Priority Driven Channel Pin Assignment

Ines Peters
Institut für Informatik
Lehrstuhl Prof. Dr. P. Molitor
Universität Halle
D–06099 Halle (Saale), Germany

Abstract

We present a polynomial time improvement of the linear channel pin assignment LCPA algorithms presented by Cai and Wong in 1990. We solve the LCPA problem according to minimum channel density under a special priority schedule subject to vertical constraints and flux. The priority driven linear channel pin assignment algorithm (PDCPA) reduces the channel height by an average of 17 % without increasing the running time[1].

1 Introduction

The channel routing problem plays an important role in the physical design of VLSI circuits. It has been extensively studied in the past ([15], [5], [19], [7], [10], [12]). Advanced (over–the–cell) channel routers consider the terminals (pins) on the two sides of the channel not completely fixed at the beginning of the routing phase ([17], [2], [6], [11], [13], [18]). This approach enables further channel density reduction. The channel pin assignment (CPA) problem subject to position, order and separation constraints has been shown to be NP–hard in general ([2]). Polynomial time algorithms for an important special case in which the relative order of the terminals on the top and on the bottom borders of the channel are completely fixed (linear channel pin assignment (LCPA) problem), have been presented in [3], [4] and [2].

We present a fast and easy schedule for the optimal Linear CPA which reduces the channel height by an average of 17 % as compared to the existing LCPA algorithm presented in [2]. The main idea of the presented algorithm is a priority driven computation of the minimum channel density. The priorities depend on the local densities ([16]), the properties of the vertical constraint graph ([5], [19]) and the flux[2] ([1]). For vertical constraints see [5], [19], [9] and for another lower bound on the channel height called flux see [1], [9].

The Figures 1 and 2 show a comparison of the LCPA algorithm presented in [2] (Figure 1) with our new approach (PDCPA) (Figure 2). The channel length is fixed (to 20 in the example) like in most practical applications. So the channel area depends only on the channel height. The minimum channel density equals five in both cases. Both LCPA and PDCPA compute a minimum channel density solution, but the channel heights differ due to the different vertical constraints graphs (shown in the lower part of the Figures).

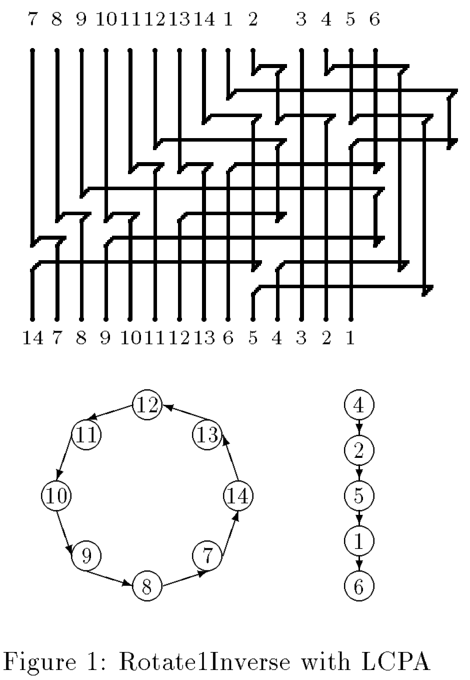

Figure 1: Rotate1Inverse with LCPA

2 Preliminaries

Given a channel with length L. Let TOP= $\{t_1, t_2, \ldots, t_p\}$ and BOTTOM= $\{b_1, b_2, \ldots, b_q\}$ be the set of terminals on the top and bottom border of the channel, respectively.

The set of nets $N = \{N_1, N_2, \ldots, N_n\}$ connecting these terminals is a partition of TOP $\cup$ BOTTOM, such that N_i contains the set of terminals of the ith

[1] This work was supported by the BMFT under 01 IS 102.

[2] The flux $f(C)$ of a channel C is defined as $f(C) =_{def} \max\{f_{local}(c) \mid c \textit{ is a flux cut of } C\}$, where a flux cut is a contiguous interval (of columns) on one of the sides of the channel and the local flux $f_{local}(c)$ of a flux cut c which spans l nontrivial columns and splits exactly n nontrivial nets is defined as the smallest integer t such that the inequality $(l-n)t + t(t+1) \geq n$ holds.

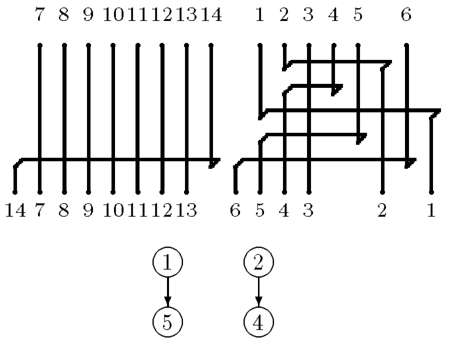

Figure 2: Rotate1Inverse with PDCPA

net, $1 \leq i \leq n$. Let net:TOP $\cup$ BOTTOM $\rightarrow N$ assign a given terminal to the corresponding net.

The order constraints of the *Linear CPA* problem are given by a fixed order of the terminals. We assume that $t_1 < t_2 < \ldots < t_p$ and $b_1 < b_2 < \ldots < b_q$, in the sense that t_i (b_i) has to be assigned to the left of t_{i+1} (b_{i+1}).

There are also position constraints, i.e. valid terminal positions, given by sets $T = \{T_k \subseteq \{1, 2, \ldots, L\}; 1 \leq k \leq p\}$ and $B = \{B_k \subseteq \{1, 2, \ldots, L\}; 1 \leq k \leq q\}$.

In order to assign all terminals t_i, b_j to a position, i.e. a certain column k, we have to distinguish between four types of decisions. They are shown in Figure 3. Let $R_1(i,j)$ ($R_2(i,j)$) be the set of nets with one terminal in $\{t_1, t_2, \ldots, t_{i-1}, b_1, b_2, \ldots, b_j\}$ ($\{t_1, t_2, \ldots, t_i, b_1, b_2, \ldots, b_{j-1}\}$) and one terminal in TOP $\cup$ BOTTOM $\backslash\{t_1, t_2, \ldots, t_i, b_1, b_2, \ldots, b_j\}$. The net net($t_i$) (net($b_j$)) also belongs to $R_1(i,j)$ ($R_2(i,j)$) if it is not trivial. In the set $R_3(i,j)$ are all nets with one terminal in $\{t_1, t_2, \ldots, t_{i-1}, b_1, b_2, \ldots, b_{j-1}\}$ and one terminal in TOP $\cup$ BOTTOM $\backslash\{t_1, t_2, \ldots, t_i, b_1, b_2, \ldots, b_j\}$ and the nets net(t_i), net(b_j) if they are not trivial and not net(t_i) = net(b_j) = $\{t_i, b_j\}$.

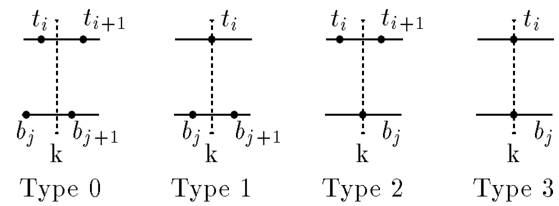

Figure 3: Four types of (i, j, k)–decisions

We define the *crossing numbers*[3] at (i,j,k) as:

$$x(i,j,k) = \begin{cases} +\infty & \text{if } k \notin T_i \\ \mid R_1(i,j) \mid & \text{otherwise;} \end{cases}$$

$$y(i,j,k) = \begin{cases} +\infty & \text{if } k \notin B_j \\ \mid R_2(i,j) \mid & \text{otherwise;} \end{cases}$$

$$z(i,j,k) = \begin{cases} +\infty & \text{if } k \notin T_i \cap B_j \\ \mid R_3(i,j) \mid & \text{otherwise,} \end{cases}$$

Let $d = (d_1, d_2) : \{1, \ldots, p\} \times \{1, \ldots, q\} \times \{1, \ldots, L\} \rightarrow \mathbb{N} \times \{0, 1, 2, 3\}$ be a function which assigns the local density and the (i, j, k)–decision type to the terminals t_i and b_j and the column k.[4]

3 The algorithm

For every column of the channel the algorithm has to decide for one of the type 0, type 1, type 2 and type 3 solutions. The decision is made according to the minimum channel density. Here we show the main part of the original LCPA algorithm (see also [2]):

for $k := 1$ **to** L **do**
 for $i := 0$ **to** p **do**
  for $j := 0$ **to** q **do**
  $D_0 := d_1(i, j, k-1)$
  $D_1 := \max\{d_1(i-1, j, k-1),\ x(i,j,k)\}$
  $D_2 := \max\{d_1(i, j-1, k-1),\ y(i,j,k)\}$
  $D_3 := \max\{d_1(i-1, j-1, k-1),\ z(i,j,k)\}$
  $d(i,j,k) := \text{Min}(D_0, D_1, D_2, D_3)$

In [2] the function **Min** returns the minimum density and the solution type. But there may be more than one solution type satisfying this supposition. At this point we introduce a priority schedule to enable the algorithm to prefer one of these. We introduce priorities $P_0, \ldots, P_3$ preferring decisions of type 0, ..., type 3, respectively. The priorities are increased or decreased according to the current situation before the new minimum function **PriorityMin** is called.

- In order to decrease channel congestion we perform the following steps according to the local densities:

  if $D_0 = D_1 = D_2 = D_3$ **then**
   if $x(i,j,k) < y(i,j,k), z(i,j,k)$
    then *Increase* P_1
   if $y(i,j,k) < x(i,j,k), z(i,j,k)$
    then *Increase* P_2
   if $z(i,j,k) \leq x(i,j,k), y(i,j,k)$
    then *Increase* P_3.

- Type 3 decisions effect vertical constraints. We prefer type 3 decisions having the same net on the top and on the bottom of the current column. Columns with different nets on the top and on the bottom cause edges in the vertical constraint graph:

  if net(t_i) = net(b_j)
   then *Increase* P_3 **else** *Decrease* P_3

[3] $x(i,j,k)$ is the local density of the channel in the case that t_i is placed to position k, b_j to a position left of k and b_{j+1} to a position right of k, y and z respectively.

[4] Note that $d_1(p,q,L)$ is the channel density and $d_2(p,q,L)$ is the decision type for the last column: $d_2(p,q,L) = 3$ means that both t_p and b_q have to be placed to position L. $d_2(p,q,L) = 2$ ($= 1$) means that just b_q (t_p) has to be placed to position L. And $d_2(p,q,L) = 0$ means that neither t_p nor b_q are assigned to L.

- Free or partly free columns are important to reduce flux. This relationship has been explained in [1]. In Figure 4 we show type 3 decisions in different contexts. We have to decide whether the terminals 3 and 4 should be put together in one column (say k). Therefore in the left most case the priority is set very high because of the free column $k-1$. In the right most case the priority is very low ($k-1$ is not free). In the other cases the priority is set to a medium value. Decisions of other types are handled in a similar way.

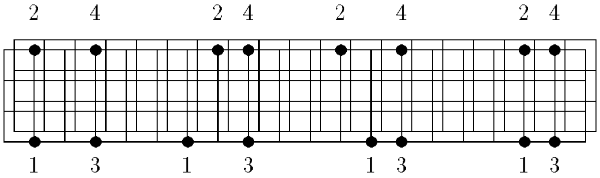

Figure 4: Type 3 decisions in different contexts

In the new algorithm PDCPA we replaced the function **Min** by a function **PriorityMin**, which returns the minimum density and the decision type according to the priorities set before. It is implemented as follows: **PriorityMin:**

Val := $\min\{D_0, D_1, D_2, D_3\}\}$
MinSet := $\{a \mid a = 0, 1, 2, 3; D_a = \mathrm{Val}\}$
if MinSet = $\{a\}$ **then return** (Val, a)
else MinSet := MinSet $\setminus \{0\}$
 if MinSet = $\{a\}$ **then return** (Val, a)
 elsif MinSet = $\{1, 2\}$ **then**
  if $P_1 > P_2$ **then return** $(\mathrm{Val}, 1)$
  elsif $P_2 > P_1$ **then return** $(\mathrm{Val}, 2)$
  else return $(\mathrm{Val}, 1), (\mathrm{Val}, 2)$ *alternating*
 elsif MinSet = $\{1, 3\}$
  if $P_1 > P_3$ **then return** $(\mathrm{Val}, 1)$
  else return $(\mathrm{Val}, 3)$
 elsif MinSet = $\{2, 3\}$
  if $P_2 > P_3$ **then return** $(\mathrm{Val}, 2)$
  else return $(\mathrm{Val}, 3)$
 elsif MinSet = $\{1, 2, 3\}$ **then**
  if $P_3 = \max\{P_1, P_2, P_3\}$
  then return $(\mathrm{Val}, 3)$
  else
   if $P_1 > P_2$ **then return** $(\mathrm{Val}, 1)$
   elsif $P_2 > P_1$ **then return** $(\mathrm{Val}, 2)$
   else return $(\mathrm{Val}, 1), (\mathrm{Val}, 2)$ *alternating*

Of course the establishing of the function $d = (d_1, d_2)$ is followed by the assignment of the terminals to the positions using backtracking. Here we use the analogous algorithm as in [2],p.13.

Obviously the proposed priority driven algorithm PDCPA runs the same time as compared to the original algorithm.

4 Experimental results

We implemented the proposed algorithm PDCPA in Modula–2 language on PC. All data are ISCAS benchmarks. For them we generated layout data and cell placements with HULDA ([9], [8]). To speed up the algorithm we partitioned the channels into several subchannels as proposed in [3]. In the last four examples we generated two–row cell placements (to get longer channels) and applied the heuristics.

Table 1: Experimental results

Circuit	**Ch**	**L**	**D**	**H1**	**H2**	**%**
CC	ch1	75	8	14	8	57.1
	ch2	75	13	15	14	93.3
CM138	ch1	38	10	14	10	71.4
CM150	ch1	83	10	12	10	83.3
	ch2	83	7	9	7	77.8
CM151	ch1	83	11	13	12	92.3
CM162	ch1	65	15	16	15	93.7
CM163	ch1	65	13	14	13	92.8
CM42	ch1	65	7	11	7	63.6
CM82	ch1	65	8	9	9	100.0
CM85	ch1	73	10	14	14	100.0
CMB	ch1	82	17	20	18	90.0
CORDIC	ch1	100	12	16	12	75.0
	ch2	100	9	11	9	81.8
CU	ch1	100	11	14	12	85.7
	ch2	100	10	13	10	76.9
DECOD	ch1	100	11	23	13	56.5
I1	ch1	100	14	16	14	87.5
MUX	ch1	107	12	13	13	100.0
	ch2	107	11	12	11	91.6
PARITY	ch1	67	11	13	11	84.6
PCLER8	ch1	102	11	15	11	73.3
	ch2	102	14	16	14	87.5
PM1	ch1	81	18	31	20	64.5
X2	ch1	90	17	20	19	95.0
Z4ML	ch1	90	11	15	11	73.3
	ch2	90	12	14	12	85.7
CC	ch1	110	19	25	22	88.0
CU	ch1	102	18	23	22	95.7
CM150	ch1	122	25	26	25	96.2
PCLER8	ch1	144	18	29	26	89.7

The experimental results are summarized in Table 1. The L means the channel length, D the channel density, H1 the channel height achieved by LCPA and H2 the channel height achieved by PDCPA. The ratio between both heights is given in %.

We obtained an average height reduction of 17.3 % for the optimal PDCPA algorithm and of 7.6 % using the PDCPA heuristics. Most channel heights achieved by PDCPA are less than the channel heights achieved by LCPA, three are equal and none is greater. In 58.0 % of all examples PDCPA reached the optimum channel height, where LCPA never did.

References

[1] B.S. Baker, S.N. Bhatt, T. Leighton, "An Approximation Algorithm for Manhattan Routing," *Advances in Computing Research*, **2**, pp. 205–229, 1984.

[2] Y. Cai, D.F. Wong, "Optimal Channel Pin Assignment," *ICCAD, To appear in IEEE Trans. on CAD*, 1990.

[3] Y. Cai, D.F. Wong, "Minimizing Channel Density by Shifting Blocks and Terminals," *Proc. ICCAD*,pp.524–527, 1991.

[4] Y. Cai, D.F. Wong, "On Shifting Blocks and Terminals to Minimize Channel Density," *To appear in IEEE Trans. on CAD*, 1992.

[5] H.H. Chen, E.S. Kuh, "Glitter: A Gridless Variable–Width Channel Router," *IEEE Trans. on CAD*, Vol. CAD–5, No. 4, pp.459–465, Oct. 1986.

[6] T. Fujii, Y. Mima, T. Matsuda, T. Yoshimura, "A Multi–Layer Channel Router with New Style of Over–the–Cell Routing," *Proc. 29th DAC*, pp.585–588, 1992.

[7] S.C. Fang, W.S. Feng, S.L. Lee, "A New Efficient Approach to Multilayer Channel Routing Problem," *Proc. 29th DAC*, pp.579–584, 1992.

[8] T. Hecker, I. Peters, B. Wartenberg, M. Weber, "An Integrated Synthesis Tool for the Generation of Space Efficient Standard Cell Layouts," *Proc. 37th IWK*, Ilmenau, Sept. 1992.

[9] T. Hecker, I. Peters, M. Weber, "A New Integrated Approach to Global and Detailed Routing of Standard Cell Layouts," *Dagstuhl Seminar 9343: Combinatorial Methods for Integrated Circuit Design*, Oct. 1993.

[10] T.T. Ho, "New Models for Four– and Five–Layer Channel Routing ," *Proc. 29th DAC*, pp.589–593, 1992.

[11] C.Y. Hou, C.Y.R. Chen, "A Pin Permutation Algorithm for Improving Over–the–Cell Channel Routing," *Proc. 29th DAC*, pp.594–599, 1992.

[12] M.S. Lin, H.W. Perng, C.Y. Hwang, Y.L. Lin, "Channel Density Reduction by Routing Over the Cells," *IEEE Trans. on CAD*, Vol 10, No. 10, Aug. 1991.

[13] S. Natarajan, N. Sherwani, N.D. Holmes, M. Sarrafzadeh, "Over–the–Cell Channel Routing For High Performance Circuits," *Proc. 29th DAC*, pp.600–603, 1992.

[14] I. Peters, "Priority Driven Channel Pin Assignment," *Manuscript*, 1994.

[15] B. Preas, M. Lorenzetti, ed., "Physical Design Automa-
tion of VLSI Systems," *The Benjamin/Cummings Publishing Company*, Inc., Menlo Park, CA 1988.

[16] C. Sechen, A. Sangiovanni–Vincentelli, "The TimberWolf Placement and Routing Package," *IEEE Journal of Solid–State Circuits*, vol. sc–20, no. 2, April 1985.

[17] T. Wang, D.F. Wong, Y. Sun, C.K. Wong, "On Over–the–Cell Channel Routing," *Proc. EuroDAC*, pp.110–115, 1993.

[18] B. Wu, N. Sherwani, N.D. Holmes, M. Sarrafzadeh, "Over–the–Cell Routers for New Cell Modell," *Proc. 29th DAC*, pp.604–607, 1992.

[19] T. Yoshimura, E.S. Kuh, "Efficient Algorithms for Channel Routing," *IEEE Trans. on CAD*, Vol CAD–1, No. 1, pp.25–35, Jan. 1982.