WIRE-ROUTING IS NP-COMPLETE

Mark R. Kramer

Jan van Leeuwen

RUU-CS-82-4

Februari 1982

Rijksuniversiteit Utrecht

Vakgroep informatica

Princetonplein 5
Postbus 80.002
3508 TA Utrecht
Telefoon 030-53 1454
The Netherlands

WIRE-ROUTING IS NP-COMPLETE

Mark R. Kramer
Jan van Leeuwen

Technical report RUU-CS-82-4
Februari 1982

Department of Computer Science
University of Utrecht
P.O. Box 80.002, 3508 TA Utrecht
the Netherlands

WIRE-ROUTING IS NP-COMPLETE

Mark R. Kramer and Jan van Leeuwen

Department of Computer Science, University of Utrecht
P.O. Box 80.002, 3508 TA Utrecht, the Netherlands

Abstract. We prove the following problem to be NP-complete: given N pairs of points on a rectangular grid, can wires be routed to connect paired points such that (i) the wires run along gridlines only and (ii) the wires do not overlap or cross. The problem remains NP-complete if wires are allowed to cross. The result shows that the general routing problem in VLSI-design is NP-complete, even in the absence of further optimality constraints.

Keywords and phrases: VLSI, chip, layout, wire, routing, NP-completeness.

1. Introduction.

The problem of finding a suitable VLSI-design is normally split in two tasks: (i) placement and (ii) routing. This leads to the following question: suppose components have been placed, how hard is it to compute a routing for the necessary connecting wires over the chip. Normally a placement of the components is not fixed unless a routing is known to exist and an effort has been made to minimize the total length of the wires used and/or the total area occupied by the design. In this note we shall prove that even the question to determine whether a routing exists at all, is NP-complete.

In 1980 C.D. Thompson [4] formulated a simple VLSI-model in which the surface of a chip is viewed as a rectangular grid of unit size cells. Cells may contain at most one processing element (point) each. Circuits are formed by connecting points by wires as required. Wires are constrained to run through the "lanes" (the columns and rows) of the grid only. Also, wires are not allowed to overlap. It follows that wires can only have orthogonal intersections (crossings) which, in reality, would correspond to passages through distinct layers of the chip surface. If wires do not cross, then a design is called "cross free". (In a cross free design no two wires can make use of a same cell.) Given a set of points and a set of prescribed connections, any admissable (cross free) drawing of the necessary wires is called a (cross free) routing.

Thompson's model has been used to study a large variety of VLSI design problems and their intrinsic complexity in general (see e.g. [3]). Using the same conventions we can now formulate the problem that we prove to be NP-complete.

ROUTING

Instance: N pairs of points on a grid.

Question: Is there a routing of the N wires connecting the pairs of points.

ROUTING is proved NP-complete by means of a polynomial transformation from 3-SAT (cf. [2]). The problem remains NP-complete if wires are not allowed to cross (CROSS FREE ROUTING). Both problems remain NP-complete if the pairs of points are required to be fully disjoint. The notations and terminology pertaining to the theory of NP-completeness closely follow Garey & Johnson [2].

2. The NP-completeness of (cross free) wire routing.

The NP-completeness proof is facilitated by considering a useful intermediate problem. Let an "obstacle" be any (connected) rectangular domain of cells.

OBSTACLE ROUTING

Instance: N pairs of points and M (rectangular) obstacles on a grid.

Question: Is there a routing of N wires connecting the pairs of points such that no wire intersects an obstacle (i.e., no wire is routed through an obstructed cell).

For technical reasons we must assume that obstacles are given by an explicit listing of the cells they cover. (We will rid ourselves of this assumption later, when we construct instances of OBSTACLE ROUTING in which the obstacles have a size that is polynomially bounded in N.) We shall prove that both OBSTACLE ROUTING and its cross free version, CROSS FREE OBSTACLE ROUTING are NP-complete.

Lemma A. (CROSS FREE) OBSTACLE ROUTING polynomially transforms to (CROSS FREE) ROUTING.

a known NP-complete problem can be polynomially transformed to them. We will transform from 3-SAT (cf. [2]).

Let an instance of 3-SAT be given. It consists of a collection C of clauses in disjunctive form that must be simultaneously satisfied, with 3 literals per clause and variables (and their negations) chosen from x_1 to x_n. To construct an equivalent instance of OBSTACLE ROUTING we need some intuitive terminology first. An "i-street" ($i \geq 1$) consists of two parallel lanes (rows or columns) of the grid, bordered by blocked cells and separated by i fully blocked lanes (see fig. 1). Only where

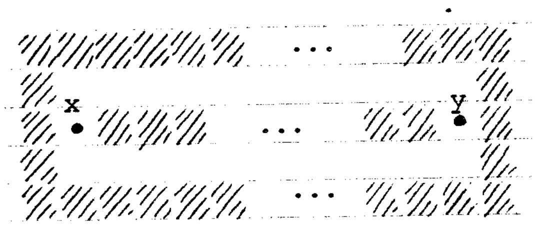


Fig. 1. A 1-street, with two points x and y.

a side-street begins (a junction) or another street intersects it (a crossing), the regular lane-structure will be modified somewhat (within the boundaries of the street though). When a wire must be routed from a point x at one end of the street to a point y at the other end (see figure 1), there essentially are only two possibilities: either through the first lane, or through the second. We will identify these options with "false" and "true" and label one lane with x^0 and the other with x^1 to distinguish them for our purposes. In a number of cases it will be necessary to let lanes switch roles. This is achieved by redirecting the wire to an intermediate point and forcing the continuation from another, such that the role-switching is effectuated. Figure 2 shows the basic inverter that can be inserted in an i-street. Inversion makes

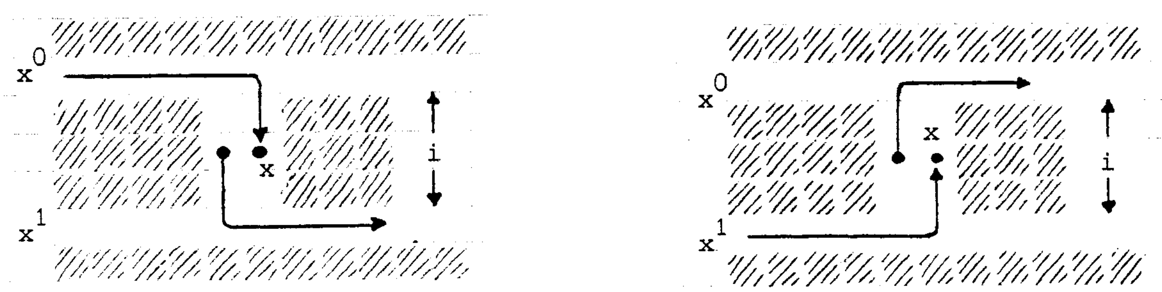
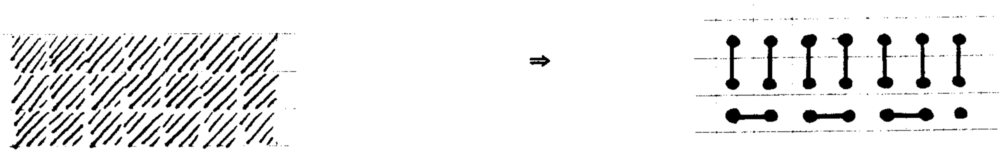


Fig. 2. An inverter.

Proof

Given an instance I of OBSTACLE ROUTING, we construct an equivalent instance I' of ROUTING. Take the same N pairs of points, but replace each of the M obstacles by an additional set of pairs (called "obstruction pairs") as follows. Put a point in each cell of the obstacle and combine



adjacent points into pairs. This can be done in such a manner that all pairs are disjoint. Only when an obstacle consists of an odd number of cells there will be one point that cannot be paired to a buddy and thus must be paired to itself. (We shall see later that this can be avoided in the application of the transformation in lemma B.) Clearly I' can be constructed in polynomial time.

To prove that I and I' are equivalent, we observe the following. Clearly obstruction pairs can be connected by drawing the "straight" wire of unit length through the common boundary of the two cells containing the pair. Thus any solution to I immediately translates into a solution of I'. (The wires connecting the obstruction pairs do not interfere with any other wires and leave a routing cross free if the given one was.) Conversely, consider any solution to I'. As the two points of any obstruction pair necessarily are in adjacent cells, we may assume that their connecting wire runs directly through the common cell boundary. (If it didn't, we could change the wiring so it does.) It means that in I' the obstruction pairs together block out certain regions (the original obstacles!) for use by other wires and leave the remaining area completely free and open. It follows that the solution to I' translates back immediately into a solution of I. (Again, if the solution to I' was cross free, then so is the solution to I.)

□

<u>Lemma</u> B. (CROSS FREE) OBSTACLE ROUTING is NP-complete.

Proof

It should be clear that both OBSTACLE ROUTING and its cross free variant are in NP. Thus, for the NP-completeness proof it suffices that it possible to switch to the truth-value assignment for $\bar{x}$ (the negation of x) when necessary. But inverters also allow for the possibility to block both lanes of a street for routing alien wires. By placing one inverter (or two, to neutralize the effect on the lane interpretation) between any two consecutive sites where a special construction has taken place, one can assure that wires can only be routed through the streets we want them to use.

Given a collection of clauses C as specified, we construct an equivalent instance of OBSTACLE ROUTING as follows. In brief, the instance will consist of n vertical 5-streets representing the n variables and $3|C|$ horizontal 2-streets that connect, in couples of 3 corresponding to the literals of a clause, to "plazas" representing the individual

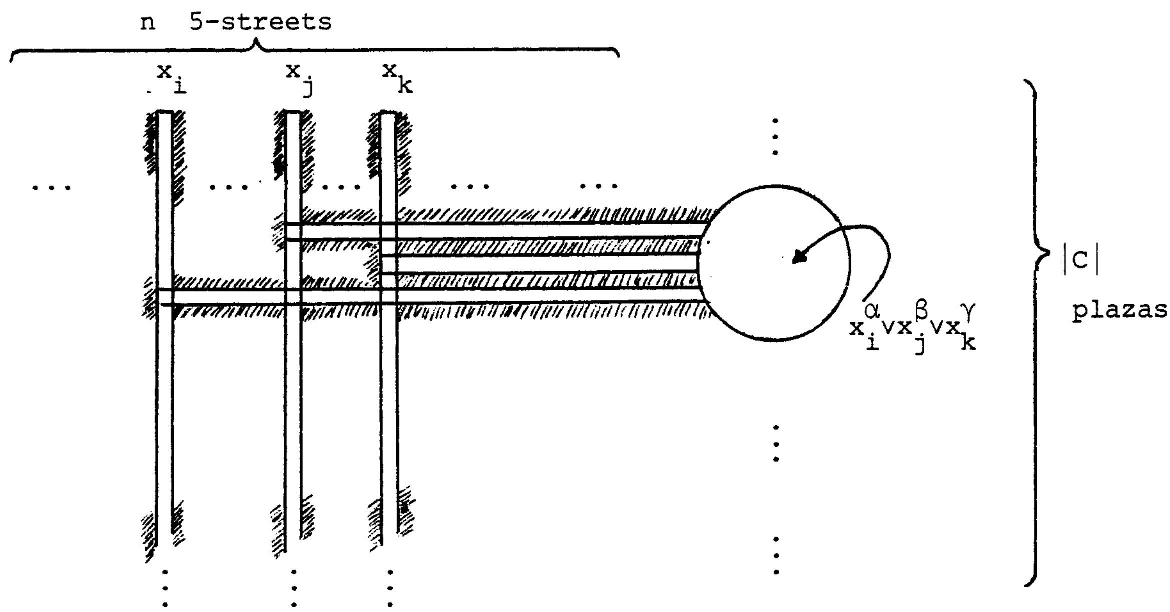


Fig. 3. Outline of the routing problem.

clauses. Horizontal streets begin at a junction with the vertical street corresponding to the proper variable x contributed to the clause. If $\bar{x}$ is to be contributed, then an inverter is put in the vertical street just before the junction to get the desired effect and another one immediately after it is used to reenact the original interpretation of the lanes. Note that horizontal streets must cross vertical ones to their right, and we shall need a special interrupt construction to let wires "cross over" while preserving the interpretation of the lanes.

A junction should be constructed such that the truth-value assignment of the corresponding variable x (as reflected by the wiring down the street) is copied into the horizontal street consistently, while a downward routing is re-established afterwards. Figure 4 shows how this can

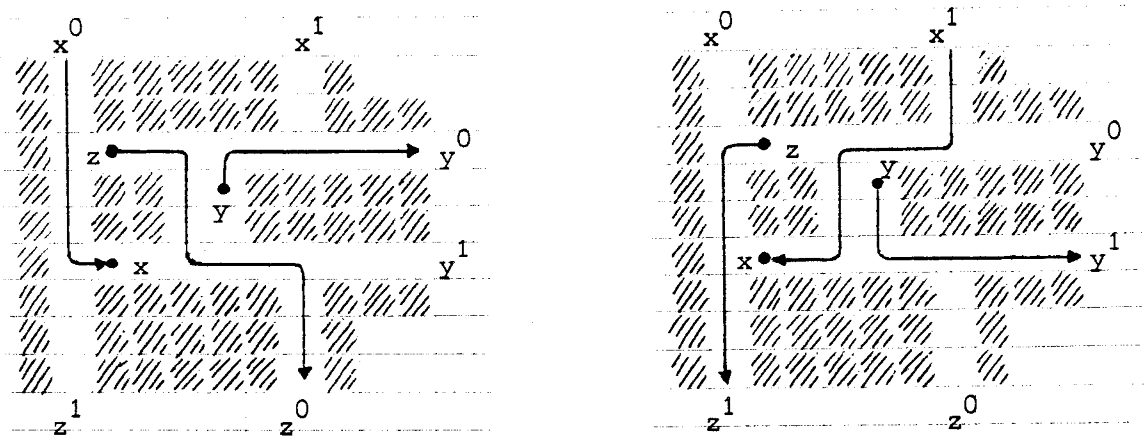


Fig. 4. A junction (x≡y≡z).

be done using only three extra points. Observe that the routing in the junction is completely determined, depending on whether the x-wire comes in through the left (x^0) or through the rigth (x^1) lane. It correctly splits off the 0-1 interpretation into the horizontal street, but in the process the wire down the vertical street has switched lanes. Thus an inverter must be put in immediately afterwards, to bring the wire back into its original lane. Note that wires cannot (and do not) cross in a junction.

To enable a horizontal street and a vertical street to cross while preserving their "value", we need a cross-over construction that interrupts (when necessary) and re-establishes the wire routing in the various intersecting lanes. In fact we need two separate cross-over constructions: one in case wires are allowed to cross (figure 5) and one in case they are not (figure 6). In figure 5 the wires are, in fact, forced to run straight on, as conveniently located pairs (s,t) and (u,v) would be blocked otherwise. While it seems that we could have let the wires cross without further steps, these two pairs are necessary to prevent one of the wires to switch lanes. One easily verifies that the routing in a cross-over is uniquely determined by the lanes through which the x-wire and the y-wire enter. Note that the cross-over makes essential use of the fact that wire-crossings are permitted. Figure 6 is more complicated and does require that the routing in one direction (down

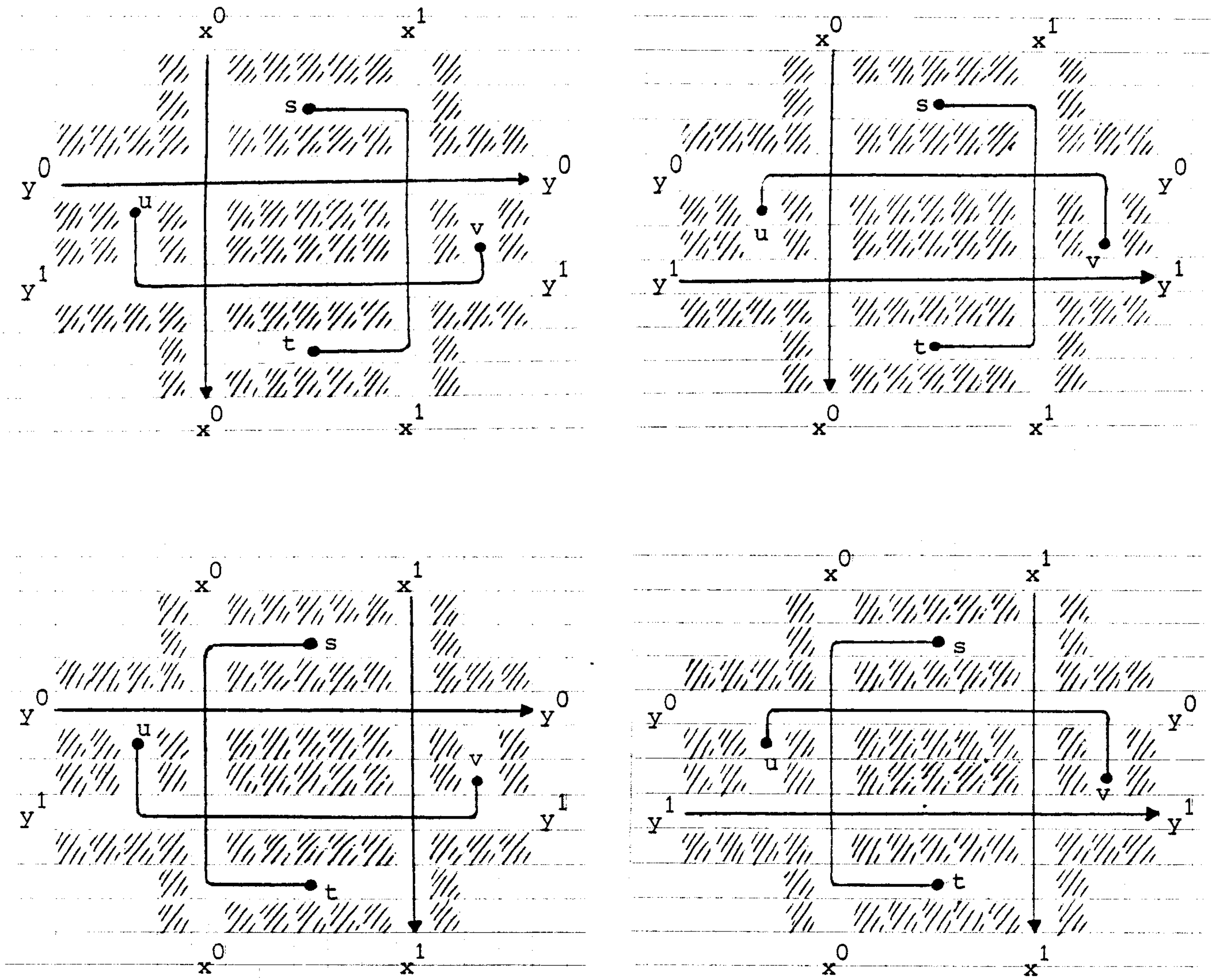


Fig. 5. A cross-over.

the vertical street, for example) is interrupted. There is no other way if wires are not allowed to cross. The extra points and the modified pairing not only guarantee that a cross free "passage" can be effectuated but force it to be as shown in the various instances of figure 6, which differ depending on where the x- and y-wire enter. Note that the x-wire (essentially turned into the z-wire) is forced to switch lanes and thus an inverter must be inserted in the vertical street, just below the cross-over. Note that figure 6 is correct by virtue of the condition that wires should not cross.

Finally plazas must be designed that property reflect the evaluation of the clauses. Thus they should allow for a (internal) routing if and only if at least one of three incoming streets brings in a wire through

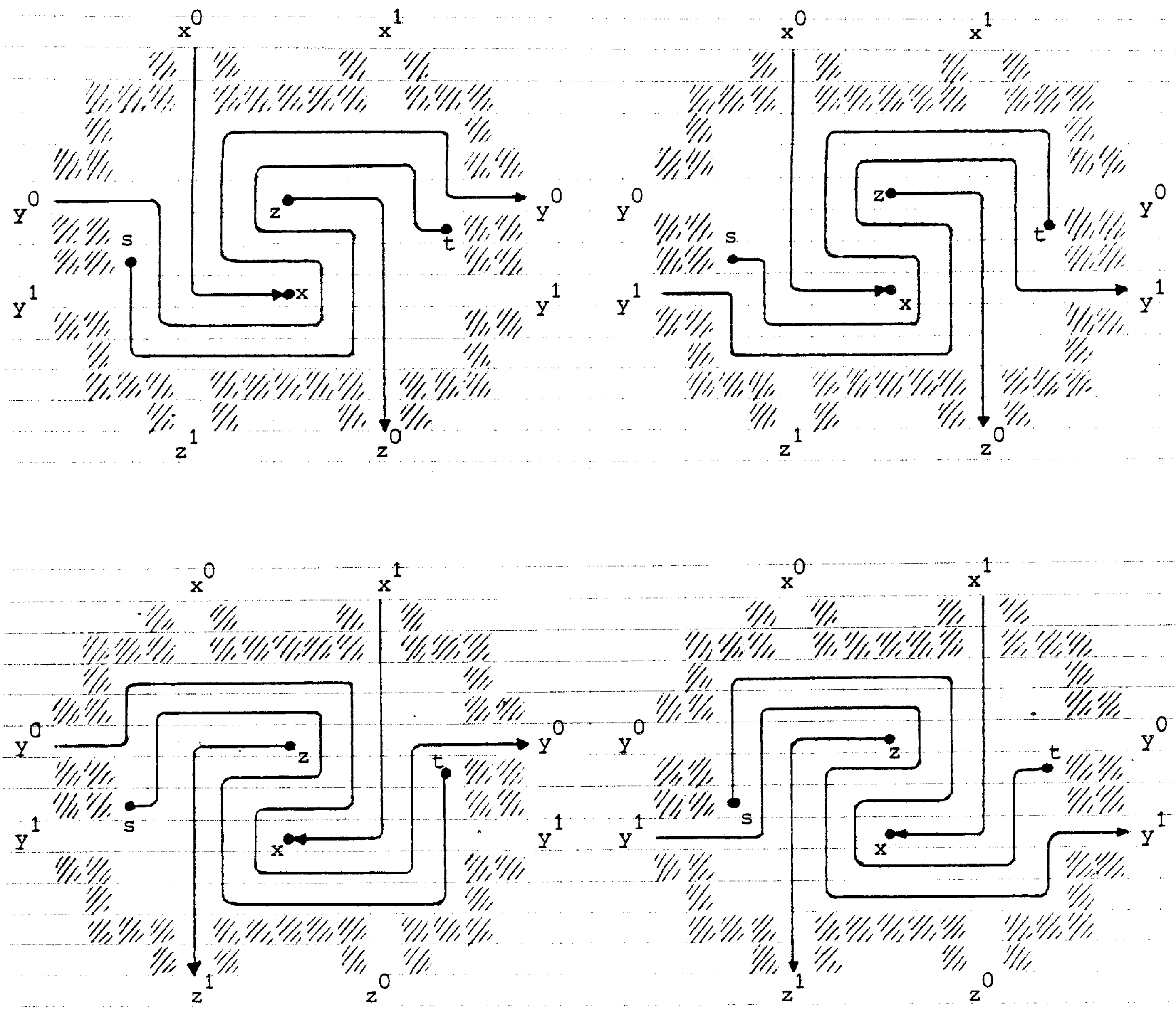


Fig. 6. A cross-over for the model in which no wire intersections are allowed (x≡z).

its "true" lane. Figure 7 shows a suitable plaza for clauses xvyvz. Note that the horizontal street corresponding to y must include an inverter just before entering the plaza, and that the horizontal street corresponding to z must be led around to enter the plaza at the required right side. (This requires an inverter in the z-street as well, as in bending around the lanes switch position.) It is necessary to put an inverter in the x-street as well, to prevent a wire that should be routed on the plaza from running into an unblocked lane. While we left it open exactly by what distance horizontal streets are to be separated (a distance of 12 blocked lanes would certainly do), figure 7 assumed rather arbitrarily a distance of 4. This can always be achieved by bending streets closer to one another. The pairs (s,t) and (u,v) are strategically chosen so as to let a routing through the narrow "gorge" of the plaza exist in case there is some room either at the s-u or at the v-t end to lead a wire around. Just in case x≡y≡z≡false(0) all this room is taken by the x-, y- and z-wires (and necessarily so, for otherwise the chances for a routing on the plaza are nil anyway) and no routing for the pairs exists, unless the overlap constraint of the model is violated.

We conclude that the instance of (CROSS FREE) OBSTACLE ROUTING is a consistent image of the instance of 3-SAT, and that the clauses of C are simultaneously satisfied if and only if a complete (cross-free) routing exists. The transformation requires the construction of $O(n.|C|)$ special elements (streets, junctions, cross-overs, inverters and plazas) which have size $O(n)$, $O(|C|)$ or $O(1)$. The entire construction is easily completed in time polynomial in the size of the given instance of 3-SAT. Thus 3-SAT polynomially transforms to (CROSS-FREE) OBSTACLE ROUTING.

□

Observe in the proof of lemma B that the instance of (CROSS FREE) OBSTACLE ROUTING obtained from the instance of 3-SAT fits in only $O(n.|C|)$ area, which means in particular that the size of the obstacles needed remains bounded by a fixed polynomial in n (as $|C|\leq 8n^3$) and thus in the number of pairs N actually constructed. The proof of Lemma B carries some similarities to a construction in [1].

<u>Theorem</u>. (CROSS FREE) ROUTING is NP-complete.

<u>Proof</u>

Clearly both ROUTING and CROSS FREE ROUTING belong to NP. The result now follows by combining lemmas A and B.

□

Note that OBSTACLE ROUTING only served as a useful intermediate problem, and that the proofs together give an immediate polynomial transformation of 3-SAT to (CROSS FREE) ROUTING. By being a bit more careful one can make sure that all intermediate obstacles have an even number of cells, which means that pairs not only are disjoint but may be assumed to consist of distinct points only. Note also that CROSS FREE ROUTING is a special case of the DISJOINT CONNECTING PATHS problem discussed in [2], which

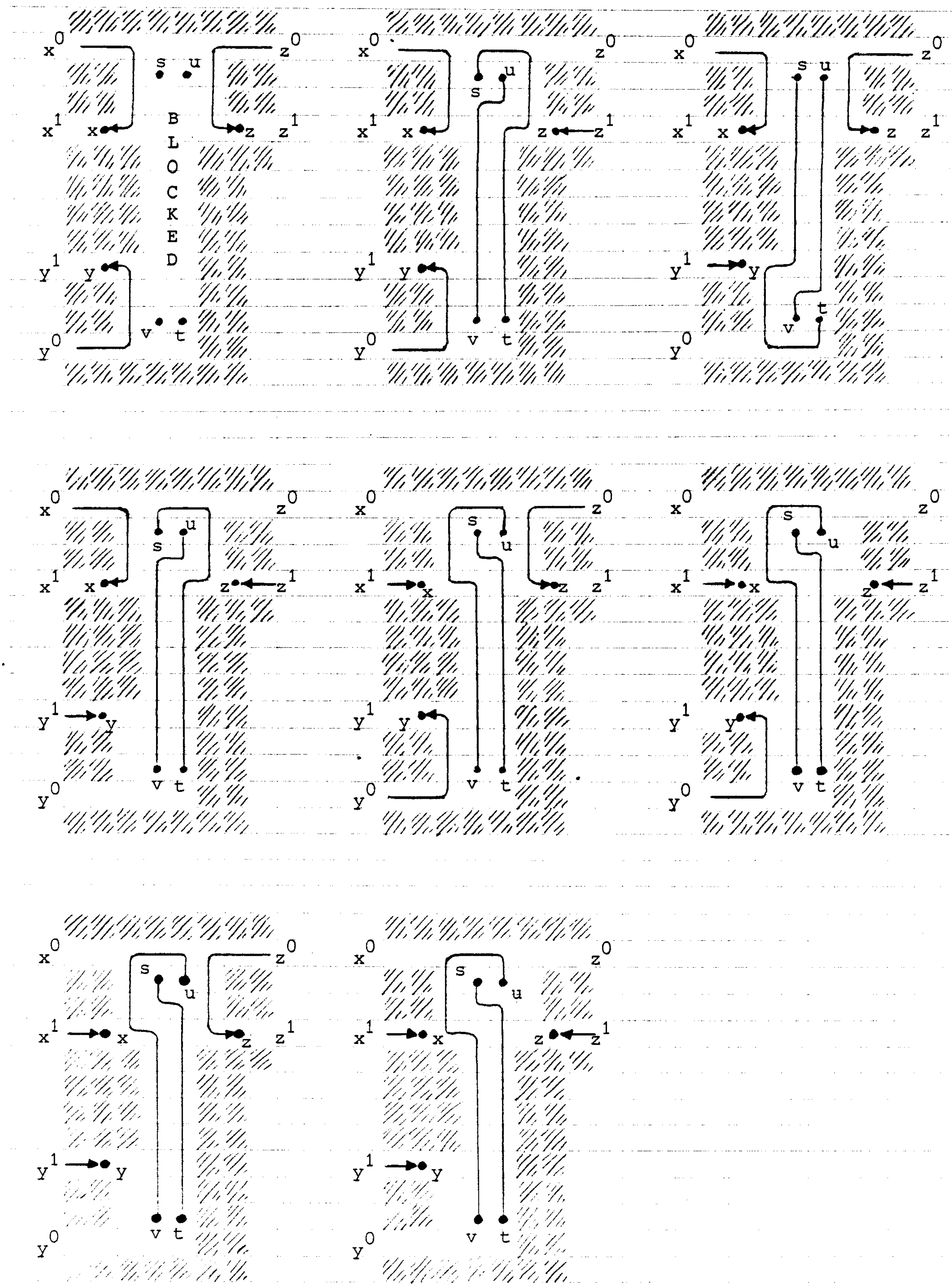


Fig. 7. A plaza for the clause x∨y∨z.

thus appears to be NP-complete even for the very restricted class of (planar) graphs that underly Thompson's model.

3. References.

[1] Fowler, R.J., M.S. Paterson and S.L. Tanimoto, Optimal packing and covering in the plane are NP-complete, Inf. Proc. Lett. 12(1981) 133-137.

[2] Garey, M.R., and D.S. Johnson, Computers and intractability: a guide to the theory of NP-completeness, W.H. Freeman, San Francisco, 1979.

[3] Kung, H.T., B. Sproull and G. Steele (eds.), VLSI systems and computations, Proc. of a CMU Conference, Computer Science Press, Rockville, Maryland, 1982.

[4] Thompson, C.D., A complexity theory for VLSI, Ph.D. Diss., Techn. Rep. CMU-CS-80-140, Dept. of Computer Science, Carnegie Mellon University, Pittsburgh, 1980.