Transistor Placement for Custom Datapath Cells

Serkan Askar Maciej Ciesielski
Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA 01003
{saskar, ciesiel}@ecs.umass.edu

Abstract

This paper addresses the problem of layout generation for datapath cells. It specifically deals with the final phase of *transistor placement* for near-custom quality datapath layout. The paper describes an important improvement to our earlier approach by addressing the routability and manufacturing issues. Both of these problems are considered in the final *post-processing* step, which modifies the computed placement by applying a series of geometry-invariant transformations, such as component mirroring and swapping. The metrics for these transformations include wiring length minimization and the minimization of the number of wells. The new results indicate that the previously reported results of geometric placement, already within acceptable area overhead, can be further improved in terms of wiring length and the number of wells.

I. Introduction

Layout generation of datapaths, characterized by highly regular structures, reduces to a careful design, often by means of hand-crafting, of individual datapath cells. This paper addresses the problem of layout automation for datapath cells on the transistor level, and specifically deals with the *transistor placement* problem. Although placement and routing are tightly interrelated, they are often performed as separate design steps in order to manage the design complexity. Another factor which justifies the separation of placement and routing is the availability of multi-layer routing technology that makes it possible to route layout structures within the diffusion-limited placement. The goal is to automate the layout design for datapath cells with near-custom quality results.

Our approach to near-custom quality transistor placement for datapath cells is based on an analytical model using mathematical programming techniques. In our earlier work [1] we presented an efficient approach to transistor placement based on a mixed integer linear programming (MILP) technique and demonstrated that it can be applied to practical datapath designs. While MILP has been previously used for floorplanning and placement [2] [3] [4] [5] [6], the novelty and originality of our method was in the efficient management of the complexity of the underlying mathematical model. In particular, the number of integer variables modeling the relative placement of transistors was drastically reduced by means of transistor merging and hierarchical grouping, making it possible to design real industrial circuits. However, the proposed model did not take into account the important issues of routability and manufacturability; routability was addressed only marginally by considering connectivity between transistors during the initial relative placement phase.

This paper describes an important improvement to our earlier work [1] by addressing the routability and manufacturing issues, and specifically the minimization of wire length and wells. Both of these problems are considered in the *post-processing* step which modifies the computed geometrical placement by applying a series of geometry-invariant transformations, such as mirroring and swapping of components. The metrics for these transformations include total (critical) wiring length minimization and, for single-well technology, the minimization of the number of wells.

The organization of the paper is as follows. Section II briefly reviews the approach taken in our earlier work and describes a number of modifications to the original problem, which allows to further reduce the number of variables in the mathematical formulation of geometrical placement. Section III then concentrates on the global formulation of the post-processing problem, including component mirroring for net-length and well minimization. Finally Section IV presents some results and conclusions.

II. Modifications to previous approach

As indicated in our previous work [1], our approach to custom datapath synthesis has been

This work has been supported by a grant from Compaq, Alpha Development Group

divided into the following five phases: component merging, initial relative placement, component grouping, geometric placement, and post-processing, shown in Fig 1. We improved our formulations and algorithms in some of these steps to handle the complexity of the problem more efficiently. The following is a brief overview of these modifications.

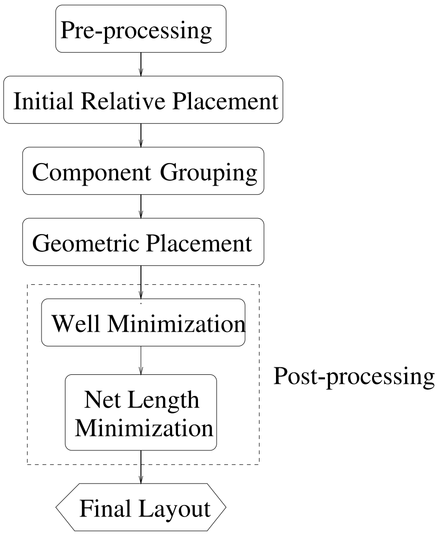


Fig. 1. Datapath synthesis flow

A. Component Merging

Identifying diffusion sharing between the components in a datapath cell is a common custom design technique that aims at minimizing the overall fanout capacitance. Our new method identifies the components with common diffusions and selects the best candidates for merging based on their dimensions. When doing this, our procedure examines several instances for each component by allowing transistor folding. It is also capable of identifying components that can be merged by their poly lines, as long as these lines have the same ordering. This is shown in Fig. 2, where one of the transistors have to be mirrored to match the ordering of poly lines. Such a merging will reduce the overhead of metal-to-poly contacts needed for routing.

B. Geometric Placement

In this phase a non-overlapping placement of components is generated using MILP. Here we describe the modifications in the basic component model and the new model for the L- and T-shape components. These new models contribute to the reduction of the number of variables in the adopted mathematical formulation of geometrical placement.

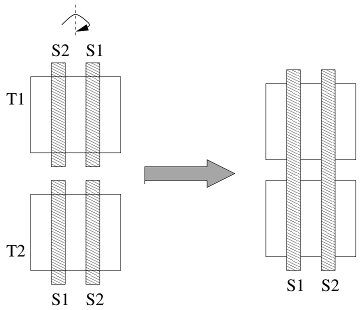


Fig. 2. Components identified for poly merging (T1 must be mirrored)

Component Modeling: In contrast to our original formulation, where each placeable component was represented as a pair of spatially related squares, here each rectangular component i is modeled as a single box whose coordinates of the top/right corner are denoted by a pair (X_i, Y_i). Both the number of equations and the number of variables used to describe the non-overlapping placement constraints are reduced roughly by half in this model.

$$X_j - X_i \geq DimX_j + L \cdot (1 - Q_{ij}) + \triangle_{sep} \quad (1)$$
$$Y_j - Y_i \geq DimY_j + L \cdot Q_{ij} + \triangle_{sep} \quad (2)$$

Here Q_{ij} is the positional relation binary variable for the pair (i, j), $\triangle_{sep}$ is the minimum required separation between the components, L is a large value, and

$$DimX_j = \begin{cases} W_j & \text{if } R_j = 1 \\ H_j & \text{if } R_j = 0 \end{cases} \quad (3)$$

$$DimY_j = \begin{cases} H_j & \text{if } R_j = 1 \\ W_j & \text{if } R_j = 0 \end{cases} \quad (4)$$

where W_j and H_j are the width and height of component j, and R_j is the rotation parameter, determining the orientation of the component w.r.to its poly line(s) (R_j=1 for vertical, and 0 for horizontal orientation).

L and T-shape components: With the new formulation we are able to properly model both the L-shape and T-shape components as a set of abutting rectangles, as shown in Fig. 3. During the geometric placement all eight orientations of these components are considered and modeled as a set of linear equations. The model requires linearization of quadratic expressions, similar to those shown in Section III.

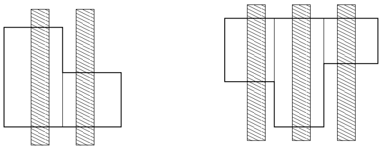


Fig. 3. Modeling of the L- and T-shape components

III. Post-processing using Global Optimization

The goal of the post-processing phase is to generate the final layout of the placed transistors at the mask level. This phase deals with the number of fine-grain layout issues, such as determining the final positioning and orientation of the placed components, assignment of N wells to P diffusion regions, etc. This is accomplished by mirroring the transistors along the X and Y axes. It should be noticed that such mirroring affects the absolute positions of component pins and the location of the P/N diffusion regions, and therefore can be used to reduce the number of wells and to simplify the routing. Minimizing the number of wells is an important issue from the manufacturing point of view. According to the experienced layout designers, it is even more important than the net-length minimization, as long as the design meets the required timing specifications. It can also simplify routing by minimizing the number of well ties.

Our approach to the generation of final transistor layout is composed of two steps. First, we compute the orientations of the components (placed during the geometric placement phase) that results in a minimum number of wells. Such generated orientation is then being used as a constraint in the second step, which globally minimizes the length of the nets.

A. Well minimization

We assume a single well technology and formulate the N-well minimization problem as a global optimization problem and solve it using MILP.

Diffusion Adjacency Graph: The first step in our procedure is to represent all possible diffusion region adjacencies between the adjacent components. We will use an *interval representation* of the components, defined as follows: each component is represented as a horizontal and vertical *interval* by projecting the component onto the X and Y axis, respectively. The length of the horizontal interval gives the width, while the length of the vertical interval specifies the height of the component. The horizontal (vertical) intervals are sorted by the reference Y (X) coordinates of the respective components, so that adjacent intervals represent adjacent components. The overlap between adjacent intervals in either direction determines possible adjacency between their N or P diffusion regions. By analyzing the adjacency information in the interval representation, we can identify the pairs of components that may contribute to the minimization of wells.

To represent the required adjacency relation between the diffusion regions of adjacent transistors we create an undirected graph, called *Diffusion Adjacency Graph (DIAG)*. Each node in the graph represents a component; an edge (i, j) is created for each pair of components that have adjacent *overlapping intervals* in either direction. Fig. 4 shows an example of a DIAG for a sample layout. Each edge $(i, j) \in DIAG$ is assigned a set of *overlap weights*, V_{ij}. Each element $v_{ij}^k \in V_{ij}$ represents one combination of adjacency of P diffusion regions between components i and j. Each such combination k corresponds to a different combination of mirroring parameters M_i, M_j. The value of v_{ij}^k is computed based on the overlap of P regions between the adjacent components, for each combination k. They can assume zero value if the P diffusion regions do not have any interval overlaps. This is illustrated in Fig. 5.

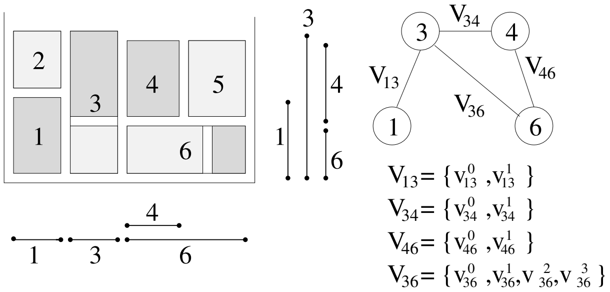


Fig. 4. Obtaining the diffusion adjacency graph from geometric placement

For a component with double diffusion there are two possible diffusion placements; this is a result of mirroring the component along the X or Y direction. In Fig. 5, mirroring of component i along the X axis gives rise to two different values of P diffusion adjacency with the single P diffusion component j, namely v_{ij}^0 and v_{ij}^1. For the graph in Fig. 4, there are four overlap weights in the set V_{36}, due to four different combinations of adjacent P diffusion regions of components 3 and 6.

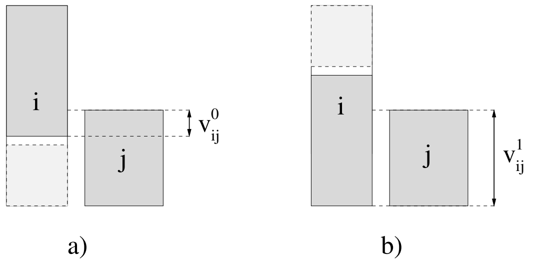


Fig. 5. Computation of interval overlaps: a) original position; b) component i mirrored in X axis

MILP Formulation: We introduce a binary variable M_i, to represent the mirroring of a double-diffusion component i along its respective axis. For single-diffusion components, the mirroring will not have any effect in the diffusion placement, so there is no need to introduce binary variables for such components. M_i will refer to the mirroring of component i along the X axis, if the rotation of the component, computed in the geometric placement, is vertical (rotation parameter $R_i = 1$); otherwise, for $R_i = 0$, M_i will refer to the mirroring in Y axis.

Assuming that only one component (i) has double diffusion, the interval overlap Z_{ij} for a component pair (i, j) can be written as follows:

$$Z_{ij} = (1 - M_i) \cdot v_{ij}^0 + M_i \cdot v_{ij}^1 \tag{5}$$

where v_{ij}^0 and v_{ij}^1 are the overlap weights for M_i = 0 and 1, respectively (see Fig. 5). The above equation can easily be extended to the case when both components have double diffusions, as shown below.

$$\begin{aligned} Z_{ij} &= (1 - M_i - M_j + M_{ij}) \cdot v_{ij}^0 + \\ &\quad (M_i - M_{ij}) \cdot v_{ij}^1 + (M_j - M_{ij}) \cdot v_{ij}^2 + \\ &\quad M_{ij} \cdot v_{ij}^3 \end{aligned} \tag{6}$$

Here $M_{ij} = M_i M_j$, which can be linearized as follows:

$$M_{ij} \leq M_i \tag{7}$$

$$M_{ij} \leq M_j \tag{8}$$

$$M_{ij} \geq M_i + M_j - 1 \tag{9}$$

The objective is to maximize the sum of the interval overlaps between all pairs of components in the DIAG.

$$max \sum_{(i,j) \in DIAG} Z_{ij} \tag{10}$$

This maximizes the adjacency between the P diffusion regions, leading to a solution with minimum number of compact wells, as shown in Fig 6.

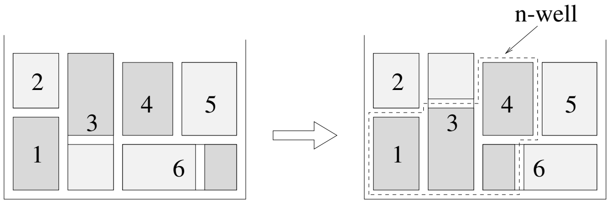


Fig. 6. Effect of post-processing on well minimization

B. Net length minimization

The length of a given net is determined by the exact locations of its terminals (pins). Mirroring of the components in X or Y axis affects the location of the pins on the component boundary. Subsequently, the length of each net depends on the final positioning of pins caused by component mirroring. The goal is to find the mirroring of components which reduces the overall net length, as illustrated in Fig. 7 We formulate the net length minimization problem as a global optimization problem using MILP.

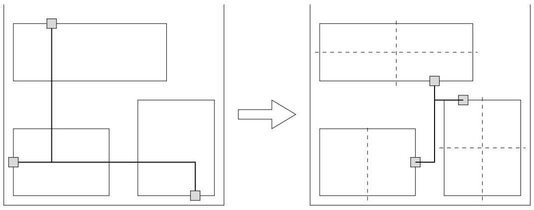
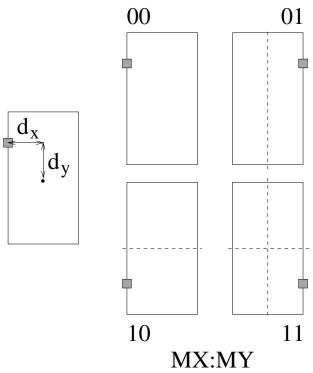


Fig. 7. Effect of mirroring on net length

Pin Modeling: Each pin i is modeled as a point with coordinates (Xp_i, Yp_i). The location of the pin is a function of the component mirroring, as shown in Fig. 8. To express the coordinates of the com-



MX	MY	X_p	Y_p
0	0	$-d_x$	d_y
0	1	d_x	d_y
1	0	$-d_x$	$-d_y$
1	1	d_x	$-d_y$

Fig. 8. Pin location as a function of mirroring in X and Y directions

ponent pins we introduce two binary variables Mx_i and My_i per component i, which represent the mirroring along the X and Y axis, respectively. The location of each pin is computed relative to the component center, whose coordinates are not affected by the mirroring. The location of pin i on component q is represented by the following linear equations:

$$\begin{aligned} Xp_i &= Xc_q - (1 - Mx_q - My_q + Mxy_q) \cdot d_x + \\ &\quad (Mx_q - Mxy_q) \cdot d_x - (My_q - Mxy_q) \cdot d_x + \\ &\quad Mxy_q \cdot d_x \end{aligned} \tag{11}$$

where Xp_i is the X location of pin i; Xc_q is the X location of the center of component q; d_x is the relative distance of the pin from the component center in X direction; and $Mxy_q = Mx_q My_q$. The latter equation can be represented as a set of linear equations, similar to the equations 7-9. Y coordinates of the pins are derived in a similar fashion.

In our optimization we approximate the length of a net using *half perimeter* of the bounding box that includes all the pins of the net. The objective is to minimize the weighted sum of the lengths of all the nets, with higher weights assigned to critical nets. Additional constraints are imposed by well minimization; mirroring binary variables, Mx_q, My_q must be fixed for some of the components q in order to preserve the solution of the well minimization.

IV. Results

We implemented our analytical approach to post-processing for global net length and well minimization and integrated it with the fully automated geometric placement tool developed earlier [1]. The minimization problems are solved using a commercial MILP solver, CPLEX. We tested the method on several datapath circuits, made available by Compaq, Alpha Development Group. The complexity of these circuits varied from 8 to 24 transistors, with the number of placeable components ranging from 9 to 15. The number of nets in those cells ranged from 12 to 35, and the number of pins from 35 to 122. The post-processing results are obtained in a few seconds using a Pentium 200MHz with 128Mb RAM.

Table I compares the final post-processing results for net length and well minimization, with the results of geometrical placement (prior to post-processing), assuming an initial reference orientation for all the components. The table shows our measure of improved routability in terms of the overall net length. The amount of the interval overlaps (used as a measure of well "packing") and the overall net lengths are given in λ.

Example $2X$ is the most complex circuit, with the largest number of transistors, pins and nets. Fig 9 shows the result of the geometric placement and the final placement after well minimization for this circuit. Only two nets are shown in the figure to demonstrate the effect of net length minimization.

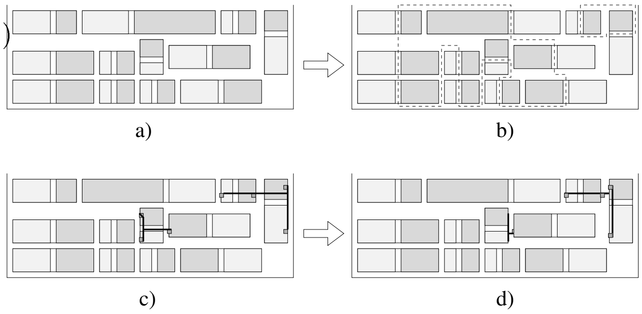


Fig. 9. Example 2X: a) before well minimization b) after well minimization c) before net length minimization d) after net length minimization, subject to fixed well configuration

We should emphasize that our datapath synthesis tool does not take into account the detailed routing. Nevertheless, for all the tested examples the geometric placement generated by the tool was routable with only minor modifications involving placing poly and metal contacts required for routing. We expect that the post-processing will significantly improve routability and minimize the length of critical nets. It should also minimize the number of the required well ties, further contributing to the improvement in routability.

V. Conclusions

We developed a systematic way to include both the routability and manufacturability measures in our analytical model for custom datapath synthesis. A single global optimization problem addresses both of these issues by means of minimizing the number of wells (for a single well technology) and the overall weighted net length, possibly subject to constraints imposed by well minimization. It is too early to say exactly by how much the described optimization will increase the routability or will reduce the delay due to net length minimization; our approach does not take detailed routing into account. Similarly, the effect of well minimization on circuit manufacturability, routability and the overall circuit layout area remains to be examined. In principle, the reduction in the number of wells, and hence the well ties, should help reduce the wiring complexity and the layout area. Due to a number of additional modi-

				Interval Overlaps [λ]		Net length [λ]	
Circuit name	# of trans.	# of nets	# of pins	Before optim.	After optim.	Before optim.	After optim.
PGK9	20	21	94	580	580	74598	71598
PB7	8	12	35	740	880	8746	8466
CS15	45	35	85	1080	1360	46640	43800
MU9	12	17	63	480	760	30780	28940
CMU8	9	14	96	620	760	68720	67140
PB8	8	12	40	400	510	12190	11020
2X	24	32	122	1120	1980	75444	74010

TABLE I
RESULTS

fications to our analytical formulation, we can now handle designs with a larger number of transistors. The results are currently being evaluated on a set of industrial datapath circuits.

Future work in our datapath synthesis research will focus on the following issues: 1) Post-processing, involving the geometry-invariant transformations by means of component swapping, followed by 2-D compaction to further reduce net lengths. 2) Addition of global routing phase to more accurately predict the effects of routing on circuit area. 3) Considering iterative improvement to MILP solutions by means of iterative group relaxation, in order to handle larger designs. 4) The concept of component grouping in the pre-processing phase needs to be explored to further reduce the number of integer variables in our formulation.

We are confident that our analytical approach combined with proper post-processing and iterative improvement will create automatically routable and fully acceptable layouts competitive with those obtained by human designers.

VI. Acknowledgement

Authors would like to thank Sam Levitin, Ken Slater and Alan Cave, of Compaq Alpha Development Group, for their invaluable feedback and support.

References

[1] S.. Askar and M. Ciesielski, "Analytical approach to custom datapath synthesis," *International Conference on Computer-Aided Design, ICCAD'99*, pp. 863–878, 1999.

[2] S. Sutanthavibul, E. Shragowitz, and J. B. Rosen, "An analytical approach to floorplan design and optimization," *IEEE Transactions on CAD*, pp. 761–769, 1991.

[3] T. Chen and Fan K. H. M., "On convex formulation of the floorplan area minimization problem," *International Symposium on Physical Design*, 1998.

[4] T. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout*, Wiley-Teubner Series, 1990.

[5] M. Sarrafzadeh and C. K. Wong, *An Introduction to VLSI Physical Design*, McGraw-Hill Series, 1994.

[6] M. S. Sait and H. Youssef, *VLSI Physical Design Automation*, IEEE Press - McGraw-Hill, 1995.