single and Double Diffusion Breaks in 14nm FinFET and Beyond," International Conference on Solid State Devices and Materials, Sendai, 2017, pp219-220.

[13] R. Xie, K. Y. Lim, M. G. Sung, and R. R.H. Kim, "Single and Double Diffusion Breaks on Integrated Circuit Products Comprised of Finfet Devices," US Patent, US20170141211A1, May 18, 2017.

[14] H. Jagannathan, S. K. Kanakasabapathy, V. K. Paruchuri, and A. Reznicek, "Fin Cut Enabling Single Diffusion Breaks," US Patent, US 9589845 B1, March 7, 2017.

[15] A. Razavieh, et al., "FinFET with Contact Over Active-Gate for 5G Ultra-Wideband Applications," VLSI Symposium, JFS2.5, 2020.

[16] M. Richards, "What to Expect at 5-nm-and-Beyond and What that Means for EDA," EE Times, March 14, 2018.

[17] L. T. Clark, V. Vashishtha, L. Shifren, A. Gujja, S. Sinha, B. Cline, et al., "ASAP7-A 7-nm FinFET Predictive Process Design Kit," Microelectronics Journal, 53(2016), pp. 105-110.

[18] ASAP7 PDK, Design Rule Manual, PDK Release 1p5.

[19] V. Vashishtha, M. Vangala, and L. T. Clark, "ASAP7 Predictive Design Kit Development and Cell Design Technology Co-optimization," ICCAD, pp. 978-984, 2017.

[20] Y. D. Chung and R. B. Lin, "Engineering a Standard Cell Library for an Industrial Router with ASAP7 PDK," ISVLSI 2020, pp. 404-409.

[21] C. W. Tai and R. B. Lin, "Morphed Standard Cell Layouts for Pin Length Reduction," ISVLSI, 2019, pp.94-99.

[22] http://www.cerc.utexas.edu/itc99-benchmarks/bench.html.

[23] A. Yazdanbakhsh, D. Mahajan, H. Esmaeilzadeh, and P. Lotfi-Kamran, "AxBench: a Multiplatform Benchmark Suite for Approximate Computing," IEEE Design & Test, vol. 34, no. 2, pp. 60-68, April 2017.

[24] https://opencores.org/projects/ orsoc_graphics_accelerator.

[25] RV32IM, Github: http://github.com/ultraembedded/riscv.

[26] https://opencores.org/projects/avs_aes.

[27] B. Landman and R. Russo. "On a Pin versus Block Relationship for Partitions of Logic Graphs," IEEE Transactions on Computers, c-20:1469–1479, 1971.

[28] http://glaros.dtc.umn.edu/gkhome/metis/hmetis/overview.