

LAYOUT TOOL FOR HIGH SPEED ELECTRONIC AND OPTICAL CIRCUITS

H. S. Façanha, R. J. Offen, D. R. Selviah and C. E. T. Oliveira

Powerful layout editing systems are available nowadays for designing integrated circuits, specially in the more popular MOS technology, such as the MAGIC system [1] from Berkeley. Besides its interactive editing capabilities, it has embodied in itself algorithms of design rule checking, circuit extraction, compaction and automatic routing. Such tool could be developed only because it employed a data structure called 'corner-stitching' [2], specially developed to represent VLSI layouts, which provided good interface with all these algorithms, although imposing a severe geometric constraint (rectangular shapes) to the layout.

At present, gallium arsenide, silicon-bipolar, integrated and holographic optics are starting to emerge as potential technologies for designing VLSI circuits. Their use will certainly require a review on the concepts of layout tools, and new algorithms may be included in such systems. This is particularly true for the design of holographic optical interconnection, which is a promising way of bringing optics into electronic circuits in order to alleviate the present VLSI communication problems. An algorithm for holographic optical interconnection is discussed in section II. For this application it is important that the data structure can represent curved shapes. A corner-stitching based data structure that can represent rectangles, trapezoids and curved shapes is discussed in section I.

I - Curved corner-stitching data structure

The curved corner-stitching data structure is represented in Fig. 1. As in the original Manhattan corner-stitching data structure [2], all space in a layout is stored as a perfectly tiled area of nonoverlapping geometric objects. Even empty spaces are explicitly represented. These rectangular, trapezoidal or curved shapes (called tiles) have pointers that connect their corners to the adjacent tiles. The lower-left corner pointers (or stitches) point to the tiles below and to the left; the upper-right corner stitches point to the tiles above and to the right. Tiles may not overlap in corner stitching based data structures. Overlapping of multiple mask layers can be achieved by creating many tile planes, one to each mask layer, and each plane with its own data structure. Alternatively, a symbolic representation can be used, when many different types of solid tiles are created, for each possible combination of mask layers.

In the original corner-stitching, each tile record contains *2 integers/record* for coordinate values, *4 pointers/linkage* and *1 integer/record* for tile type, using a total of *7 words/tile*. It is worth pointing out that only 2 coordinate values (the x and y coordinates) of the bottom-left corner of a tile are required to be stored. The other vertices of a rectangle can be found by looking at the co-ordinates of its neighbours.

H. S. Façanha and D. R. Selviah are with the Dept. of Electrical and Electronic Engineering, University College London. R. J. Offen is with the Dept. of Electrical and Electronic Engineering and Computer Science, University College London. C. E. T. Oliveira is with the Dept. of Computer Science, University College London.

In the curved corner-stitching, isoparametric polynomials of the first and second orders were used to map the sides of trapezoids and curved shapes into rectangles, as illustrated in Fig. 2. In this case the number of co-ordinates stored per tile has to increase, requiring *8 words/tile* for the trapezoidal tile and *11 words/tile* in the case of the curved tile. For the trapezoidal case only the bottom-left x and y co-ordinates and the top-left x coordinate need to be stored. For the curved tile these co-ordinates must also be stored, plus the x and y co-ordinates for the point in the middle of the left curved sides and the y coordinate for the middle point in the right curved side. The remaining coordinate values can be derived from the neighbours co-ordinates.

This data structure, as well as the original corner-stitching, is very attractive for representing VLSI circuits, since it performs the neighbour finding operation in constant time, because neighbourhood information is explicitly stored in the data structure. Corner-stitching based data structures make possible the integration of algorithms such as design rule checking, circuit extraction and compaction in the editing system. Besides, they provide information about empty spaces, which is extremely important for routing algorithms.

Following the presentation of the curved corner-stitching it is now possible to investigate a holographic interconnection algorithm that is appropriate for integration into an editing layout system. Using the discussed data structure, the curved shapes of the holographic pattern can be represented in the same database as the rest of the integrated circuit layout data.

II - Holographic routing algorithm

A basic holographic interconnection system is shown in Fig 3. It corresponds to the interconnection of one optical transmitter (laser diode, modulator) to one optical receiver (photodetector or modulator). The reflective hologram containing a certain interconnection pattern is placed above the integrated circuit with the optical components to be interconnected. The light rays exiting the optical transmitter reach the hologram in a certain angle; then the light is reflected back to the integrated circuit to the location where the optical receiver is placed. The hologram is designed in such a way that the desired optical interconnection is achieved. This kind of hologram, where no lens is required between the integrated circuit and the hologram is called Fresnel hologram. In the other case, where the lens is used, it is called Fourier Transform hologram. The technique discussed next concerns only the Fresnel type of hologram.

A Fresnel hologram for optical interconnection is described by a well-known Fresnel zone plate equation, considering off-axis, off-line hologram. A description of such holographic method can be found in [3]. Basically speaking, it is represented by a set of holographic interference fringes, whose locus of points of maximum irradiance occur at the holographic plane at

$$r-r_p = [(x-x_r)^2 + (y-y_r)^2 + z_r^2]^{1/2} - [(x-x_p)^2 + (y-y_p)^2 + z_p^2]^{1/2} = N w , \quad (1)$$

where r and p represent the reference point source (the optical transmitter) and the object point source (the optical receiver) respectively; x and y represent the co-ordinates of these optical components; w is the light wavelength and N is an integer.

In order to obtain the holographic pattern corresponding to the interconnection of one optical transmitter to one optical receiver (one-to-one interconnection), this equation is normally sampled with a sampling grid resolution calculated for achieving some system specifications, as described in [4]. The implementation of an one-to- n interconnection is achieved by superimposing (one-to-one) interconnection Fresnel holograms, each one corresponding to the interconnection of one pair of transmitter-receiver.

By carrying on the sampling procedure as discussed in [4], it can be recognised that the holographic pattern corresponds to sets of concentric ellipses, each set corresponding to a one-to-one interconnection. Fig. 4 illustrates a one-to-four interconnection hologram. We have been investigating the implementation of such holograms not by sampling, but by directly designing the elliptical shapes that form the holographic pattern, using a plotter device or an electron beam machine.

In fact, equation (1) can be manipulated to take the form of a conic equation:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0, \quad (2)$$

where

$$A = 4N^2 w^2 - 4(x_r - x_p)^2, \quad (3)$$

$$B = -8(x_r - x_p)(y_r - y_p), \quad (4)$$

$$C = 4N^2 w^2 - 4(y_r - y_p)^2, \quad (5)$$

$$D = -4N^2 w^2(x_r + x_p) - 4(x_r - x_p)(x_p^2 + y_p^2 - x_r^2 - y_r^2), \quad (6)$$

$$E = -4N^2 w^2(y_r + y_p) - 4(y_r - y_p)(x_p^2 + y_p^2 - x_r^2 - y_r^2), \quad (7)$$

$$F = 4N^2 w^2(x_p^2 + y_p^2 + z_p^2) - [N^2 w^2 + (x_p^2 + y_p^2 - x_r^2 - y_r^2)]^2. \quad (8)$$

The implementation of a program to design the holographic pattern of (2) is described in [3]. What is worthwhile mentioning is the simplicity of this algorithm, which makes it appropriate for incorporation into an integrated circuit layout tool. The only parameters required are the co-ordinates of the optical components involved in the interconnection, besides some constants, such as the wavelength and the distance between the integrated circuit containing the optical components and the hologram. The information about the positions of integrated circuit devices is usually readily available in existing integrated circuit layout tools, since this information is required for circuit extraction and automatic electronic routing algorithms.

III - Towards a layout tool with capability of designing optical interconnects

Using the curved corner-stitching data structure it is possible to develop an integrated circuit layout tool with capabilities of implementing optical interconnects, by using simple 'holographic routing' algorithms, such as the one mentioned in section II. The curved corner-stitching data structure will have the same performance as the original corner-stitching when representing the rectangular layout shapes. However, it is estimated that for representing trapezoidal shapes it will take up to twice the time, when compared to the rectangles, and twice to ten times for the curved shapes. The curved corner-stitching can easily interface with such 'holographic routing' algorithm. This data structure can represent a hierarchical layout as separate planes of symbolic or mask layers. Therefore, the elliptical fringes that compose the holographic patterns can be stored into a separate plane, as is already done with the metal layers that contain the patterns of electronic wiring. The location of the optical components to be connected is readily obtainable from corner-stitching based editors, since this information is normally provided to automatic routing and circuit extraction algorithms. And the information about separation/spaces between objects that is provided to design rule checking and compaction algorithms can be very helpful when designing the holograms: similar design rules must

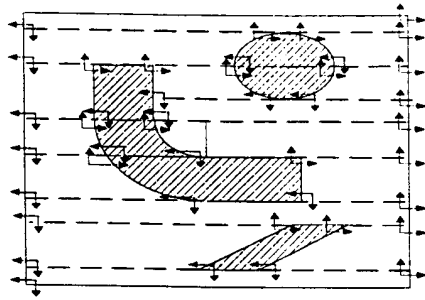


Fig. 1 Curved corner-stitching

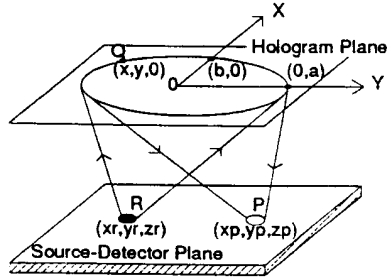


Fig. 3 Basic holographic system connecting two optical components.

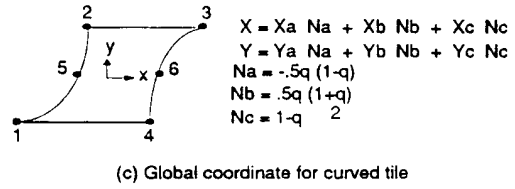
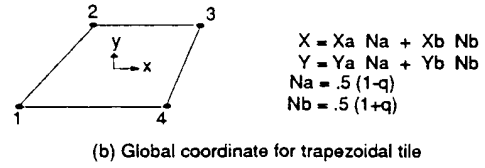
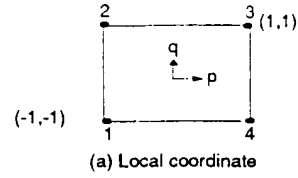


Fig. 2 Isoparametric tiles representation: in the eqs. above "a" corresponds to point 1 or 4 and "b" to point 2 or 3.

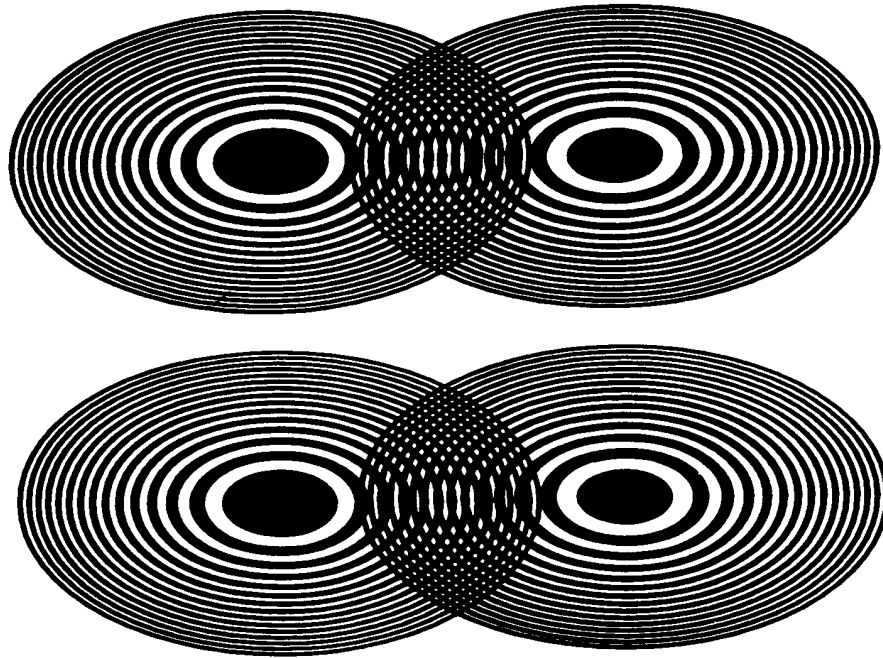


Fig 4 One-to-four interconnection hologram

be observed when choosing the distance between optical components, since the hologram size and position are attached to them and overlapping of holograms may cause the optical channels to interfere with each other, causing unacceptable aberrations in the receiving signal.

Acknowledgements

H. S. Façanha wishes to thank the CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) of Brazil for the sponsorship of this work, in the form of a PhD scholarship.

References

1. OUSTERHOUT, J.K. et al.: "MAGIC: A VLSI layout system", Proc. 21st ACM-IEEE Design Automation Conference, 1984, pp. 152-159.
2. OUSTERHOUT, J.K.: "Corner-stitching: A data-structuring technique for VLSI layout tools", IEEE Trans. on CAD, vol. CAD-3, no. 1, pp. 87-100, Jan. 1984.
3. FAÇANHA, H.S. et al.: "Design of Fresnel holograms for optical interconnection of VLSI", Proc. 2nd. International Conference on Holographic Systems, Components and Applications, Sept. 1989, pp. 213-217.
4. FELDMAN, M.R. and GUEST, C.C.: "Computer generated holographic optical elements for optical interconnection of very large scale integrated circuits", Applied Optics, vol. 26, no. 20, Oct. 1987, pp. 4377-4384.