TRANSFORMING DISFIGURED AND DISORIENTED AREAS INTO ROUTABLE SWITCHBOXES[1]

M. STARKEY
T.M. CARTER

UUCS-91-003

Department of Computer Science
University of Utah
Salt Lake City, UT 84112, USA

March 7, 1991

Abstract

Routing an entire circuit requires partitioning the circuit (routing area) into smaller, localized routing areas. Using non-rectangular, rotated switchbox shapes (and therefore non-manhattan routing layout) has the potential to simplify the partitioning of the circuit into routable areas and to use "dead space" on a chip for routing. The method described in this paper for generating non-rectangular, rotated switchboxes borrows ideas from computer graphics.

[1]This work supported in part by grant J-FBI-89-102. This papter also appears in the Proceedings of the First Great Lakes Symposium on VLSI, March 1991.

Transforming Disfigured and Disoriented Areas Into Routable Switchboxes[1]

M. Starkey
Department of Computer Science
University of Utah
Salt Lake City, UT 84112

T. M. Carter
Department of Computer Science
University of Utah
Salt Lake City, UT 84112

Abstract

Routing an entire circuit requires partitioning the circuit (routing area) into smaller, localized routing areas. Using non-rectangular, rotated switchbox shapes (and therefore non-manhattan routing layout) has the potential to simplify the partitioning of the circuit into routable areas and to use "dead space" on a chip for routing. The method described in this paper for generating non-rectangular, rotated switchboxes borrows ideas from computer graphics.

1 Introduction

Automated wire routing is an important problem in the design of VLSI circuits. It involves finding the solution to two problems: subdividing the circuit into smaller, routable pieces and routing the wires. Usually, wire routing is performed on a square grid, thus providing the rectangular switchboxes that are used in the many good routing algorithms presently available. For some circuits, partitioning would actually be simplified if non-rectangular switchboxes could be used. For example, although much effort is used to rectangularize circuit modules, some elements are generally conceived as being non-rectangular. For example, the parallel multiplier is generally drawn conceptually as a parallelogram rather than as a rectangle (see Figure 1). The internal routing of the multiplier circuit can be easily converted to form a rectangular module. Alternatively, non-rectangular switchboxes can be used for routing outside the multiplier. Figure 2 shows some potential non-rectangular switchboxes created by the placement of a multiplier implemented as a parallelogram.

Routing algorithms are generally restricted to routing in rectangular areas [1, 2, 3, 4, 5, 6]. In addition, there are some algorithms that deal with angled routing [7, 8]. To avoid rewriting these for many different cases, we propose a method for transforming non-rectangular, four-sided switchboxes into rectangular switchboxes and back again, thereby permitting conventional routing algorithms to deal only with rectangles and manhattan routing. The method is borrowed from computer graphics and is known as transformational geometry which includes the *perspective projection.*

[1] This paper also appears in the Proceedings of the First Great Lakes Symposium on VLSI, March 1991.

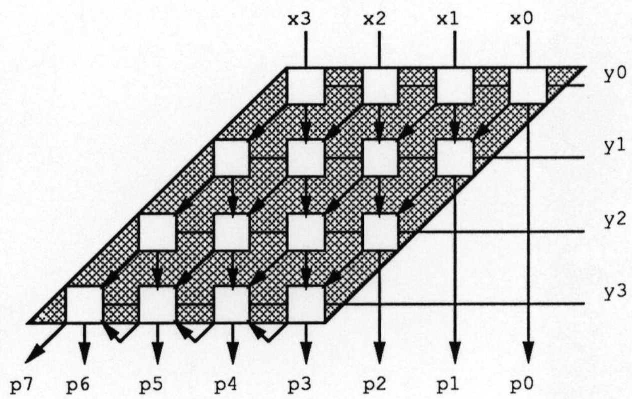

Figure 1: Parallel Multiplier Circuit

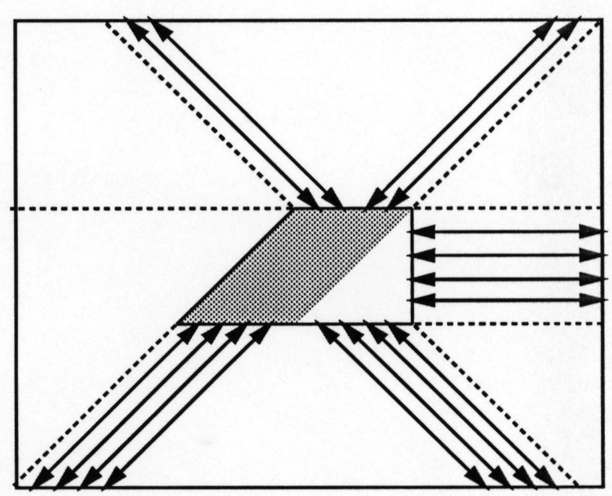

Figure 2: Switchboxes Caused by placing a Parallel Multiplier

2 Coordinate Transformation by Matrix Algebra

Computer graphics involves manipulating models of 3-D shapes to appear as 2-D objects on a computer screen. These transformations of 3-D shapes involve the use of matrices that describe rotation, translation, shear, mirroring and perspective, and vectors that describe the vertices of shapes. All of the manipulations are performed in a *homogeneous* coordinate system involving one extra dimension. A 3-D homogeneous coordinate is a vector of length 4 and its 3-D transformation matrices are 4-by-4. Foley and Van Dam describe this in great detail in [9].

Many simple transformation matrices can be multiplied to form a single matrix that embodies all of the successive transformations. Thus, a drawing composed of many lines (wires) can be transformed using a single matrix multiplication for each point once the complete transformation matrix has been composed.

As noted above, the dimension of the graphics area determines the size of the transformation matrices as well as the number of elements in the coordinate vectors. Switchboxes are modelled as 2-D objects in a 3-D space (that is, the z coordinate values are generally all 0). Using a 3-D model permits more complex transformations such as the perspective projection. Homogeneous coordinates for 3-D space are formed by appending a fourth value, w (always equal to 1), to the normal x, y and z values:

$$(x, y, z) \Rightarrow \begin{bmatrix} x & y & z & w \end{bmatrix}$$

This vector is then multiplied by a 4-by-4 transformation matrix and the values in the vector are divided by w (usually 1).

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{x'}{w'} & \frac{y'}{w'} & \frac{z'}{w'} & 1 \end{bmatrix} \tag{1}$$

2.1 Single Transformations

The standard, single transformation matrices for rotation, translation, shear and perspective projection are shown in Figure 3.

a)

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

b)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_{translate} & y_{translate} & 0 & 1 \end{bmatrix}$$

c)

$$\begin{bmatrix} 1 & y_{shear} & 0 & 0 \\ x_{shear} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

d)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{d} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 3: Single Transformation Matrices
a) Rotation of θ degrees around the z axis
b) Translation of $x + x_{translate}, y + y_{translate}$
c) Shear of $x + x_{shear}y, y + y_{shear}x$
d) Perspective projection

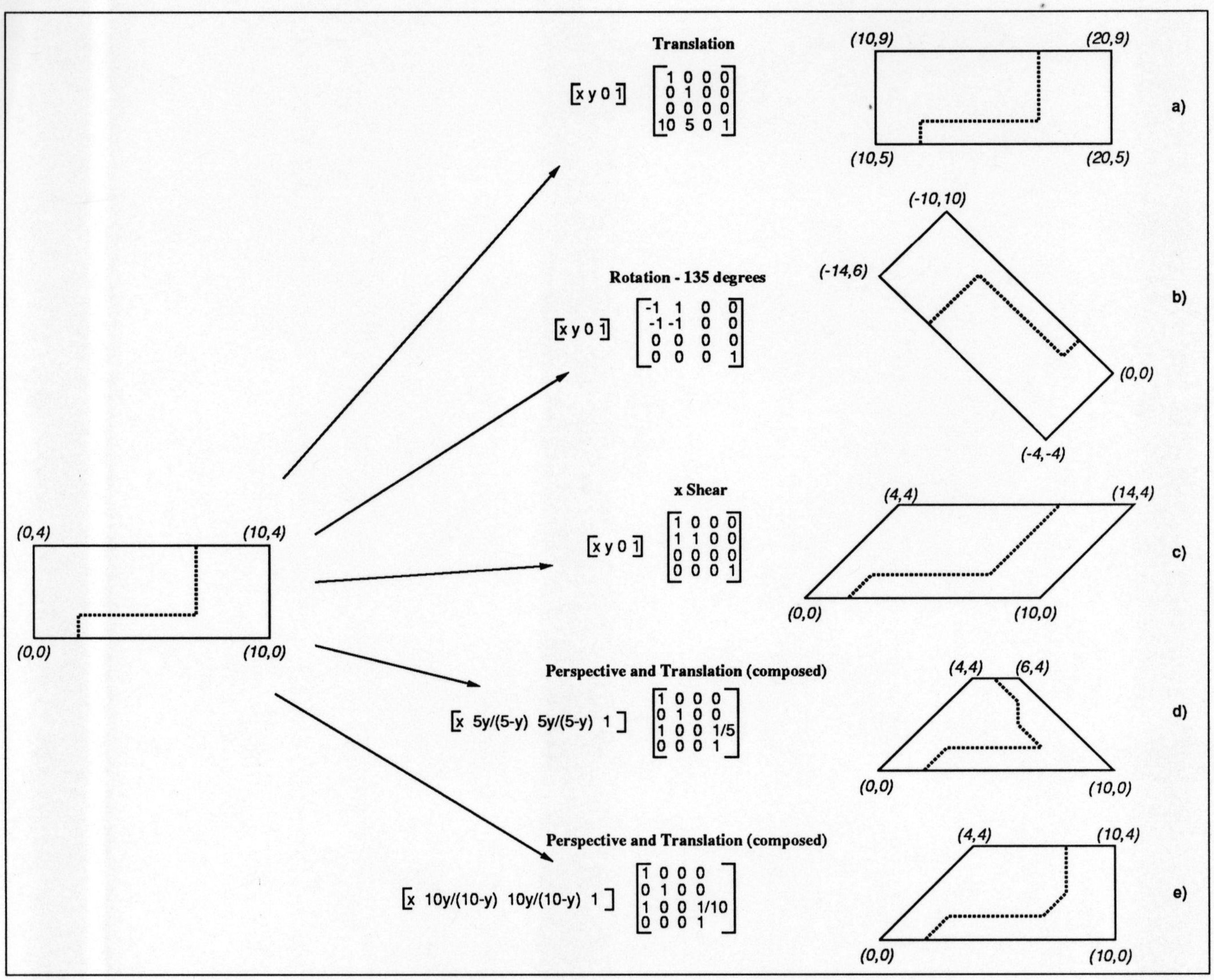

Figure 4: Examples of transformations of a rectangle.

Switchbox shapes defined by all of these operations may appear in circuit layouts. Figure 4 demonstrates all the possible four sided shapes provided routing is available on a 45 degree angle. Figure 4a shows a simple translation, 4b a rotation about (0,0), 4c a shear in the x direction. Figures 4d and 4e provide examples of more complex transformations. These shapes are created by perspective projections.

2.2 Simple 2-D Transformations

When non-rectangular switchboxes are encountered, the transformation(s) required to render that switchbox as a rectangle must be discovered. The simplest transformations to discover and use are rotation, translation and shear.

Given an original, non-rectangular switchbox (the *real* routing area) in which coordinates are denoted $[x_i, y_i, z_i, w_i]$, we desire to produce a rectangular switchbox (the *desired* routing area) in which coordinates are denoted $[x'_i, y'_i, z'_i, w'_i]$. By definition, the switchbox resides in the xy plane for these simple transformations and $z_i = 0$. The transformations ensure that $z'_i = 0$ also. If the elements of a transformation matrix are denoted M_{ij}, then

$$\begin{aligned} M_{11}x_i + M_{21}y_i + M_{31}z_i + M_{41} &= x'_i && (2)\\ M_{12}x_i + M_{22}y_i + M_{32}z_i + M_{42} &= y'_i && (3)\\ M_{13}x_i + M_{23}y_i + M_{33}z_i + M_{43} &= z'_i && (4)\\ M_{14}x_i + M_{24}y_i + M_{34}z_i + M_{44} &= w'_i. && (5) \end{aligned}$$

Given the four vertices of a real switchbox (it must be a parallelogram), the simple, 2-D transformation matrix may be discovered by solving the system

$$L = \left[\begin{array}{cccc|c|c} x_1 & y_1 & z_1 & 1 & x'_1 & y'_1 \\ x_2 & y_2 & z_2 & 1 & x'_2 & y'_2 \\ x_3 & y_3 & z_3 & 1 & x'_3 & y'_3 \\ x_4 & y_4 & z_4 & 1 & x'_4 & y'_4 \end{array}\right] \qquad (6)$$

and then substituting values from the solved system into the transformation matrix as follows (without perspective projection):

$$T = \begin{bmatrix} L_{15} & L_{16} & 0 & 0 \\ L_{25} & L_{26} & 0 & 0 \\ L_{35} & L_{36} & 1 & 0 \\ L_{45} & L_{46} & 0 & 1 \end{bmatrix} \qquad (7)$$

The desired, non-rectangular area can now be routed with traditional algorithms when the shape is a parallelogram (a sheared rectangle) with applied rotations and transformations. When the non-rectangular routing area is a trapezoid, a perspective projection is required.

2.3 Perspective Projection Transformations

A number of methods are used to enhance the realism of images displayed on a computer screen. One of these is perspective projection (Figure 5). Drawing objects on the screen (the *projection plane*) that are farther away from the viewing point (the *centre of projection*) smaller than those that are closer gives the impression that these objects are more distant. Therefore, a rectangular area having one edge slanted away from the viewing point, will be drawn as a trapezoid on the screen to provide this appearance.

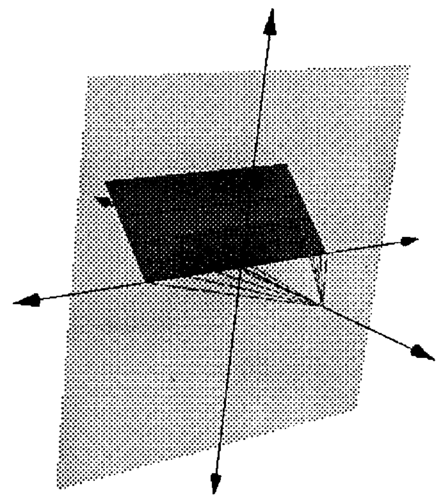

Figure 5: Perspective Projection

The transformation for calculating the projected points is defined as

$$\begin{bmatrix} x_i & y_i & z_i & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{d} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{x_i}{\frac{z_i}{d}+1} & \frac{y_i}{\frac{z_i}{d}+1} & 0 & 1 \end{bmatrix} \qquad (8)$$

In an image, the distance from the projection plane to the centre of projection is usually known and the transformation matrix is easily created. In transforming non-rectangular switchboxes to rectangular ones, only the vertices of the projection are known — the task is to discover the transformation matrix that transforms a rectangle into the trapezoidal perspective projection. By applying the inverse of this transformation matrix, we can transform a trapezoidal routing area into a rectangular routing area. Once routed, the rectangular routing area and all wires within it are then transformed back into the trapezoidal area using the original transformation matrix.

Figure 6 shows all that is known about the two areas. Since perspective projection is a 3-D transformation, z_i and z'_i are now included. Since the real area (Figure 6b) is on the projection plane, the z'_i values are all 0. Limiting the projection to produce only vertical, horizontal and 45 degree edges (we achieve this by angling the rectangle 45 degrees away from the projection plane), and

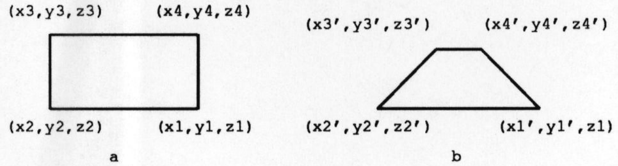

Figure 6: Projection of the desired area (a) to the real area (b)

constraining one edge (we choose the bottom) of the desired rectangular area to be on the projection plane, we get

$$
\begin{gathered}
x_4 = x_1 = x_1' \\
x_3 = x_2 = x_2' \\
y_1 = y_1' = y_2 = y_2' = z_1 = z_1' = z_2 = z_2' = 0 \\
y_3 = y_4 = z_3 = z_4 \\
y_3' = y_4' = z_3' = z_4'.
\end{gathered}
$$

We know the values for x_1, x_2, x_3, x_4, y_1, y_2, z_1 and z_2. We need only solve for d (from equation 8) which requires z_3.

Using equation 8,

$$z_3 = y_3 = \left(\frac{z_3}{d} + 1\right) y_3' = \frac{x_3}{x_3'} y_3' = \frac{x_2'}{x_3'} y_3' \tag{9}$$

$$d = \frac{x_3'}{x_3 - x_3'} z_3 = \frac{x_2'}{x_2' - x_3'} y_3' \tag{10}$$

Routing usually assumes uniform spacing of the horizontal and vertical wires. Given the orientation of the trapezoidal switchbox with parallel edges parallel to the x axis, the uniformly spaced horizontal wires on the real switchbox will be mapped by non-uniformly spaced wires on the desired rectangular switchbox. Uniformly spaced vertical wires on the desired rectangular switchbox will be mapped to angled wires (at 45 degrees) on the real trapezoidal switchbox when routing is complete and the transformation is applied.

Viewing vectors emanating from the centre of projection and passing through routing grid points in the real routing area provide the desired mapping on the desired routing area (Figure 7). Constraining the system such at $y_i = z_i$ and defining $y_{switchbox}$ as the spacing between horizontal wires in the real switchbox, the equation that describes these vectors is

$$y = z = \frac{y_{switchbox}}{d} z + y_{switchbox} \tag{11}$$

Solving for y, we obtain

$$y = \frac{d \cdot y_{switchbox}}{d - y_{switchbox}} \tag{12}$$

and the point in the desired area that corresponds to a point in the real switchbox is

$$\left[x \frac{d \cdot y_{switchbox}}{d - y_{switchbox}} \frac{d \cdot y_{switchbox}}{d - y_{switchbox}} 1 \right].$$

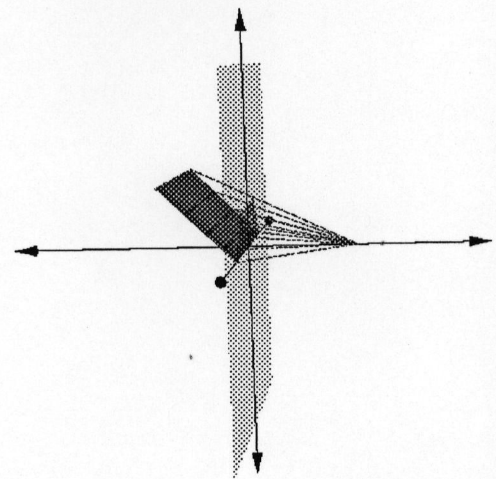

Figure 7: Side view of a perspective projection

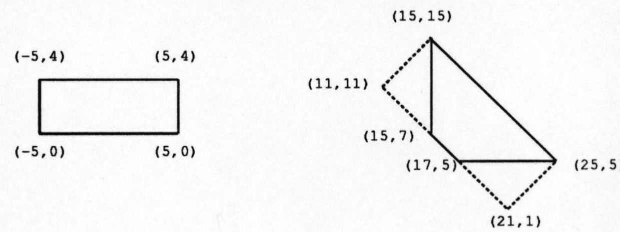

Figure 8: An example of a complex transformation

Examples are presented in Figures 4d and 4e which demonstrate sample equations which are required.

2.4 Composed Switchbox Transformations for Routing

The examples in Figure 4 involve only single transformations of a rectangular area. The example in this section, as shown in Figure 8 involves the composition of a number of transformations, including a rotation, a translation and a perspective projection. We briefly show the mathematics involved in obtaining the transformation.

Using equation 7, we form a system of linear equations that describe all of the transformations except the perspective projection, obtaining

$$L_{Figure8} = \left[\begin{array}{cccc|c|c} 5 & 0 & 0 & 1 & 15 & 15 \\ -5 & 0 & 0 & 1 & 25 & 5 \\ -5 & 4 & 0 & 1 & 21 & 1 \\ 5 & 4 & 0 & 1 & 11 & 11 \end{array} \right]$$

for which the solution is

$$L_{Figure8} = \left[\begin{array}{cccc|c|c} 1 & 0 & 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 20 & 10 \end{array} \right].$$

Thus, the transformation matrix representing the rotation and translation is

$$T_{Figure8} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 20 & 10 & 0 & 1 \end{bmatrix}$$

The inverse transformation is used to place the real points into the correct orientation. The inverse transformation matrix for the rotation and translation is

$$T^{-1}_{Figure8} = \begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 5 & 15 & 0 & 1 \end{bmatrix}$$

Now we compute the required perspective projection. From equation 10,

$$d = \frac{-5}{(-5)-(-1)} \cdot 4 = 5$$

and the perspective projection matrix is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{5} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Combining the two matrices, we obtain

$$\begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{5} \\ 20 & 10 & 0 & 1 \end{bmatrix}$$

which will be used to transform the routed area and its wires back to its original trapezoidal shape.

Now, the multiplication required is:

$$\begin{bmatrix} x & \frac{5y}{5-y} & \frac{5y}{5-y} & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{5} \\ 20 & 10 & 0 & 1 \end{bmatrix} \quad (13)$$

3 Conclusions

This powerful transformational approach to routing non-rectangular switchboxes achieves its results by permitting routers to work in their accustomed cartesian domain using "orthogonal" wires and rectangular switchboxes. We have purposely limited it in this presentation to work on 45 degree angles although the general method can be used for any angle.

It has the potential to simplify module design where the module is best implemented in a non-rectangular shape. It permits routing to be done using algorithms that are limited to rectangular shapes. These rectangular routing areas can then be transformed to non-rectangular shapes that border the non-rectangular module.

It will be particularly effective where three layers of routing are available since there can be sets of vertical, horizontal and diagonal wires. For example, the horizontal wires could be on the middle layer, the vertical wires on the top layer and the diagonal wires on the bottom layer. If the desired routing algorithm(s) can handle obstructions, the non-rectangular switchbox can be routed using the bottom (diagonal) and middle (horizontal) layers. Then, a rectangular switchbox superimposed over the non-rectangular switchbox can be routed using the top (vertical) and middle (horizontal) layers.

As routing methods are developed to deal with rectilinear shapes other than rectangles (say L- or T-shaped areas), this transformational approach can also be applied to these algorithms permitting them to route complex shapes under any combination of the transformations presented here: rotation, translation, shear and perspective projection.

References

[1] H. Shin and A. Sangiovanni-Vincentelli, "A detailed router based on incremental routing modifications: Mighty," *IEEE Transactions on Computer-Aided Design*, vol. CAD-6, pp. 944–955, November 1987.

[2] J. P. Cohoon and P. L. Heck, "Beaver: A computational-geometry-based tool for switchbox routing," *IEEE Transactions on Computer-Aided Design*, vol. 7, pp. 684–697, June 1988.

[3] M. Burstein and R. Pelavin, "Hierarchical wire routing," in *VLSI Circuit Layout: Theory and Design* (T. C. Hu and E. S. Kuh, eds.), pp. 191–202, IEEE Press, 1985.

[4] R. L. Rivest and C. M. Fiduccia, "A "greedy" channel router," in *19th Design Automation Conference*, pp. 418–424, 1982.

[5] G. Hamachi and J. K. Ousterhout, "A switchbox router with obstacle avoidance," in *21st Design Automation Conference*, pp. 173–179, 1984.

[6] C. Y. Lee, "An algorithm for path connections and its applications," *IRE Transactions on Electronic Computers*, vol. 10, pp. 346–365, September 1961.

[7] I. G. Tollis, "Techniques for wiring in non-square grids," in *ISCAS'89*, pp. 66–69, 1989.

[8] F. P. Preparata and J. Witold Lipski, "Optimal three-layer channel routing," *IEEE Transactions on Computers*, vol. C-33, pp. 427–437, May 1984.

[9] J. D. Foley and A. V. Dam, *Fundamentals of Interactive Computer Graphics*, ch. 8, pp. 267–279. Addison-Wesley Publishing Company, 1984.