Library Cell Layout with Alt-PSM Compliance and Composability

Ke Cao, Puneet Dhawan and Jiang Hu
Department of Electrical Engineering, Texas A&M University, College Station, TX 77843, U.S.A.

Abstract

The sustained miniaturization of VLSI feature size presents great challenges to sub-wavelength photolithography and requests usage of many Resolution Enhancement Techniques (RET). The difficulty and feasibility of deploying the RET such as Alternating Phase Shifting Mask (Alt-PSM) depend heavily on circuit layout. In this paper, we propose a Boolean satisfiability (SAT) based library cell layout method that can achieve Alt-PSM compliance and composability in a constructive manner. Compared to previously reported post processing approach, our method often leads to further cell area efficiency improvement.

I. Introduction

With sustained progress of the VLSI technology, the minimum transistor feature size has decreased to $90nm$ which is remarkably below the lithography wavelength of $157nm$. In the subwavelength lithography, strong diffractive effects may cause severe mismatch between mask shapes and printed shapes. Alternating Phase Shifting Mask(Alt-PSM) (Figure 1(a)) is a popular technique to alleviate this printability problem. In Alt-PSM, lights with opposite phase are shed on two sides of a thin critical feature. In the feature region, the destructive interference between the opposite-phase lights can make the printed shape sharper.

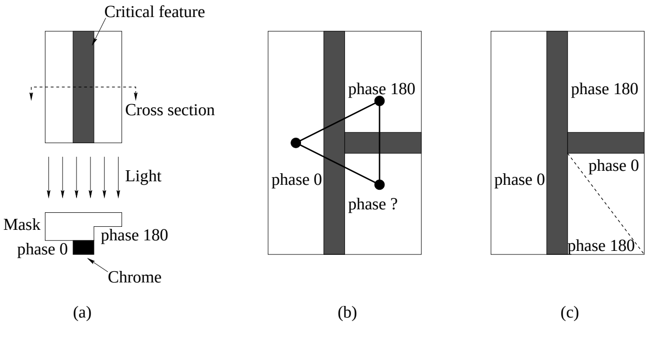

Fig. 1. (a) Alternating Phase Shifting Mask(Alt-PSM). (b) Phase conflict occurs for a T-shaped critical feature. (c) Phase conflict removal by splitting a phase region.

Even though the Alt-PSM technique is carried out in the stage of mask design and lithography, it needs to be considered in circuit layout as well. For example, a T-shaped layout like in Figure 1(b) may make phase assignment infeasible. In [1], a region where a phase conflict occurs is split to enable Alt-PSM as shown in Figure 1(c). As the phases of lights are opposite along two sides of the splitting line (the dashed line in Figure 1(c)), unwanted features may be left there and need to be trimmed by another exposure. The second exposure will increase the already expensive mask cost and cause mis-alignment risk. In [2], graph based algorithms are proposed to modify existing layout for Alt-PSM compliance. This algorithm can achieve global optimality, but demands large CPU time if applied on an entire chip layout. In standard cell designs, one speedup method is to exploit the repetition usage of library cells. The Alt-PSM compliance for each library cell does not need to be obtained repeatedly, it can be achieved once in library cell designs. However, placing Alt-PSM compliant cells adjacent to each other may cause new phase conflict. For example, in Figure 2, when the two Alt-PSM compliant cells are placed next to each other, a phase conflict happens between the two regions indicated by the arrow. In [3], a network flow based layout modification algorithm is suggested to achieve Alt-PSM composability for library cells.

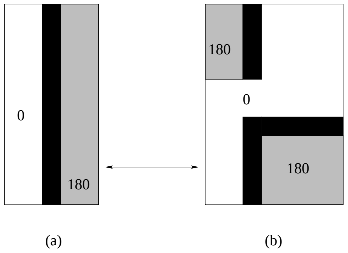

Fig. 2. The cell layout Alt-PSM composability problem.

As the works of [2, 3] are all post processings on existing layouts, they may not work well if the initial layout is poor. In this work, we propose to obain Alt-PSM compliance and composability constructively in library cell layout. Further, we consider the transistor placement and intra-cell routing at the same time so that the layout of poly layer and metal layer can match with each other. Recently, the minimum width cell layout problem is tackled[4] with Boolean satisfiability(SAT) method. In this work, we also propose to achieve Alt-PSM compliant and composable cell layout by using SAT. Generally speaking, SAT based methods can provide high quality solutions at expensive computation cost. Since a library cell is designed once and employed many times, its quality is usually more important than the runtime to generate it. We transform the constraints for the library cell lay-

0-7803-8736-8/05/$20.00 ©2005 IEEE.

out to a SAT formulation and then use Siege SAT solver[5] to search for solution. Our formulation handles multiple sized transistors and takes care of transistor folding for large devices. Experimental results show that our method often leads to further cell area reduction compared to previous post processing approach.

II. Algorithm Overview

The layout style employed in this work follows the convention of [4] and is illustrated by redrawing the example of [4] in Figure 3. The overview of our algorithm flow is shown in Figure 4. Step 1 is very similar to transistor placement in[4] except that we handle transistor folding. The PSM constraint for Poly can be enforced in either placement or routing stage. As the enforcement of this constraint depends on a known transistor placement, it is first considered in Step 2. If there are multiple solutions from Step 1, we choose one satisfying the PSM constraint for Poly. If there is no satisfactory solution from Step 1, this constraint will be handed with other constraints in the routing stage. These constraints make Poly and Metal 1 to be PSM compliant. Moreover, they ensure that the same phase can be assigned to the peripheral region of the cell so that the PSM composability is achieved.

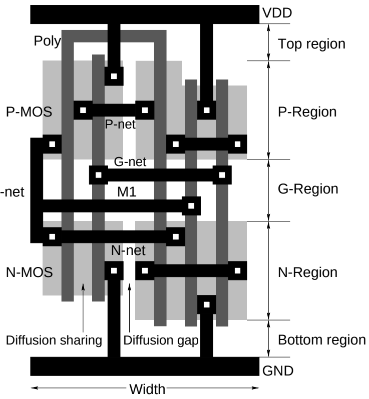

Fig. 3. Library cell layout style.

III. Transistor Placement

Following placement formulation from[4], N P-MOS and N N-MOS devices need to be placed in minimum number of columns so that the resultant placement is PSM clean and composable. Each transistor's placement is defined by a set of Boolean variables of length P, where $P = \lceil logW \rceil$, W being the number of columns. A single Boolean variable f is needed to define flip of each transistor. Total number of variables thus needed for placement is $2N(\lceil logW \rceil + 1)$. The placement constraints of overlap, unused column, vertical gate and neighboring are formulated same as [4] and will not be repeated here.

Transistor Folding Constraints: Smaller transistors resulting from splitting MOSFETs larger than maximum allowable width should be placed together:

$$\bigwedge_{k=0}^{W-B} [C_n(i,k) \wedge C_n(i+1,k+1) \wedge ....C_n(i+B-1,k+B-1)] = 1$$

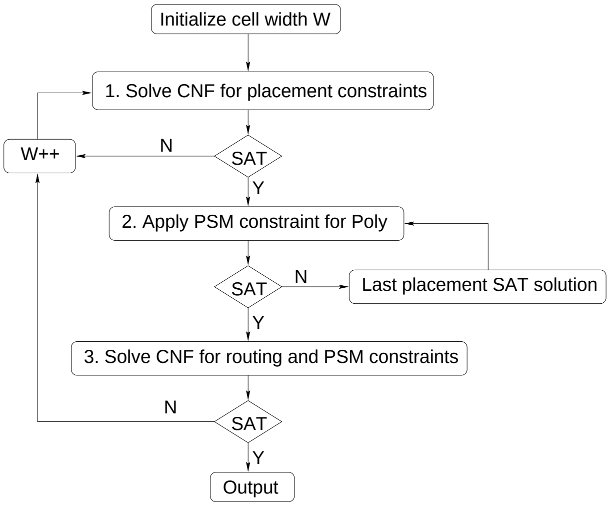
Initialize cell width W
1. Solve CNF for placement constraints
W++
N
SAT
Y
2. Apply PSM constraint for Poly
SAT
N
Last placement SAT solution
Y
3. Solve CNF for routing and PSM constraints
N
SAT
Y
Output

Fig. 4. PSM driven library cell layout algorithm flow.

where B is the number of fingers a larger device is split into. Flip of neighboring folded transistors should be opposite :

$$(f(i) \oplus f(i+1)) \wedge (f(i+1) \oplus f(i+2)) \wedge ...(f(i+B-2) \oplus f(i+B-1)) = 1$$

PSM Constraint for Poly: Since the cell layout style is very regular, the PSM constraint is necessary only for a few pre-identified scenarios. For Poly layout, the PSM constraint requires that the distance between any two transistors with connected gates should not be even if all columns between them are occupied. For the example in Figure 5(a), the transistors on two sides are connected with Poly at the top and the middle column is occupied by another transistor, thus, a phase error occurs. In Figure 5(b), there is no phase error due to the empty column in the middle. Since it is not known if the columns in-between are all occupied before the placement is finished, this constraint is applied after the placement as Step 2 in Figure 4. This constraint is optional in Step 2 of Figure 4, since the phase error such as in Figure 5(a) can be avoided through Metal 1 routing later as shown in Figure 5(c). However, if there are multiple solutions from placement, we choose the one satisfying this constraint so that there will be less constraint for the subsequent routing stage. This constraint can be expressed as following for SAT formulation.

$$\overline{Q_{ij}} \vee (n_{ip} \oplus n_{jp}) = 1$$

where $Q_{ij} = 1$ if the gates of transistor i and j are connected, 0 otherwise. n_{ip}/n_{jp} is the least significant bit of placement vector of NMOS i/j. Similar constraint is applied for PMOS devices.

IV. Routing

The routing style is similar to [4] except that the Poly connections are handled differently for the concern of PSM. A connection between two gate terminals is called a Poly connection if the net is placed in Top/Bottom Region, a G-Net if in G-Region. The PSM related con-

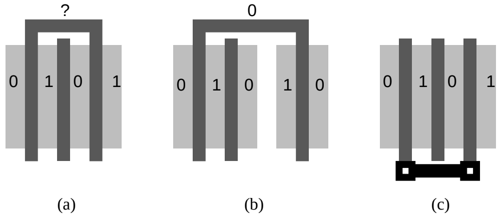

Fig. 5. PSM constraint for Poly. '?' indicates phase conflict.

straints are described below, other constraints are the same as [4].

Poly/G-Net Connection Constraints: Poly/G-Net connection between two gate terminals is considered as a single instance. Let x_{ij} represents the variable that assumes value of 1 if a gate connection between i and j is necessary, otherwise 0. Let a_{ij} be 1 if gate connection between i and j uses the G-Region, otherwise 0. All instances in Poly/G-Net with multiple gate terminals are treated as separate connections. The terminals in a net should be connected by minimum connections without redundancy. For example in a net with gate terminals i, j and $k (i < j < k)$that need to be connected together. There should be no redundant connections:

$$x_{ij} \oplus x_{jk} \oplus x_{ik} = 1$$

Terminals i, j and k should be connected together:

$$(x_{ij} \wedge x_{jk}) \vee (x_{ij} \wedge x_{ik}) \vee (x_{ik} \wedge x_{jk}) = 1$$

Similar constraints can be written for nets with more than 3 connections. Variable x_{ij} forms part of the solution space signifying which gate-gate connection instances are finally used.

V-Net Pass Constraints: A V-Net along with all its gate terminals should lie on same side of another V-Net's source-drain column. This is represented by:

$$G_{i,j} \oplus G_{ik,j} = 0; k \in V_{ig}$$

where $G_{i,j}$ is the relative position of source-drain column of V-Net i w.r.t. source-drain column of V-Net j, i.e., $G_{i,j}$ is 0(1) if column for j is to the left(right) of the column for i. $G_{ik,j}$ is the relative position of k^{th} gate terminal in V-Net i w.r.t. V-Net j. V_{ig} is the set of gate terminals in V-Net i.

A G-Net with terminals say $i \leftrightarrow j$ should not overlap the source-drain column of any V-Net k i.e.

$$x_{ij} a_{ij} (G_{i,j} \oplus G_{j,k}) = 0$$

where $G_{i,j}$ is 0(1) if column for j is to the left(right) of the column for i.

If a gate connection $i \leftrightarrow j$ presents in G-Region, it should not overlap any gate terminal connection of a V-Net v, i.e.

$$x_{ij} a_{ij} C_{ij,kv} \overline{[(g_{ij,1} \oplus g_{k,1}) \vee ....(g_{ij,M} \oplus g_{k,M})]} = 0 \forall k \in V_{vg}$$

where $C_{ij,kv}$ is 1 if interval $i \leftrightarrow j$ overlaps with interval $k \leftrightarrow v$, 0 otherwise, $g_{ij,1}..g_{ij,M}$ constitute the bit vector defining the row of net $i \leftrightarrow j$.

If connection to gate terminals of two V-Nets overlap, it should not be placed in same row in G-Region. Any V-Net i with gate terminal j should not overlap another V-Net l with gate terminal k :

$$(G_{i,k} \oplus G_{j,k}) \overline{[(g_{ij,1} \oplus g_{kl,1} \vee ..(g_{ij,M} \oplus g_{kl,M}))]} = 0 \forall j \in V_{ig}, k \in V_{lg}$$

If any N/P Net with left right terminals i and j is placed in row 0 of N/P-Region, it should not overlap the columns occupied by a V-Net y with source/drain connection terminals $k \leftrightarrow l$ in row 0 of N/P region:

$$N_{ij} [(G_{i,y} \oplus G_{j,y}) \vee C_{ij,kl}] = 0$$

where $N_{ij} = 1$, if N/P net is in row 0 of N/P region, 0 otherwise. $G_{i,y}, G_{j,y}$ are relative position of N/P net terminals $i \leftrightarrow j$ w.r.t. V-Net y. $C_{ij,kl} = 1$, if interval $(i \leftrightarrow j) \bigcap (k \leftrightarrow l) \neq \phi$

Two V-Nets i and j should not overlap in N/P Region.

$$G_{i,j} \oplus G_{ik,j} = 0; Iv_i \cap Iv_j = \phi$$

where k $\in$ Source/Drain Terminals of V-Net i. Iv is the source/drain terminal connection interval[4] of V-Net.

Poly Routing Overlap Constraints: Two poly instances of different gate-gate connections with terminals $i \leftrightarrow j$ and $k \leftrightarrow l$ placed in Top/Bottom of G Region can not overlap:

$$x_{ij} x_{kl} \overline{a_{ij}}\ \overline{a_{kl}}\ \overline{[(g_{ij,1} \oplus g_{kl,1}) \vee ...(g_{ij,M} \oplus g_{kl,M})]} = 0$$
$$I_{poly}(i,j) \cap I_{poly}(k,l) \neq \phi$$

where g_{ij}/g_k define the placement of a poly connection. I_{poly} is the interval of poly connection.

Two same poly net instances of same gate-gate connection with terminals $i \leftrightarrow j$ and $k \leftrightarrow l$, both can not be placed in either top or bottom of G region to avoid T connections that lead to PSM phase conflicts:

$$x_{ij} x_{kl} \overline{a_{ij}}\ \overline{a_{kl}}\ \overline{[(g_{ij,1} \oplus g_{kl,1}) \vee ...(g_{ij,M} \oplus g_{kl,M})]} = 0$$

Poly PSM Phase Error Constraint: If all columns between two poly net instances are occupied by transistor gates, the number of polys in such interval should be even.

$$x_{ij} x_{kl} \overline{a_{ij}}\ C_{ij} \overline{(l_b \oplus r_b)} = 0$$

$C_{ij} = 1$ if all columns between i and j are occupied, 0 otherwise. l_b/r_b are the least significant bits of left/right column of a poly connection. This constraint is same as the PSM Constraint in Placement stage, but since the devices have already been placed, now the occupancy of connected gates is known.

PSM Composability Constraint: For any two poly net instances $i \leftrightarrow j$ $(i < j)$ and $k \leftrightarrow l$ $(k < l)$, the column number of one poly instance should have same least significant bit as other instance, if all the intervals between them are occupied.

$$x_{ij} \overline{a_{ij}} x_{kl} \overline{a_{kl}} \wedge [C_{ik}(b_i \oplus b_k) \vee C_{il} \overline{(b_i \oplus b_l)}$$
$$\vee C_{jk} \overline{(b_j \oplus b_k)} \vee C_{jl}(b_j \oplus b_l)] = 0$$

where b_i repressents the least significant bit of the column variable for transistor i.

Metal 1 Layer PSM Constraint: To avoid T-type connections that introduce phase assignment problem in

TABLE I
Experimental Results

	Non PSM		Our PSM aware layout						Post Processing
Cell	# cols	CPU	# tran	# Pvar	# Rvar	# cols	CPU	% area inc	% area inc
and8	9	0.70	16	90	12	9	0.81	0.00	0.00
xnor	6	0.21	10	40	18	6	0.30	0.00	5.90
aoi22	6	0.02	10	40	7	6	0.05	0.00	0.00
nd2ab	5	0.12	8	32	17	5	0.32	0.00	0.00
nor2	6	0.16	12	48	21	6	0.23	0.00	7.10
mux2	7	0.43	10	40	23	7	0.54	0.00	5.35
nand3	5	0.08	10	40	14	5	0.12	0.00	12.36
xor2	8	0.43	12	48	30	8	0.58	3.12	6.10
cgi2	6	0.06	10	40	18	6	0.08	0.00	10.2
nor8	8	0.51	18	64	18	8	0.53	0.00	0.00
nanf21	5	0.05	10	40	17	5	0.09	0.00	6.810
blf41	7	0.42	12	48	16	7	0.53	0.00	6.10

Metal 1 layer, the width of T's top can be increased to make it a non critical feature, thus providing conflict free phase assignment to T's trunk. In V-Nets such situations are very likely. To overcome this, if a V-Net has a T connection, the width of T's top is increased beyond critical feature width. To provide enough space and design rule compliance, both columns in vicinity of such V-Nets can not be occupied. At least one column should be empty:

$$\overline{V_{c-1} \wedge V_{c+1}} = 1$$

where $V_{c-1} = 1$ if the column before V-Net source-drain column is occupied, 0 otherwise. $V_{c+1} = 1$ if the column after V-Net source-drain is occupied, 0 otherwise. However if V-Net with T connection is placed at column 0, column 1 should be unoccupied and similarly if in last column, last but one should be empty.

V. Experiments

The constraints for Placement and Routing described above are transformed into our SAT formulations and solved using Siege Variant 4[5] SAT Solver. Siege was observed to be faster and more efficient for this problem than other common SAT solvers like ZChaff and Berkmin561. Another major advantage of Siege being its output dependence on input seed. This helps to avoid adding additional clauses at each placement run to suppress an existing placement result in order to get a new one. Our formulation has been implemented in 'C++' and experiments were conducted on Dell PE2650 Linux machine with Xeon 2.4GHz processor and 2GB of RAM. A timeout of 3000s was used in our experiment.

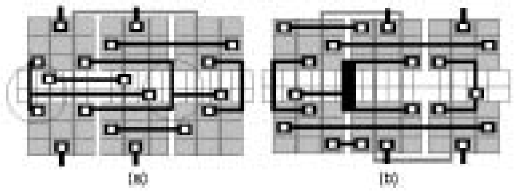

Fig. 6. (a) XOR gate layout: T joints in Metal 1 layer that pose phase conflict problem. (b) PSM clean layout.

The experimental results are summarized in Table 1. The results of cell layout without considering PSM are shown in column 2 and 3. The *#cols* in the layout indicates the cell area size. The number of rows in N/P region are assumed to be 3, for G region 2 and one row each in Top and Bottom Region. The number of transistors for each cell is listed in column 4. The number of variables for placement and routing are in column 5 and 6, respectively, for our PSM driven layout. The final area of our method is in column 7. The percentage increase over the non-PSM version is in column 9. We can see that we can usually achieve PSM compliance and composability without area increase. If we perform post-processing[2, 3], on the non-PSM layout result, there are more area increase as shown in the right-most column of Table 1. Figure 6 shows a comparison of layouts with and without PSM considerations in placement and routing phases for a 2-input XOR gate.

VI. Conclusion

We propose a new approach to handle PSM Compliance and Composability problem due to decreasing feature size. Our SAT based formulation for library cell layout is PSM aware and is a step in direction to integrate manufacturability concerns in the design process. This work results in minimum width layout that are free of phase conflicts both within and between library cells, and of comparatively less area as compared to those obtained from post processing of layouts for any phase assignment problems.

References

[1] C. Pierrat, M. Cote, and K. Patterson. New alternating phase-shifting mask conversion methodology using phase conflict resolution. *Proceedings of SPIE*, 4691(31), 2002.

[2] A. B. Kahng, S. Vaya, and A. Zelikovsky. New graph bipartizations for double-exposure, bright field alternating phase-shift mask layout. In *Proc. of ASP-DAC*, pages 133–138, 2001.

[3] K. Cao, J. Hu, and M. Cheng. Layout modification for library cell alt-psm composability. In *Proceedings of SPIE International Symposium on Microlithography*, 2004.

[4] T. Iizuka, M. Ikeda, and K. Asada. High speed layout synthesis for minimum-width CMOS logic cells via Boolean satisfiability. In *Proc. of ASP-DAC*, pages 149–154, 2004.

[5] E. Zarpas, Siege vs. zChaff and Berkmin561 on BMC Formulas from the IBM Formal Verification Benchmarks Library, July 2004. *http://www.haifa.il.ibm.com/projects/verification/ RB_Homepage/bmcbenchmarks_illustrations.html*