Abstract
A new area optimizing transistor placement algorithm for CMOS complex gate layout synthesis is presented. The algorithm handles dual planar CMOS circuits and regards cell width reduction by systematical jog insertion. A study comprising 3503 series-parallel and 423 non-series-parallel CMOS circuits shows, that cell area can be reduced by up to 52% through placement optimization, compared to previous approaches.

1 Introduction

Complex gates (also referred to as functional cells) implement multilevel logic in a single gate. The use of complex gates reduces the overall circuit size and often improves performance compared to circuits using more primitive gates like NAND and NOR alone. A larger library of cells or the means of on-the-fly generation of complex gates gives circuit synthesis systems a higher degree of freedom in optimizing chip area, gate count and power consumption of digital circuits.

The primary layout optimization objective at cell level is to find an appropriate transistor placement which minimizes the required amount of wiring. In their seminal paper, Uehara and vanCleemput [Uv81] proposed a transistor placement algorithm for static CMOS layout generation in a row-based layout style, which we will refer to as the traditional one-dimensional gate-matrix (TGM) layout style here. A considerable amount of research has been conducted to improve transistor placement while essentially maintaining the layout style, e.g. [MD88, CH88, MH92]. The jogged gate matrix (JOGM) layout style, proposed by [Hi93], modifies some layout style assumptions by systematically inserting jogs into transistor gates.

Our transistor placement algorithm takes into account the physical layout structure dimensions derived from the design rules of the target fabrication process. Thus, a physical width-to-height trade-off can be considered which for the first time enables true area optimization. Moreover, arbitrary dual planar circuits can be handled.

Section 2 introduces the layout optimization problem and describes the impact of transistor placement on cell dimensions. Section 3 presents our transistor placement approach. Experimental results are provided in section 4.

2 Layout optimization problem

2.1 Basic assumptions

We make some assumptions on the circuit structure and the one-dimensional gate-matrix layout style:
(a1) Static CMOS complex gates are used. The PMOS and NMOS part of the circuit are each others dual.
(a2) Transistors are placed in two parallel horizontal rows, one for PMOS and one for NMOS.
(a3) Complementary transistor gate terminals are placed in the same order in the PMOS and NMOS row.

Former work used a more restrictive version of (a3) which assumed equidistant transistor gate spacings and strict alignment of PMOS and NMOS transistor gates.

2.2 Cell width

Due to assumptions (a2) and (a3) cell width optimization in a one-dimensional gate-matrix layout style translates to finding an input signal order that maximizes diffusion abutment of adjacent transistors. Hence width-increasing diffusion gaps are avoided. Fig. 1a shows a width optimal TGM layout containing no diffusion gaps. Many width-only optimization algorithms have been presented, e.g. in [Uv81, NB85, MH92].

2.3 Cell area

In full custom layout, cell area usually is the primary optimization goal. This requires width and height optimization. A simple measure for cell height is the number of horizontal metal tracks required for intra-cell routing. Fig. 1b shows an area optimal TGM layout implementing the same function as in Fig. 1a. Cell width is increased due to the inserted diffusion gap but the number of required metal tracks is decreased by 2, illustrating the width-to-height trade-off situation. If the cell is designed for use in a standard-cell like environment, where cell height is uniform, height reduction can be turned into increased cell transparency (i.e. free horizontal metal tracks), easing over-the-cell routing and providing a higher transistor channel width variation range.

The cell area minimization problem for series-parallel circuits is known to be NP-hard [Ch91]. Area minimizing algorithms have been reported, e.g. in [MD88, MH92].
2.4 Netlist reordering

Consideration of all distinct netlists implementing a given switching function with the same number of transistors, generally referred to as reordering, often yields smaller layouts. This is not surprising, since the suitability for layout optimization obviously depends on the netlist structure. Fig. 1 displays two distinct netlists, where the corresponding layout optimization potential differs. However, reordering also has an impact on circuit performance [CC92].

2.5 Problem classification

Basically, we adopt the problem classification scheme of [MH92], which uses the letters W, H and R to indicate Width and Height minimization and the employment of netlist Reordering. We additionally introduce the letter A to denote true Area optimization. This leads to the following problem classification (all problems may be modified by appending an 'R' to denote reordering):

W Width-only minimization. The optimal layout is one of minimal width.

WH Width before height minimization. The optimal layout is one of minimal height among all layouts of minimal width.

A True area minimization. The optimal layout is one of minimal area. Width is not necessarily minimal, since a width-to-height trade-off can occur.

3 Transistor placement algorithm

3.1 General strategy

Our approach is a branch-and-bound strategy using a depth-first tree search. A placement is constructed incrementally from left to right by a recursive procedure which alternately appends transistors \( t_p \) and \( t_n \) to two sequences, representing the PMOS and NMOS row respectively (see pseudo code in Fig. 2). After each appendage of a transistor, a cost \( c_{\text{bound}} \) is calculated which is a lower bound on the cost of any layout that can be constructed by completing this partial placement. If \( c_{\text{bound}} \) is not greater than the cost \( c_{\text{best}} \) for the best complete layout found so far, the procedure continues recursively at the next position. Otherwise, the last appendage is undone and the next candidate is tried.

Transistors which abut at the current end of the respective sequence are preferred in statements S1 and S3 (Fig. 2) to achieve fairly good placements as early as possible, which is essential for efficacious search tree pruning. Through S2, we can exploit the multiple occurrence of a gate input signal for placement optimization. In this case, complementary transistors are not necessarily placed at the same position in the p- and n-sequences.

The placement algorithm can be easily adapted to our variety of optimization goals by using a different cost calculation. This is done by separately estimating lower bounds for cell width and height (see Tab 1).

![Fig. 1: Two CMOS complex gate layouts and distinct netlists implementing OUT = ¬((c*(def + ghi) + ab + jklm))]
3.2 Layout structure dimensions

For true area (A) optimization, we have to measure cell width and cell height precisely to permit width-to-height trade-off consideration during transistor placement. Hence, some exact physical layout structure dimensions of transistors, diffusion separations and horizontal metal wires have to be taken into account.

Cell width is influenced by the physical horizontal transistor lengths \( L_T \) and diffusion separation lengths \( L_{DS} \). They are measured as the horizontal distance between the respective source and drain contact centers, as illustrated in Fig. 3. If the layout style relies on strict vertical alignment of complementary transistor gates (i.e. TGM), a diffusion separation in either row propagates its horizontal length \( L_{DS} \) to the opposite row (Fig. 3a), leading to a rather large \( L_T \). If the layout style uses a transistor grid but relaxes the strict vertical alignment, the cumulated horizontal length \( L_T(i) \) of \( i \) transistors with no diffusion separation is given by \( i \cdot L_T(1) \) (Fig. 3b). If the transistor grid is omitted and systematical jog insertion after [Hi93] is employed, the horizontal transistor length is further reduced in a predictable way, since all diagonal gate segments possess the same horizontal length (Fig. 3c). The layout structure dimension settings adapt our algorithm to the layout style.

3.3 Cell width calculation

We calculate the respective costs \( c_{p\_row} \) and \( c_{n\_row} \) of the PMOS and NMOS rows by separately summing all horizontal lengths of transistors and diffusion separations \( L_T(i) \) and \( L_{DS} \) in the respective row. Cell width is then given by

\[
\text{cell width } = \max\{c_r | r \in \{p\_row, n\_row\}\}.
\]

A row with no diffusion separation is of constant width which depends only on layout dimensions. The width cost \( c_r \) of such a row \( r \) is denoted \( c \). Whenever a diffusion separation is inserted into the row, a penalty cost \( c_p \) is added. \( c_r \) can then be computed as

\[
c_r = c + \sum c_p.
\]

So, in fact, we minimize the sum of penalty costs in the rows while \( c_r \) is only needed to determine the width-to-height trade-off during computation of \( c_{\text{bound}} \). A penalty cost arises, if the transistor selected in S1 or S3 cannot be abutted with the transistor at the end of the sequence because of assumption (a3). This forces the insertion of a diffusion separation. Two penalty costs \( c_p \) are distinguished, depending on the respective circumstances:

- A diffusion separation is inserted between two transistors which require a contact between them. This occurs where the diffusion terminal at the current sequence end is part of a net with more than two terminals. The penalty is the cost of the required diffusion separation:

\[
c_p = L_{DS}.
\]

- A diffusion separation is inserted between two transistors which do not require a contact between them (Fig. 3c, position X). This separation splits a net of two terminals which could otherwise be abutted at very little cost. The penalty is the cost difference between two transistors plus a diffusion separation and two abutted no-contact-requiring transistors:

\[
c_p = 2 \cdot L_T(1) + L_{DS} - L_T(2).
\]

3.4 Cell height calculation

Cell height mostly depends on the number \( m \) of horizontal metal tracks required for intra-cell routing. Additionally the metal wiring pitch derived from the target process physical design rules is taken into account to precisely determine width-to-height trade-off. Height is calculated as

\[
\text{cell height } = m \cdot \text{pitch}.
\]

\( m \) is determined by a simple horizontal sweep across the row, looking for the maximum number of metal segments crossing the same column. For a partially constructed row a possible extension of a metal track over the current right end of the row must be detected in order not to
underestimate \( m \). For each net for which a metal track has been allocated, we determine whether there still remain currently unplaced transistors to be connected to that net. If this is the case, the respective metal track will extend over the current right end of the row to reach the remaining transistors.

4 Experimental results

4.1 Complex gate benchmarks

**SP_4_4**: This is the set of all 3503 circuits of distinct functionality that can be implemented with a series-parallel (SP) transistor network using no more than 4 in-series connected transistors in the longest pull-up/down path between output and power supply nodes of the gate. These circuits have been considered to be of 'practical size'.

**NSP_4_4**: This set of 144 circuits is defined like SP_4_4 except that it contains non-series-parallel (NSP) networks. These circuits were manually derived by Maziasz and Hayes [MH92].

**NSP_HC**: A concise table of circuits implementing all switching functions with four inputs with the respective minimal number of transistors (also referred to as Harvard Circuits) has been presented by Ninomiya [Ni61]. 302 of them are non-series-parallel but 23 of the latter are non-s/t-planar and therefore do not possess a dual counterpart [La76]. The remaining 279 circuits constitute the NSP_HC circuit set. In some of them, the same input appears more than once on the pull-up or pull-down path. Hence, the NSP_HC circuits contain up to 6 in-series connected transistors.

4.2 Benchmark results

The following table shows the scope of the transistor placement algorithms presented in [MH92] and in this paper.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>W</th>
<th>WH</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>[MH92]</td>
<td>yes</td>
<td>yes</td>
<td>yes no no</td>
</tr>
<tr>
<td>ours</td>
<td>yes</td>
<td>yes</td>
<td>yes yes yes</td>
</tr>
</tbody>
</table>

Tab. 2: Problems addressed by transistor placement algorithms for SP and NSP CMOS circuits.

[MH92] have proposed \( R^- \), \( P^- \) and \( HR-TrailTrace \) addressing the W problem for SP and NSP and the WH problem for SP circuits. They compared extensively the results of their algorithms to previous approaches, yielding equal or superior results in virtually all cases. Hence, we will restrict our comparison to these.

**Series-parallel CMOS circuits**: As stated before, netlist reordering can considerably reduce cell area. However, if the circuit structure was designed with respect to performance by a circuit synthesis system like [At93], reordering is likely to be constrained. Therefore, in contrast to \( R^- \) and \( HR-TrailTrace \), we split reordering and transistor placement into separate tasks, which allows a circuit synthesis system to provide a set of 'electrically acceptable' circuits. For comparability, we have computed all distinct transistor netlists for the SP_4_4 benchmark, which took only 10 user-CPU seconds on a Sparc IPC. The maximum number of netlists found for a single function was 144, with the average being 19.

In order to compare WHR- and AR-optimization, TGM layout structure dimensions were assumed. Whereas [MH92] believed that the WHR-Problem is synonymous to the AR-Problem in most or all cases (based on an investigation of 30 circuit examples revealing no differences), our analysis disproves this conjecture. Tab. 3 shows, that for 726 (21%) of the SP_4_4 circuits such a trade-off situation evolves. The maximum area improvement was 39%, with an average of 11.6% for all improvable layouts.

<table>
<thead>
<tr>
<th>Height decrease</th>
<th>Width increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2777</td>
</tr>
<tr>
<td>1</td>
<td>554 14</td>
</tr>
<tr>
<td>2</td>
<td>126 28</td>
</tr>
<tr>
<td>3</td>
<td>2 2</td>
</tr>
</tbody>
</table>

Tab. 3: WHR- vs. AR-optimization of SP_4_4 circuits, layout width-to-height trade-off in terms of diffusion separations and metal tracks.

**Non-series-parallel CMOS circuits**: Previous approaches did not address the WH- or A-optimization problems for non-series-parallel circuits. Thus, the respective height and width distributions for the NSP_HC and NSP_4_4 circuits (tables 4 and 5) could not be obtained before. Since WH-optimization yields less width and A-optimization less height, the distributions for both methods are included.

<table>
<thead>
<tr>
<th>Number of tracks</th>
<th>NSP_HC circuits</th>
<th>NSP_4_4 circuits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WH</td>
<td>A</td>
</tr>
<tr>
<td>5</td>
<td>104</td>
<td>140</td>
</tr>
<tr>
<td>6</td>
<td>149</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>44</td>
<td>26</td>
</tr>
<tr>
<td>8</td>
<td>52</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

Tab. 4: Height distribution for WH- and A-optimized NSP_HC and NSP_4_4 layouts in terms of metal tracks.
Tab. 5: Width distribution for WH- and A-optimized NSP_HC and NSP_4_4 layouts in terms of diffusion separations.

<table>
<thead>
<tr>
<th>Number of d. sep.</th>
<th>NSP_HC circuits</th>
<th>NSP_4_4 circuits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WH A</td>
<td>WH A</td>
</tr>
<tr>
<td>0</td>
<td>76 52</td>
<td>16 12</td>
</tr>
<tr>
<td>1</td>
<td>187 127</td>
<td>126 98</td>
</tr>
<tr>
<td>2</td>
<td>16 60</td>
<td>2 34</td>
</tr>
<tr>
<td>3</td>
<td>- 38</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>- 2</td>
<td>-</td>
</tr>
</tbody>
</table>

Tab. 6: WH- vs. A-optimization of NSP_HC/NSP_4_4 circuits, layout width-to-height trade-off in terms of diffusion separations and metal tracks.

<table>
<thead>
<tr>
<th>Height decrease</th>
<th>Width increase</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0  1  2</td>
</tr>
<tr>
<td>0</td>
<td>162/108</td>
</tr>
<tr>
<td>1</td>
<td>- 39/10</td>
</tr>
<tr>
<td>2</td>
<td>- 39/26 11/-</td>
</tr>
<tr>
<td>3</td>
<td>- - 1/-</td>
</tr>
<tr>
<td>4</td>
<td>- 6/- 14/-</td>
</tr>
<tr>
<td>5</td>
<td>- - 7/-</td>
</tr>
</tbody>
</table>

For these non-series-parallel circuits, no reordering was performed due to the lack of an appropriate method. However, reordering could be done by manually entering the set of reordered netlists for each non-series-parallel circuit.

Tab. 6 displays the width-to-height trade-off situation in the NSP_HC/NSP_4_4 circuit sets. In 42%/25% of the circuits, a trade-off occurred, achieving up to 52%/24% area reduction (average of improvable: 20%/16%).

The A-optimization method usually prefers cell height if applied to large circuits, because cell width is greater than cell height for these circuits. The WH-optimization method tends to run into local optima in preferring width first, leading to a remarkable height increase in quite a few cases. The overall maximum height is reduced by as many as 4 (!) tracks for the NSP_HC circuits and 2 tracks for the usually smaller NSP_4_4 circuits. Therefore, A-optimization is suitable for layouts to be used within a standard-cell like environment, where cell height of the tallest cell may determine the height of the entire row.

**Effect of transistor gate alignment and gridding:** Fig. 4a shows the WH-optimized layout of NSP_HC circuit no. 102 with strictly aligned and gridded transistors. Dropping the alignment restriction (implying that diffusion separations in the P- and N-part of the layout may be at distinct positions) leads to a solution with cell height reduced by 4 tracks (Fig. 4b) at no width increase. Additionally, further width reduction arises from the employment of systematical jog insertion into transistor gates (Fig. 4c) using a non-gridded layout style. However, the layout has a slightly increased height compared to Fig. 4b. This is a side effect of JOGM layout generation, which can occur if rather small transistors are used. The amount of height increase is not predictable during transistor placement.

### 4.3 Program statistics

The transistor placement was implemented in the C++ programming language under Linux and SunOS. Tab. 7 summarizes the execution times for the SP_4_4, NSP_4_4 and NSP_HC benchmarks. Assuming, that a Sparc IPC is about 20 times faster than the µVAXII used by [MH92], our algorithm performs equally to the R-, HR- and P-TrailTrace algorithms in terms of execution time.

As surveyed in [MH92], former work believed this exhaustive approach to be impracticable because of the potentially large number of placements which can be formed. Indeed, in the worst case, all input signal orders may be examined. However this approach guarantees the best possible results and our experiments show, that all
cells of practical size are constructed within reasonable computation time. The key to computational efficiency is the way of search tree traversal and pruning.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Benchmark</th>
<th>W</th>
<th>WH</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP_4_4</td>
<td>[MH92] ours</td>
<td>30h</td>
<td>220h</td>
<td>8.5h</td>
</tr>
<tr>
<td>NSP_4_4</td>
<td>[MH92] ours</td>
<td>232s</td>
<td>-</td>
<td>19.4s</td>
</tr>
<tr>
<td>NSP_HC</td>
<td>ours</td>
<td>-</td>
<td>-</td>
<td>500s</td>
</tr>
</tbody>
</table>

Tab. 7: Benchmark execution times.

[MH92] on µVAXII (~ 1MIPS), ours on Sparc IPC (~20 MIPS).

5 Conclusion and future work

A transistor placement algorithm using a branch-and-bound strategy was presented for dual planar CMOS complex gate layout optimization. Being part of a layout synthesizer for non-gridded complex gate layout, it takes into account physical design rules as well as some layout style assumptions. Depending on the designation of the cell layout, preference can be given to width or area optimization. A comprehensive study of virtually all practical sized CMOS circuits shows, that results equal or superior to previous approaches in terms of cell area are derived in affordable computation time.

Future work will concentrate on a module generator for non-gridded CMOS cell layout.

Acknowledgment

The authors would like to thank Dr. Maziasz for providing the 144 non-series-parallel netlists.

References


