Layout design for bipolar integrated circuits

LAYOUT DESIGN FOR BIPOLAR INTEGRATED CIRCUITS

R.H.J.M. OTTEN EN M.C. VAN LIER

LAYOUT DESIGN FOR BIPOLAR INTEGRATED CIRCUITS

PROEFSCHRIFT

TER VERKRIJGING VAN DE GRAAD VAN DOCTOR IN DE TECHNISCHE WETENSCHAPPEN AAN DE TECHNISCHE HOGESCHOOL EINDHOVEN, OP GEZAG VAN DE RECTOR MAGNIFICUS, PROF.DR. P. VAN DER LEEDEN, VOOR EEN COMMISSIE AANGEWEZEN DOOR HET COLLEGE VAN DEKANEN IN HET OPENBAAR TE VERDEDIGEN OP VRIJDAG 17 DECEMBER 1976 TE 14.00 UUR.

DOOR

MARINUS CHRISTIANUS VAN LIER

GEBOREN TE EINDHOVEN

DRUK: WIBRO HELMOND

DIT PROEFSCHRIFT IS GOEDGEKEURD
DOOR DE PROMOTOREN

Prof.Dr.-Ing. J.Jess
en
Prof.Dr. H.Groendijk

LAYOUT DESIGN FOR BIPOLAR INTEGRATED CIRCUITS

PROEFSCHRIFT

DOOR

R.H.J.M. OTTEN EN M.C. VAN LIER

CONTENTS

PREFACE ix

1. INTRODUCTION

1.1. Objectives 1
1.2. Approach 2

2. BIPOLAR INTEGRATION TECHNIQUE

2.1. The basic steps of the fabrication process 5
2.2. The integrated components 8
2.3. The layout of a circuit 14
2.4. The isolation of the components 16
2.5. The component library 17

3. OUTLINE OF THE DESIGN PROCEDURE

3.1. The potential graph 21
3.2. Considerations for planarization 32
3.3. The isolated regions 37
3.4. Wirability 44

4. PLANARIZATION OF THE POTENTIAL GRAPH

4.1. Introduction 55
4.2. The edge-weighed potential graph 57
4.3. Considerations for the construction of a suitable planarization algorithm 60
4.4. The path-embedding planarization method 65
4.5. The cascade-embedding planarization method 73
4.6. The permutation planarization method 83
4.7. The behaviour of the described planarization methods 85
4.8. Measures to guarantee the extra requirements 91
4.9. Working out the determined set of modifications 94
4.10. Example 96

5. CROSSING DIFFUSIONS

5.1. Searching optimal curves 103
5.2. Example 108

6. DISTRIBUTION OF THE COMPONENTS OVER ISOLATED REGIONS

6.1. Construction of the IR-compatibility classes 111
6.2. Construction of the final island partition 114
6.3. The island interconnection graph 122
6.4. Subsets of island components that are embedded in rectangular regions 129
6.5. Example 132

7. THE DEEP P-DIFFUSION

7.1. The generation of the neighbour relations 139
7.2. The simplex-tableau 143
7.3. Considerations for the implementation 147
7.4. Example 150

8. THE COMPONENTS

8.1. Preliminary remark 153
8.2. Placement of components 154
8.3. The shape of the resistor diffusions 156

9. THE ALUMINIUM MASK

9.1. The wiring procedure 163
9.2. Grid expansion, wire shifting and region shrinking 167
9.3. Admissible interconnection sequences 173
9.4. The interconnections between islands 174

10. CONCLUSIONS AND FINAL REMARKS

10.1. Conclusions 177
10.2. Final remarks 177

APPENDICES

A. A précis of graph theory 184

B. Some planarity testing algorithms 203

C. Searching optimal sequences 211

D. A theory of draingraphs 213

E. Simplex method 220

F. Representation of planar graphs 223

G. The component library 236

PREFACE

In these theses we describe the essential parts of an automatic design of layouts for bipolar integrated circuits, results of a research project running now for five years at the Eindhoven University of Technology.Communicating results in the field of layout design appears to be problematic. Though a thorough analysis of the obstacles in the impartation of this knowledge to others is beyond our competence, a little reflection upon the subject made us believe that the difficulties are mainly caused by two circumstances.

Firstly, the unbelief in the possibility of automatizing layout design, especially for bipolar integrated circuits, is incredibly wide-spread in the world of integrated circuit manufacture. Some ten years ago things were quite different and many research projects with objectives similar to ours were started, but the premature abortion of these projects or the passing on to less ambitious purposes seem to have abated the credit of a such-like project considerably, and now it only meets compassionate headshaking. However, such an attitude is not uncommon when new things are introduced. It was already explicitely stated by Machiavelli: ".....nothing is more difficult to arrange, more doubtful of success, and more dangerous to carry through than initiating changes.....". The second circumstance can also be described in Florentine terms: being a Guelph to the Ghibellines and a Ghibelline to the Guelphs. In developing our concepts we mainly drew from two entirely different fields, namely technology and mathematics or more particularly graphtheory and semiconductor technology, without adding anything new to either field. Consequently, the whole thing seems to be too technological for the mathematician and too mathematical for the technologist. Besides, none of the two is the ultimate addressee. This is the electronic designer, and usage of his special knowledge is only marginal.

Probably the only way of changing the first circumstance is to deliver a program capable of generating feasible layouts for given circuits. As for the second problem we tried to make these theses almost self-

contained. We only assume that the reader is familiar with the rudiments of set theory and relational structures, quantifier and junctor notations, and some basic electronic circuit theory. Bipolar integration technique is reviewed in chapter 2. Some notions are precisely defined in that chapter such that technologists are also advised to read that chapter. Before reading chapter 3 the reader should at least be familiar with the graphtheoretical notations. They are contained in the appendices A and D together with the graphtheory so far as necessary for usage in these theses. Of course referencing to standard texts was the alternative, but

- planarity is mostly treated in the later chapters if at all,
- fixing our notations is necessary anyway considering the lack of unanimity in the standard texts, and
- the very important equivalences of theorem 8 are correctly proved for the first time in this appendix, at least as far as we know.

In appendix B several planarity algorithms are concisely described, because some ideas from these algorithms are worked into the fully automatic planarization method of chapter 4. The basic procedure for finding optimal sequences of entities over which a neighbour relation is defined, and that is used many times throughout the whole design process, can be found in appendix C. It is a more general form of an algorithm that is the best known among layout designers, and commonly referred to as the "Lee-algorithm". The appendices D and E form the background of chapter 7. Appendix F is rather different from the others, since it deals with an efficient generation of representations of planar graphs in the plane, which has no function in a completely automatic design. The motives for still including it are two-fold: it enables one to compare automatic planarization with interactive planarization and it meets the urgent requirement of many applications of planar graphs, namely to have a method of obtaining surveyable representations that can be fastly generated.

Perhaps two words used in these theses deserve some attention, the first one because it cannot be found in any dictionary, and the second one because it has so many different meanings that the one we attach to it must be stated precisely.

fif is a coordinative conjunction linking two statements and expressing the equivalence of these two statements. As most coordinative conjunctions "fif" may be used repetitive in order to link more than two statements.

A *model* of a certain source object is a structure consisting of some entities and one or more relations defined over them which can be uniquely derived from its source object. Thus it is not required that two distinct source objects lead to distinguishable models, which means that a model need not incorporate all aspects of its source object. On the contrary, it should be divested of its irrelevant factors. Which factors are irrelevant depends on the information we want to obtain from the model. Adequacy of a model with respect to a property its source object possibly has, is the quality that a set of requirements can be listed such that the fact that the model satisfies these requirements is a necessary and sufficient condition for the source object to enjoy that particular property. Coherence of a model is a measure for the "density" of its relations. When the relations are almost empty, no really brightening statements about the source object can be derived from the model that cannot be easily recognized in the source object itself. The same is true for very "congested" relations. In useful models coherence is neither too low nor too high; where the useful range of coherence is, depends on our ability to derive information from the model in a practical way.

As to the authorship of the various parts of these theses: the chapters 2,4 and 6, section 8.3 and the appendices B and G are written by M.C.van Lier, the chapters 3,5,7 and 9, the sections 8.1 and 8.2 and the appendices A,C,D,E and F are written by R.H.J.M.Otten. Besides, the author is marked by an initial (L and O respectively) at the bottom of each page.

We are grateful to our colleagues at the Eindhoven University of Technology, especially H.O.Koopmans and F.A.Martis for their programming work, H.Donkers for providing the artwork in these theses and C.C.C.Vogels-Schermeij for typing the manuscript. Further we wish to thank the students that have contributed to the project.

Marinus C.van Lier and Ralph H.J.M.Otten

Eindhoven, The Netherlands

1. INTRODUCTION

1.1. Objectives.

The aim of the research project of which the essential results are compiled in these theses, was to examine the possibilities of complete automatization of layout design for integrated circuits, i.e. a design by a computer without any man interaction starting from data the electronic designer has at his disposal. The resulting layouts must be comparable with manually or interactively obtained layouts as to production costs and performance. The automatic design has two important advantages over other design methods:

1. the layout expert is no longer necessary which means that
   - costs are reduced
   - communication difficulties between the electronic designer and the layout specialist are avoided, and
   - design errors are prevented;
2. the time necessary for the design of the layout is considerably reduced.

Of paramount importance is that no relevant degrees of freedom are lost in the automatized design. This freedom, however, is predominantly determined by the chosen technology. Consequently, making a universal layout program is not possible. We decided to take bipolar integrated circuits, because

- layout design for this kind of circuits was considered to be more difficult to automatize, especially because of the great variety in component structures and the often very irregular pattern of interconnections, and
- these circuits are produced in relatively small numbers (mass manufacture is not paying, because of the diversity in requirements) which makes a fast design method highly desirable.

In these theses we describe procedures that together perform all the essential design tasks. The obvious complement to these descriptions is a working program. The majority of the procedures are implemented and

combined, but placement is not implemented in the way described here, and wiring is being tested at the moment. Since it was not possible to put the program in practice (i.e. in an environment in which these circuits are manufactured), the various parameters could not be properly adjusted and the necessary adaptations could not be determined. Thus, the program is not yet ripe for application in industry, but what should become apparent from these theses is that it is practical to design feasible layouts automatically from data such as the types of components used, the interconnections between these components, and the extra requirements of the electronic designer.

1.2. Approach.

Of course, we were not the first to attempt to automatize the layout design for bipolar integrated circuits. Several groups of investigators, in industry as well as in scientific institutes, put much effort in such a project, but - as far as we know - none of them was successful. The reasons for these failures can be brought down to the following:

1. much effort was lost in striving after elegance and sophistication leading to cumbersome concepts that became hindrances in later stages of the design,
2. the whole problem was serially decomposed into a number of tasks in which the respective problems were treated without taking into account criterions based upon convenience in solving later problems, and this resulted in
   - much freedom in the earlier procedures which left little to go by, and
   - many difficulties in the later procedures, and
3. availability of a graphics display (beside the introduction of interactive means, it makes also the expert layout designer and such an apparatus necessary, and thus the costs of the design are increased).

We therefore use a rather simple structure, a graph, which remains the central entity throughout the design. It can be easily constructed from the data supplied by the electronic designer. In the various procedures this graph is extended, reduced and modified, and the greater

part of the information needed by these procedures is obtained from this graph or its derivatives. Further, we tried to use all freedom available in order to diminish the pressure upon the later procedures of the design. All the problems occurring during the execution of the program are solved as soon as the necessary information is present in the adequate form such that the results can be used immediately in subsequent parts of the program.

It is not worth-while to go further into the structure of the program. Its outline is given in chapter 3. The various subtasks and the information flow are represented in the scheme of fig. 10.1.

2. BIPOLAR INTEGRATION TECHNIQUE

2.1. The basic steps of the fabrication process.

The fabrication of a great number of identical bipolar integrated circuits starts with slicing and polishing a wafer of p-type silicon, which will serve as the substrate of the circuits.
Next, an n-type "epitaxial" layer is grown on the surface of the wafer. The crystal structure of the substrate is continued in the epitaxial layer by passing a gas containing both silicon atoms and n-type impurities over it in a heated environment.
In the next steps a number of diffusions are performed into the epitaxial layer (p-type or n-type). With the first of these diffusions, the socalled deep p-diffusion, channels are created, that reach the p-type substrate. Into the regions of n-type epitaxial material, generated by this deep p-diffusion, p- and n-type impurities are diffused, thus realizing the other p-type and n-type layers of which the integrated components consist.
The diffusions only have to take place on certain areas of the wafer, and therefore a photolithographic process is applied before the execution of each diffusion step. This process involves the following actions (see fig. 2.1):

- a thin silicon dioxide coating is grown by exposing the wafer to an oxygen atmosphere of about 1000° C.
- this silicon dioxide layer is covered with a special kind of photosensitive emulsion (photoresist)
- the wafer is locally exposed to ultra-violet light. This is done by placing a photographic mask on top of the photoresist. The transparant windows of the mask determine the areas that will be exposed to the ultra-violet light
- the unexposed parts of the emulsion are dissolved, thus leaving selected areas of the silicon dioxide uncovered
- in the uncovered areas the oxide is etched away
- the photoresist is removed.

After this process a part of the wafer is covered with silicon dioxide, which acts as a barrier to the diffusion of dopants. Each

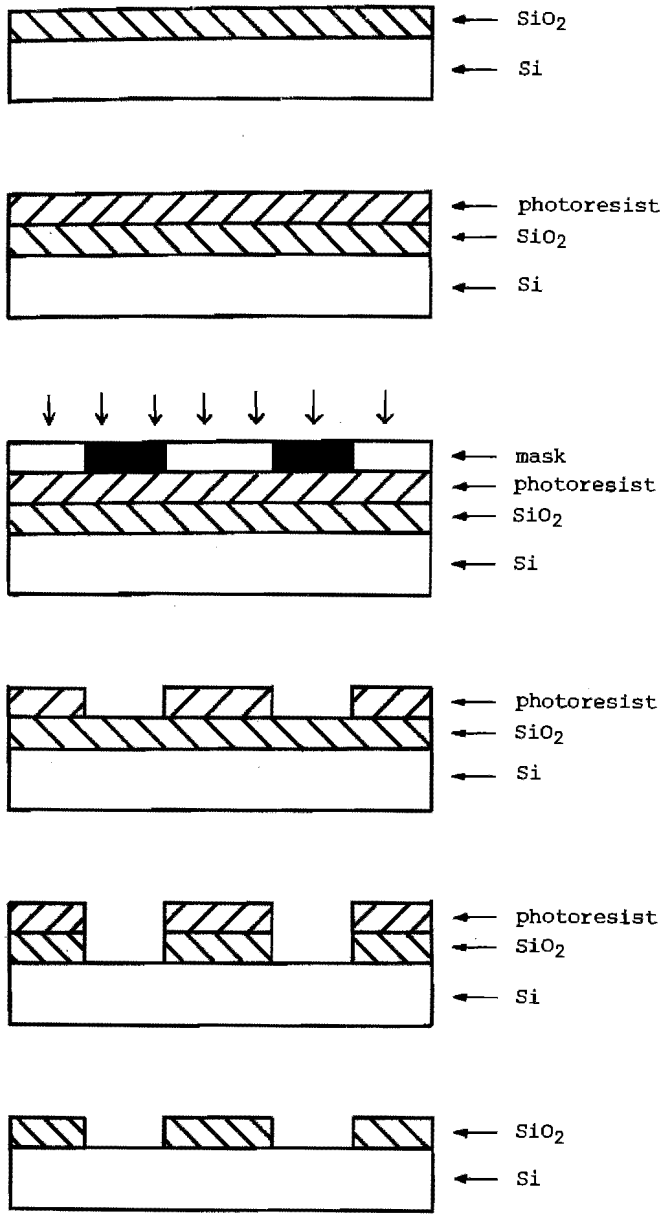


Fig. 2.1. The photolithographic process.

diffusion step has its own "process parameters", determining the diffusion profile, i.e. the "depth" of the diffusion layer and its concentration of impurities.
In order to isolate the completed (pn-structured) semiconductor material from the aluminium interconnection pattern that will be added in the last step, a new SiO_2 layer is grown and the contact holes are etched in this layer. Then a thin film of aluminium is evaporated over the wafer; certain areas of this aluminium are thereupon etched away in order to obtain the desired pattern for component interconnection. After testing the completed circuits on the wafer, they are separated into individual "chips" by scribing and breaking the wafer. Finally the chip is "packaged": it is bounded to a lead frame, its bonding pads are wired to the pins of its house, and the house is hermetically sealed or encapsulated.

2.2. The integrated components.

In this section we will discuss the component structures that are commonly used in realizing a circuit with the described manufacturing process.

npn-transistors

The process parameters of the diffusions in the fabrication process are chosen such that a good performance of the npn-transistors is obtained. In fig. 2.2. the configuration of an npn-transistor is depicted. The n-type epitaxial layer serves as the collector region of this transistor. The base region is obtained by a shallow p-diffusion in the epitaxial layer, and the n^+-type emitter region is diffused into the p-type base region. The fact that the impurity concentration of the emitter diffusion is much higher than that of base and collector region, is indicated by the plus sign.
In order to make an ohmic contact with the collector region, the area under its contact hole is also made n^+-type. The pn-junction between the collector region and the substrate is reverse-biased by giving the substrate the most negative potential. The capacitance and breakdown voltage of this junction is optimized by using highly resistive material for the substrate. It is possible to reduce the collector series

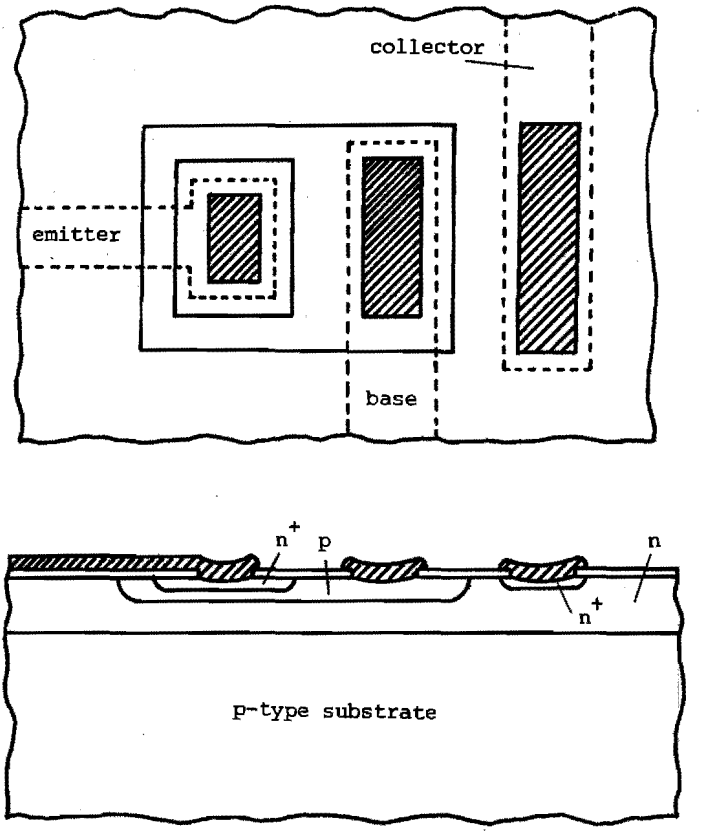


Fig. 2.2. Top view and cross section of an integrated npn-transistor.

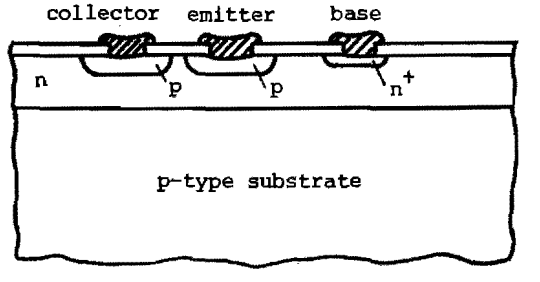


Fig. 2.3a. Cross section of a lateral pnp-transistor.

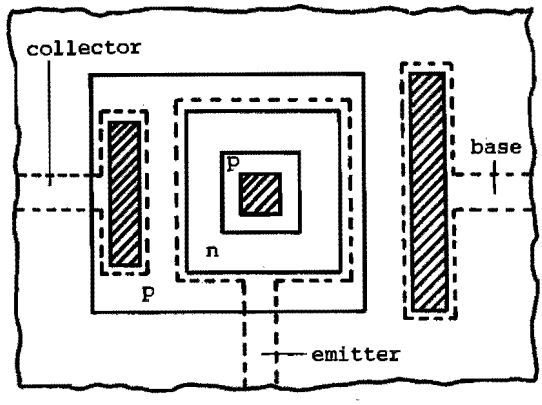


Fig. 2.3b. Top view of an improved lateral pnp-transistor.

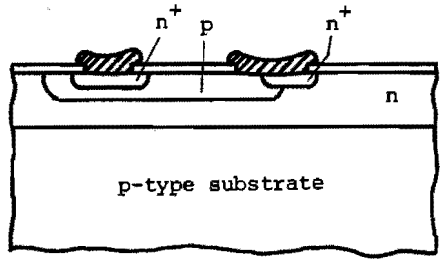


Fig. 2.4a. Cross section of a base-emitter diode.

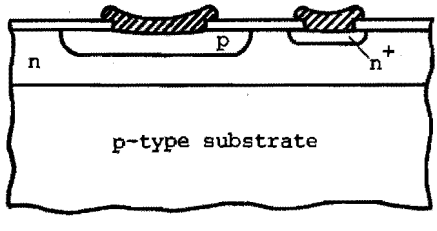


Fig. 2.4b. Cross section of a collector-base diode.

resistance by extending the basic process and placing an extra n^+-diffusion in the substrate (before growing the epitaxial layer) over the whole collector area of the transistor ("buried layer"). More diffusion steps may be added in order to create special devices (e.g. an extra "side wall" diffusion to reduce the collector resistance further, or an extra emitter diffusion to narrow the base width (and thus achieving a high current gain).

pnp-transistors

With the same p-type diffusion ("base diffusion") and n^+-type diffusion ("emitter diffusion") that have been used for the construction of the npn-transistors, it is possible to construct a pnp-transistor of the "lateral" type. A cross section of such a transistor is shown in fig. 2.3a.
Since the average base width of this pnp-transistor is very large its gain is rather small. The effective base width can be reduced by shaping the collector such that it surrounds the emitter completely (fig. 2.3b).
The gain can be increased further by reducing the downward injection introduced by the parasitic pnp-transistor having the substrate as collector. This can be achieved by placing a "buried" layer between epitaxial layer and substrate.
A different type of the pnp-transistor is the "substrate pnp-transistor", which is formed by the base diffusion, the n-epitaxial layer, and the substrate. With this type a better performance can be obtained, but then the control over the epitaxial layer thickness has to be tighter, since it is directly related to the effective base width of the transistor. Therefore it might be recommendable to avoid substrate transistors completely. Since the substrate is connected with the most negative potential in the circuit, this substrate transistor can only be used for transistors of which the collector is connected with this potential.

diodes

For the construction of the diodes the base and emitter diffusions already mentioned can be utilized. Those yield:
- the base-emitter diode (fig. 2.4a).

Since the collector region is also present, its bias is of importance. The base and collector region are usually connected. If the diode is forward-biased, the component works like an npn-transistor in its active region. In reverse-biased state the breakdown voltage for this type is about 6 volts, so that for higher reverse operating voltages we have to use the base-collector junction:

- the base-collector diode (fig. 2.4b).

This diode has the disadvantage that if it is forward-biased, the parasitic pnp-transistor is in its active region. The gain of this pnp-transistor can be reduced by addition of the buried layer.

resistors

Resistors can be made by using the resistive nature of doped semiconductor layers. It is most common to use the base-diffused layer, since its resistivity is convenient, and the tolerances and temperature coefficients are acceptable ("p-type diffused resistor"). The high impurity concentration of the emitter diffusion makes it only suitable for the realization of resistors with very low values (n^+-type diffused resistors).

The value of a resistor is of course dependent on the length, width and depth of its resistance layer. A convenient measure of the value of a diffused resistor (if the resistivity of the material and the thickness of the layer is fixed), is the "sheet resistance" in ohms per square: the resistance of a square area is independent of the length of its sides.In order to obtain the required value of a resistor, the appropriate number of "squares" has to be added together in series between its contacts. In designing large-valued resistors one often applies bends in the resistor path (see fig. 2.5). Due to current crowding around the inside corner, one has to correct the ("effective") contribution of this bend to the total resistor value.

The (design of the) geometry of a diffused resistor with a large value, is very flexible. This flexibility is used to fit the resistors in the regions assigned to them, even if the regions have a somewhat uncommon shape. As a consequence the resistor diffusion may be given a rather intricate shape. The determination of this shape is called the "meandering of the resistor".

In order to avoid parasitic transistor activities we embed the resistor

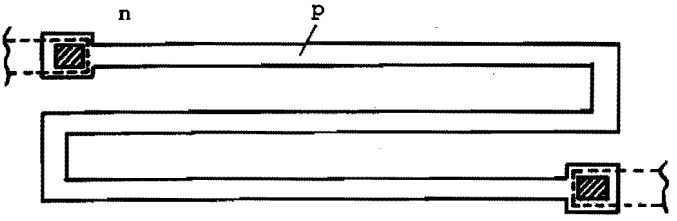


Fig. 2.5. Top view of a typical (meandered) p-diffused resistor.

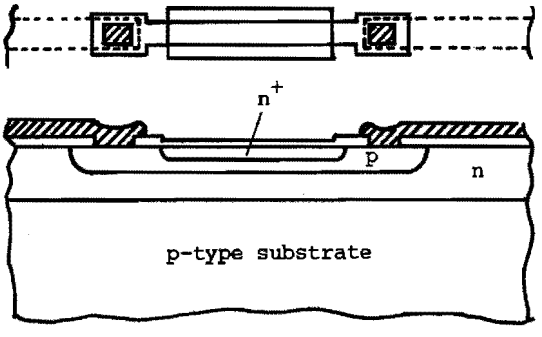


Fig. 2.6. Top view and cross section of a base-emitter pinch resistor.

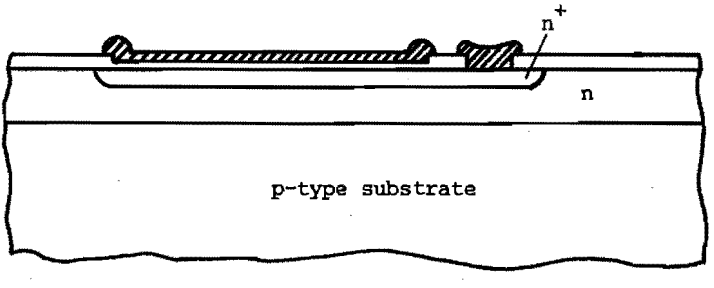


Fig. 2.7. Cross section of a metal-oxide-silicon capacitor.

diffusion in a layer having a potential such that the intermediate pn-junction is reverse-biased.
It is possible to reduce the relatively large area occupied by (large-valued) resistors, by realizing them as "pinch resistors" (see fig. 2.6). This is a base-diffused resistor of which the cross section is drastically decreased by placing the emitter diffusion over it. However, this type of component has some properties which restrict its application: it has a non-linear voltage-current characteristic, its breakdown voltage is very low, its temperature coefficient is rather high, and a large tolerance has to be allowed.

capacitors

It is possible to use the reverse-biased pn-junction as capacitor, but its performance is voltage-dependent, and the capacitance per unit area is low.
A device having somewhat better properties is the "metal-oxide-silicon" capacitor. The top electrode is formed by the aluminium interconnection layer, while the n^+ emitter diffusion acts as bottom electrode (see fig. 2.7). The silicon dioxide layer serves as a dielectric, and has to be kept thin in order to provide a suitable capacitance per unit area. Therefore the application of this device requires the etching of the oxide in the capacitor area, and the growing of a new, thin oxide layer.

From the discussion of the component structures it has become apparent that there are two diffusion steps (the p-type base diffusion, and the n^+-type emitter diffusion) that are indispensable for the realization of the different components. Some of the properties of the devices can be improved by making extra (diffusion) steps in the fabrication of the device. In order to symplify the text and the pictures in these theses we will not mention and depict the extra mask configurations that have to be used if it is decided to employ these extra steps. The application of these (extra) steps does not essentially affect any algorithm of the design procedure to be described.

From the discussion so far it is clear that the process parameters have big influence on the properties of the different components. We

will assume that all the process parameters are fixed and will refer to the described process, having these fixed parameters, as the "standard process".
In the standard process five masks have to be used in executing the following indispensable (photolithographical) steps
- the deep p-type diffusion (DP)
- the shallow p-type diffusion (SP)
- the shallow n^+-type diffusion (SN)
- the etching of the contact holes (CO)
- the etching of the aluminium interconnection pattern (IN).

The configurations of these masks determine the "layout" of a chip that has been manufactured by the standard process.

2.3. The layout of a circuit.

An electrical circuit specification is any collection consisting of the following items:

1. a set of components, specified either by reference to a model stored in a library, or completely described in terms of parameters of the applied process;
2. a partition over the set of component contacts, which is such that all the contacts contained in the same block, should always be at the same potential;
3. a predescribed circuit performance with certain tolerances.

When a circuit is integrated by the process described in the preceding sections, the components are usually realized completely by the part that consists of semiconductor material, whereas the interconnections, that provide the potential correspondences between the contacts in the same block, can be recognized in the aluminium configuration. For this reason the part of the integrated chip that is separated from the aluminium by the silicon dioxide layer is called the "component layer", and the part that consists of the aluminium interconnections is called the "wiring layer".

Remark: Distinguishing between the two layers on the described grounds is not always correct. On the one hand, the component layer is

used for the accommodation of a small diffusion conductor, the "crossunder", which makes it possible that leads cross each other without making contact. On the other hand, the aluminium layer is sometimes used as a part of a component as for example in case of a metal-oxide-silicon capacitor.

The layout of a circuit is any set of data that completely specify all the masks necessary for integrating the circuit such that:
1. all the components of the circuit can be recognized;
2. all the potential correspondences are realized by conducting leads;
3. the circuit behaves within the given tolerances.

The special characteristics of the applied manufacturing process make that the design of the circuit (both the electrical diagram and the layout) is highly dependent on the (technological) factors that have an influence on: - the electrical performance of the integrated circuit
- the manufacturing cost of the integrated circuit.
An important design objective is the minimization of the chip area, since the overall yield of chips decreases if the occupied area increases due to the occurrence of defects in the semiconductor crystal. The circuit designer will therefore try to avoid components that require relatively large areas on the chip such as large-valued diffusion resistors and capacitors. Some other factors that may have consequences for the design of the electrical circuit, are:
- the tolerances of the resistors are rather high, and therefore the electrical behaviour (e.g. the biasing of the active components) should not be critically dependent on the values of these resistors;
- "matched components" can be realized more easily. By placing the components close to each other and in the same orientation, this "matching" can even be improved ;
- non-ideal properties of components (parasitic effects, temperature effects, breakdown voltages, etc. have to be taken into account, too).

The applied technology also has its consequences for the design of the layout. One has to strive after minimization of the chip area (as has been mentioned). But also other requirements may establish constraints on the layout design:

- for connecting the circuit to the pins of the housing it is advantageous to have the "bonding pads" on the periphery of the chip. An additional reason for doing this is to have the components on the chip not placed too far from each other, and thus obtain low temperature differences between the components;
- there may be several demands from the circuit designer that are important for an adequate electrical performance of the circuit. One may wish certain components to be placed close to each other (matching) or far from each other (thermal effects, parasitic coupling). For certain component contacts it may be important to have very little loss of potential along the interconnection lead (no crossunder!), or between certain potentials capacitive coupling must be avoided (input and output potential). In general the length of the interconnection leads is to be kept as small as possible;
- one has to exploit possibilities to reduce the total chip area, e.g. the total area occupied by the isolation diffusion can be reduced by using the minimum number of isolated regions that is possible, and by giving these regions an approximately squared form. Furthermore one has to be careful with measures that consume extra area, such as: the application of crossunders, increasing the width and length of a (low-valued) resistor in order to move its contacts farther from each other, and realizing a resistor as diffused resistor, even though the design would admit a pinch resistor;
- standardization rules may dictate that the terminals of the integrated circuit occur in a previously specified sequence;
- it may occur that the designer requires to use a special, predesigned layout for some part of the electrical circuit. The "terminal"-potential-leads leave this "layout-block" in a certain sequence. This constrains the layout of the rest of the circuit.

2.4. The isolation of the components.

Isolation between the components can be obtained by pn-junctions that are reverse-biased. The substrate is connected with the (most) negative supply voltage and thus its junction with the epitaxial layer will always be reverse-biased. The same holds for the junction between the deep p-diffusion channels (reaching the substrate) and the epitaxial layer.

By surrounding a certain region of the chip by a deep p-type "isolation channel" it is possible to isolate this region from the rest of the chip. Such an isolated region (IR) is called an "island", and the "island potential" is defined as the potential of its n-type epitaxial layer.
Not all components have to be placed in separate islands. The voltage states of the electrical circuit at any moment may be such that for certain components, even if they are part of the same island, the required electrical isolation is automatically guaranteed. Such components are called "IR-compatible". For example, an npn-transistor of which the collector is connected with the (most) positive supply voltage is IR-compatible with any p-type diffused resistor. The components of the circuit have to be distributed over a number of isolated regions, and thus a partition of the set of components has to be determined.
The components of which the epitaxial layer is a part of their structure, determine the island potential of the isolated region they are in, and are called "epitaxial components" (EP-components).
EP-components cannot be embedded in the same island if their epitaxial parts are to be connected with different potentials.
Components of which the n-type epitaxial layer is not a part of their structure are called "non-epitaxial components" (NEP-components).
For all voltage states of the circuit, the potential of their fundamental layer (i.e. the layer that, on the bottom side, is surrounded by the epitaxial layer) has to be less than or equal to the island potential.
These rules determine a number of maximal sets of components of which the elements are mutually IR-compatible (the "IR-compatibility classes" [2.2]).
The freedom which is left after observing the rules, can be used to satisfy other desires (see chapter 6).

2.5. The component library.

In section 2.2. we have seen that the properties of the various components are rather dependent on the geometries of the different parts in the component structure.
The design of the typical diffusion patterns of the components, and

their dimensions, are in some ways constrained. First of all some of the desired component properties can demand a special form (rectangular or circular, length/width ratio) or a minimum area (adjusted to handle the expected current) for some of the diffusion regions of the component pattern. Furthermore, it is advantageous to minimize the total area that is occupied by the component (yield, parasitic effects). However, due to tolerances in the mask alignment and in the etch and diffusion processes, a number of requirements on the minimum spacings between the different mask patterns, and on the minimum widths of certain of these mask configurations, have to be satisfied. These requirements are fixed in a so-called "clearance list".
For the convenience of the designer a "component library" is arranged, containing the geometrical information of a series of "standard components". It enables the user to give simple definitions of the integrated components that are desired in the realization of the electrical circuit. The geometry of such standard components, and their properties need to be considered and determined only once, and adequate analysis and breadboard models can be made available for a performance analysis of the electrical circuit and the layout that is designed.

For some component types (for example the resistors), there are only very few design rules in the clearance list that have to be satisfied in designing their geometrical configuration. Furthermore, their actual shape is very flexible, and does not influence the electrical performance very much. Therefore, for these components, only few parameters have been fixed in the library (e.g. sheetresistance, meander spacing, shape of a contact, etc.). Their actual configurations have to be determined in a separate procedure. The flexibility in the design of their shape can be used to satisfy other design objectives.

The library has to be easily extendable with "new" standard components such that, if necessary, the circuit designer has the freedom to utilize or create new component configurations having special properties. Appendix G gives the geometrical parameters of some standard components which are contained in our library. This data is stored and manipulated by sequences of line and circle segments.

References.

[2.1] H.R.Camenzind, "Electronic Integrated Systems Design", Microelectronics Series, Van Nostrand Reinhold Company, New York, 1972.

[2.2] D.Ferrari, "On the selection of isolated regions in computer aided design of integrated circuits", IEEE Transactions on Circuit Theory, CT-17, pp. 134-136, 1970.

3. OUTLINE OF THE DESIGN PROCEDURE

3.1. The potential graph.

A direct translation of the ideas on integration reviewed in the preceding chapter leads to a more or less stylistic layout. We start with an inquiry into the existence of such a layout, taking into account certain constraints if required. We therefore give a precise definition of this layout.

A layout of a circuit is called a formal layout if

A1: there is a component layer in which every component has its own domain without overlapping domains of other components,

A2: there is a wiring layer in which all interconnections between components are realized and no interconnection path crosses a component,

A3: every component contact is made by exactly one contact hole which is the end of an interconnection path,

A4: all contacts of a component can be reached simultaneously from given points at the boundary of its domain by interconnection leads.

Clearly, the requirements for a formal layout are too rigid to be conclusive about the existence of a layout of a given circuit: components are not allowed to share certain diffusions, an interconnection lead only enters a component domain to reach a contact hole, a diffusion is contacted via exactly one hole in the silicon dioxide layer, etc.. However, this concept gives a convenient starting point for the discussion on the existence of a layout for a circuit. Before entering this discussion, we make some remarks about A4. This requirement is included in the definition to make sure that, no matter from which direction the component domain is approached, if it can be reached, the proper contact can also be reached. This is necessary because of the finite distances between contacts and the width of the aluminium interconnection leads. Thus A4 (and partly also A3) is a restriction on the types of components that are allowed in the circuit. For the moment we

consider desires concerning components as special requirements and later on we shall take these into account, by dropping A4 and introducing some constraints.

The problem now is to formulate a criterion for the existence of a formal layout of a given circuit. The available data consist of a list of components and their mutual connections, i.e. which contacts of which components must always be at the same potential. These data are usually conveniently displayed in a schematic diagram, where components are represented by suitable symbols and the interconnections by trees, the "potential trees". These entities are injectively mapped onto a set of vertices: α assigns to every component of the circuit a c-vertex, and β assigns to every potential of the circuit a t-vertex. The set of c-vertices is denoted by G_C and the set of t-vertices by G_T. $G_T \cap G_C = \emptyset$. Also the relation that a component $c\alpha^{-1}$ has one of its contacts at a certain potential $t\beta^{-1}$ is easily recognized in the schematic diagram. In such a case we write $c \perp t$.

Remark: In the description so far the notion of an incidence structure [3.1] or hypergraph [3.2] forces itself upon us, for consider the components as points, the potentials as blocks and the elements of $\alpha \perp \beta^{-1}$ as flags. Indeed some authors have thought of this as the adequate formulation of the problem, but they only introduced a cumbersome concept without a single advantage over the following graph-theoretical approach [3.3].

The elementary potential graph (G,U) of a circuit is the graph with $G = G_C \cup G_T$ and $U = \{[g_1,g_2] \mid g_1 \perp g_2\}$.

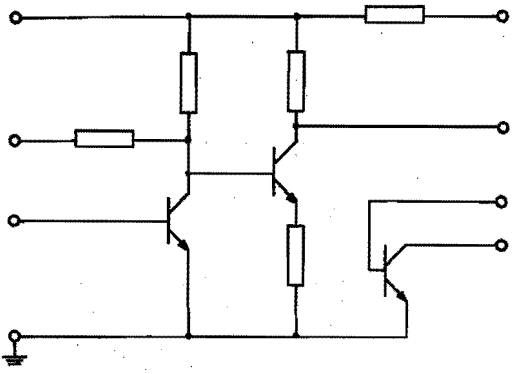
The most important property of this graph is that its planarity is necessary and sufficient for the existence of a formal layout of the circuit concerned.

Suppose we have a formal layout. Draw in every aluminium configuration a tree with vertices of degree 1 in the contact holes and in which the sum of the lengths of its edges is minimum. If terminal contacts are interconnected with only one component, the tree must have a vertex in this contact. Internal edges are edges $[x,y]$ with $x\gamma_U \neq 1$ and $y\gamma_U \neq 1$. Draw in every component domain a closed Jordan curve which has

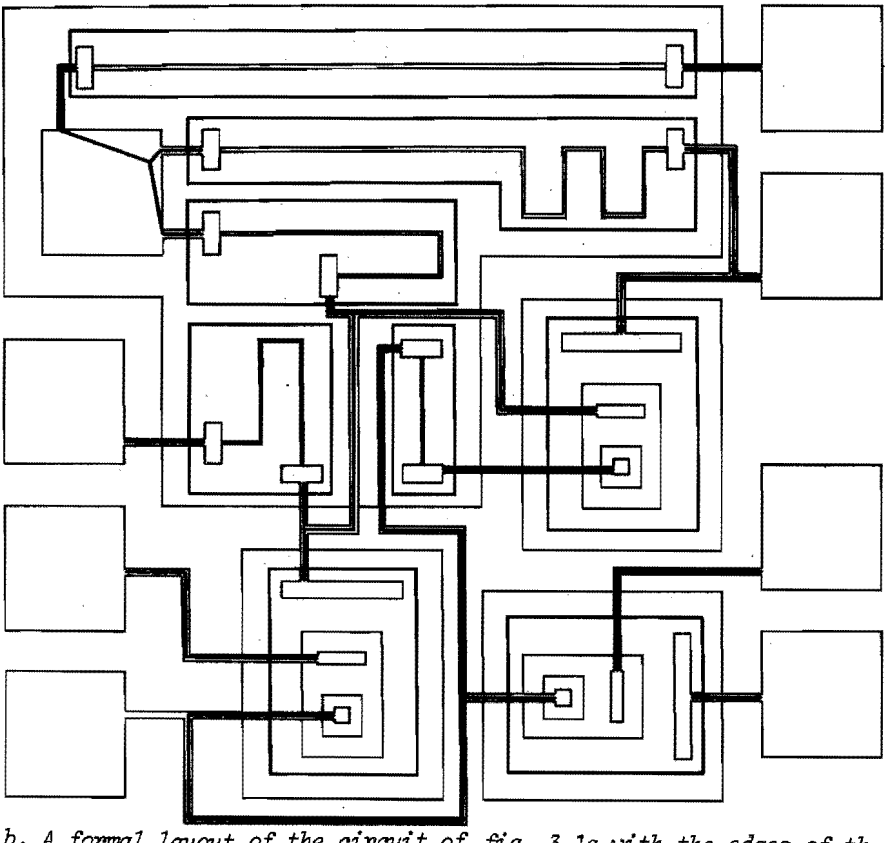
all the contacts of this component in its interior. With every component now corresponds a circuit consisting of as many edges as contacts and an equal number of edges associated with a vertex of degree 1 in its interior: we call these edges domain edges. In this way we obtained a plane graph and after contraction of all its internal edges and all its domain edges this graph is isomorphic to the elementary potential graph of the circuit. Since a planar graph remains planar when some of its edges are contracted, the elementary potential graph is planar (fig. 3.1).

Conversely, suppose that the potential graph (G,U) is planar, i.e. there exists a plane representation (G^*,U^*) of it. For every $c^* \epsilon G^*$ there exists a (maximal) real number r such that $\Omega_r(c^*)$ contains no element of $G^* \setminus \{c^*\}$ and $\Omega_{\frac{1}{2}r}(c^*)$ contains no point of $S_{(G^*,U^*)}$ except points in $\{x^* | \exists_{y^* \epsilon c^* \rho_{U^*}} [x^* \epsilon [c^*,y^*]]\}$, the so-called star of c^*. Further, with every component there exists a (minimal) disk with radius R in which the component fits. Let the desired width of the aluminium interconnections be D_1 and the required distance between two interconnections be D_2. For every c-vertex c^* we now can calculate a number $F(c^*) := 2(R+c^*\gamma_{U^*}(D_1+D_2))/r$. Let the shortest distance between two points belonging to the stars of two distinct t-vertices and to the exterior of all neighbourhoods $\Omega_{\frac{1}{2}r}(c^*)$, c^* being a c-vertex, be d. Choose the greatest number of the set $\{F(c^*) | c^* \epsilon G_C^*\} \cup \{2(D_1+D_2)/d\}$ as a factor for blowing up $S_{(G^*,U^*)}$ and thus obtaining (G_1^*,U_1^*). Place the components in the neighbourhood $\Omega_{\frac{1}{2}r_1}(c_1^*)$ of the corresponding c-vertex, design a wiring that contains all points of $S_{(G_1^*,U_1^*)}$ outside these neighbourhoods and make the necessary contacts by interconnecting the contact holes with the reached point of the boundary of the neighbourhood (fig. 3.1).

Apart from the one-layer constraint there may be some other requirements, emanating from standardization and performance considerations. For example, some components must be placed close to each other with identical geometry and orientation to obtain only small differences in the influence of temperature and mask-alignment errors on those

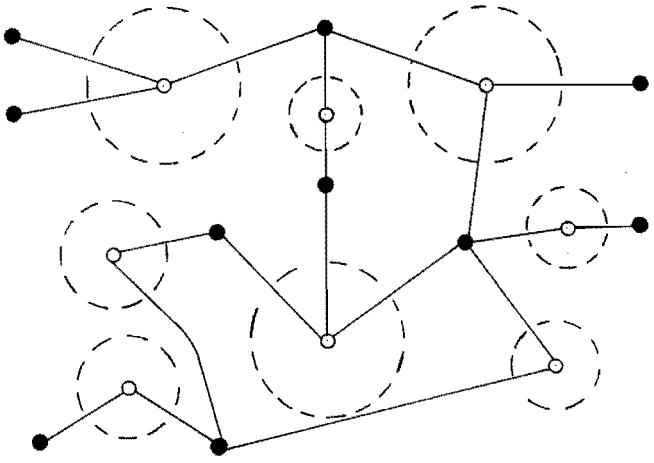


a. A schematic diagram.

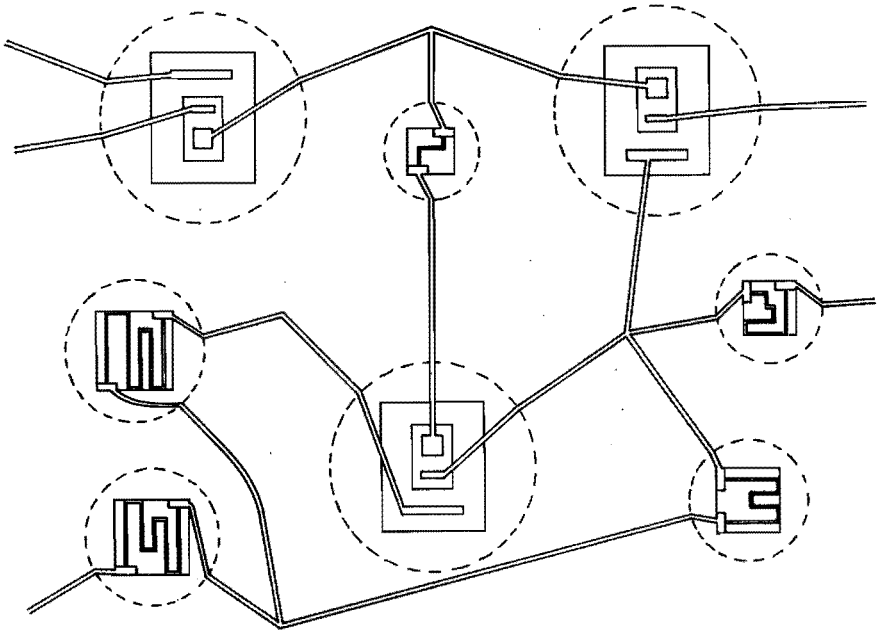


b. A formal layout of the circuit of fig. 3.1a with the edges of the potential graph, the internal edges and the domain edges.

Fig. 3.1. The meaning of the potential graph.



c. The potential graph.



d. A formal layout of the circuit of fig. 3.1a obtained by scaling the representation of the potential graph in fig. 3.1c.

components. Because of these same thermal effects there is some advantage in keeping the bonding pads at the periphery of the chip; otherwise components are placed unnecessarily far from each other. Besides, bonding becomes much easier, when the pads are along the edges of the chip. Often the sequence in which the pads occur at the periphery is specified by the designer for reasons of convention or interchangeability with some other integrated circuit. Sometimes the geometry of a certain component is such that the interconnection leads must enter the domain in a special sequence to reach the contact holes.

The question now is, whether it is possible to extend and/or modify the elementary potential graph of a given circuit, such that the planarity of this new graph is necessary and sufficient for the existence of a constrained formal layout, i.e. a layout satisfying A1, A2, A3 and one or more of the requirements mentioned in the above paragraph. Three of them are restated in B1, B2 and B3:

B1: Bonding pads are to be placed in the border of the chip.

B2: Bonding pads are to be placed along the edges of the chip in a specified sequence.

B3: Interconnection leads are to arrive in the domain of a certain component in a special sequence.

It is obvious that we can satisfy B1 fif there exists a plane representation of the elementary potential graph for which the t-vertices corresponding with the potentials that must be available at the terminals of the chip, are all on the boundary of some face. In other words there exists a formal layout only constrained by B1 fif the elementary potential graph is H-accessible, H being the set of t-vertices described in the preceding sentence. In order to examine the graph upon the desired properties we extend the elementary potential graph (G,U) by adding a vertex $h_o \notin G$ and edges between h_o and the vertices in H, thus obtaining the graph (G_1,U_1). We now assert that the elementary potential graph (G,U) is H-accessible fif $(G_1,U_1) = (G \cup \{h_o \mid h_o \notin G\}, U \cup \{[h_o,h] \mid h \in H\})$ is planar.

Suppose that (G_1,U_1) is planar, and (G_1^*,U_1^*) is a plane representation of it. A face w containing h_o^* contains two of the new edges, $[h_o^*,h_1^*]$

and $[h_o^*,h_j^*]$, and thus h_i^* and h_j^* can be connected by an open Jordan curve which, except for the images of 0 and 1, is contained in w. The Jordan curves, thus added, form together a closed Jordan curve with h_o^* either in its interior or in its exterior. This curve and the subset containing h_o^* can be mapped onto a circular disk by a topological equivalence ϕ and since any point of the boundary of this disk can be connected with the center of the disk, their images under ϕ^{-1} form the open Jordan curves of which the existence proves the fact that the points of H^* are on the same face boundary.

Conversely, if (G,U) is H-accessible, it has a plane H^*-periphere representation (G^*,U^*). Thus, the vertices in H^* all occur on the face boundary of some face w of (G^*,U^*) and they can be connected with an arbitrary point x^* in w by open Jordan curves which are disjoint except for the point x^* itself. Consider x^* as a new vertex and the curves as new edges and we have a plane representation of (G_1,U_1).

B_2 and B_3 can be treated in analogous ways, after the elementary potential graph is extended as in the case of B_1 (B_1 is a subcase of B_2!). The corresponding requirements for the elementary potential graph are expressible in terms of face and wheel consecutivity, but by starting with (G_1,U_1) both come to wheel consecutivity requirements. Let us write $t_1 \sim t_2$, $t_1 \in G_T$ and $t_2 \in G_T$, if their images must be consecutive in a representation. We propose the following extension of the elementary potential graph (G,U):
in case B_2 must be observed we start from (G_1,U_1) and we add an edge $[t_1,t_2]$ for every "consecutivity relation" $t_1 \sim t_2$, if only requirements of B3-type must be observed we start from (G,U) and add also an edge $[t_1,t_2]$ for every element of the "consecutivity relations", $t_1 \sim t_2$. Thus we obtain (G_2,U_2). The new edges are called consecutivity edges. Considering the definitions of consecutivity it is not difficult to see the truth of the statement: there exists a plane representation of the elementary potential graph (G,U) satisfying a face consecutivity relation on $H \subset G_T$ and wheel consecutivity relations for some components fif $(G_2,U_2) = (G \cup \{h_o \mid h_o \notin G\},\ U \cup \{[h_o,h] \mid h \in H\} \cup \{[t_1,t_2] \mid t_1 \sim t_2\}) \in PL$.

Another possible requirement was mentioned in this section, namely that some components must be laid out close to each other and with the same - often special - geometry. This problem can be adequately solved by considering these components together as one "supercomponent"; its layout must be among the input data of the design procedure. This supercomponent must often be treated with the B3-constraint.

The graph in which all these requirements are incorporated in the way described in the above paragraphs is called the potential graph of the circuit. Its planarity is necessary and sufficient for the existence of a constrained formal layout. Before we give an example of the construction of the potential graph for the μA 725 (fig. 3.2), a few remarks are in order. Firstly, the elementary potential graph is a bipartite graph, and the potential graph with consecutivity edges is not. If there is some advantage in keeping the graph bipartite, one should replace every consecutivity edge by two edges and a vertex of degree 2. The simple graph of the potential graph and the simple graph of this "bipartite potential graph" are isomorphic to each other. Thus the bipartite potential graph is planar fif the potential graph is planar.

From the necessity and sufficiency of the condition for the existence of a constrained formal layout, we conclude that the potential graph is an adequate model in examining this existence. But it is not the graph with minimal complexity having this feature, (complexity being the number of edges divided by the number of vertices) if $(\overline{G},\overline{U})$ is 2-connected, which is the only practical case. For if (G,U) is 2-connected and H-accessible, $H \subseteq G$, there is only one face consecutivity relation on H possible.

Since this statement is obvious for $|H|<4$, we suppose $|H|\geq 4$. Further, we assume that there are two H-periphere plane representations of (G,U), (G^*,U^*) in which h_1^* and h_2^* are face consecutive, and (G^{**},U^{**}) in which the corresponding vertices h_1^{**} and h_2^{**} are not face consecutive. This means that there is a path $(P^*,P^*[h_1^*,h_2^*])$ in $(C_W^*,C_W^*[\ ])$ in which no element of $H^*\setminus\{h_1^*,h_2^*\}$ is contained. $(P^{**},P^{**}[h_1^{**},h_2^{**}])$ is the corresponding path in (G^{**},U^{**}), in which

at least one edge is not in the same face boundary as H^{**}. $(P_1^{**},P_1^{**}[h_1^{**},h_2^{**}])$ and $(P_2^{**},P_2^{**}[h_1^{**},h_2^{**}])$ are two paths forming this face boundary. Suppose $h_3^{**} \epsilon P_1^{**}$ and $h_4^{**} \epsilon P_2^{**}$. In (G^*,U^*) we can easily find a path $(P_3^*,P_3^*[h_3^*,h_4^*])$, not containing a vertex of P^*, for example in $(C_w^*,C_w^*[\ ])$. In (G^{**},U^{**}) such a path cannot exist, since it would be alternating with the part of $(P^{**},P^{**}[h_1^{**},h_2^{**}])$ which is not on the face boundary on which H^{**} is in some subgraph of (G^{**},U^{**}) with respect to this face boundary.

This observation makes the usefulness of the extension of (G,U) with consecutivity edges questionable. Nevertheless we maintain these edges, because the potential graph is in general not planar. This means that we have to change our primordial ideas about the integration of the circuit, but we have to preserve the consecutivity requirements. The presence of these consecutivity edges makes sure that we take these requirements into account.

The fact that the potential graph never is planar, may raise the question why we introduced this model at all. As mentioned before, the existence of a formal layout is not necessary for the existence of a layout of the circuit, since the many possibilities technology leaves us are not included in the model. The reasons for starting with the potential graph and not with a model in which all possibilities are incorporated, are:

(1) the optimal solution will be shown to be very close to the formal model, i.e. the number of modifications applied to the original concept is much smaller than the total number of admissible modifications

(2) a model that approaches the problem from the other side, cannot be expected to be a manageable structure.

The latter statement is of course,not easily substantiated, since it is impossible to examine all practicable models. Beside its plausibility, there are the attempts with other models registrated in literature. Even a model incorporating only one of the many kinds of modifications, namely crossing resistors with interconnection leads, turned out to be so incoherent, that it could not be treated automatically [3.6].

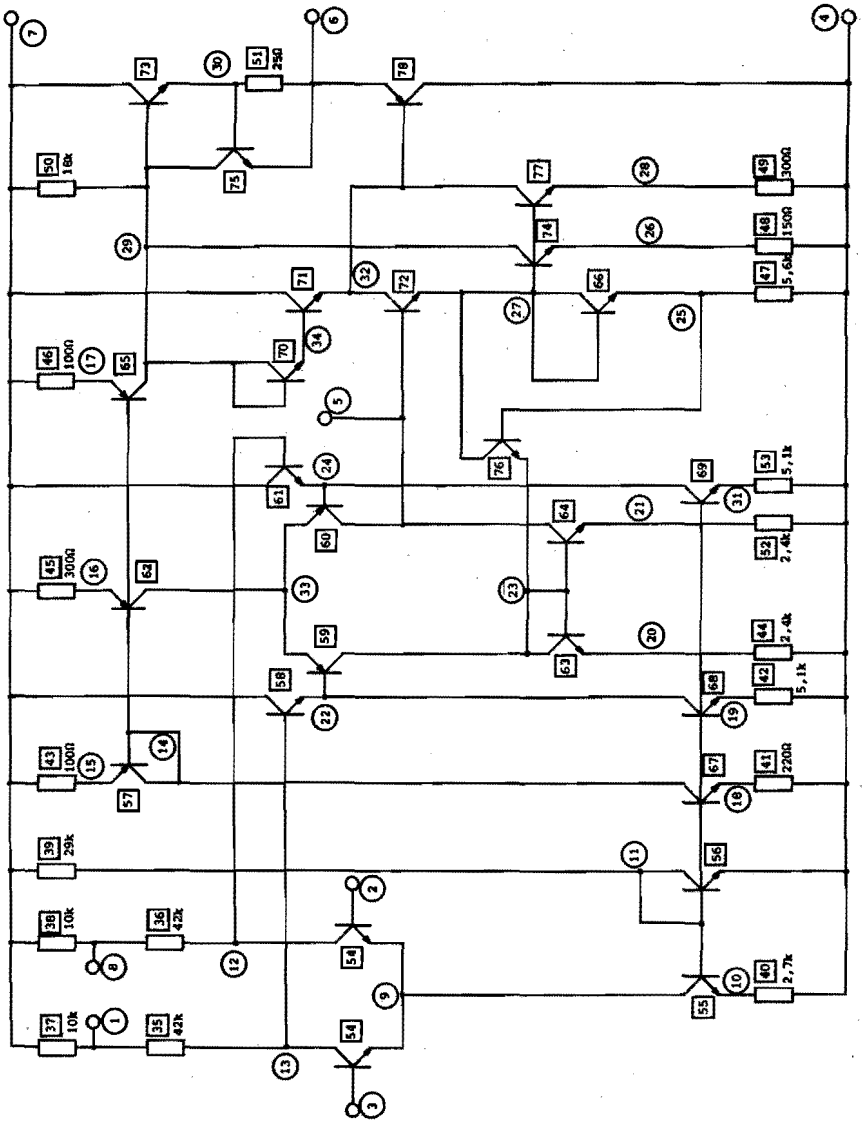
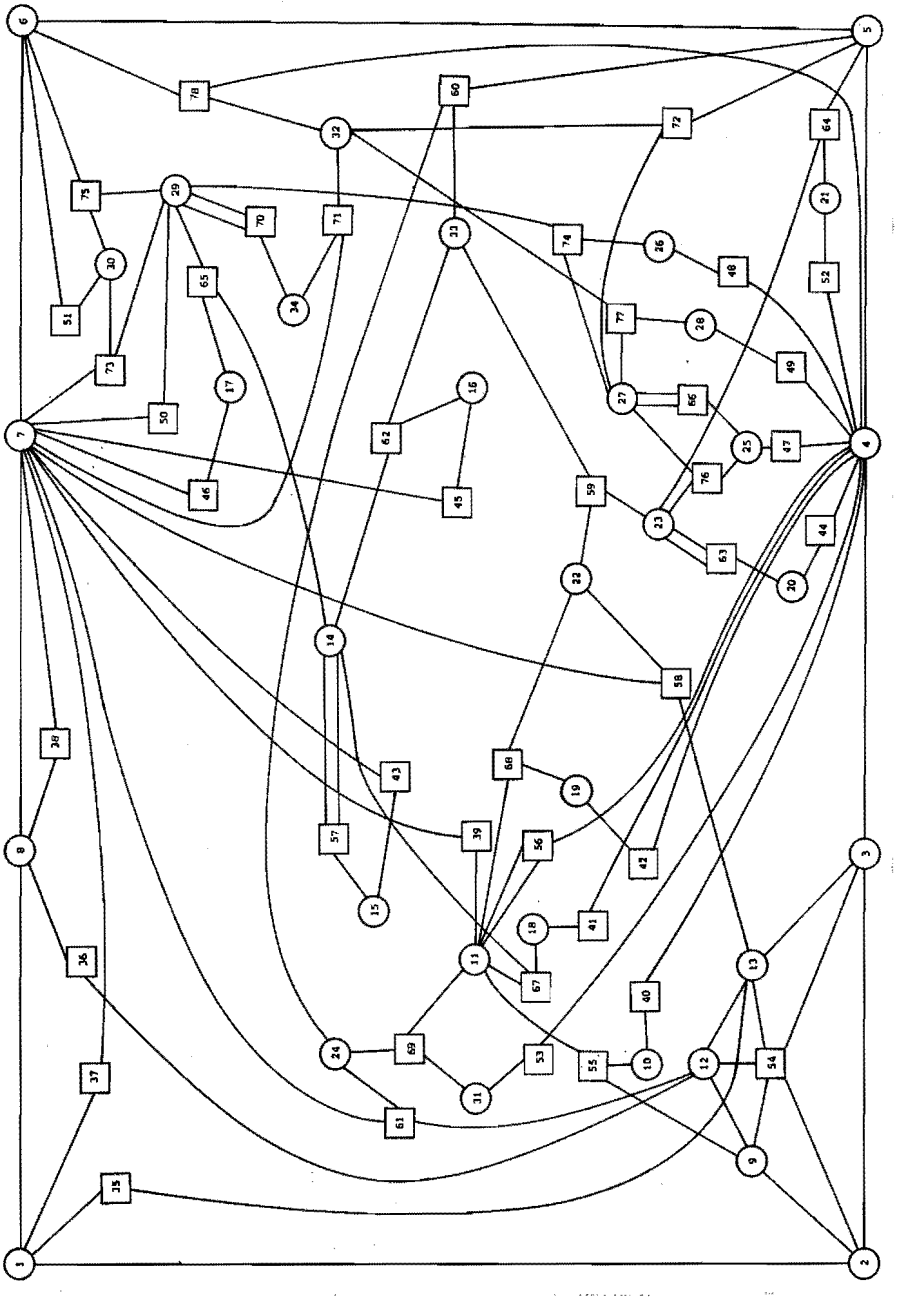


Fig. 3.2. The schematic diagram of the operational amplifier μA725 and its potential graph. The transistors of the input stage are treated as one "supercomponent".



Remark: It certainly is possible to include more requirements into the model. For example one may take into account the interchangeability of equivalent terminals (input-terminals of a logic gate for example). One may also wish to implement interchangeability of groups of terminals belonging to identical subcircuits. (It is, however, doubtful whether this can be implied by the planarity of a graph. This claim does occur in literature, but the implementation proposed in [3.7] is certainly not correct.) These constraints, however, are seldom relevant in bipolar integrated circuit design.

3.2. Considerations for planarization.

In general the potential graph of a circuit is not planar, which means that the constrained formal layout does not exist. From the necessity of planarity it is easy to conclude that we have to slacken the requirements A1, A2 and A3. Thus, modifications are necessary. These modifications, however, should never be such that the number of masks is increased or that the circuit behaviour is no longer within given tolerances. The following paragraphs are an inquiry into the freedom left by the chosen technology which can be used to come to a layout of the circuit observing the two absolute rules:
C1: the circuit performance must be within given tolerances
C2: the number of masks is not allowed to increase.

(1) The island potential.
The partitioning of the IC-chip by the deep p-diffusion in isolated regions has some consequences for the planarization of the potential graph. The potential of the epitaxial layer in an isolated region, the "island potential", is realized of course by making only one contact between the island and the respecting potential tree, but it is sometimes advantageous to have this potential available at several points on the chip, without need for aluminium leads. The greater the island area is, the more convenient it mostly is with regard to this consideration. For the potential graph this may have two kinds of consequences:
(a) All the edges symbolizing the connection between the potential

$t\beta^{-1}$ which is the same potential as that of the epitaxial layer of the isolated region containing some of the EP-components, and these components, may be deleted, since the island needs only once to be connected, and interconnections crossing the island between this contact and the other contacts of the components in that island, are allowed.

(b) Furthermore we have the possibility of splitting such a t-vertex t in as many vertices $(t_1,t_2,\ldots,t_p)$ as we wish, provided that in the resulting graph (G',U') $\bigcup_{i=1}^{p} t_i'\rho'=t\rho$. This corresponds with the idea of making arbitrarily many contacts with the epitaxial layer of the island. In order to obtain better potential correspondence between these "new" potential trees they may be connected by n^+-diffusions, since the conductivity of this diffusion is better than that of the epitaxial layer.

In both cases the influence is reduced, when a buried layer is present.

(2) The substrate potential.

Since the p-substrate always is at the lowest potential of the circuit and the deep p-diffusion reaches the p-substrate, we have this potential at our disposal by making a contact with an "isolation channel". In the graph this means that the vertex t associated with the negative dc-voltage may be split into arbitrarily many t-vertices $(t_1,t_2,\ldots,t_p)$ such that in the resulting graph (G',U') we have $\bigcup_{i=1}^{p} t_i'\rho'=t\rho$.

The substrate is sometimes a part of the component, which means that the edge symbolizing the connection of that part with the lowest potential (in such a case this edge must exist) must be deleted. We meet this situation, for example, in the case of a pnp-transistor of which the collector is connected with the negative supply voltage. The occurrence of such a "substrate transistor" in the circuit has consequences for the process. The control of the epitaxial layer thickness must be tighter, since it is directly related to the effective base width of the transistor. So it might be of interest to avoid substrate transistors completely, but if there has to be one substrate transistor in the circuit, replacing

a lateral pnp-transistor with the collector at the lowest potential by its substrate version has no disadvantages; on the contrary, the latter can handle higher currents than the lateral type of comparable geometry, and also the current gain and frequency response are somewhat better.

(3) Components with more than two contacts.
In planarizing the potential graph sometimes the following operation might be helpful. Suppose we have a component with more than two contacts. Associated with this component is a c-vertex c connected with as many t-vertices $t_1, t_2, \ldots, t_p$ as the component has contacts. Now we come to another graph (G'U') which has one extra t-vertex t'_{p+1} and satisfies:
$\forall_{1\leq i\leq p}[i\neq j \rightarrow t_i\rho=t'_i\rho']$, $t'_j\rho' \cup t'_{p+1}\rho' = t_j\rho$ and $t'_j\rho' \cap t'_{p+1}\rho' = \{c\}$
In (G',U') the vertices t'_j and t'_{p+1} represent the same potential which became available at two "sides" of the component via the contact of the component. Once applying this modification to a certain component, another application of it on another contact of the same component will often be effectless.
A transistor is - in many cases - a component with three contacts. Which of these contacts can be treated as described in the above depends on the geometry the transistor is supposed to get. The collector contact of an npn-transistor, for example cannot be considered for the modification in question. In the Fairchild realization of the µA 709 this modification is applied to the base contact of the boot-strapping transistor of the output stage and also to the emitter contact of the non-inverting input transistor of the first differential pair.

(4) Components with greater distances between their contacts.
When a component has a contact or a group of contacts which may be positioned so far from its other contacts that one or more aluminium interconnections may cross the component, this may be used to planarize the graph. In the graph this is reflected by splitting the c-vertex associated with the component in question. If desired, the edges ending in a vertex of degree 1 after this operation are deleted. The component that lends itself outstandingly to the

application of this modification is the resistor.

(5) Diffusion conductors.

The modification that is nearest to a real crossing is the small diffusion conductor in an interconnection, the cross-under. Other interconnection wires can cross this extra diffusion. Of course, every graph can be planarized by unlimited application of it, since it corresponds to splitting a t-vertex t associated with the potential tree that has got the diffusion conductor in two vertices t_1' and t_2' such that $t_1'\rho' \cup t_2'\rho' = t\rho$. However, an extra resistance is introduced in the potential tree $t\beta^{-1}$. Whether one may allow this, depends on the influence such an insertion has on the performance of the circuit. In a base connection even resistances of 300Ω are sometimes permitted (for example, Fairchild μA 725, first amplifier stage). Of course, using the n^+-diffusion with 2-5Ω/□ results in smaller resistances, but such a diffusion conductor may have a resistance which is still not negligible (its value depends on the number of crossings it must allow and its width). Furthermore, it often needs a separate island (if possible one may place it in the isolation channels) and the value is voltage dependent. Often the diffusion conductor has not a big influence on the circuit performance, but sometimes the consequences may be severe (for example, in the emitter connection of one of the transistors of a differential amplifier stage). It is difficult to take this into account a priori and since it always means a loss of area, there exists an inclination to avoid the cross-under if possible.

Determinative in the choice of the modifications must always be their influence on the electrical behaviour. To find out this influence one has to model the modification and simulate the network with an analysis program. It is, however, not expedient to carry out an analysis for every modification which might planarize the potential graph. Except for some applications of the cross-under, network sensitivities are perhaps the adequate means to weigh the modifications against each other. Absence of an analysis program makes a fixed grading of the modifications necessary with which one enters the planarization proce-

dure. Of course, special requirements and area considerations also create preferences in applying certain types of modifications.

Many of the described modifications introduce crossings of metal interconnections over diffusions, which may give additional problems to consider. Such a crossing always means a parasitic capacitance between this diffusion and the interconnection, which may cause unallowed coupling of signals. One cannot always adequately prevent these couplings in an automatic layout procedure, and in many cases a final test must decide whether the responses of the circuit are still acceptable. The procedure performing this test should be capable of deriving its models from the layout. Another influence such a crossing may have is an increase of the sheet resistance, when the metal of the interconnection has a potential rather different from that of the diffusion. As for the n-epitaxial layer this effect is often held off by the buried layer; for the other lightly doped areas this situation does not often cause serious difficulties. A third effect is that of surface inversion. When, for example, a metal conductor crosses an isolated region which has a much higher voltage than the crossing metal path, a p-channel is created under the interconnection. When this lead also crosses a meander or several p-diffusions the result is an electrical connection between these diffusions.

Beside these technological problems, also some layout difficulties are introduced. Allowing a resistor to be crossed by splitting the corresponding c-vertex, means that control over the number of crossings is lost. The consequential problems are not insuperable. In the last extremity one can always make a cross-under in series with the resistor, but whatever the measure is, the chip area is increased.

In order to come to a good layout with regard to the mentioned problems we want a procedure that selects which potential trees are to cross a certain diffusion. Priorities derived from the considerations in the two preceding paragraphs should be observed and overlapping of diffusions must be avoided. Such a procedure is given in chapter 5.

3.3. The isolated regions.

As explained in section 2.4 the components of an integrated circuit are distributed over isolated regions. This distribution induces a partition over the set of components and this partition must be determined by the program according to some rules and observing some desires. Let us first repeat the two main rules that must be obeyed:

D1: Components of which the epitaxial layer is a part of the component structure (EP-components) cannot be embedded into the same isolated region if these parts are to be connected with different potential trees.

D2: Components of which the epitaxial layer is not a part of the component structure (NEP-components) cannot be embedded in an island of which the potential is lower than that of the deepest diffusion of that component.

Some components may impose further constraints on the determination of the partition. For example the base-emitter pinch resistor has a low breakdown voltage of about 6V; the potential of the island in which it is to be embedded, should never be more than 6V higher, than the most negative contact of the resistor. Furthermore, there are components forming a diffused island such as the resistor made by the shallow n-diffusion into a deep p-diffusion. Supercomponents may include several islands (for example the input transistors of the μA 725). Diffused islands and supercomponents are called solitary EP-components and enter the partition as blocks with one element.

The given rules only determine a set system over the set of components consisting of so-called IR-compatibility classes [3.8]. The mentioned partition must be a refinement of this set system. The freedom that is left after obeying the rules, can be used to satisfy desires which will be listed below. First, we make some remarks about the determination of the IR-compatibility classes. It is clear that the behaviour of several potential differences must be known. (Otherwise we must choose for a partition with all the NEP-components in the island of which the potential is the positive supply voltage.) This means that a simulation program must be available which can simulate the circuit in the operating states. As soon as one of the potential

differences undergoes a sign change the respective component must be removed from the concerned block. Starting with classes containing compatible EP-components and all NEP-components, or only a solitary EP-component, we must end up with the desired set system. If there is no EP-component of which the epitaxial layer is at the positive supply voltage, a class must be added containing all NEP-components.

Beside these rules, decisions taken during the planarization of the potential graph have their impact on the final partition, especially the modification discussed under (1a) in the preceding section. For if the edge symbolizing the connection between some potential tree and the epitaxial layer part of some component is deleted, this component must be in the same block as another EP-component having its epitaxial layer part at the same potential or the concerned isolated region must get the correct island potential via a separate contact hole.

Other objectives in determining the island partition may ensue from the following considerations:
In general, the total chip area is decreased by choosing a partition with a minimum number of blocks, which means a minimum number of isolated regions, because the area occupied by isolation channels is in most cases smaller then. Also the total parasitic capacitance between the islands and the substrate is lower then. If the capacitance between an EP-component and the substrate must be kept small, its island area must be kept small, i.e. one should not add NEP-components to the block of that EP-component. The influence of other parasitic junction capacitors can be reduced by keeping the reverse biased voltage high. If the planarization procedure has made it necessary to have an island potential available at several points on the chip, without aluminium interconnections between these points, it is convenient to give this island a larger area, and thus to strive after many components in the block belonging to that island potential. Total interconnection length must also be minimized, because of the chip area the wiring needs, several parasitic effects such as delay and signal coupling, etc.

After deciding which components are to be placed in the same island,

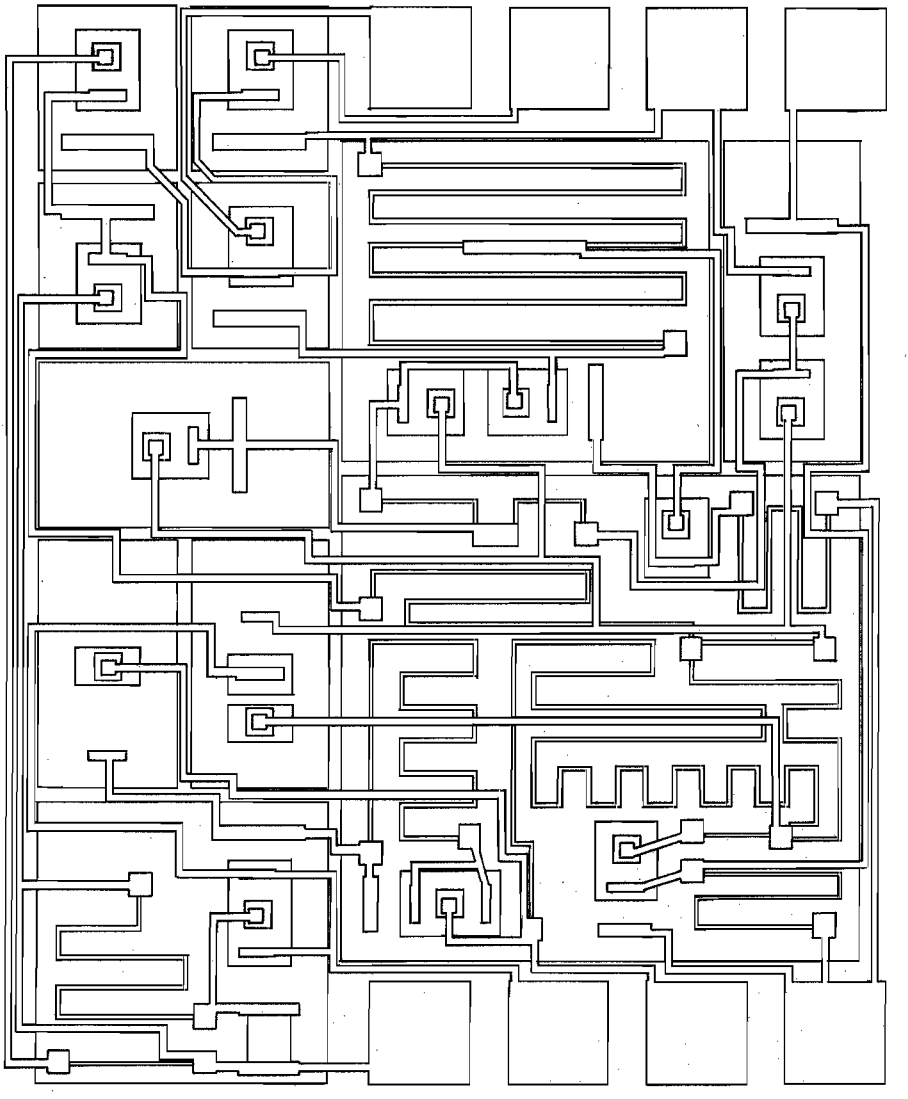


Fig. 3.3. A layout of the operational amplifier μA709 in which all the islands have a rectangular form.

the program has to determine the form of the islands. An island is either rectangular or a union of rectangles, which is a consequence of the practice of making only right angles between isolation channels. However, in existing layouts the bigger islands often are unions of many rectangles. Things would be greatly simplified if every island can be laid out as one rectangle or a union of a small number of rectangles. The possibility of such a layout for the µA 709 is shown by fig. 3.3. Every island in this layout even has got a rectangular form with the total chip area equal to the Fairchild realization. In general, we admit island figures consisting of a few adjoining rectangles. How to divide a block of the island partition in such a case, is decided on grounds of the information contained in the planarized potential graph and the results of the procedure which determines the crossing interconnection leads over the diffusions. It is described in section 6.4.

Let us take an arbitrary rectangle partition of a rectangle. In every rectangle we draw an oriented curve from the upper side to the lower side and we assign two positive real numbers to this curve, namely the length and the width of the rectangle. No information is lost when we now delete the vertical line segments. And again no information is lost, when we contract the horizontal line segments to points. What is left now, is a digraph with two numbers assigned to each arc. Another digraph would have been obtained by an analogous procedure interchanging "horizontal" and "vertical". This digraph contains exactly the same information. Both constructions are illustrated in fig. 3.4.

The digraph thus constructed has a number of special properties. First of all it is planar which is obvious from the construction. There is exactly one source and exactly one sink, namely the vertex associated with the upper side of the outer rectangle and the one associated with the lower side of the outer rectangle. Since it is not possible to return to the same horizontal line segment when we go from upper sides to lower sides the digraph must be acyclic. Thus the digraph is a plane drain. Besides, its source and sink are at the same face boundary, which makes it a drain representation. The other digraph constructed by

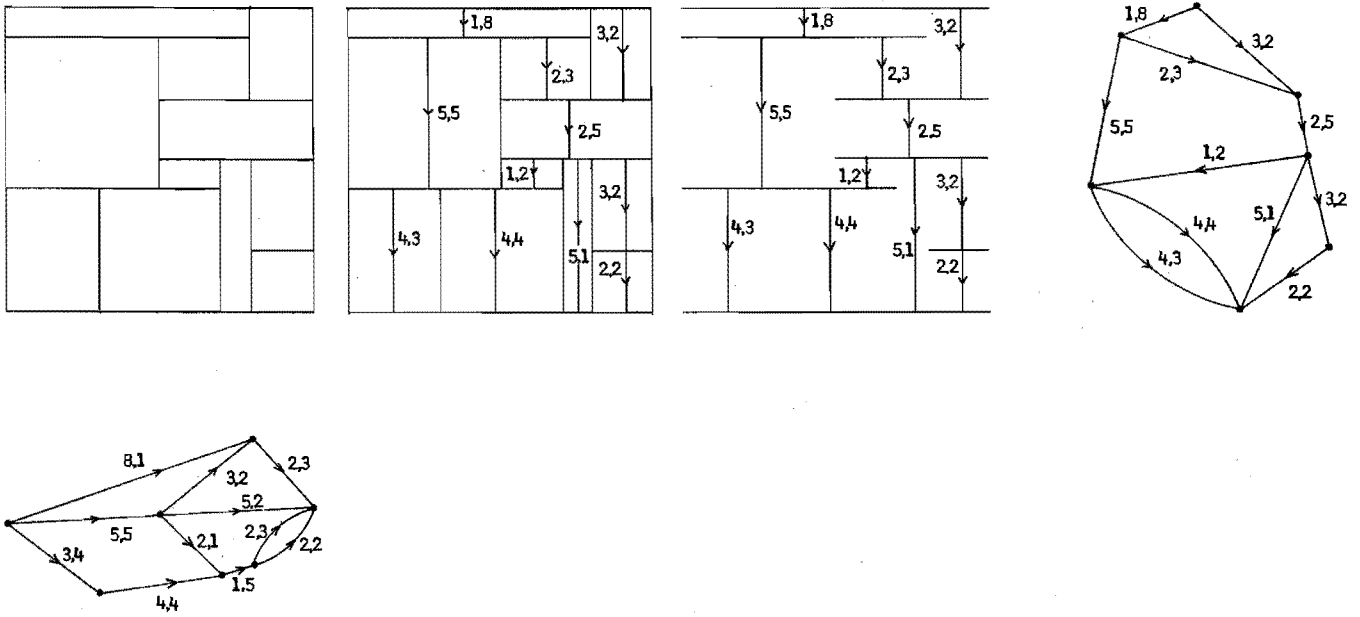


Fig. 3.4. The construction of the digraphs starting from a given rectangle partition.

deletion of the horizontal sides and contraction of the vertical line segments is the dual of the above drain representation. To recognize another property the following observations may be helpful. There are a number of rectangles that have a part of a certain horizontal line segment as an upper side and a number of rectangles that have a part of that same horizontal line segment as a lower side. The sum of the widths of the rectangles in the first set is, of course, equal to the sum of the widths of the rectangles in the other set. In the drain this means that the sum of the second co-ordinates of the arcs pointing to a vertex is equal to the sum of the second co-ordinates of the arcs pointing from this vertex. An analogous reasoning gives a suchlike quality of the drain for the first co-ordinates: going from a certain horizontal line segment to another from upper sides to lower sides, no matter which series of rectangles is traversed, the sum of their lengths must always be the same. In the drain representation this implies that the sum of the first co-ordinates of the arcs of the two chains forming a face boundary must be equal. The latter properties will not fail to remind of the Kirchhoff current and voltage law for electrical networks.

What should be kept in mind from the preceding two paragraphs is that the drain representation and the partitioned rectangle are two equivalent structures for representing the same information [3.9]. However, in this stage of the design we have neither a rectangle partition nor a drain representation with pairs of numbers associated with its arcs. So it must be constructed from the data available from preceding procedures, taking into account the following facts and objectives.

E1: Each rectangle needs a certain minimum area.
This minimum depends on the components that must be placed in this rectangle and the interconnection leads it must allow. The area a certain component needs can be obtained from the component library.

E2: The rectangles that together form one isolated region must be adjoining.
Otherwise we have to connect the various regions in the n-epitaxial layer with the same potential, which may result in many difficulties,

when designing the aluminium mask.

E3: Rectangles that contain components that should undergo the same thermal effects are preferably placed close to each other.

E4: Rectangles that contain components that have a relatively high power consumption are preferably placed far from each other.

E5: If there are many interconnections between the components of two blocks, the corresponding rectangles are preferably placed close to each other.

E6: There is some advantage in making endeavour to approach a square form for an isolated region.
The area occupied by isolated regions is smaller then and the capacitance between the epitaxial layer and the substrate is reduced. Further it is obvious that for example a transistor which is ten times as long as it is wide is not realistic.

E7: The total chip area should be as small as possible.
The yield of the production of an integrated circuit is directly related to the chip area. The number of circuits per wafer is greater and the number of errors caused by crystal errors and contaminated surfaces, is proportional to the chip area.

E8: The chip should not be too oblong.
Otherwise we have a greater risk of breaking the chip when snapping the wafer.

The procedure that must determine the geometry of the islands consists of two subsequent steps described in chapter 7. The first step is the construction of an acceptable drain representation, which takes into account the minimum areas of the rectangles and the preferred neighbour relations, acting as if every rectangle must be a square. From this drain representation a set of linear equations is derived forming a tableau on which the simplex algorithm is applied in order to minimize the total chip area.

After the determination of the drain representation we have some information about the number of interconnection leads that must pass between two rectangles. Depending on this number, we can add extra area to the desired minimum before entering the second step, the simplex algorithm.

3.4. Wirability.

The last part of the design procedure has to fix the exact positions and orientations of the components on the chip and to determine the configuration of the aluminium interconnections. Existing design systems facing the same problems, decompose the design serially into a placement and wire-routing problem. This concept is so generally recognized that almost all papers in the field of layout design deal with only one part of the problem, instead of describing a complete system. Nevertheless it goes without saying that the two design functions, placement and wire-routing, are very contingent on each other. An ill-chosen placement may even disable the routing procedure completely. The goal of any solution to the placement problem has therefore always been to reduce the difficulties in the design of the ensuing interconnection pattern. For an automatic design system this is not sufficient, because in that case the applied algorithms must provide the assurance of finding a satisfactory solution if it exists. None of the methods recorded in literature have adequate capacity to ensure the outcome of an acceptable layout, and this incapability of satisfying the fundamental requirement of any automatic design makes them all unsuitable for insertion in the layout design system we are describing. Notwithstanding that, we will have a look at previous attempts, hoping to pick up some ideas that can be of use in the algorithm to be developed.

Let us first try to come to a more precise formulation of the problem. The preceding part of the program has delivered a rectangle partitioned into smaller rectangles. We cover this rectangle with a grid structure of square cells of which the sides must be at least as large as the width of the interconnection leads plus the required spacing between them. (We presume that all interconnection leads have the same width

and take this width equal to the required spacing between them). The length of these sides is called the grid constant. We have chosen the grid constant to be three times the width of a base-diffused resistor used for greater resistance values. This means that the grid constant will be about 30 μ and the width of an interconnection lead ca 15 μ. This enables us to avoid that an interconnection lead will exactly follow a resistor over a large distance. (fig. 3.5).

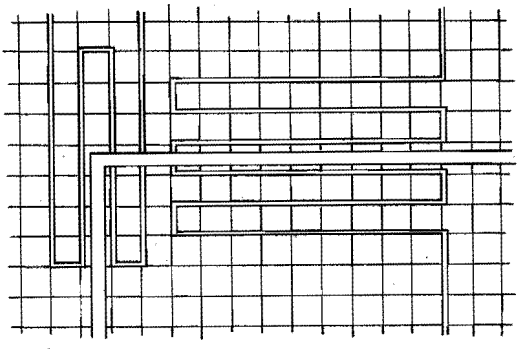


Fig. 3.5.
By keeping the resistor diffusions at the bottom and the left hand side of the cells one can avoid that interconnections "cover" these diffusion over large distances.

(This would make the resistance dependent on the potential of the lead). We now shift the boundaries of the rectangles such that they coincide with the lines of the grid structure and leave a space of one grid constant between each pair of neighbouring rectangles (the isolation channels). The components, except those having very flexible forms such as large resistors and capacitors, must be placed as rectangles in the grid structure with some cells along its sides marked as "containing a contact". The potential that has to reach this contact must also be specified. A part of the remaining area is destined for the non-placed components of which the exact shape still must be determined. The interconnections must be fixed as chains of cells between the contacts.

If we decompose the task serially into a placement and wire-routing problem, the solution of the first part leaves us with a rectangular array of elementary cells in which each cell is

a) available for all interconnections to be routed or
b) a terminal point for an interconnection to be routed and thus non-available for the other routes or
c) not available for some interconnections (components that are to be crossed by certain potential trees, determined in an earlier stage of the design) or

d) not available for interconnections to be routed.
The question now is whether this always gives a wirable situation in the sense that a set of non-interfering paths meeting all constraints can be exhibited, and if so, whether there is a procedure that finds such a wiring. Let us therefore examine some attempts dealing with the single-layer-wiring problem recorded in literature.

The procedures handling this part of layout design, fall into two categories. One category contains the route-by-route procedures in which the paths are considered in some sequence and another path is not started before the actual one is established. The cells of the established routes become obstacles (non-available cells) for the subsequent paths. In the other category we find the procedures working on several paths simultaneously, i.e. the procedures establish parts of other interconnections before they have completely determined an earlier started route. In the multi-layer-wiring problem there are some algorithms belonging to the second category, but in case only one layer is available there are very few procedures in the second category [3.11]. Their main disadvantage is that paths requiring some detours to avoid non-available cells, can hardly be generated. No successful implementation of such a parallel procedure, in the sense of having a high "percent completion" is known to us.

Thus, almost all of the procedures for finding interconnection paths are of the route-by-route type, which gives the additional problem of determining the sequence in which the interconnections are to be routed. Experiments have shown that decisions concerning this sequence based on Manhattan distances between the points to be interconnected do not have significantly higher percent completions than random sequences. Nevertheless we should keep in mind that a sequence must be chosen, and if possible we must use this free choice to make the outcome of a solution certain (if it exists!).

The most widely used algorithm for finding paths in a grid structure as described in the above is the one that can be derived from the path optimization routine of appendix C, and which is commonly referred to as the "Lee-algorithm" [3.12]. Suppose we have a grid of R rows and K

columns of squares. We identify the cells with their row and column number. To apply the procedure of appendix C one takes:

$$C:=\{(r,k)\mid 1\le r\le R \wedge 1\le k\le K\}$$

$$S:=\{-1,1\}$$

$$\forall_{c\in C}[(c\sigma=1 \leftrightarrow c \text{ is available}) \wedge (c\sigma=-1 \leftrightarrow c \text{ is not available})]$$

$$\forall_{(c_i,c_j)\in C\times C}[(c_i,c_j)\in\eta \leftrightarrow (c_i\sigma\neq -1 \wedge c_j\sigma\neq -1 \wedge |r_i-r_j|+|k_i-k_j|=1)]$$

$$\forall_{(a,s,b)\in\mu}[a+s=b]$$

The operation of the algorithm is shown in fig. 3.6. The algorithm

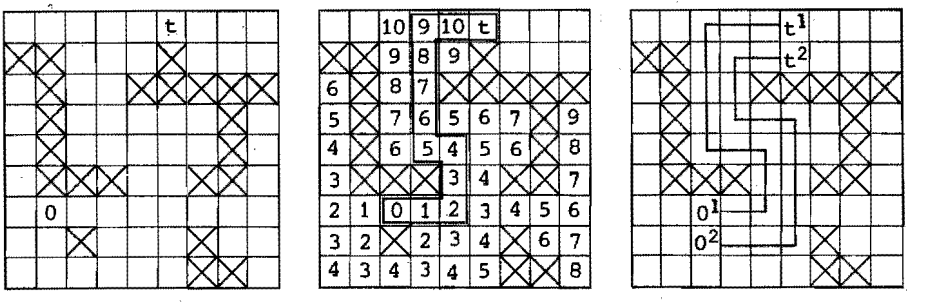


Fig. 3.6. The "Lee-algorithm".

has the salient feature, that, when it is confronted with an array in which the actual route exists, it will always find it, and when more than one path exists, it will establish the shortest path available. That the latter property is not always desirable shows the last grid of fig. 3.6. It is clear that there exists a wiring connecting o^1 with t^1 and o^2 with t^2; but when for one pair, no matter which pair is chosen, the shortest path is established, no path for the second pair can be found. So one might abandon the principle of accepting only the shortest path available at the moment in order to come to a better overall performance.

It is clear that on an open grid structure (i.e. almost all cells available) the algorithm is very slow, especially when the path which is searched for, is quite long. Several attempts to reduce the size of the search are recorded in literature. Most of them let the first routes be generated by more directed, but not exhaustive algorithms

and leave the remaining interconnections to the Lee-algorithm. Another possibility is two-ended search, which means that two search-waves, one from the origin(s) and one from the target(s), are expanded. [3.13] A more important contribution was the introduction of the predictor-cost function or even the depth-first-predictor search [3.14] (the latter method cannot be fitted into the scheme of the algorithm of appendix C). But the simplest way is to apply the Lee-algorithm only to labyrinth-like problems. If the algorithm is only offered grids in which the avarage value of $|c_n|$ is close to 2, none of the mentioned versions gives a more efficient program than the Lee-algorithm does.

Much more important than time saving is to raise the percent completion. At this point it is fit to remark that the algorithm of appendix C (and the algorithm described by Lee as well) can handle more weighing relations than the simple "shortest-distance-to-an-origin" weight. For this, it is necessary to extend the cell alphabet S in order to distinguish more kinds of cells than available and non-available cells. In that case we can use the labeling relation σ to see to it that cells in the neighbourhood of obstacles have inflated weights assigned to them. The general idea behind this is to avoid adjacent non-available cells as much as possible, because they can cut off access to regions of the grid structure in which there are still cells to be interconnected with cells outside such a region. This certainly improves the performance of the algorithm, though storage and time consumption are increased. However, adjacency avoidance does not assure the complete wiring if it exists. [3.15]

Another method to increase the number of routed interconnections is to use the freedom in choosing the sequence. We have already said that Manhattan distance does not give a satisfactory indication as to which interconnection must be routed first. The best ordening is to route first those interconnections that will probably interfere least with the routes still to be determined. The difficulty, however, is to measure this probability before any interconnection is routed. The first idea derived from this principle is to encourage the routing of interconnections in dense regions before those in sparse regions in order to route to a point while it is still accessible. It is, of course,

recommendable to apply this rule in an adaptive manner, i.e. from time to time the next wires to be routed are chosen. Experiments recorded in literature show that a combination of the above-mentioned adjacency avoidance and the following sequence determination gives the best results. For every pair of cells, (r_1,k_1) and (r_2,k_2), to be interconnected the "primary" rectangle, defined as the rectangle with corner co-ordinates (r_1,k_1), (r_1,k_2), (r_2,k_1) and (r_2,k_2), is used to establish an interference measure with respect to the other interconnections. This number is based on the type of conflict and the number of shortest paths inhibited if the other interconnection is routed first. [3.16]

Still more freedom can be discovered, when examining the operation of the Lee-algorithm, namely in tracing the route after the search is finished. Mostly there will be several paths of the same minimum length available. Among these paths the trace procedure can as well choose the best path according to some additional criterion, for example reducing the chance of blocking future wires. How to formulate such a criterion, depends on which wires are still to be routed, thus on the sequence in which the interconnections are routed.

Furthermore, we have of course the possibility of rerouting, especially applied by people having a graphical display at their disposal. In fully automatic design systems iterative routing schemes are seldom applied. One more or less successful technique is based on adjusting relative penalties for length, adjacency, and intersection until no improvement is obtained. The procedure starts with routing all interconnections with only a penalty on the length. In each iteration the penalty on intersections is increased and the penalty on the length is decreased, while the adjacency penalty is constant after the second iteration. Though the result is better than the one obtained by a one-pass routing procedure, the iterative scheme is very slow. [3.17]

After this certainly incomplete excursion through single-layer wiring it is appropriate to summarize the conclusions.

1. First completing the whole placement and then solving the ensuing wiring problem is probably not the best decomposition of the problem.

Though complete interaction cannot be efficiently realized, there is certainly advantage in having established a part of the wiring, before finishing the placement.

2. Placement must be performed in such a way that the probability of obtaining wirable grids is very high, for in that case the placement need not to be adjusted very often.

3. Apply the path algorithm - if possible - only in labyrinth-like situations.

4. Try to avoid adjacency among obstacle cells if it is possible that future wires have to be inserted between them.

5. Longer paths that probably do not interfere with future wires, have to be preferred over the shortest path with a greater chance of blocking not-yet-established routes.

6. It must be possible to abolish a non-wirable situation, either by rerouting or by changing the grid.

7. The sequence in which the interconnections are to be treated must be chosen such that:
   a. from time to time it can be checked whether the wirability is spoiled by the last few wires; in that case they have to be rerouted
   b. non-wirable situations can be detected, and must be abolished
   c. tracing can be done such that it leaves the maximum space for future wires.

As to placement three categories of methods can be recognized:
1. constructive, initial placement
2. iterative placement improvement
3. branch-and-bound.

The first category comprise the methods in which the placement configuration is formed by adjoining components to the set of already placed components, by alternately selecting and positioning components.

The rules governing this selection and positioning determine the method. These methods do not consume much computer time, but for highly critical problems they are not well suited and therefore, in general, combined with iterative placement improvement. In the latter type of procedures each stage of the operation of the algorithm consists of selecting one or more components for change of position. According to some criterion the repositioning is performed or the old configuration is retained. These iterative methods demand more computer time. Branch-and-bound is a well-known programming technique that - when exhaustively applied - delivers an optimal result. The efficiency of the method depends on the definition of the optimum and the "preclusion" strategies (i.e. terminating the search with certain configurations as a part of the whole solution). [3.18].

References.

[3.1] P.Dembowski, "Finite geometries", Springer-Verlag, Berlin, 1968.

[3.2] C.Berge, "Graphes et hypergraphes", Dunod, Paris, 1970.

[3.3] W.L.Engl, D.A.Mlynski, "Topological synthesis procedure for circuit integration", IEEE International Solid State Circuits Conference, University of Pennsylvania 1969, Digest of Technical Papers, pp. 138-139, New York, 1969.

[3.4] A.J.Goldstein, D.G.Schweikert, "A proper model for testing the planarity of electrical circuits", Bell Syst. Tech. J., vol. 52, pp. 135-142, Jan. 1973.

[3.5] W.L.Engl, D.A.Mlynski, "Embedding a graph in a plane with certain constraints", IEEE Trans. Circuit Theory, vol. CT-17, pp. 250-252, May 1970.

[3.6] K.C.Kwok, "Topologische Widerstandseinbettung in integrierten Schaltungen", Ph.D. dissertation, Rheinisch-Westfälische Technische Hochschule Aachen", July 1972.

[3.7] W.M.Vancleemput, J.G.Linders, "An improved graph-theoretic model for the circuit layout problem", Proc. 11th Design Automation Workshop, pp. 82-90, Denver, Colorado, June 1974.

[3.8] D.Ferrari, "On the selection of isolated regions in computer-aided design of integrated circuits", IEEE Trans. Circuit Theory, vol. CT-17, pp. 134-136, February 1970.

[3.9] R.L.Brooks, C.A.B.Smith, A.H.Stone, W.T.Tutte, "The dissection of rectangles into squares", Duke Math. J., Vol. 7, pp. 312-340, 1940.

[3.10] T.Ohtsuki, N.Sugiyama, H.Kawaniski, "An optimization technique for integrated circuit layout design", Proceedings of ICCST-Kyoto, Kyoto, Japan, pp. 67-68, Sept. 1970.

[3.11] C.J.Fisk, D.L.Caskey, L.E.West, "ACCEL-Automated circuit card etching layout", Proc. IEEE, Vol. 55, pp. 1971-1982, November 1967.

[3.12] C.Y.Lee, "An algorithm for path connections, and its applications" IRE Trans. on Elec. Compu., Vol. 10, pp. 346-365, September 1961.

[3.13] I.S. Pohl, "Bidirectional and heuristic search in path problems", Ph.D. dissertation, Stanford University, Stanford, 1969.

[3.14] F.Rubin, "The Lee path connection algorithm", IEEE Trans. on Comput., Vol. C-23, September 1974.

[3.15] S.B.Akers, "Routing", "Design of digital systems", Vol. 1, chapter 6, Prentice-Hall Inc., Englewood Cliffs, New Jersey, pp. 283-334, 1972.

[3.16] L.C.Abel, "On the automated layout of multi-layer planar wiring and related graph colouring problem", Illinois University, Urbana, Illinois, January 1972, Coordinated Science Labarotory Report no. R-546.

[3.17] F.Rubin, "Printed wire routing for multilayer circuit boards", Ph.D. dissertation, Syracuse University, Syracuse, June 1972.

[3.18] M.Hanan, J.M.Kurtzberg, "Placement techniques", Design automation of digital systems, Vol. 1, chapter 5, Prentice-Hall Inc., Englewood Cliffs, New Jersey, pp. 213-282, 1972.

4. PLANARIZATION OF THE POTENTIAL GRAPH

4.1. Introduction.

In the preceding chapter a graph theoretical model has been discussed, which will be used as a starting point for the construction of a layout of the given electrical circuit.
The planarity of the potential graph is a necessary and sufficient condition for the existence of a layout satisfying the requirements A1 up to A4 and B1 up to B3. Once we have constructed a plane representation of the potential graph, it can be used for the realization of a formal layout by constructing a wiring pattern according to the description in section 3.1. In all subsequent parts of the layout procedure, it is possible to exploit the information that is included in the plane representation of the potential graph about the wiring pattern. Thus the actual wiring design may be facilitated, and the number of difficulties in this procedure decreased.

A plane representation of the potential graph only exists if this graph is planar. In general the potential graph of a circuit will be non-planar. The requirements A1 up to A4 are in that case too stringent (a formal layout does not exist), and a number of deviations from these requirements have to be admitted. In chapter 3 it is considered which possibilities the bipolar technology offers in deviating from the formal layout. The use of some of the applicable deviations may have a very unfavourable effect on the performance of the circuit if they are applied to a certain part of the circuit, or may increase the cost of the integrated circuit. The application of these deviations has to be avoided if possible. The number of possibilities to obtain an acceptable layout of the circuit depends on the structure of the circuit and the number of extra requirements on the part of the circuit designer. In general there are so many acceptable solutions that a rather superficial search for a set of modifications suffices. The layouts that are completely designed by hand are obtained by a superficial search. Therefore it will not be worth-while to develop expensive (i.e. time- and/or storage-consuming) algorithms that find an "optimal" solution

(i.e. the set of modifications implying minimal cost and degradation of performance). The aim is rather to develop an efficient algorithm that constructs a plane representation of a graph that is derived from the potential graph by applying an acceptable set of deviations from the formal layout. This graph will be called "modified potential graph".

Remark: If an optimum solution had to be determined, we would have been in serious trouble, since there is no other method at hand than generating all possible solutions, and store the best ones. However,this method would take ages of computation time, even for a very small practical circuit.

An easy way to relate all (but one of) the applicable deviations and their effects to the potential graph, is defining weights on the edges and vertices of the graph. In order to simplify the planarization considerably only the edges of the potential graph are weighed (section 4.2).
In the past a number of planarity testing algorithms have been described in literature. Simply performing such a test, however, only supplies the answer whether the graph is planar or non-planar. In case the graph is planar, some of the methods construct a plane representation of the graph. However, the knowledge whether the potential graph is planar or not does not solve our problems. What we need is a "planarization algorithm", that modifies the non-planar potential graph by the application of an acceptable set of deviations from the formal layout. The planarity tests that have been published, contain some basic principles and methods to detect non-planarities, that may advantageously be used and combined with procedures that try to minimize the total weight of the edges that will not be embedded in the plane. Therefore we will discuss some of the properties of these methods (section 4.3). In the subsequent sections some new planarization algorithms are described, and their performance is discussed. Furthermore a procedure is given for improving a representation once obtained, particularly if not all extra requirements have been satisfied in the first attempt. In section 4.9 the way to work out the consequences of the set of deviations (associated with the set of non-embedded edges) is described. Thus it is

achieved that their application is ensured in later steps of the layout design procedure.
In the last section we communicate some results of the planarization process for the operational amplifier μA 725.

4.2. The edge-weighed potential graph.

During the determination of an appropriate set of deviations from the formal layout indications as to the effect of the application of such a deviation to a certain part of the circuit on performance and cost have to be known. These effects can be given by the circuit designer, and a circuit analysis program may be utilized to determine this information. If a separate analysis procedure is not available one can grade the different "kinds" of deviations by assigning to them their individual cost: the higher the cost, the less attractive it is to apply the respecting kind of deviation. Some of the costs can be made dependent on parameters of the part of the circuit under consideration (such as the respecting component value or the possible area of the island in question). The designer has the opportunity to specify the cost associated with each kind of deviation. One may also include the possibility to give a special cost when applying a certain kind of deviation to a certain part of the circuit. For example, the cost of introducing a cross-under in the metal lead having the positive supply potential can be made extra high.

All the applicable deviations, except the one which is considered in the preceding chapter under "3" (component contact doubling) correspond, in the potential graph, with the deletion of some edge(s) or with the splitting of some vertex ("modifications" in the graph). The deletion of some particular edge or the splitting of some vertex can be achieved by several kinds of deviations. For each edge and vertex in the graph one can determine the "cheapest" deviation (i.e. the one having minimum cost) causing the deletion of this edge or the splitting of this vertex, and then assign the appropriate cost to it. The planarization algorithm can be considerably simplified if the number of possible types of modifications in the graph is restricted, for example by translating the splitting of vertices into an edge-deletion modification.

The disconnection of edges from a certain vertex by splitting the vertex will be handled by deleting the edges. Later on, an extra vertex (vertices) will have to be added, and the edge relation adjusted according to the splitting of the original vertex. It is not probable that the new (splitted) vertices have a very high degree, but due to the fact that this type of modification implies the addition of extra components or contacts that require extra chip area, the application of it has to be avoided as much as possible anyway. Thus, it is decided to associate with each edge the cost ("weight") that corresponds to the cheapest deviation causing the deletion of this edge or the disconnection from one of its end vertices. Also a code, corresponding with the kind of deviation is stored.

Remark: The consecutivity edges between the t-vertices that represent the potentials of the terminal contacts of a (super)component, make it difficult to associate some deviation with an edge that symbolizes the connection between the component and one of its contact potentials. Deleting such an edge from the representation of the graph does not disconnect the (super)component from the respecting t-vertex since the consecutivity edges are still connected with this t-vertex. This problem can be avoided by using another model for these (super)components (see fig. 4.1). The (super)component itself is now represented by a wheel.

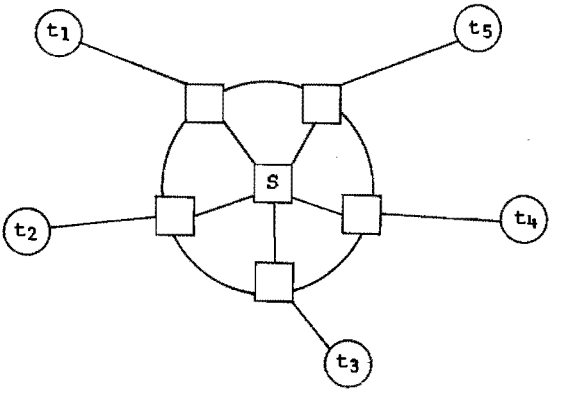


Fig. 4.1. The representation of a 5-terminal-(super)component by a wheel.

The hub and the edges that are associated with it, represent the inner region of the (super)component. The vertices on the rim of the wheel represent the terminals of the (super)component, and the edges of the rim take care of the required sequence of the terminal contacts (consecutivity edges). The connection of the terminals with their potentials is symbolized by the edges between the wheel and the respecting t-vertices. The replacement of the original wheel in the model by this subgraph does not have consequences for the planarity of the potential graph.

Since the wheel consecutivity of the terminal contact vertices has to be ensured, the edges of the wheels are given a special (high-valued) weight and code to prevent their deletion.

In fact all components in the potential graph could be represented by a wheel, but if the number of contacts of a component is less than or equal to 3, the consecutivity edges are not essential, and thus the wheel can be contracted into one (c-)vertex.

Let us assume that we are able to construct a plane representation of a subgraph (G_s,U_s) of the potential graph (G,U). Then, from the set of edges that are not in the subgraph $(U\backslash U_s)$, a set of deviations from the formal layout can be derived that modify the potential graph such that it can be embedded in a planar way. In the constructed plane representation the consequences of the deviations that are associated with the modifications have to be worked out:

- extra t-vertices (due to deviations that cause the splitting of a t-vertex) may have to be added, and the edge relation adjusted accordingly;
- new c-vertices, representing cross-unders or extra island contacts may have to be introduced, and the edge relation extended accordingly;
- component types may have to be changed as a consequence of the deviations from the formal layout.

This process will be described in section 4.9.

4.3. Considerations for the construction of a suitable planarization algorithm.

By a planarization algorithm we mean an algorithm that constructs a plane representation of a graph that is derived from the original graph (G,U) by applying a set of admissible modifications (edge deletion modification and component contact doubling modification). It will be the objective to keep the total cost of the set of modifications low. Before we develop a planarization algorithm fitting our purposes, we will discuss a number of tests on planarity that have been used in literature, and mention their (dis)advantages in using the principles in a planarization algorithm. Unless otherwise mentioned, we consider 2-connected graphs.

There are three basic theorems on the planarity of some graph (Kuratowski [4.1, appendix A (theorem 8)], Whitney [4.2] and McLane [4.3]), but planarity tests based on these theorems [4.4 up to 4.7], seem to be more of theoretical interest. They are not efficient and it is difficult to obtain indications how to optimize the total weight of the set of non-embedded edges of a non-planar graph.

A very extensive group of planarity testing algorithms repeatedly uses the "alternation principle" (appendix A) to detect non-planarity of the graph.
A great number of authors [4.8 up to 4.15] in some way use an algorithm that is originally described by Auslander and Parter [4.8] (also known as "pseudo-hamiltonian-method"). This method looks for a circuit in the graph, and tests whether its bridges can be partitioned in two sets, such that the attachment sets of the bridges in the same set are mutually non-alternating. This is achieved by a test on the bipartiteness of the associated alternation graph. Once the embedding of the circuit is completed, the procedure attempts to embed the first set of bridges in the inner, and the other set in the outer region of the circuit. The plane representation of any individual bridge is handled by recursively applying the same algorithm to the subgraph consisting of the union of the circuit and the bridge ("decomposed subgraph"). In this way a set of circuits is recursively

defined that will be referred to as "initial circuits". Bader [4.9] already indicates that a non-planar graph can be planarized by abolishing the alternations that have been discovered in each step by deleting a number of edges. It is obvious that in order to maintain the validity of the theory no edge from any of the initial circuits may be deleted. Pernards [4.16, 4.17] has used the principles of the pseudo-hamiltonian-method for the construction of a planarization algorithm. He tests the bipartiteness of the alternation graph by an exhaustive search of all circuits of odd length in this graph. Each time such a circuit is found it is tried to abolish one of the alternations among the attachment sets in question, using only the "cheap" modifications. This part is done automatically; in a second, interactive, part the rest of the non-planarities are abolished. In the automatical part little effort is bestowed on optimizing the total number of non-embedded edges, and additionally, no attempt is made to limit the number of cheap modifications that cannot be used anymore because they are in one of the initial circuits. An evaluation of the results of this partly interactive approach shows a definite impact on these limitations (see section 4.7).

Another iterative planarity test using the alternation principle, is described by Yoshida and Ohta [4.18]. It is particularly suitable to test whether there exists a plane representation of a graph, in which a certain circuit is a face boundary, for example the outer face boundary. A path (P,P[x,y]) is searched for between two vertices x and y of the circuit, and the bridges with respect to this path are determined. If there exists a bridge containing the complete (outer) circuit the algorithm terminates, since in that case the required plane representation cannot be accomplished. If this is not the case it is tried to partition the set of bridges in two sets such that the bridges in the same set have mutually non-alternating attachment sets. The (outer) circuit is, by the vertices x and y, divided into two parts. These two parts are subgraphs of exactly two bridges which of course may not be in the same set. The same procedure is now applied to the two subgraphs that are formed by joining the bridges of the same set together with the path. The outer circuit of the new subgraph consists of the path (P,P[x,y]) and the path that was part of the original circuit. If a new subgraph consists of only the (outer) circuit, it will not be

partitioned anymore. The circuit forms a face boundary in the plane representation of the original graph.
If we want to abolish the non-planarities we should not delete the edges that are contained in the outer circuit or in the constructed path (P,P[x,y]) of the subgraphs that are treated.

There are a lot of planarity tests that are of the constructive type. In the planarity test of Goldstein [4.19 up to 4.22], the plane representation partly completed is in each step extended with some path, that has both its end points on the boundary of the same face. First priority is given to the embedding of paths that are contained in bridges that can be embedded in exactly one of the faces (i.e. there is only one face boundary that contains the complete attachment set of the bridge). In this way it is achieved that one can conclude for non-planarity of the graph in case there exists a bridge for which no face boundary can be found that contains its complete attachment set. Now the test on alternation of attachment sets of different bridges can be avoided and the test becomes very efficient. However, this principle may work out very disadvantageous in a planarization procedure. By embedding an edge having low weight (but first priority), it may become necessary to delete a high weighted edge.
Hotz [4.23] has described a similar constructive embedding method for three-connected graphs. Because of the uniqueness of the plane representation of a 3-connected graph (theorem 12 of appendix A) it is not necessary to determine priorities for the embedding of the paths. However, in modifying a non-planar graph, the 3-connectivity of the (sub)graph(s) can easily be spoiled, and it is laborious to keep track of the 3-connectivity constantly.
This is also one of the reasons to reject the laborious method of Bruno et al.[4.24]. They reduce a 3-connected graph by two operations (deletion and contraction of an edge) into a wheel. Next they try to construct a plane representation by performing the inverse operations in reversed sequence.
Hopcroft and Tarjan [4.25] implemented a constructive method that, starting with a circuit, in each step embeds a path without taking into account the priority rule which is applied in Goldstein's method. The consequence of this is, that the constructed plane representation

repeatedly may have to be corrected in order to be capable of embedding new paths in a planar way (embedded in other regions). They succeeded in implementing this method in a very efficient way (special data structures, path sequence, etc.), and mention computation times in the order of the number of vertices plus the number of edges (|G|+|U|) (Rubin's implementation of Goldstein's algorithm [4.22] appeared to be of the same order). The planarity test is organized in such a way that an eventual alternation is discovered very early in the search process, but if an alternation is found one has only few indications as how to optimize the non-planarity.

Next, we mention the constructive tests, that in each step embed a number of paths or edges, and come to a decision about planarity by manipulating a formula.

In the algorithm of Lempel et al.([4.26], appendix F), the vertices are numbered and the edges directed from the vertex having lower number to the one having higher number. In numbering the vertices (1 up to n) it can be achieved that for each vertex i (1<i<n) there is at least one edge that has i as end-vertex, and at least one edge that has i as start-vertex. A "drain" has been constructed, which has exactly one source (only edges leaving) and exactly one sink (only edges entering). Starting with the source and its associated edges, the embedding is in each step extended with the tree of edges that leave from the next vertex in the sequence. Before bringing in the next tree, let us say of vertex i, the end points of the already embedded trees that are associated with i have to be identified to one point. From the manipulations with the formula developed up to the current state, it can be decided whether this is possible. Each manipulation corresponds with a permutation of subgraphs around some articulation vertex or a reflection of a subgraph around some articulation vertex.

The formula manipulation can be implemented rather efficiently. During the procedure, there is only one face in which the not yet embedded part of the graph is to be considered, which relieves the considerations for the modifications that have to be applied. The method lends itself to interactive planarization (appendix F).

The algorithm of Klemm [4.27] develops a special representation of the graph in segments. The vertices of the graph are embedded on a number of parallel straight lines in the plane. By adding extra vertices it

can be accomplished that there are only edges between two vertices that are embedded on two consecutive parallel lines. The edges are embedded in the region between two consecutive lines (the segment region). The neighbourships of the vertices that are embedded on the same line are determined by the development of a formula. In case of non-planarity of the graph the constructed formula for the first segment does not meet certain requirements. It seems difficult to gather from this formula indications as to the optimization of the set of non-embedded edges. However, the special way of representing the graph may be utilized in a planarization algorithm that simultaneously attempts to optimize the plane representations of the segments (section 4.4).
Finally we want to mention the planarity test which is published by Knauer [4.28]. The graph has to be decomposed in 3-connected subgraphs, and the efficiency of the method is rather poor. These were the reasons for not considering the method further for application in a planarization procedure.

Beside the planarity tests, some algorithms have been published that try to minimize the number of crossings in embedding a non-planar graph. In spite of the fact that this is not our criterion, these methods may be of interest for us.
Nicholson [4.29] represents all the vertices in a certain sequence on a (horizontal) straight line α. The edges of the graph are partitioned in two sets (the upper and lower), and each of these sets is embedded in one of the half planes that are created by the straight line α (upper and lower). Two edges of the same set, of which the associated end vertices do alternate with respect to the sequence of vertices on α, have to cross each other when they are represented in the same half plane. The number of crossings may be decreased by changing the position of one of the vertices on the straight line α, and revising the distribution of its associated edges over the upper and lower set. This is exhaustively tried (first for the vertex of which the associated set of edges has the highest total number of crossings), until no decrease can be obtained by changing the position of only one vertex. The computation time of the method increases very quickly with the number of vertices of the graph, but the obtained number of crossings is

low (in many cases minimal). Such an approach for planarization does not seem to be so efficient, but because of its easy way of implementing, it may be used to construct material for comparison in order to get some indications about the level of optimality that is reached with the other planarization algorithms that are developed.
Ferrari and Mezzalira [4.30] base their algorithm on the pseudo-hamiltonian planarity test of Auslander and Parter. For each (decomposed) subgraph that is to be treated, the number of crossings (between attachment edges) of all possible pairs of bridges is determined. Next, all possible partitions of the set of bridges in two classes are considered. This is done exhaustively until the sum of the number of crossings of all pairs of bridges in the same class is minimal (the admitted number is repeatedly increased). This process of considering all the partitions of the set of bridges in two classes is rather time consuming.

4.4. The path-embedding planarization method.

In this method a plane representation of a graph (G_s,U_s) which is derived from the graph (G,U), is constructed in steps. The subrepresentation that is obtained after step i will be denoted by (G_i^*,U_i^*), its associated graph by (G_i,U_i). Vertices and edges are called "embedded" or "non-embedded" after the i-th step according to whether they are or are not in G_i or U_i.
The embedded vertices $g\in G_i$ that are associated with non-embedded edges $u\in U\setminus U_i$ form the attachment set G_i^{att}. The elements of G_i^{att} are called attachment vertices. The non-embedded edges that have an attachment vertex as end vertex are called attachment edges. The minimal (G_i,U_i)-bounded subgraphs in the complement $(G\rceil G_i,U\setminus U_i)$ of (G_i,U_i) in (G,U) form the set of bridges BR_{U_i}.
The set of faces in (G_i^*,U_i^*) is denoted by W_i. Let us for the time being consider only one face $w\in W_i$, and the bridges that contain attachment vertices of C_w. Delete all the attachment edges of these bridges, that are incident to attachment vertices that are not in C_w. Two bridges (G_{b_1},U_{b_1}) and (G_{b_2},U_{b_2}) are $C_w[\ ]$-alternating if there is no path in $(C_w,C_w[\ ])$ containing all the attachment vertices of (G_{b_1},U_{b_1}) and none of the attachment vertices of (G_{b_2},U_{b_2}), except its end vertices.

Two bridges that are $C_w[\ ]$-alternating, cannot be embedded both in w in a planar way (theorem 7 in appendix A).
From every attachment vertex of a bridge, there starts at least one path, a so-called "attachment path" of the bridge. The path is proceeded until a vertex is met having degree $\neq 2$ (the "hinge"). The union of the attachment paths that have the same hinge, is called a "pier" of the bridge. Bridges that consist of one path, have by definition only one pier (if the path contains non-embedded vertices, one of them is defined to be the hinge of the pier). The part of the bridge that is not contained in any of its piers, is called the "inner part".
The piers are connected subgraphs of the bridge, and we therefore can conclude:

If two (or more) piers of a certain bridge are $C_w[\ ]$-alternating, then the bridge cannot be embedded in face w in a planar way.

Let us first sum up the important steps of the algorithm. The representation (G_i^*, U_i^*) achieved after the i-th step of the algorithm is extended by embedding some paths of the (non-embedded) bridges. We will refer to the whole process of determining these paths, together with the actual extension of the representation, as "extension step". During an extension step each bridge will be considered for (partial) embedding in only one of the faces. Therefore, first of all it is decided for each bridge, in which face it will be considered for embedding ("bridge-to-face assignment"). Next, the alternations between each pair of piers of the same bridge are abolished in such a way that the total sum of the least-weighed edge of the "non-embeddable" attachment paths of the piers in question is minimum. These paths will no longer be considered as paths for embedding in this particular extension step. Thereupon the alternations between the sets of remaining attachment paths of the bridges are abolished (the inner part is considered as a vertex). Now, all the attachment paths that are left, can be embedded in the appropriate faces without having to introduce any crossing. However, since all the decisions that are made during the bridge-to-face assignment and the abolishments of the alternations, need not be optimal, one may act very cautiously and

embed not all possible but only a few of the attachment paths.

Any plane representation of some subgraph of (G,U) will do as a starting point for the algorithm. Thus, an edge may be chosen as initial representation. In that case we should choose for an edge having high edge weight, since an "embedded" edge will not be deleted anymore in later phases of this procedure. If the terminals have to be on the chip in a prespecified sequence, we start with the consecutivity edges between the vertices that are associated with these terminals. In order to ensure the possibility to keep the terminals on the boundary of the graph, we added - in section 3.1 - an extra vertex, together with edges between this vertex and all the terminals. However, during the construction this can easily be achieved by excluding the interior of the outer face for embedding. Thus the hub and its associated edges need not be added. The procedure stops the extension of the current plane subrepresentation (G_i^*,U_i^*) as soon as all bridges of the complement of (G_i,U_i) consist of only one edge, for which there is no face boundary that contains both end vertices: In each extension step the following steps are executed

-1- Construction of the bridges.

Applying "depth-first-search" [4.32] we first build the bridges that contain at least one non-embedded vertex of degree ≥ 3 (such vertex is taken as the vertex to start the search). If an embedded vertex is encountered during the search, it is stored as an attachment vertex of the bridge in question. The hinge of the respecting attachment path is the vertex of degree ≥ 3 that is last encountered in the search so far.To any attachment path an "attachment weight" is assigned. This is equal to the weight of the edge with the smallest weight among the edges of the attachment path in question. For any attachment path the attachment weight is computed by updating during the search. If all bridges containing vertices of degree ≥ 3 have been considered, the algorithm turns to the construction of bridges containing no vertices of degree ≥ 3, the so-called "path bridges".

-2- The bridge-to-face assignment.

In the subsequent part of the extension step, we will consider each bridge for (partial) embedding in one of the faces. For every bridge we first determine the face(s) for which the sum of the attachment weights of all the associated attachment paths is the highest. Only this set of faces will be considered as being possibly assigned to the bridge in question. The cardinality of the set is called the "embedding number" of the bridge.

It is very easy to take account of the component contact doubling modification. If a certain attachment path cannot be attached to a certain face, we determine whether this becomes possible by the admissible doubling of some component contact. Of course the attachment weight in question has to be greater than the respecting modification cost of the contact doubling. This cost will be substracted from the total attachment weight that is associated with the face.

In determining the assignment of the faces to the bridges it is worth-while to take account of the alternations between the sets of attachment paths that can be associated with each face. For each pair of bridges that can be assigned to the same face an "alternation cost" can be computed, e.g. by determining the lowest weighed set of attachment paths the deletion of which abolishes the alternations of the bridges with respect to the concerned face.

A final assignment having a sub-optimal "total alternation cost" may be constructed in the following way:

a) construct an initial assignment:
   with increasing embedding number, assign the bridges to the admitted face in which the alternation cost with the already assigned faces is minimal

b) optimize the constructed initial assignment by repeatedly changing (if possible) the current assignment of one of the bridges, under the condition that the total alternation cost decreases. Since the number of possible assignments is finite, this process will be finite.

The "inner parts" of the bridges will not be considered in abolishing the alternations. Therefore it is recommendable to extend the current plane representations very cautiously (embed only very few

high weighed paths). In that case the number of "extension steps" is rather high. The determining and updating of the alternation costs for all possible choices makes the assignment procedure time consuming. Because of these facts, it was decided to apply the following, very simple assignment procedure (without the determination of any alternation cost), which appeared to be satisfactory:

assign the bridges (with increasing embedding number) to the admissible face having the lowest "embedding weight". With embedding weight we mean the total sum of the attachment weights of the attachment paths of bridges that are already assigned to the face. Since the bridges that have only one attachment vertex associated with the face do not alternate with any other bridge, their contribution is not included in the embedding weight.

-3- The abolishment of the alternations between piers of the same bridge.

In order to ensure that in the embedding of the attachment paths of a bridge (in the face that is assigned to the bridge), the (sub) representation remains plane, there is searched for alternations between the piers in question. If an alternation occurs, then the lowest weighed set of attachment paths is determined, deletion of which will abolish the alternations. Let us discuss the process of determining this set for two $C_w[\ ]$-alternating piers (G_A,U_A) and (G_B,U_B).

Let the face w be stored as the sequence of vertices

$$v_1,v_2,\ldots,v_{|C_w|}$$

Let the sequences of attachment vertices on this face be denoted:

$v_{a1},\ldots,v_{ai},\ldots,v_{aa}$ for pier (G_A,U_A)

$v_{b1},\ldots,v_{bj},\ldots,v_{bb}$ for pier (G_B,U_B)

where $a1 \leq ai \leq aa$

$b1 \leq bj \leq bb$

$a1 \leq b1$ (no restriction)

There are two cases in which no alternation occurs:

case 1: there is no v_{ai} for which $b1<ai<bb$

case 2: there is no v_{bj} for which $a1<bj<aa$

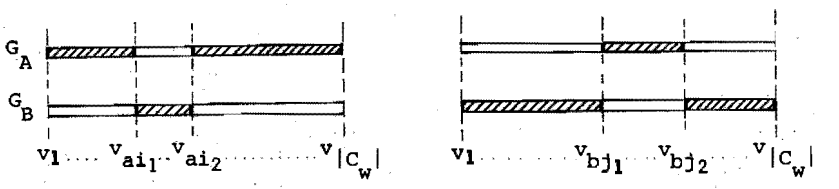


Fig. 4.2. The two cases for which the two graphs (G_A,U_A) and (G_B,U_B) are not C_w-alternating. The embeddable attachment paths occur only in the shaded area of the faceboundary.

The following numbers are associated with the vertices v_i of the face:

LEFT A[i]= the sum of the attachment weights of the attachment paths of pier (G_A,U_A), that are connected with the vertices v_j, where $j \leq i$, of the sequence in question.

RIGHT A[i]= idem, except $j \geq i$.

With respect to pier (G_B,U_B) , LEFT B[i] and RIGHT B[i] are defined in the same way.

Furthermore define LEFT A[0]= LEFT B[0]= 0.

The optimal abolishment satisfying case 1 is obtained by determining the value ai_1 and ai_2 $(2 \leq ai_2 \leq |C_w|)$ for which the following expression is maximal (see fig. 4.2):

$$\text{LEFT B}[ai_2]+\text{RIGHT A}[ai_2]+ \max_{1 \leq ai_1 < ai_2} (\text{LEFT A}[ai_1]-\text{LEFT B}[ai_1-1])$$

The optimal abolishment satisfying case 2 is obtained by determining the values bj_1 and bj_2 $(2 \leq bj_2 \leq |C_w|)$ for which the following expression is maximal:

$$\text{LEFT A}[bj_2]+\text{RIGHT B}[bj_2]+ \max_{1 \leq bj_1 < bj_2} (\text{LEFT B}[bj_1]-\text{LEFT A}[bj_1-1])$$

The solution of the case having maximal attachment weight "placed" fixes the attachment paths that are considered to be non-embeddable during the rest of the extension step (the attachment paths in the non-shaded area in fig. 4.2).

-4- The abolishment of alternations between bridges.

The same procedure as in "-3-" is used to abolish the alternations between the sets of ("embeddable") attachment paths of different bridges being assigned to the same face.

-5- The embedding of a number of (attachment) paths.

We are certain now that all remaining attachment paths can be embedded in a plane way in the face they are assigned to. In the preceding steps, decisions have been taken which need not to be entirely optimal, and which will not be adjusted before the next extension step. If we refresh these decisions as soon as possible (after each little change in the current situation), a better final result may be accomplished. Thus one comes up with the following very cautious embedding-strategy: "embed the attachment path having maximal attachment weight". A non-cautious, but much faster strategy would be: "embed all the embeddable attachment paths", but of course all kinds of intermediate forms can be applied.

Remark: If the attachment of a path can only be accomplished in a certain face by the application of the component-contact-doubling modification, the doubling is only performed in case the strategy chooses this attachment path for embedding.

Some difficulties may arise if certain "dangling" paths are embedded (i.e. paths having only one attachment vertex). In the subsequent extension steps, the attachment vertex of such a path is contained at least twice in the sequence of face vertices. If such a vertex is at the same time an attachment vertex of some bridge(s) in these subsequent steps, we would have to determine to which of the identical vertices in the sequence of face vertices the attachment path in question has to be attached such that there is minimal alternation cost. Furthermore, if a number of dangling paths have to be embedded that have the same attachment vertex, it may be difficult to determine the best sequence for them to leave the attachment vertex. A dangling attachment path does not give these difficulties (is not "troublesome") if another attachment

path of the same pier is embedded at the same time, or if there is no other non-embedded edge incident to its attachment vertex. The problems just mentioned are avoided by as long as possible deferring the embedding of these troublesome attachment paths (in the non-cautious strategy) or by continuing the troublesome attachment path until another attachment vertex on the face is reached, and consider this "continued" (non-dangling) path for embedding (in the cautious strategy; for the construction of such a path see below). If only troublesome paths are left in a face, in both strategies "continued" paths are considered for embedding. If such a path does not exist, the troublesome attachment paths can be embedded without difficulties, since all the bridges have only one edge attached to the face assigned to them. The construction of a non-dangling "continued" path is performed as follows. The objective is to find a path in the bridge in question starting with the troublesome path and reaching the face in some other attachment vertex of the bridge (a "target"), such that the least-weighted edge is maximum (or almost maximum). An exhaustive search for all possible paths may become time consuming (for larger bridges). In order to speed up the search, we abandon the assurance to find an optimal path: we avoid that searches are exploited that fail because of the impossibility to reach a target without crossing vertices that are already on the path. Let the starting vertex (origin) be the attachment vertex in question, and the targets be the other attachment vertices of embeddable attachment paths of the face. The vertices of the bridge are partitioned into a set of ordered classes: the class with the lowest "class number" consists of the origin, and the class having the highest class number consists of the targets. We now search for an optimal path between the origin and one of the targets, on the restriction that the class number of the class to which the subsequent path vertices belong, is increasing. The class partitioning starts with assigning the origin to the first class. To the i-th class it assigns all non-classified vertices of the bridge (except the targets), that are connected with a vertex of the (i-1)-th class via an edge. All the targets form the last class.
The path construction algorithm is recursive and has to be started

in the origin. Each search is only continued (recursive call) via vertices that are an element of a higher class than the vertex reached.

For every vertex y, the "path weight" PW[y] and the "trace-label" TL[y] are computed.

PW[y]= the minimal edge weight in the current path between origin and vertex y (this path can be constructed by tracing back via the trace-labels)

TL[y]= the vertex that precedes y in the current path from origin to y.

The pathweight is maximized during the search: if the vertex y is reached via a better path, TL[y] is changed appropriately. A path search need not be continued if the path weight becomes less or equal than the best solution (path weight of a target) that is already found ("SOLUTIONPW").

In pseudo-Algol the recursive algorithm has the following form:

```
procedure PATHSEARCH (x)
begin  for  all y connected to x  ([x,y]∈U) do
       if  CLASSNUMBER[y]>CLASSNUMBER[x]  then
       begin  NEWPATHWEIGHT:= minimum (PW[x], WEIGHT[x,y]);
            if  NEWPATHWEIGHT>PW[y]
                and NEWPATHWEIGHT>SOLUTIONPW then
            begin  PW[y]:= NEWPATHWEIGHT ; TL[y]:= x;
                 if y∈{targets} then
                 begin SOLUTIONPW:= NEWPATHWEIGHT; TARGET:= y
                 end
                                  else PATHSEARCH(y)
end  end  end;
```

4.5. The cascade-embedding planarization method.

Let us assume that we have a bipartite graph (G,U). Then it is easy to partition the set of vertices G into a number of subsets G_i (classes), such that there are only edges between two vertices that are in two consecutive classes. Let s∈G, then the classes are obtained by executing the following steps:

1) i:=0 ; G_o:={s}

2) $i:=i+1$

3) $G_i:=\{y \mid ([x,y]\in U)\wedge(0\le j<i)\wedge(y\notin G_j)\wedge(x\in G_{i-1})\}$

4) if $G_i\neq\emptyset$ then goto step 2, else $k:=i-1$.

Let us now represent the graph (G,U) on the plane Π in a special way. Consider $k+1$ parallel straight lines α_i $(0\le i\le k)$ in this plane Π. Let all α_i $(0<i\le k)$ be situated in Π in the same half plane that is determined by α_o, and let their distance to α_o increase with i. Represent all vertices of G_i on α_i, thus defining a linear order relation R_i on the set of vertices G_i $(0\le i\le k)$. Next all edges are represented as open Jordan curves, such that they do not intersect the lines α_i.
Now, we have obtained a so-called "(k)-leveled representation" (G^*,U^*) of the graph (G,U). The subgraph (S_i,U_i) is called i-th segment (graph) if

$S_i = G_{i-1}\cup G_i$ and

$U_i\subseteq U$ the set of edges that are associated with a vertex of G_{i-1} and of G_i. The embedding of a segment (S_i,U_i) is called a "segment representation" if:

- the sets of vertices G_{i-1} and G_i are placed on the parallel straight lines α_{i-1} and α_i
- the edges of U_i do not intersect these straight lines α_{i-1} and α_i, except for the end vertices.

The segment representation is called plane if the edges of U_i do not intersect each other.
Let us orient α_{i-1} and α_i in the same way and number the vertices on each line in accordance with their order relations R_{i-1} and R_i. Let for each edge $u\in U_i$, the function $f_i(u)$ give the number of the associated vertex on α_{i-1} and let the function $g_i(u)$ give the number of the associated vertex on α_i. Then the following can be stated [27]:

There exists a plane segment representation of (S_i,U_i) fif there exist ordering relations R_{i-1} and R_i, and a numbering of the edges such that $f_i(u)$ and $g_i(u)$ are both monotone non-decreasing functions ("segment criterion").

- Assume that we have a plane segment representation of the segment (S_i,U_i). Then we can introduce a Cartesian X-Y-coordinate-system

such that the line α_{i-1} is fixed by the equation $x=x_o$ and α_i by $x=x_2$. With y_j^p we denote the y-coordinate of the crossing point between the straight line $x=x_p$ and the Jordan curve representing edge $u_j \epsilon U_i$. Let us number the vertices of G_{i-1} according to the ordering of their y-coordinates, and let us do the same for the vertices of G_i. Since the edges do not cross each other, except possibly in their end points, we can number the edges according to the ordering of the y-coordinates on $x=x_1$, where $x_0<x_1<x_2$. Let $n_i(u)$ denote the number of edge $u \epsilon U_i$ and thus

$$\forall_{u_j \epsilon U_i} \forall_{u_p \epsilon U_i \setminus \{u_j\}} [n_i(u_j)<n_i(u_p) \leftrightarrow y_j^1<y_p^1]$$

Since the edges can only cross in their end points,

$$\forall_{u_j \epsilon U_i} \forall_{u_p \epsilon U_i \setminus \{u_j\}} [y_j^1<y_p^1 \leftrightarrow y_j^0 \leq y_p^0 \wedge y_j^2 \leq y_p^2]$$

Now define $f_i(u_j)=y_j^0$ and $g_i(u_j)=y_j^2$, then

$$\forall_{u_j \epsilon U_i} \forall_{u_p \epsilon U_i \setminus \{u_j\}} [n_i(u_j)<n_i(u_p) \leftrightarrow f_i(u_j) \leq f_i(u_p) \wedge g_i(u_j) \leq g_i(u_p)]$$

- Conversely, let us have a numbering of the edges, denoted by the function $n_i(u)$, such that $f_i(u)$ and $g_i(u)$ are both monotone non-decreasing functions.

Let the vertices on α_{i-1} be ordered

$$g_1, g_2, \ldots, g_{j_1}, \ldots, g_{p_1}, \ldots, g_a \quad \text{, where } a=|G_{i-1}| \text{ ,}$$

and let the vertices on α_i be ordered

$$g_1', g_2', \ldots, g_{j_2}', \ldots, g_{p_2}', \ldots, g_b' \quad \text{, where } b=|G_i|$$

Consider the region on the inside of the closed Jordan curve $J_0=(C_0^*, C_0^*[\ ])$ that is determined by the straight line segments between the pairs of vertices:

$$(g_1,g_2),\ (g_2,g_3), \ldots, (g_a,g_b'), (g_b',g_{b-1}'), \ldots, (g_2',g_1'), (g_1',g_1)$$

For two edges of U_i, $u_j=[g_{j_1}, g_{j_2}']$ and $u_p=[g_{p_1}, g_{p_2}']$,

$$n_i(u_j)<n_i(u_p) \leftrightarrow (j_1 \leq p_1) \wedge (j_2 \leq p_2) \quad \text{holds}$$

The closed Jordan curve J_0 can be partitioned in two sequences of

straight line segments $(g_{j_1},g_{j_1+1}),\ldots,(g_a,g'_b),\ldots,(g'_{j_2+1},g'_{j_2})$.
containing g_{p_1} and g'_{p_2}

and $(g'_{j_2},g'_{j_2-1}),\ldots,(g'_1,g_1),\ldots,(g_{j_1-1},g_{j_1})$.

not containing g_{p_1} and g'_{p_2}, except possibly on the ends.
No pair of edges of U_i is $C_0[\ ]$-alternating. Furthermore, for all edges (bridges) $u\in U_i$ the graph $(C_0,C_0[\ ]\cup\{u\})$ is planar, and thus $(C_0,C_0[\ ]\cup U_i)$ is planar (theorem 8 of appendix A). According to theorem 7 of appendix A, there exists a plane representation of $(C_0,U_i\cup C_0[\ ])$, in which $(C_0^*,C_0^*[\ ])$ is face boundary. The edges of U_i can be embedded in a planar way in the inner region determined by J_0, and thus a plane segment embedding of (S_i,U_i) does exist.

In a plane k-leveled-representation of (G,U), all order relations of R_i $(0\le i\le k)$ are such, that for every segment the segment criterion holds.

Let us now give the edges of the graph an orientation: every edge [x,y] with $x\in G_{i-1}$ and $y\in G_i$ $(1\le i\le k)$ is directed from x to y. The digraph thus created is denoted by (G,V).
Consider the order relation R_i defined on G_i. Let G_i^{i+1} be the vertices of G_i that are associated with the edges of the (i+1)-th segment. Then two vertices x and y of G_i^{i+1} with xR_iy are called "G_i^{i+1}-adjacent" if there is no vertex $z\in G_i^{i+1}$ for which $xR_iz\wedge zR_iy$. A subset $X\subset G_i^{i+1}$ is called "G_i^{i+1}-grouped" if the graph (X,N) is a path, where

$$N=\{[x_1,x_2]\mid (x_1\in X)\wedge(x_2\in X)\wedge(x_1 \text{ and } x_2 \text{ are } G_i^{i+1}\text{-adjacent})\}$$

A G_i-ancestor set of $x\in G$ in the digraph (G,V), denoted $ANC_i(x)$, is defined as

$$ANC_i(x)=\{y\mid (y\in G_i)\wedge(\text{there exists a chain } (C,C[y,x>) \text{ in } (G,V))\}$$

If sinks occur only in the (highest) class G_k, then in a plane k-leveled-representation of (G,U), all G_i-ancestor sets of all vertices $x\in G_j$ $(i+1\le j\le k)$ are G_i^{i+1}-grouped $(0\le i<k)$.

Assume that the G_i-ancestor set of $x\in G_j$ is not G_i^{i+1}-grouped. Then there would be at least one vertex $c\in G_i$, such that

$aR_ic\wedge cR_ib$, where $a\in ANC_i(x)$, $b\in ANC_i(x)$ and $c\notin ANC_i(x)$.

Let $(C_1,C_1[a,x>)$ and $(C_2,C_2[b,x>)$ be two chains in (G,V).
Since there are no sinks in the classes G_p $(i\leq p<j)$, and $c\notin ANC_i(x)$, there exists at least one vertex $y\in G_j$, $x\neq y$, such that there exists a a chain $(C_3,C_3[c,y>)$ in (G,V).
The chains $(C_1,C_1[a,x>)$ and $(C_3,C_3[c,y>)$ are disjoint, since otherwise there would exist a chain in (G,V) between c and x, which would mean that $c\in ANC_i(x)$. For the same reason $(C_2,C_2[b,x>)$ and $(C_3,C_3[c,y>)$ are disjoint.
Consider the region between the straight lines α_i and α_j in which the chains have to be embedded and the sequence in which the vertices of G_i and G_j occur on the boundary $(C_\alpha,C_\alpha[\ ])$ of this region. Two cases can be distinguished: x is on α_j ordered before y (xR_jy), or y is on α_j ordered before x (yR_jx). Assume xR_jy, then the sequence of vertices is:

$$g_1,\ldots,a,\ldots,c,\ldots,b,\ldots,g_{|G_i|},g'_{|G_j|},\ldots,y,\ldots,x,\ldots,g'_1$$

Since $(C_2,C_2[b,x>)$ and $(C_3,C_3[c,y>)$ are $C_\alpha[\ ]$-alternating the leveled representation would not be plane, which is a contradiction with our hypothesis.
In case yR_jx the two chains $(C_1,C_1[a,x>)$ and $(C_3,C_3[c,y>)$ would alternate.

This planarization algorithm will construct in the i-th step a plane segment representation of a subgraph of the i-th segment (where $i=1,2,\ldots,k$). By the construction of the i-th plane (sub)segment, the order relation R_i on the set G_i is fixed. Since this ordering influences the segment representations that will be constructed in the subsequent steps, the non-embedded part has to be considered in optimizing the i-th segment:

- simultaneously with the construction of the i-th plane (sub)segment, the tentative representation of the (i+1)-th segment is considered;
- the G_i-ancestor sets of all vertices $g\in G_j$ $(i+1<j\leq k)$ are preferably kept G_i^{i+1}-grouped.

In embedding the edges of a segment we have restricted ourselves to the regions between the two parallel lines in question. However, in case there are sources or sinks in the plane representation, it may be possible to embed a not yet embedded edge in a planar way, by admitting

these edges to leave the segment region (see fig. 4.3).

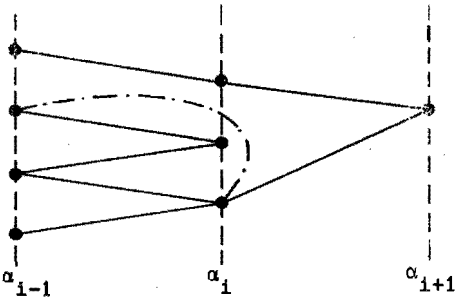


Fig. 4.3. The embedding of edge [x,y] in a planar way, by leaving the segment region between α_{i-1} and α_i.

By taking the following measures, it can be ensured that there are no (real) sinks or sources in the segments of graph (G,U).

- Choose an edge [s,t] of the graph (G,U).
- Construct a drain of (G,U\{[s,t]}), having source s and sink t.
- By replacing some of the arcs by chains of (new) arcs (new vertices have to be introduced), it can be accomplished that all the chains between the source and an arbitrary vertex have equal length ("stretching"). It is now easy to fix the segments, such that G_0={s} and G_k={t}. The edge [s,t] will be embedded in the outer face of the plane segment representation of a subgraph of (G,U\{[s,t]}).

However, in order to satisfy the segment criteria, it may be advantageous, or even necessary, to create sources and/or sinks. The utilization of the possibilities to represent the not yet embedded edges (after the construction of all the plane segments) by leaving the segment regions is achieved by the execution of a suitable embedding procedure at the end (for example the "path-embedding planarization procedure" of section 4.3).

There are some other reasons to apply such a final embedding procedure:

1. The number of extra vertices (and edges) that have to be added in stretching the graph, may grow considerably, and give rise to a large increase of the computation time. Therefore it is decided to use the simple construction of the segments as described in the

first paragraph of this section, which generally will come out with several sinks.

2. The process of constructing the plane segment representations is not very suitable for the implementation of the component contact doubling modification. In the final embedding procedure the application of this modification can be handled.

Let us now describe the different stages of the algorithm.

-1- First the set of vertices G is partitioned in a number of classes (say k+1). The partition is obtained with the simple algorithm described at the beginning of this section. The graph (G,U) has to be bipartite, which means that in the potential graph all consecutivity edges have to be replaced by a path of two edges (new vertex).

In case that the bonding pads of the circuit have to be positioned on the periphery of the chip in a previously specified sequence the outer wheel need not be added during the construction of the plane leveled representation if we take the set of terminal vertices as the class G_0, and embed these vertices on α_0 in the prespecified sequence (thus fixing the order relation R_0 of the first segment).

If no sequence is prescribed for the terminals, we choose as starting vertex the vertex that is connected with all terminal vertices and that represents the "outer world".

-2- Next, plane segment representations are constructed of subgraphs of the segments (S_i,U_i). This is done in the sequence i=1,2,...,k.

Let us consider the embedding of the i-th segment.

With the i-th segment a matrix can be associated, the "segment matrix" M_i^{i-1}:

- the columns correspond to the vertices of G_{i-1}
- the rows correspond to the vertices of G_i
- if there exists an edge $u=[x,y]\in U_i$, then the matrix entry associated with x and y has as value the weight of edge u.

If the sequence of columns and rows is according to the order relations R_{i-1} and R_i, then the segment criterion gives the following requirement for the matrix entries that are associated with the edges to be embedded:

if we number these entries such that, with increasing entry number, the associated column number does not decrease, then the row number of the entry should not decrease either.

Thus, only the set of edges that corresponds with a "cascade" of matrix entries, can form a plane segment representation with the order relations R_{i-1} and R_i. A procedure is developed in which the rows and columns will be ordered such that a cascade is constructed containing very much weight (also the structures of other segments are simultaneously taken into account). The cascade is constructed by "placing" rows and columns in a certain sequence. The (partially constructed) cascade has two ends, the upper end U ("left side") and the lower L ("right side"). During the construction it is possible to extend this cascade at each end in two directions:

- the row direction (extension by placing a row)
- the column direction (extension by placing a column).

The set G_i is contained in two segments. In determining the order relation R_i we, therefore, construct two cascades associated with the two segment matrices M_i^{i-1}, and M_i^{i+1}, where, for the sake of convenience, in both cases the vertices of G_i are associated with the rows of the matrices. Since the order relation R_{i-1} has already been determined (and R_{i+1} not), the treatment of the two matrices is somewhat different. We take account of the rest of the non-embedded edges by considering the measure in which the G_i-ancestor sets are G_i^{i+1}-grouped.

Let us assume that both cascades are partially constructed: the entries of the cascades correspond with a number of "placed" rows, and a number of "placed" columns in M_i^{i-1} and in M_i^{i+1}. Since the sequence of columns in M_i^{i-1} is already determined, the set of non-placed columns in M_i^{i-1} is partitioned in a "left non-placed part" and a "right non-placed part" (see fig. 4.4).

We first determine for both segments the column or row direction in which a cascade preferably should be extended. Let us consider the upper "end" (row r^* and column c^*):

- as direction weight we take the sum of the (left) non-placed

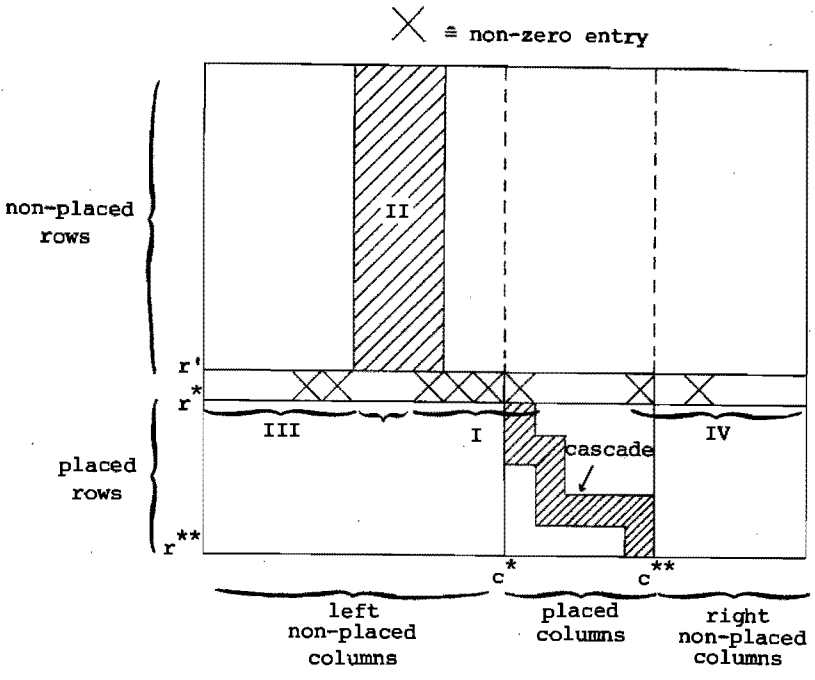


Fig. 4.4. Considering the non-placed row r' for placement on the upper side of the row sequence (M_i^{i-1}).

matrix entries in the considered direction;

- because of the predetermined sequence of columns in M_i^{i-1} one may, as a refinement, in M_i^{i-1} in the row direction only consider that part that probably will be placed (if the sum of the weights in part II, denoted sum(II), is greater than the sum of the weights in part III of row r', it is probable that part III will not be placed in the cascade if r' is placed as next row).

If the column direction of an "end" is favourable, then a column is placed on this "end" of the cascade:

- in M_i^{i-1} of course the column preceding c^* in the predescribed column sequence;
- in M_i^{i+1} we choose the column c' with $M[r^*,c'] \neq 0$ with minimal "column weight" (i.e. the sum of the weights of the non-placed column entries).

This process is repeated until the row direction is favourable in all ends of both cascades.

Thereupon it is determined which (non-placed) row is most suitable for extending the two cascades at one of the sides (upper or lower). We compute for every non-placed row an "upper row weight" and a "lower row weight". The row with maximal row weight is placed at the appropriate end of both cascades. Each row weight consists of three contributions, W_1,W_2 and W_3, emanating from M_i^{i-1}, M_i^{i+1} and the G_i-ancestor sets respectively:

- The contributions W_1 and W_2:

  They are made linearly dependent on the matrix entry which is considered for placement in the cascade ($M[r',c^*]$).

  Let (left) "rowsum" be the sum of the weights in the (left) non-placed part of the row r'. Since the rows with low rowsum are preferred to be placed first, a factor (F<1) is included in the expression of W_1 and W_2, that decreases with increasing rowsum. The rows of which the (left) rowsum is equal to zero (and $M[r',c^*]\neq 0$) should have first priority and no mutual preferences. Therefore an upper bound M_m for $M[r',c^*]$ in the expression, is introduced: $M_m=\min\{M[r',c^*] \mid M[r',c^*]\neq 0 \wedge$(left) rowsum $=0\}$. Because of the prescribed sequence of columns in M_i^{i-1}, it may sometimes be disadvantageous to place a row r' having $M[r',c^*]\neq 0$. The method can be refined by taking account of the part of the left rowsum of which the placement is not likely when extending the cascade with row r'. We therefore determine for the left non-placed columns c' having $M[r',c']\neq 0$, the sum of the weights in part II of the matrix M_i^{i-1} (see fig. 4.4). If the sum of the weights of part III of row r' (denoted sum(III)) is greater than sum(II), then it is likely that part III is not placed in the cascade. If sum(I)<sum(III) the row r' should be placed later on, and therefore W_1 is made negative in this case:

  W_1:= sum(I)-sum(III).

  If a part II of the matrix is met, for which

  sum(I)<sum(II)<sum(III), the cascade will probably not contain entries of part II. The row r' should be placed later on, so that we still have the possibility to place the entries of part II and III in the cascade

  W_1:= sum(I)-sum(II).

  Furthermore we take into account whether it is better to place the row on the other side of the cascade (right rowsum>left rowsum).

- The contribution W_3:

  The G_i-ancestor sets are handled via the "ancestor set matrix". With every ancestor set a column is associated, and again the rows correspond to the vertices of G_i. A matrix entry has value 1 if the corresponding row vertex is element of the corresponding ancestor set. In extending the sequence of rows with a non-placed row r', we distinguish 3 possibilities for each ancestor set $ANC_i(x)$:

  a) an interruption of the sequence of one's in the column is obtained

  b) an extension of the sequence of one's in the column is obtained

  c) otherwise.

  The different cases have different contributions to W_3: $-C_x$, $+C_x$ and 0 respectively, where C_x can be made dependent on the weights associated with the arcs pointing to vertex x.

  If a certain ancestor set $ANC_i(x)$ is interrupted or extended, then the contribution of ancestor sets having $ANC_i(x)$ as subset, is made zero.

-3- After the construction of the plane leveled representation, a final embedding procedure is employed, that utilizes the possibilities to extend the representation further by admitting the edges to leave the segment regions. The path-embedding method of section 4.3 is very suitable, since any plane representation can be used as initial representation. Furthermore the component contact doubling modification is considered in this procedure.

In order to keep the terminal vertices in the prespecified sequence on the outer face, the outer wheel is added again. If a subgraph of the representation is not connected to the part that contains the first segment, this subgraph is deleted. Possible troublesome dangling paths are avoided by deleting the dangling paths from the representation (exhaustive deletion of all vertices of degree 1).

4.6. The permutation planarization method.

In the first part of this algorithm a special plane representation will be constructed in which

- the vertices are embedded on a (vertical) straight line α
- each embedded edge is a Jordan curve that only has its end vertices in common with the straight line α.

During the construction, each edge of the graph (G,U) is an element of one of the following three sets:

a) the set U_L which consists of the edges that are embedded in the left-hand-side half plane determined by α

b) the set U_R which consists of the edges that are embedded in the right-hand-side half plane determined by α

c) the set U_N, which consists of the edges that are not embedded in the plane.

Since the representation has to be kept plane, the alternations among the sets of end vertices of the edges, with respect to the sequence of the vertices on α, will be considered. The edges in U_L (U_R) are not allowed to alternate mutually.

The total weight of the edges in U_N is called the total modification cost of the (current) partitioning of the set of edges U in the three subsets U_L, U_R and U_N. To each vertex v a "vertex cost" $v\gamma$ can be related, that is equal to the total weight of the non-embedded edges associated with this vertex. In the main step of the algorithm the total alternation cost of the current plane representation (partitioning of U) is exhaustively decreased by changing the position of one vertex in the sequence of vertices on α and adjusting the partitioning of the set U.

The following stages are distinguished in the method.

1) Construction of the initial representation.

   a) if the terminals of the electrical circuit have to occur on the periphery of the chip in a prespecified order (requirements B1 and B2), the concerned c-vertices are embedded on α in this sequence, and the consecutivity edges are all contained in U_L. The position of these vertices will not be changed in the subsequent steps and the consecutivity edges will be kept in U_L. The hub of the outer wheel, and the edges associated with it, need not be added then, since all other vertices will never be embedded inside the circuit of consecutivity edges (they are embedded on α). If B1 and B2 are not required, one can start with the embedding of one edge (in U_L), e.g. an edge having maximal weight.
   Thus, $U_R=\emptyset$ and $U_N=U\backslash U_L$.

   b) Repeatedly embed the other vertices on α (first the non-embedded one having maximal current vertex cost).

Embed such a vertex v on a suitable place of the sequence: Consider the set of edges $U_v=\{[v,v']\mid (v' \text{ is already embedded}) \wedge ([v,v'] \in U_N)\}$. Embed the vertex v on α (add v to the sequence of embedded vertices) such that the total weight of the edges that can be embedded in a planar way (do not alternate with the already embedded edges in U_L or U_R (such that no alternation is intro-assigned to the set U_L or U_R (such that no alternation is introduced), and deleted from U_N.

2) Repeatedly try to decrease the current alternation cost by changing the position of one vertex in the sequence of vertices on α and adjusting the partitioning of the set of edges.
For each vertex v (first the one having maximal vertex cost) each possible position p in the sequence is considered.

Let U_v be the set of edges that are associated with v. For each edge $u \in U_v$ the minimum alternation cost C_u with respect to position p is determined: C_u = minimum ($u\omega_e$, C_u^L, C_u^R), where

$u\omega_e$ = edge weight of u

C_u^L = the total weight of the edges in U_L that alternate with u

C_u^R = the total weight of the edges in U_R that alternate with u.

In this way the position is determined for which the sum of these minimum alternation costs of all edges of U_v is minimal.

The sequence of the vertices on α, and the partitioning of the edges is adjusted to the situation corresponding with this minimum (vertex) cost. This step is repeated until for none of the vertices a decrease in the total alternation proces is obtained.

3) It may be possible to embed some of the non-embedded edges in a planar way by permitting them to cross the straight line α. Therefore the path-embedding planarization method of section 4.3 is performed on the obtained plane representation (dangling paths deleted). In this way also the "doubling of component contacts" is exploited as a modification.

4.7. The behaviour of the described planarization methods.

In this section we give some results obtained by the implementations

of the described planarization methods in the preceding sections.
For the path-embedding planarization method we developed two versions of the algorithm:

-A1- implying a non-cautious embedding strategy: "all embeddable attachment paths are embedded in each extension step".

-A2- implying a cautious embedding strategy: "only the maximally weighed embeddable attachment path is embedded in each face".

Because this method is also used as final embedding procedure for the other methods, we created also two versions for these methods:

-B1- the cascade-embedding method, having A1 as final embedding procedure.

-B2- the cascade-embedding method, having A2 as final embedding procedure.

-C1- the permutation method, having A1 as final embedding procedure.

-C2- the permutation method, having A2 as final embedding procedure.

The examples have been chosen such that it was possible to compare the results with results that have been obtained in literature (Fairchild layouts, semi-automatically or completely interactively obtained solutions). The various versions are tested on the following four examples:

the operational amplifiers	µA 709	(53 vertices, 83 edges)
	µA 725	(79 vertices, 124 edges)
	µA 741	(61 vertices, 104 edges)
and the camera timing device	TAA 580	(32 vertices, 53 edges)

We have distinguished ten different "kinds" of admissible deviations from the formal layout. They are listed in table 4.1, together with their costs.
The resulting total modification costs that are obtained by applying the various methods on the given examples are depicted in table 4.2.
In all cases the versions with the cautious embedding strategy gives lower or equal modification costs compared to the versions in which the non-cautious embedding strategy is applied.
In three of the four examples (especially the largest two examples), method B_2 behaves better than A_2 and C_2. There are no significant differences in the resulting costs and all the solutions can be used for practical realization.

kind of the deviation	cost
1. introducing a cross-under	99
2. replacing a pinch resistor by a diffused resistor (and cross between the contacts)	98
3. crossing between the contacts of a capacitor	97
4. splitting an island potential contact (the cost decreases with increasing island area)	40↔80
5. negative supply-potential contact splitting	40
6. replacing a lateral pnp-transistor by a substrate-pnp	30
7. crossing a low-valued p-type diffused resistor (<1kΩ) (increasing cost according as the value decreases)	10↔25
8. crossing a high-valued p-type diffused resistor (>1kΩ)	10
9. deleting a muliple island potential component contact of an EP-component (npn-collector edge or pnp-base edge)	5
10. doubling a transistor contact potential	1

Table 4.1. Deviations from the formal layout.

method / circuit	A1	B1	C1	A2	B2	C2
µA 709	143	213	95	86	213	95
µA 725	364	220	112	279	92	112
µA 741	571	377	438	406	299	438
TAA 580	184	167	53	27	27	53
sum of costs	1262	977	698	798	631	698

Table 4.2. The total modification costs of the obtained solutions.

For each example the set of applied modifications given by the solution obtained by method B2 is depicted in table 4.3. The applied modifications in existing realizations (Fairchild), and some solutions that are obtained by interactive planarization ([4.16,4.31]), are also given. The completely automatically obtained solutions certainly compete with these data.

	µA 709			µA 725		µA 741			TAA 580	
	B2	[16]	FAIR-CHILD	B2	FAIR-CHILD	B2	[16]	FAIR-CHILD	B2	[31]
cross-under						1	3	4		
island potential contact splitting	2		1		3	3		3		
substrate-pnp	1	1	1		1	1	3	1		
resistor	2	13	8	3	8	1	7	3	1	5
multiple island potential contact	2	1	3	10	10	6	8	5	3	2
transistor contact doubling	1	2	2	1	4	4	2	1	2	1

Table 4.3. Differentiation as to the nature of the applied modifications.

The methods are implemented on the Burroughs B6700. The computation times for the examples are depicted in table 4.4.
As expected, the non-cautious embedding strategy of method A1 is least time consuming(the construction makes progress very fast), however its results are of lower quality compared to those of the other methods.

method / circuit	A1	B1	C1	A2	B2	C2
µA 709	5	7	44	28	17	55
µA 725	9	20	121	57	54	143
µA 741	10	18	74	50	47	97
TAA 580	3	5	7	14	16	10

Table 4.4. Computation times for the different methods (seconds on B6700).

All methods have acceptable computation times, that are small in comparison to the times necessary to develop the solution

by hand or interactively.
The storage requirement for the given examples was, in all cases, less than 10 kilo words.

In order to get some further indications about the quality of the constructed solutions we employed a procedure that tries to optimize the alternation cost of obtained solutions further. In this procedure it is examined for each non-embedded edge [x,y], whether it is possible to embed the edge in a planar way by leaving a number of embedded edges out of the current representation, such that the total weight of the deleted edges is less than the weight of [x,y]. The lowest weighted set of edges that has to be deleted in order to embed [x,y] in a planar way is called the [x,y]-crossing set of the considered representation. The path optimization procedure of appendix C is apt to be used in finding a curve between x and y that crosses a number of edges of which the sum of their edge weights is minimal.
Let W_s be the set of faces of (G_s^*, U_s^*), and let ω_e relate the weights with the edges of the graph. The "cells" of the path search will be the edges of U_s^*:

$$C = \text{set of cells} \triangleq U_s^*$$

Each cell is labeled (by the relation σ), with the weight that is associated with the edge:

$$S = R_+ \quad , \text{ and } \quad \sigma \triangleq \omega_e$$

Two edges that subsequently will be crossed by the curve that is to be constructed, are part of the face boundary of the face that is crossed by the curve, and thus the neighbour relation η defined on $C \times C$, is:

$$(c_i, c_j) \in \eta \leftrightarrow \exists_{w \in W_s} [c_i \in C_w[\ ] \wedge c_j \in C_w[\ ]]$$

If some edge is reached that is on the face boundary of a face that also contains the vertex y, the search can be stopped, since the curve can be connected with y without crossing any other edge. Therefore, as the set of targets is taken:

$$C_t = \text{set of targets} \triangleq \{u \mid \exists_{w \in W_s} [y \in C_w \wedge u \in C_w[\ ]]\}$$

As set of origins we take a set of edges that have x as end vertex, but certainly need not be crossed by the curve:

$$C_o = \text{set of origins} \triangleq \{u \mid \exists_{c \in G_s^*}[u=[x,c]]\}$$

The weighing relation is defined by simply adding the edge weight associated with the edge that is reached by the search:

$$(a,s,b) \in \mu \leftrightarrow b=a+s$$

If $C_o \cap C_t \neq \emptyset$ then the search need not be performed, since x and y are already on the same face, and [x,y] can be embedded without having to delete any edge of the current plane representation. If $C_o \cap C_t = \emptyset$ the search is performed, and an optimal sequence of cells $c_0, c_1, \ldots, c_p$ is found, where $c_o \in C_o$ and $c_p \in C_t$. The edges $c_1, \ldots, c_p$ have to be deleted from (G_s^*, U_s^*) in order to be able to embed [x,y] in a planar way.

Remark: The search can be made faster by decreasing the number of cells (and neighbours of the cells) by the exhaustive application of the following two steps before the search:

- replace the edges $u_1=[c,c_1]$ and $u_2=[c,c_2]$ that are associated with the vertex $c \in G_s^* \setminus \{x,y\}$ that has degree 2 in the plane representation, by one edge $[c_1,c_2]$, and associate with this edge the minimum of the edge weights $u_1\omega_e$ and $u_2\omega_e$.
- delete the edge $u \in U_s^*$ that is associated with some vertex $c \in G_s^* \setminus \{x,y\}$ having degree 1 in the plane embedding.

The above described process is applied for all the non-embedded edges [x,y] of the graph (G,U) (first the maximally weighted edges are treated). If the total weight of the edges in the [x,y]-crossing set in the current representation is less than the weight of [x,y], the appropriate changes in the representation are performed. We will refer to the process as "cutting-in" process. The optimization of the representation stops, when for none of the non-embedded edges, improvement of the representation is obtained. In table 4.5 we depict for the various examples the total modification costs. The computation times for the execution of the optimization process are about 1 to 5 seconds (B6700), and are of course dependent on the number of non-embedded edges.

method / circuit	A1	B1	C1	A2	B2	C2
µA 709	143	213	95	86	213	95
µA 725	305*	97*	112	101*	92	107*
µA 741	433*	284*	397*	406	299	397*
TAA 580	105*	64*	53	27	27	53
sum of costs	986*	618*	657*	620*	631	652*

Table 4.5. The total modification costs after the optimization process. (means that the modification cost is decreased by the process).*

The differences between the methods have decreased considerably.

4.8. Measures to guarantee the extra requirements.

For a number of edges their embedding in the plane representation has to be ensured. We will refer to these edges as "restriction edges". First of all there are the restriction edges that belong to the "outer wheel", which is incorporated in the potential graph because of the extra requirements for the terminals of the chip (B1 and B2). Secondly, there are the restriction edges that belong to the "inner wheels" representing the (super)components that have a prescribed sequence of contacts.
Until now we have only taken measures to ensure the embedding of the outer wheel. However, in using some planarization methods it still might occur in some cases that an inner wheel is not embedded completely. In particular this will be the case for the cascade planarization method, because of the special representation that is used there. In figure 4.5 an example is depicted showing a class partitioning such that a plane leveled representation cannot contain all the edges of the wheel. In figure 4.6. a situation in which an inner wheel will not be embedded completely, is depicted for the path-embedding procedure having the non-cautious embedding strategy.

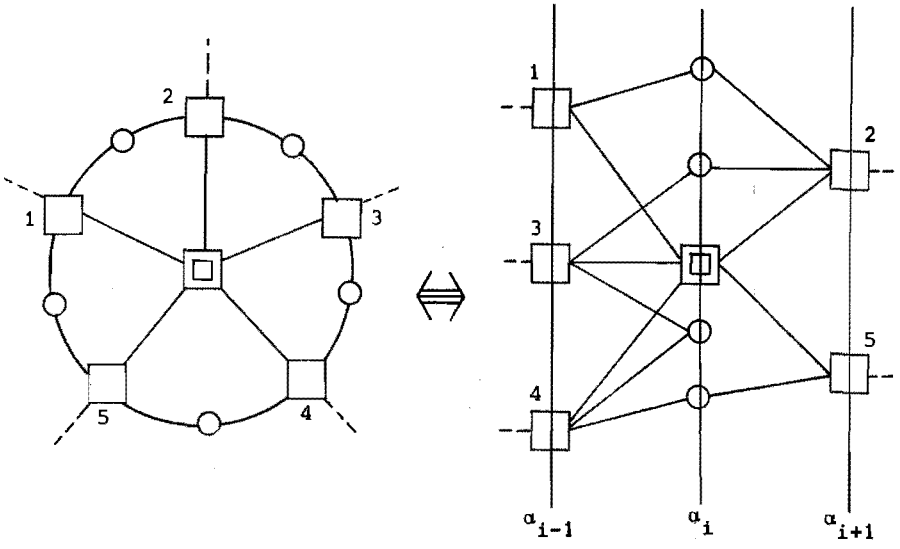


Fig. 4.5. Situation in which an inner wheel will not be completely embedded (cascade-embedding planarization method).

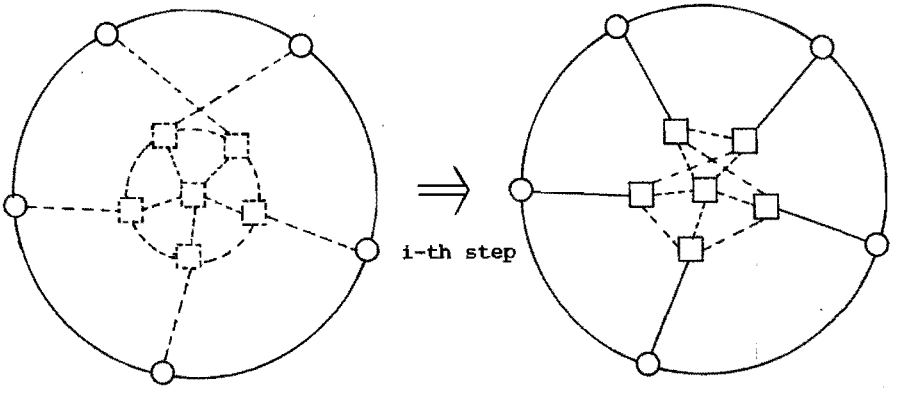


Fig. 4.6. Situation in which an inner wheel will not be completely embedded (non-cautious path-embedding planarization method). The dashed edges are not yet embedded.

It is possible to take special measures for the various methods such that the embedding of the inner wheels is guaranteed. For example, one can ensure the embedding of the complete wheel as soon as one of the terminal contact vertices of the wheel is embedded. In that case we should have indications as to the orientation in which the wheel can be embedded best. It may be rather difficult to obtain these indications. Furthermore, the occurrence of inner wheels will be rather seldom, which makes it not recommendable to charge the normal case (no inner wheels!) with the process of keeping track whether a wheel vertex is actually considered for embedding, or not.
An easy way to guarantee the embedding of the complete inner wheels is by using the cutting-in process of section 4.7. This procedure can be applied to any plane representation, no matter which planarization procedure has been applied. If an inner wheel is not completely contained in the plane representation, its embedding is accomplished by the following steps.

a) Perform the cutting-in process for each of the non-embedded consecutivity edges of the wheel. In order to avoid that other, already embedded, wheels partially have to be deleted, it has to be ensured that no subgraphs are embedded in a face that is only bounded by edges of an inner wheel. This can easily be incorporated in the path-embedding planarization method: do not consider the embedding of paths in the faces that contain a hub and that have only three edges on the face boundary. Furthermore, the edges of a wheel obtain such a high weight, that they will never be contained in a crossing set determined by the cutting-in process.
b) Delete from the current representation, all the edges and vertices that are in the inside region of the constructed rim of the wheel in question.
c) Embed the hub of the wheel, together with its associated edges (the spokes) in the created face of consecutivity edges.
d) In step -b- we may have deleted some vertices. We take care of their embedding by applying the path-embedding planarization procedure on the current representation.

In this way the embedding of all inner wheels can always be accomplished, because each wheel is planar and all wheels are mutually disjoint.

4.9. Working out the determined set of modifications.

With each non-embedded edge, a certain deviation from the formal layout is associated (by the modification code). The following adjustments are performed in order to ensure the application of these deviations.

1) Let U_t be the set of non-embedded edges that are associated with the island potential splitting modification, and that have the island potential vertex t as end vertex. Then, consider the set of c-vertices that are associated with these edges: $C_t=\{c|[c,t]\in U_t\}$ and determine a face w of the plane representation that has the highest number of vertices of C_t on its boundary. The face boundary of this face is denoted $(C_w^*,C_w^*[\ ])$. A new t-vertex t' is introduced, representing the same (island) potential as t. The edge relation of the potential graph is changed such that each edge [c,t], with $c\in C_w$, is replaced by an edge [c,t']. These edges are, together with the new vertex t', embedded in the face w. Furthermore, it is checked whether t' is connected with a c-vertex representing an EP-component of the island. If this is not the case, a new c-vertex c' is introduced and embedded, representing an extra island contact, together with an additional edge [c',t']. The same check is executed for vertex t. This process is repeated until all edges of U_t are replaced (and embedded).

2) If there are non-embedded edges having a modification code that corresponds with the splitting of the substrate potential, a similar process as in the preceding case is performed.

3) If the code corresponds with the modification that changes the type of a component, such that the substrate becomes part of its structure (e.g. using the substrate pnp-transistor instead of the lateral pnp), the type of this component is adjusted appropriately.

4) Let the code of the non-embedded edge [c,t] correspond with the modification allowing to cross the component represented by c, between its contacts. Then delete the c-vertex c together with its

associated edge, from the representation. In a subsequent step of the layout procedure it will be determined by which potentials the components will be crossed (chapter 5), and according to the number of crossing potentials some component parameters may then be changed.

5) If the code of the non-embedded edge [c,t] corresponds with the cross-under modification, we introduce a new c-vertex c' representing the cross-under, and a new t-vertex t' representing the same potential as t does. The edge [c,t] is replaced by the edges [c,t'], [t',c'], and [c',t]. The vertex c' and the edges associated with it, are not (yet) represented (however t' and [c,t'] are embedded). In the subsequent procedure that determines the potentials that will pass between the contacts of the cross-under, the component parameters of this component can be fixed.

6) If the code of a non-embedded edge corresponds with the deletion of some (multiple) island potential contact of an EP-component, the component type of the component in question is changed to the appropriate type in which the island potential contact is not incorporated. If in this way the island potential contacts of all the EP-components in the island are deleted, then a loose island contact (this is a separate component in the library!) may have to be added. If there are still edges associated with the t-vertex representing the island potential in question (thus connecting the t-vertex with a c-vertex representing an EP-component of an other island, or some NEP-component), a new c-vertex c' is embedded, together with an edge [t,c']. The c-vertex c' represents the loose island potential contact.

Remark: The contraction of the edges of inner wheels if they are present, and the deletion of the outer wheel, is not yet performed, since their presence can with advantage be used in the procedures of the chapter 5 and 6.

Beside the described adjustments in the data, due to the different deviations that have been determined by the planarization algorithm we prefer to delete the superfluous multiple contacts of the same island potential. The occupied chip area can be reduced by contacting the

island potential as few times as possible. Each deletion of such a contact has as consequence the deletion of an edge incident to each t-vertex representing the island potential, and to a c-vertex representing an EP-component of the island. Furthermore, the component type of the component in question has to be changed into an appropriate type in which the island potential contact is not incorporated.
If a t-vertex, representing an island potential, is connected with more than one EP-component of the island, we delete all but one of the island potential contacts. Here we have used the fact that the layout will be realized with the minimum number of islands (see chapter 6).
It may be possible that by the deletion of some island contacts, an applied component contact doubling becomes unnecessary. In this case the consequences of this modification for the potential graph are cancelled.

The plane representation thus obtained is denoted by (G_s^*, U_s^*).

4.10 Example.

Once again we choose the μA725 to illustrate the result of the planarization process, and to give the plane representation that will be used in the subsequent parts of the layout design procedure. Contrary to section 4.7. we will now consider the extra requirement of having a prespecified layout for the two input transistors because we want to apply for these components the highly symmetrical layout that has been used in the Fairchild realization of the μA725. Then, the interconnection leads enter the region of the supercomponent (corresponding with c-vertex 54 of the potential graph depicted in figure 3.2) in the sequence 2,3,9,12,13 (t-vertices representing potentials). Application of the various implemented planarization algorithms to the potential graph containing the wheel of supercomponent 54 resulted in the following costs:

method	A_1	B_1	C_1	A_2	B_2	C_2
total modification cost	232	170	92	91	106	87
computation time (sec.)	20	30	100	78	66	130

The versions using the cautious embedding strategy give the better solutions compared with the ones using a non-cautious strategy. The cautious version having the lowest computation time is version B_2, and its solution will be considered for the illustration of the various parts of the layout design process. The obtained plane representation before working out the consequences of the applied modifications, is depicted in figure 4.7. The following modifications are applied:

- deletion of edge [4,78] (implying a change in type of transistor 78 from lateral pnp to substrate pnp);
- deletion of the edges [21,52],[4,42],[31,53],[8,36] (which means that the resistors 52,42,53 and 36 are to be crossed);
- deletion of the edges [7,61],[22,59],[24,60],[29,74],[32,72], [32,77] and [14,67]. (i.e. the island potential contacts of the transistors 59,60,61,67,72,74 and 77 are not to be laid out);
- addition of the t-vertex 29' and edge [29',73]; furthermore the edges [29,70] and [29,75] are replaced by [29',70] and [29',75]. (i.e. contacting the base of transistor 73 on two sides).

The consequences of these various deviations from the formal layout are processed by the steps of section 4.9..The resulting plane representation is depicted in figure 4.8..Beside the deletion of the edges that are incident to the c-vertices representing the resistors that have to be crossed, a number of edges have been deleted, due to the necessity of only one island potential contact for each island. Of course the types of the various components in question have to be changed in the appropriate type that does not include the island potential contact. The deletion of these edges make the contact doubling of transistor 73 superfluous: vertex 29' can be deleted an [29',70] can be replaced by [29,70].

References.

[4.1] C.Kuratowski, "Sur le problème des courbes gauches en topologie", Fund. Math., Vol. 15, pp. 271-283, 1930.

[4.2] H.Whitney, "Non-separable and planar graphs", Trans. Am. Math. Soc., 34, pp. 339-362, 1932.

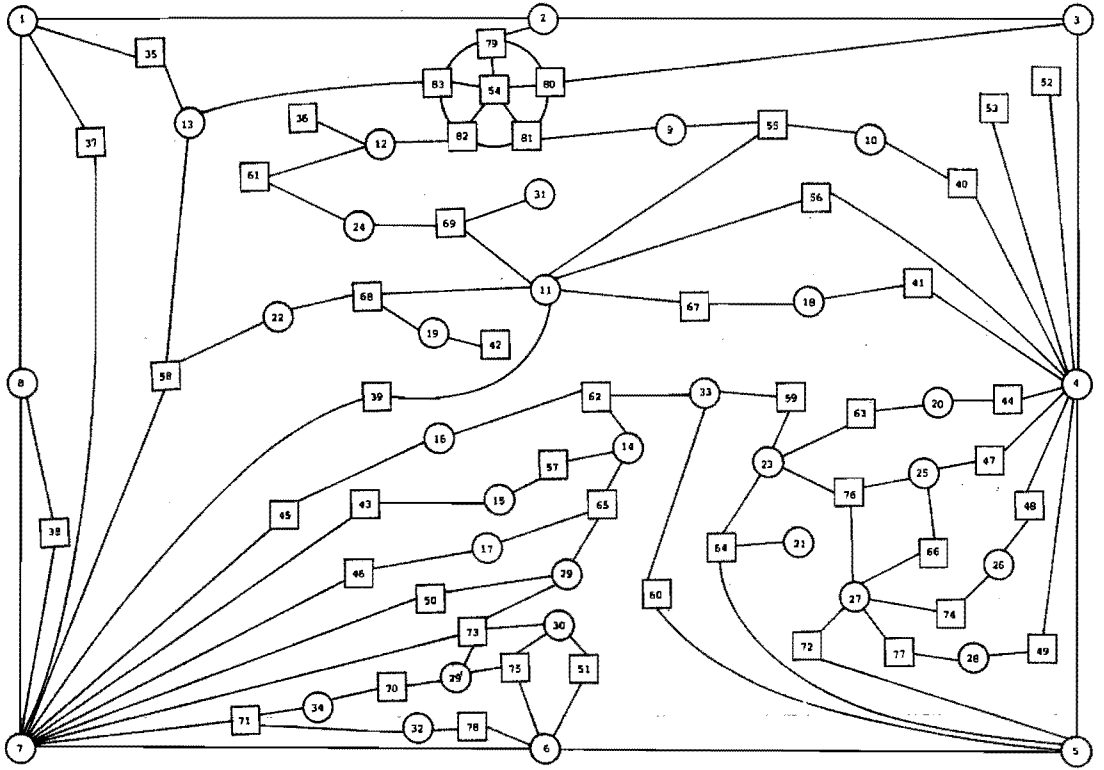


Fig. 4.7. The planarized potential graph for the μA725 (with supercomponent).

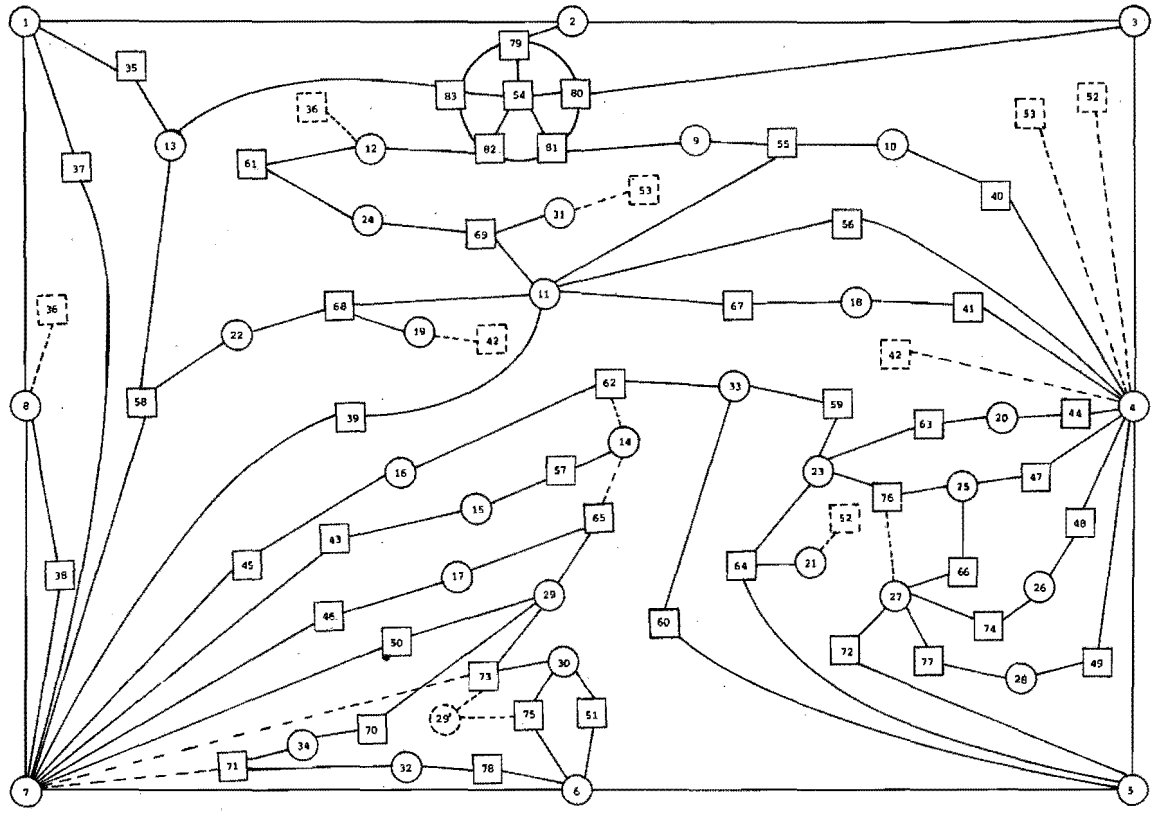


Fig. 4.8. The planarized potential graph (μA725) after working out the consequences of the applied modifications (the dashed edges are not included in the plane representation).

[4.3] S.MacLane, "A structural characterisation of planar combinatorial graphs", Duke Math. J., Vol. 3, nr. 3, pp. 460-472, 1937.

[4.4] C.Picard, "Graphes complémentaires et graphes planaires", Rev. Franc. Recherche Operationelle 8 (1964), pp.329-343.

[4.5] M.Vidyasagar, "An algebraic method for finding a dual graph of a given graph", IEEE Trans. on Circuit Theory, CT-17, pp. 434-436, August 1970.

[4.6] J.J.Dambit, "Einbettung von Graphen in die Ebene", Wissensch. Zeitschrift der T.H. Ilmenau, nr. 3, pp. 63-70, 1971.

[4.7] W.R.Dunn and S.P.Chan, "An algorithm for testing the planarity of a graph", IEEE Trans. on Circuit Theory, CT-15, pp. 166-168, June 1968.

[4.8] L.Auslander and S.V.Parter, "On embedding graphs in the sphere", J. of Math. & Mech., Vol. 10, nr. 3, pp. 517-523, 1961.

[4.9] W.Bader, "Das topologische Problem der gedruckten Schaltung und seine Lösung", Archiv für Elektrotechnik, Vol. 49, nr. 1, pp. 2-12, 1964.

[4.10] G.J.Fisher and O.Wing, "A correspondence between a class of planar graphs and bipartite graphs", IEEE Trans. on Circuit Theory, CT-12, nr. 2, pp. 266-267, 1965.

[4.11] P.M.Lin, "On methods of detecting planar graphs", 8-th Midwest Symposium on Circuit Theory, Colorado St. Univ., Fort Collins, pp. 1.0-1.9, 1965.

[4.12] G.J.Fisher and O.Wing, "Computer recognition and extraction of planar graphs from the incidence matrix", IEEE Trans. on Circuit Theory, CT-13, nr. 2, pp. 156-163, 1966.

[4.13] M.Saito and K.Muraki, "A condition for the planarity of electronic circuits", Electronics and Communication in Japan, 51-C, pp. 166-167, 1968.

[4.14] P.Dierick, R.Lavic and J.L.Martin, "Tracé automatique de masques de circuits integrés hybrides", L'onde electrique, Vol. 49, fasc. 1, pp. 98-103, 1969.

[4.15] S.H.Chung and P.H.Roe, "Algorithms for testing the planarity of a graph", 13-th Midwest Symposium on Circuit Theory, pp. VII 4.1-4.12, 1970.

[4.16] P.Pernards, "Elektrotechnisch sinvolle Planarisierung von Schaltungsgraphen", Ph.D. dissertation, T.H.Aachen, May 1973.

[4.17] D.A.Mlynski and P.Pernards, "Planarization of circuit graphs", Proceedings of the 6-th Hawaii Intern. Conf. on System Sciences, pp. 389-392, 1973.

[4.18] K.Yoshida and T.Ohta, "Topological layout of a monolithic IC", Electronics and Communications in Japan, Vol. 52-C, No. 12, pp. 173-178, 1969.

[4.19] A.J.Goldstein, "An efficient and constructive algorithm for testing whether a graph can be imbedded in a plane", Graph and Combinatorics Conf., Office of Naval Res. Logistics Proj., Contract Nr. NdNR 1858-(21), Dept. of Math., Princeton Univ., 1963.

[4.20] G.Demoucron, Y.Malgrange, and R.Pertuiset, "Graphes planaires: reconnaissance et construction de representations planaires topologique", Revue Francaise de Récherche Opérationelle, Vol. 8, nr. 3, pp. 34-47, 1964.

[4.21] R.W.Shirey, "Implementation and analysis of efficient planarity testing algorithms", Ph.D.-dissertation, Computer Science Department, Univ. of Wisconsin, 1969.

[4.22] F.Rubin, "An improved algorithm for testing the planarity of a graph", IEEE Trans. on Computers, Vol. C-24, no. 2, February 1975.

[4.23] G.Hotz, "Einbettung von Streckenkomplexen in die Ebene", Math. Annalen, Vol. 167, pp. 460-472, 1966.

[4.24] J.Bruno, K.Steiglitz, L.Weinberg, "A new planarity test based on 3-connectivity", IEEE Trans. on Circuit Theory, Vol. CT-17, nr. 2, pp. 197-206, May 1970.

[4.25] J.Hopcroft and R.Tarjan, "Efficient planarity testing", J. of the ACM, Vol. 21, No. 4, pp. 549-568, October 1974.

[4.26] A.Lempel, S.Even and I.Cederbaum, "An algorithm for planarity testing of graphs", in Theory of Graphs, Intern. Symp., Rome, July 1966, Paris (Dunod), pp. 215-232.

[4.27] U.Klemm, "Ein algorithmisches Planaritätskriterium", Computing 3, pp. 194-204 and 245-257, 1968.

[4.28] B.Knauer, "Normalformen planarer Graphen", Computing 10, pp. 121-136 & 137-152.

[4.29] T.A.J.Nicholson, "Permutation procedure for minimizing the number of crossings in a network", Proc. of the IEE, nr. 1, Vol. 115, pp. 21-26, 1968.

[4.30] D.Ferrari and L.Mezzalira, "On drawing a graph with the minimum number of crossings", Relazione interna n. 69-11, Laboratorio di Calcolatori, Politecnico di Milano, 1969.

[4.31] H.Klamet, "Computer-Aided Design of the Layout of Integrated Circuits (CADLIC)", Proc. of the Intern. Computing Symposium 1973, 451-458 (1974).

[4.32] R.Tarjan, "Depth-first search and linear graph algorithms", SIAM J. Comput., 1(1972), pp. 146-159.

5. CROSSING DIFFUSIONS

5.1. Searching optimal curves.

After the planarization of the potential graph some components and connections are no longer represented in the modified potential graph. For those modification that allow crossings of metal interconnections over diffusions and isolated regions, we have to determine which metal interconnections cross these diffusions and regions, taking into account that overlap of diffusions must be avoided.

Thus we have a plane representation of a planarized potential graph (G_s^*,U_s^*) with the set of faces denoted by W, and we want to determine the crossings of metal interconnections over a certain diffusion that has to be inserted in the plane representation. This means that we have to find a curve between two vertices, an origin and a target - such that further it only has t-vertices in common with $S_{(G_s^*,U_s^*)}$. The path optimization algorithm of appendix C is apt to solve this problem. For take for the set of cells C the vertices of G_s^*, for the cell-alphabet S the non-negative real numbers R_+, define the neighbour relation η by $(g_1,g_2)\in\eta \leftrightarrow \exists_{w\in W}[g_1\in C_w \wedge g_2\in C_w]$, and let the weighing relation be an addition: $(a,s,b)\in\mu \rightarrow b=a+s$. Further, we give the c-vertices - except origins and targets, and sometimes some other c-vertices - a much higher "weight" than t-vertices. The relation σ assigns these weights to the vertices in order to avoid c-vertices on the curve and to grade t-vertices according to certain priorities emanating from the considerations in section 3.2.

Of course the inserted diffusions should not overlap each other, and therefore we extend the set of cells and change the neighbour relation after each insertion of a curve, in such a way that the curves still to be found cannot cross the curves already traced (they may "touch" each other, i.e. they may pass through the same t-vertex). We will describe this new neighbour relation η' after introducing some new notations and explaining how η is stored.

The vertices C_w of a face boundary of (G_s^*, U_s^*) are stored in a sequence N^w in clockwise succession:

$$N^w = [g_i, g_{i+1}, \ldots, g_x, \ldots, g_y, \ldots, g_j]$$

Any two vertices of such a sequence, g_x and g_y, determine two subsequences:

$$N^w_{g_x,g_y} = [g_{x+1}, \ldots, g_{y-1}] \quad \text{and} \quad N^w_{g_y,g_x} = [g_{y+1}, \ldots, g_j, g_i, \ldots, g_{x-1}]$$

Furthermore, for each vertex $g \in G^*$ a sequence W^g is stored, representing the faces that contain g, also in clockwise succession:

$$W^g = [w_i, w_{i+1}, \ldots, w_x, \ldots, w_y, \ldots, w_j]$$

Any two faces in W^g, w_x and w_y, determine two subsequences:

$$W^g_{w_x,w_y} = [w_{x+1}, \ldots, w_{y-1}] \quad \text{and} \quad W^g_{w_y,w_x} = [w_{y+1}, \ldots, w_j, w_i, \ldots, w_{x-1}]$$

and $W^g_{w_x}$ denotes the sequence $[w_{x+1}, \ldots, w_j, w_i, \ldots, w_{x-1}]$

Let us assume we have found a curve between an origin g_1 and a target g_k, passing through the t-vertices $g_2, g_3, \ldots, g_{k-1}$ and the faces $w_1, w_2, \ldots, w_{k-1}$. For the next curve to be found the set of cells will be extended with new cells: $g_2', g_3', \ldots, g_{k-1}'$, where g_i' corresponds with the same potential and gets the same weight as g_i. The new neighbour relation η' can be derived from the new sequences which will be made as follows:

1) Each N^{w_i}, $1 \le i \le k-1$, is to be replaced by two sequences:

$$N^{w_i'} := [g_{i+1}, N^{w_i}_{g_{i+1},g_i}, g_i] \quad \text{and} \quad N^{w_i''} := [g_i', N^{w_i}_{g_i,g_{i+1}}, g_{i+1}']$$

with $g_1' = g_1$ and $g_k' = g_k$.

2) The cell g_i is to be replaced by g_i' in all sequences N^w for which

$$w \in W^{g_i}_{w_{i-1},w_i}, \quad 2 \le i \le k-1$$

3) Each W^{g_i}, $2 \le i \le k-1$, is to be replaced by two sequences:

$$W^{g'_i} := [w''_{i-1}, W^{g_i}_{w_{i-1},w_i}, w''_i] \quad \text{and} \quad W^{g_i} := [w'_i, W_{w_i,w_{i-1}}, w'_{i-1}]$$

4) Furthermore, $W^{g_1} := [w'_1, W^{g_1}_{w_1}, w''_1]$ and $W^{g_k} := [w''_{k-1}, W^{g_k}_{w_{k-1}}, w'_{k-1}]$

After making these changes the next curve may be searched for. Since new cells have been created the same potential is now associated with more than one cell. In the algorithm we accounted for this situation, since it searches for the optimal curve between a cell in C_o, the set of origins, and a cell in C_t, the set of targets.

Face boundaries formed by consecutivity edges and edges associated with the extra vertex h_o are not used in deriving neighbour relations; else the bonding pads are possibly no longer at the periphery. There always is a solution (i.e. the set L in the path optimization algorithm never is empty).

This is quite obvious, for take the plane representation of (G^*,U^*) and connect the origin and the target by an arbitrary curve in the interior region of the Jordan curve formed by the consecutivity edges. For every point this curve has in common with $S_{(G^*,U^*)}$ we can find a t-vertex via which a curve can go from the one face to the other: if the point is on the edge u we choose the t-vertex associated with u (this vertex must exist because there are only edges between a t-vertex and a c-vertex and between two t-vertices). In case it meets another curve already inserted, it can make a detour via the t-vertices of this curve and the t-vertices connected with the c-vertex that was an origin or a target.

Preferences regarding the crossings of interconnection leads over diffusions can be conveniently translated into a gradation in the weights assigned to the t-vertices by σ. If for example a resistor,

deleted by the planarization procedure, must be inserted and the island potential in which this resistor is to be diffused is much lower than the potential corresponding with a t-vertex, we can assign a much higher weight to this vertex in order to reduce the chance of inversion problems. Detrimental parasitic capacitances or increases of sheet resistances may also be countered by giving the critical cells higher weights. If it is desirable to avoid that particular leads cross a diffusion immediately along each other, this can be achieved simply by changing the neighbour relation into $\eta \backslash \{(g_1, g_2)\}$ where g_1 and g_2 are the vertices corresponding with those interconnection leads. In view of the wiring to be designed in a later stage of the procedure, it is recommendable to give a lower weight to the potentials connected with a component which is in the same island as the diffusion to be crossed. This is, of course, only possible if the distribution of the components over the isolated regions is known, thus after the procedures described in the sections 6.1 and 6.2. Of course, the sequence in which the curves are inserted, has an influence on the result. If there are already some curves inserted, the next curve will possibly get a higher total weight, then it would have had if it were the first curve to be inserted. In any case the total weight will never be lower than the one found on the initial model. It is therefore expedient to insert first the more critical components, for example small resistors before the greater ones.

Our starting point was a certain plane representation of the planarized potential graph, which is in general not 3-connected. Thus there exist other plane representations that might give better results when applying the above method. Though it is practicable to generate all representations and use them as the initial model, it is clear that this would be very inefficient. It is, however, quite easy to evade the restriction emanating from the assignment of a vertex of degree 1 to a certain face boundary, namely by giving the vertex connected with it a zero weight in case the vertex of degree 1 is an origin or a target.

Remark: It is perhaps somewhat obscure how we avoided overlap between diffusions by changing the stored sequences, because it is not easy to visualize the effect in the representation. Nevertheless, fig. 5.1 might help if one realizes that the vertices on the

boundary of the shaded face need not be neighbours. A pair of them only is in η' if they are on the boundary of some other face.

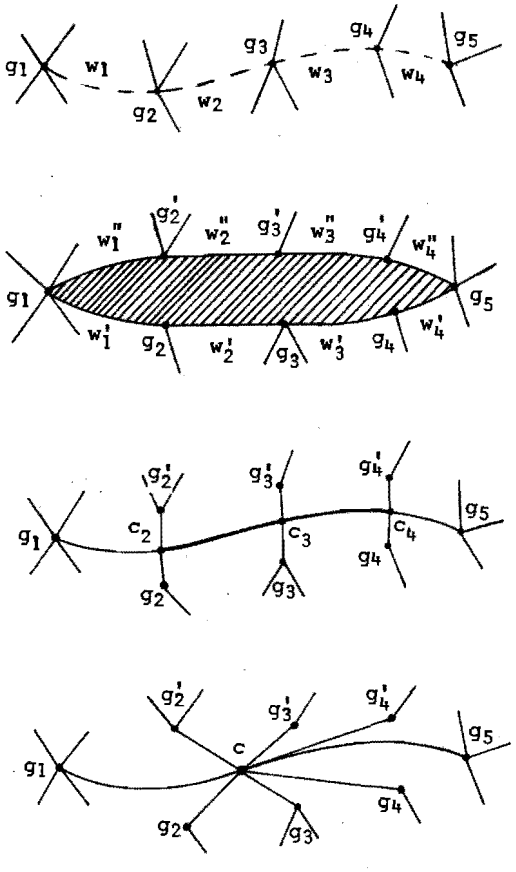


Fig. 5.1. Two visualizations of how overlap of diffusions can be avoided.

Another way of illustrating the main ideas (and also of implementing them), which will get more attention in chapter 6, is the following: Suppose we found a curve between g_1 and g_k passing through $g_2, g_3, \ldots, g_{k-1}$. Consider the curve as a part of the plane representation. Split each of the vertices $g_2, g_3, \ldots, g_{k-1}$ in three vertices (g_i, g_i' and c_i) and connect c_i

with the other two as shown in fig. 5.1. After contraction of all the edges of the curve except the ones associated with g_l and g_k we have an adequate model to continue inserting other diffusions. We associate the vertex generated by these contractions with the component of which the diffusion was inserted. Again, all the components of the circuit are represented in the resulting graph (G_p,U_p). Beside edges symbolizing potential correspondences, there are now edges indicating a crossing too. In this way the information about the crossings is fixed in the graph (G_p,U_p), and can be used by the procedures to be described in the next chapter.

Remark: Strictly, speaking, $(C_w,C_w[])$ is only defined in case the graph is 2-connected, which means that the interior or the exterior of $(C_w^*,C_w^*[])$ is empty. In this chapter we used the notation $(C_w,C_w[])$ though the modified potential graph is not always 2-connected. It is, however, clear that lower connectivity does not affect the results: C_w must be considered as the set of vertices on the face boundary.

5.2. Example.

We let the procedure operate upon the planarized potential graph of the μA725 (fig. 4.7). Four resistor diffusions must be inserted in a sequence corresponding with increasing lengths. Further, higher potentials were preferred over lower potentials (weights between 1 and 2). The result is visualized in fig. 5.2 in the second way, described in the first remark of the preceding section.

References.

[5.1] N.A.Rose, "Computer-aided design of printed wiring boards", Ph.D. dissertation, University of Edinburgh, Edinburgh, Scotland, April 1970.

[5.2] A.J.Fletcher, "Computer programs for the layout of integrated circuits", Ph.D. dissertation, Victoria University of Manchester, Manchester, England, May 1970.

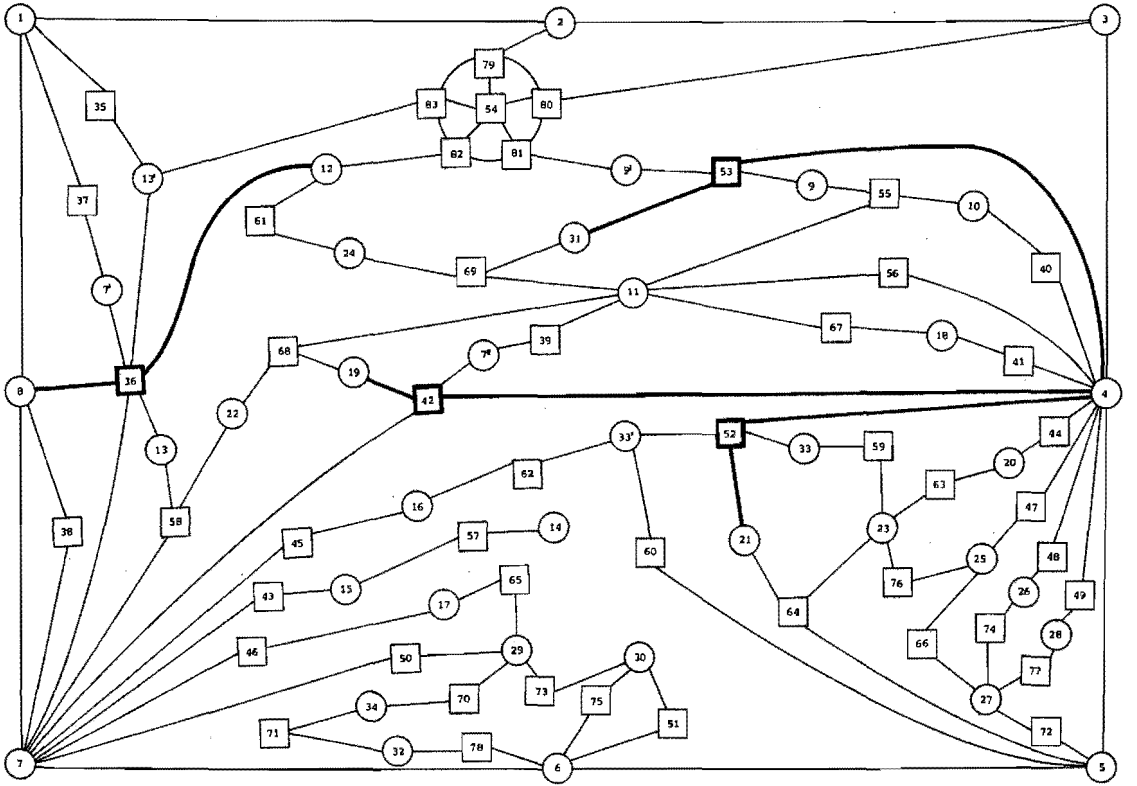


Fig. 5.2. Result of the procedure that inserts the resistors.

[5.3] K.C.Kwok, "Topologische Widerstandseinbettung in integrierten Schaltungen", Ph.D. dissertation, Rheinisch-Westfälische Technische Hochschule Aachen, Aachen, July 1972.

6. DISTRIBUTION OF THE COMPONENTS OVER ISOLATED REGIONS

6.1. Construction of the IR-compatibility classes.

As mentioned in chapter 2, the electrical isolation between the different components is, in bipolar technology, provided by reverse-biased pn-junctions. The n-epitaxial layer of the chip is divided by the deep p-diffusion in isolated regions (also called "islands"). The "island potential" is defined as the potential of its epitaxial layer. The deep p-type isolation channels have a depth greater than the epitaxial layer thickness, and therefore reach the p-type substrate. Since this substrate is connected to the negative supply voltage, the islands are isolated from each other.
The structure of the components and the voltage states of the circuit can be such that electrical isolation between certain components is always assured. Components that are allowed to be placed in the same island are called IR-compatible.
When distributing the components over the isolated regions the following rules have to be observed.

D1: Components of which the epitaxial layer is part of the component structure (the "epitaxial" or EP-components) cannot be embedded in the same isolated regions in case these epitaxial parts are to be connected with different potentials.

D2: Components of which the epitaxial layer is not a part of the component structure (the "non-epitaxial" or NEP-components) cannot be embedded in an island of which the potential is lower than the potential of the deepest diffusion of that component.

These rules determine a number of maximal sets of components the elements of which are mutually IR-compatible (the so-called "IR-compatibility classes" [6.1]).

In table 6.1. the characterizations of the commonly used component types are given.
The n-type resistor can be embedded in the deep p-diffusion, thus forming a separate island. In this case it forms a separate IR-compatibility class, and the component is then considered as "solitary"

component type	NEP or EP	epitaxial contact
resistor (p-type)	NEP	-
resistor (n-type)	(N)EP	-
resistor (pinch)	NEP	-
capacitor (C/B)	EP	t^+
capacitor (E/B)	NEP	-
capacitor (MOS)	EP	t
transistor (npn)	EP	collector
transistor (pnp)	EP	base
diode (B/E)	NEP	-
diode (B/C)	EP	cathode

where EP = "epitaxial" component
NEP = "non-epitaxial" component
/ = junction
MOS = metal-oxide-silicon
E = emitter
B = base
C = collector
t^+ = most positive contact
t = one of the contacts

TABLE 6.1. Characterization of components.

EP-component. However, another possibility is to diffuse the n-type resistor in a p-type area, and shortcircuit the n^+ and p-areas via t^- or connect the p-area with the negative power supply voltage (which needs an extra edge in the potential graph). It then has to be considered as a NEP-component.

For a "supercomponent", which represents a predetermined layout of a certain part of the circuit, a separate region has to be reserved. In its associated "island" no other component is allowed to be embedded, and therefore such supercomponent is treated as a "solitary" EP-component.

Loose island contacts are also treated as EP-components. They have a predescribed geometry which is present in the component library, and

can only be embedded in an island having an island potential which is equal to its contact potential.

In order to be able to construct the set of compatibility classes, at least a simulation program has to be available, which can simulate the circuit in the operating states. It is necessary to know the operating states of the circuit, since it has to be determined whether a certain pn-junction always is reverse-biased or not.
The IR-compatibility classes will be represented by the "compatibility matrix":

- its rows correspond to the compatibility classes
- its columns correspond to the components
- a matrix entry is 1 if the concerned component is contained in the concerned class (else the value is 0).

Thus, compatibility[i,c]=1 means that component c is contained in the i-th compatibility class. Two components are called "i-compatible" if they are contained both in the i-th compatibility class.
The matrix is constructed in the following way. First all EP-components are considered. A new row (class) is constructed if an EP-component is encountered which is "solitary" or which has an epitaxial contact potential (island potential) that is different from the island potentials that are associated with the already constructed rows. Next, for the NEP-components it is checked whether the potential of one of its contacts to the p-type layer is higher than the island potential which is associated with the considered compatibility class. If this is the case for some voltage state of the circuit, the NEP-component is excluded from the respecting class by giving the associated matrix entry the value 0 (in the other case value 1 is assigned).

Remark: For the pinch resistor one has also to test whether the breakdown voltage is exceeded. If this is the case, the component is also excluded from the respecting class.

It is possible that some NEP-component cannot be placed in any of the constructed classes. Then a new row (class) is created, and the positive power supply voltage becomes the island potential of this class. Since this is the most positive voltage of the circuit, all NEP-components

can be placed in this class (all row entries are given value 1).

From the obtained compatibility matrix all possible island distributions can be derived by constructing all partitions of the set of components such that the components in the component subsets of the partition only contain mutually IR-compatible components. The minimum number of islands that can occur in such a partition, is equal to the number of compatibility classes.

6.2. Construction of the final island partition.

In the preceding section we have determined the minimum number of islands that is necessary in order to obtain proper isolation between the components of the circuit: for each compatibility class at least one isolated region has to be created.
As a consequence of some of the modifications applied during the planarization of the potential graph, certain EP-components have to be embedded in the same island. These requirements can easily be satisfied by choosing a partition with the minimum number of isolated regions. Another, important reason for doing this, is that in general the total chip area decreases, since the area occupied by "isolation channels" in most cases can be reduced then (and also parasitic capacitances may be decreased). Furthermore, it may be possible to save area by deleting a number of the epitaxial contacts of the EP-components that are situated in the same island. Because of these reasons we associate with each compatibility class exactly one island. The EP-components are contained in exactly one compatibility class, and will be embedded in the island associated with the compatibility class in question.
Each NEP-component can be element of several IR-compatibility classes (several 1-entries may occur in the corresponding column of the compatibility matrix). We are free to choose one of the islands that is associated with these classes, for their final embedding. In doing so we can try to satisfy other desires, e.g.

-1- if the parasitic capacitance between a certain island potential and the substrate potential (= negative supply voltage) is desired to be minimal, this island should be kept as small as possible. Immediately after the construction of the compatibility matrix

each entry having value 1 in the row corresponding with the island in question is replaced by 0 if it is not the only 1-entry in the column.

-2- In order to reduce the chip area, parasitic effects and delay times, the length of the interconnection leads should be kept low. This is achieved by placing the components that have to be interconnected, close to each other. If components are part of the same island, we generally have the opportunity to place them close to each other. If, however, they are in different islands this might become difficult, since it will not always be possible to make these islands adjoining (sometimes there are very many of such "desired neighbourships" for an island, and not all can be satisfied). In general the distribution of the components over the islands will be such that a number of potential leads have to cross certain island areas without actually contacting a component of such an island. In other words, certain groups of components in the same island may be separated from each other by a number of interconnection leads that are not related to the island. Depending on the number of such crossing leads, and of course their actual width and spacing, we will need an amount of extra chip area for this island. In constructing the final island partition the objective will be to embed components that are interconnected with each other, in the same island, and to keep the number of crossing potential leads over the islands small.

We therefore introduce the notion of "distance" of a set C_i of c-vertices in the plane representation of the modified potential graph, to another set C_j of c-vertices. It is defined as the minimum number of t-vertices that have to be crossed in the considered plane representation, in connecting (some element of) C_i with (some element of) C_j via a curve that is only allowed to cross t-vertices. If $C_j=\emptyset$, the "distance" of C_i is by definition zero. In assigning each NEP-component c to one of its "admissible" islands I (compatibility[I,c]=1), we will try to improve the distances between the c-vertices that represent the components that are currently assigned to the islands. The sequence in which the various distance optimizations are applied will be dependent on preferences that are due to the next two objectives.

-3- The capacitance of a reverse-biased pn-junction is voltage dependent. Parasitic effects due to that capacitance can be reduced by embedding the components in islands having a high island potential.

-4- As a consequence of modifications that are applied in the planarization procedure a certain island potential may have to be available at several points on the chip (a number of extra island contacts is introduced). The greater the island area, the more convenient it is to find a suitable place for these contacts without creating much chip area that will not be occupied (particularly if the island contains resistors having a flexible geometry). While assigning components to islands we can prefer the islands having many epitaxial contacts.

-5- If a p-type resistor is assigned to an island having high island potential, "surface-inversion" may have a detrimental influence, in case that the resistor diffusion is crossed more than once by some metal lead having a much lower potential. The occurrence of this effect mainly appears if meandered resistors are crossed by a metal lead, perpendicular to the direction of meandering. The chance on surface-inversion would be decreased by not assigning the high valued resistors that have to be crossed by some potential lead(s), to an island having a high island potential. However, for a reliable solution of the problem we have to take special measures in the other parts of the layout procedure (wiring, meandering the resistors). This objective has been given the lowest priority.

In the following paragraphs of this section we will describe the procedure that is followed to assign each component c to one of the admissible islands I (compatibility[I,c]=1). In fixing the island partition the most important objective will be to decrease the distances of the (sets of) c-vertices that represent components that are currently assigned to the same island (the desire mentioned under -2-). It is preferred to assign components to the same island if they have to be interconnected with each other via a metal lead. Our starting point is the plane representation (G_p^*, U_p^*) of the modified potential graph that is obtained after the insertion of all the circuit components

that have to be crossed by a metal lead (see chapter 5). The set of vertices G^*_p consists of a set of c-vertices G^*_{pC} and a set of t-vertices G^*_{pT}, where $G^*_{pC} \cap G^*_{pT} = \emptyset$. The set of faces of the representation is denoted W_p. During the assignment procedure no changes are made in this representation, though this might be a possibility to improve the distance of some c-vertex (representing a vertex already assigned to island I) to the rest of the c-vertices representing components of island I. The consideration of all possible plane representations of (G_p, U_p) would be too time consuming.
Furthermore, the information contained in the compatibility matrix (which possibly has been modified by the first of the listed objectives) will be used.
In the assignment procedure the following steps can be distinguished.

-a- First of all, assign all the components that are admissible in exactly one island (exactly one 1-entry in the associated column of the compatibility matrix) to the respective island. Thus, all the EP-components, but also certain NEP-components, will be assigned to an island.

-b- Now, for each island I a number of so-called "island vertex groups" is constructed. These groups contain only c-vertices representing components that are already assigned to the same island in the previous step. A c-vertex c is included in a certain island vertex group C^I_i of island I if one of the following two properties is satisfied:

1. the vertex c is connected (via a t-vertex) with one of the c-vertices that are already in C^I_i ;
2. the distance of c to one of the vertices in C^I_i is zero.

Let $G^I := \{c \mid (c \in G_{pC}) \wedge (c\alpha^{-1}$ assigned to island I)$\}$, then the various island vertex groups of an island I are constructed by the following steps:

1. i:=0
2. i:=i+1
3. Choose $c_1 \in G^I$, and $C^I_i := \{c_1\}$
4. $G^I := G^I \setminus \{c_1\}$

5. If $\exists_{c\in G^I}\ \exists_{c'\in C_i^I}[(\exists_{t\in G^*_{pT}}[[c',t]\in U^*_p \wedge [c,t]\in U^*_p]) \vee (\exists_{w\in W_p}[c\in C_w \wedge c'\in C_w])]$

then

begin $C_i^I := C_i^I \cup \{c\}$; $G^I := G^I \setminus \{c\}$; goto step 5 end

6. If $G^I \neq \emptyset$ then goto step 2.

For each island vertex group C_o of an island I, its (group-)distance (denoted $C_o\delta$) is determined to the set C_t of vertices that are elements of the other groups that are associated with the same island I. If C_t is empty, the distance $C_o\delta$ is zero, but otherwise we compute it by using the path optimization procedure of appendix C by taking:

C= set of cells = G^*_p

S= $\{s_c, s_t\}$ where $s_c \in R_+$, $s_t \in R_+$ and $s_c >> s_t$

$(c_i,c_j)\in\eta \leftrightarrow (\exists_{w\in W_p}[c_i\in C_w \wedge c_j \in C_w])$

set of origins = C_o

set of targets = C_t

$(a,s,b)\in\mu \leftrightarrow b=a+s$

$(c_i,s)\in\sigma \leftrightarrow (c_i \in G^*_{pC} \wedge s=s_c) \vee (c_i \in G^*_{pT} \wedge s=s_t)$

In order to avoid c-vertices on the curve, the c-vertices have been given a much higher "weight" than the t-vertices (relation σ). These c-vertices have not been ignored in the search, because the neighbour relation η uses the information of the faces. The faces are stored by sequences of vertices, and include the t-vertices.

If the number of cells in the obtained curve is p, then the distance $C_o\delta$ which is associated with C_o is equal to p-2.

-c- In this step, a "distance-improvement procedure" is applied on all island vertex groups C_i, in a sequence of decreasing distance $C_i\delta$. In case some groups have equal distance, these groups can be

treated in a preferred order based on the objectives 3 and 4.
Let us discuss the process attempting to improve the distance of group C_o. Let C_o be an island vertex group of island I, and C_1 up to C_z be all the other island vertex groups of this island.
For the group C_o we first perform an "extension search", determining the set of "tentative extenders" of C_o, denoted C_o^{ext}. Such a set consists of c-vertices representing a component c that is not yet assigned to any island, and for which compatibility[I,c]=1.
A vertex c is added to the current set C_o^{ext} if it is connected (via a t-vertex) with, or has distance zero to some element of the set $C_o \cup C_o^{ext}$. This can easily be accomplished by the cell mass determination of the path optimization procedure of appendix C:

C= set of cells = $\{c \mid c \in G^*_{pC} \wedge \text{compatibility}[I,c]=1\}$

S= $\{1\}$

$(c_i,c_j) \in \eta \leftrightarrow (\exists_{t \in G^*_{pT}} [[c_i,t] \in U^*_p \wedge [c_j,t] \in U^*_p]) \vee$

$\vee (\exists_{w \in W_p} [c_i \in C_w \wedge c_j \in C_w])$

set of origins = C_o

set of targets = $\bigcup_{1 \le j \le z} C_j$

$(a,s,b) \in \mu \leftrightarrow b = a+s$

The c-vertices that have got a cell mass $cv \neq \infty$, are the "tentative extenders" of the group C_o. A trace-back relation, denoted λ_o, can be associated with this search, such that it relates any cell that has been reached, with one of the neighbour cells having minimum cell mass. With this relation we can construct an (optimal) curve from a tentative extender to one of the origins, such that this curve contains the minimum number of c-vertices. If a target $c_p \in C_j$ $(1 \le j \le z)$ is reached, only the tentative extenders in $c_p \hat{\lambda}_o$ are actually assigned to C_o ($\hat{\lambda}_o$ is the transitive closure of λ_o). Furthermore, C_o and C_j are joint into one island vertex group. If no target has been found a similar extension search is executed, but now with

set of origins = $\bigcup_{1\le i\le z} C_i$

set of targets = $\emptyset$

The set of vertices that have now got a cell mass, is denoted C^{ext}, and its trace-back relation λ.

This search is succeeded by a "distance-search" between the auxiliary groups

$C_o^{aux}=C_o \cup C_o^{ext}$ and $C^{aux}=(\bigcup_{1\le i\le z} C_i)\cup C^{ext}$:

C= set of cells = G_p^*

$S=\{s_c,s_t\}$ where $s_c\in R^+$, $s_t\in R^+$ and $s_c >> s_t$

$(c_i,c_j)\in\eta \leftrightarrow \exists_{w\in W_p}[c_i\in C_w \wedge c_j\in C_w]$

set of origins = C_o^{aux}

set of targets = C^{aux}

$(a,s,b)\in\mu \leftrightarrow b=a+s$

$(c_i,s)\in\sigma \leftrightarrow (c_i\in G^*_{pC} \wedge s=s_c) \vee (c_i\in G^*_{pT} \wedge s=s_t)$

Let the optimal sequence of cells found by this algorithm be $c_1,c_2,\ldots,c_p$, then the distance between the groups is equal to p-2, and $c_1\in C_o^{aux}$ and $c_p\in C^{aux}$. If the distance is less than the original distance of C_o to $\bigcup_{1\le i\le z} C_i$, then we actually assign those components to the island I that are represented by the tentative extenders in the set $\{c \mid (c\in\{c_1\}\cup c_1\hat{\lambda}_o)\vee(c\in\{c_p\}\cup c_p\hat{\lambda})$, where $\hat{\lambda}_o$ and $\hat{\lambda}$ are the transitive closures of λ_o and λ. The assigned vertices of C_o^{aux} are incorporated in C_o, and the assigned vertices of C^{aux} are incorporated in that particular group C_i $(1\le i\le z)$ of which $(c_p\hat{\lambda})\cap C_i\neq\emptyset$.

Remark: In this step we assign only a part of the tentative extenders in order to have the possibility to use the other ones in subsequent distance-improvement steps. The cells c_1 and c_p can be determined such that the minimum number of tentative extenders is assigned.

However, in order to keep the number of interconnections between the various islands low, we prefer to embed components that are directly interconnected, in the same island. One may therefore (during the extension search) give first priority to extensions with cells that are directly interconnected with a neighbour cell having the current minimum cell mass cv. If absolute preference is given to these extensions, the number of assignments need not be minimal anymore.

Thus we have minimized the distance of C_o to $\bigcup_{1\leq j\leq z} C_j$ via the assignment of a number of NEP-components. However, in extending C_o and/or C_i the distances that are associated with the other groups C_j $(1\leq j\leq z)$ can also have been changed. In order to update these distances (used for the determination of the next group on which the distance-improvement procedure will be applied), maximally two (namely if $z>1$) distance-searches have to be performed: the first with C_o as set of origins and the second with C_i. The search is continued until all other groups are reached, and each time a group is reached its distance is updated if necessary.

Remark: It occasionally can occur that some c-vertex, representing a component not yet assigned to any island, does not belong (anymore) to the set of tentative extenders of any group. It is possible to apply a similar distance-improvement process for such a component, but in practice the chance of real improvement is rather small (since there are several islands admitted for the embedding of its associated component, and the minimum distance of the component to some component that does belong to one of these islands will in general be small; furthermore the number of non-assigned components that can be used for the improvement may be decreased in the former step). Assigning these components to islands is accomplished in step -e-.

-d- In the previous step we have been cautious in assigning NEP-components to an island. In general only a part of the tentative extenders, have actually been embedded in one of the admissible

islands. During this step the rest of these tentative extenders will be assigned to an island. First priority is given to those "extenders" that are connected (via a t-vertex) with a c-vertex representing a component that is already assigned to an admissible island. The islands are, in this step, all the time treated in a sequence that is according to the island preference derived from the desires 3 and 4. The island vertex groups are extended in the following way.

1. First of all, for each island vertex group the following extension is exhaustively executed. Let us consider some vertex c of the group (associated with island I) that is connected via a t-vertex with some vertex c', representing a component that is not yet assigned to any island. If compatibility$[I,c'\alpha^{-1}]=1$, then the vertex c' is incorporated in the group, and the component represented by this vertex is assigned to the island I in question.
2. Next, we extend each vertex group (let us say associated with island I) with a number of maximal sets that have the following properties:
   - its elements are c-vertices representing components that are not yet assigned to any island, and that are I-compatible;
   - each c-vertex is interconnected via a t-vertex with at least one other c-vertex of the set (except if the set consists of only one vertex);
   - the set has distance zero to the island vertex group in question.

   This process is repeated until for none of the island vertex groups, such a set can be found.

-e- For each remaining c-vertex c, representing a component not yet assigned to any island, a distance-search is performed to the set of c-vertices representing the components of all the admissible islands I (compatibility$[I,c\alpha^{-1}]=1$). The component in question ($c\alpha^{-1}$) is assigned to the island of the target-component that has been reached.

6.3. The island interconnection graph.

In placing the components on the chip we want to take into account the spacings that have to occur between certain (sets of) components in

order to make it possible for a number of potential leads to pass between the contacts of these (sets of) components. Before actually placing the components into their islands , and before constructing the geometries of these islands, we have to determine certain information about the potential leads that will pass between certain sets of components, so that area then can be reserved for these leads. Thus, the chance that the wiring procedure is getting stuck because of the fact that a wire path cannot pass between two obstacles (wire blocking) is decreased. Wire blocking should only appear sporadically, since the special measures that have to be taken in creating enough space between the obstacles, may introduce an extra amount of chip area that will not be occupied.

In the preceding section it was an objective to embed interconnected sets of components in the same island. These components can be placed near to each other in the island, and their interconnection leads can be kept short. The interconnection leads between components in different islands will generally be the leads that are rather lengthy, since it is not always possible to keep the islands in question adjacent to each other. For the interconnection leads between the different islands, by preference the space between the islands is used (the area of the isolation channels has to be present anyway). The wiring between the islands will be determined in a separate execution of the wiring procedure (see chapter 9). Therefore we have to fix the information about the potentials that leave the island areas, and about the sequence in which they leave. This information will be determined and stored via the construction of a plane representation of an auxiliary graph, the so-called "island interconnection graph". This graph may also be used to determine some information about the number of wires passing between the various adjacent islands. Extra area can then be reserved for these wires.

In some cases the interconnection leads will have to cross certain islands, without actually contacting any of the components in this island. Decisions have to be made about the potentials that will cross a certain island. One could easily think that the information obtained in the previous section fixes this information: the t-vertices crossed by the curves that are determined by the various distance searches represent the potentials that have to cross the island between the groups of components associated with the various island vertex groups.

These curves in fact "represent" the epitaxial layer of the island in question. Therefore, two curves are not allowed to cross each other if they represent epitaxial layers of different islands. During the determination of the island partition the curves between the various ("growing") island vertex groups are continuously changing. This is one of the reasons that this restriction (that the curves are not allowed to cross each other), is not included already in that procedure. Another reason is that, in fixing the potentials crossing the islands, we want to take acçount of the chip area occupied by these crossings. This will be done by introducing a new type of component, which in the modified potential graph is represented by a c-vertex. Such a component is referred to as a "crossor": it occupies a certain crossing-area, which divides the rest of the respective island I in two parts. Let the sets of components that are embedded in such a part, be denoted C_1^I and C_2^I, and let the total area of the components that are elements of $C_1^I(C_2^I)$ be denoted $A_1^I(A_2^I)$. In general the area of the crossor will be small if one of the areas A_1^I or A_2^I is small (see fig. 6.1). Therefore it seems reasonable to make the area that will be reserved for the crossor, by some way dependent on the minimum of the areas A_1^I and A_2^I .

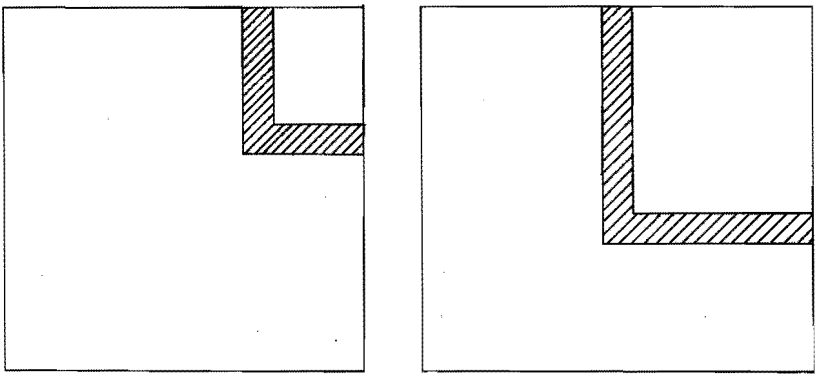


Fig. 6.1. The crossing area (shaded) is small if one of the two parts A_1^I and A_2^I is small.

During the determination of the island partition (previous section) it is difficult to take account of the areas of the various islands (or groups of components of the islands), since then the distribution of

the components over the islands is not yet completely determined!

Let us now define the island interconnection graph (G_{ISL},U_{ISL}) of which a plane representation (G^*_{ISL},I^*_{ISL}) will be constructed and used in determining the wiring between the islands.
The set of vertices G_{ISL} contains two kinds of vertices:
- "island vertices" (i-vertices) corresponding with the islands of the chip; the set of island vertices is denoted G_I $(G_I \subset G_{ISL})$
- "interconnecting-tree vertices" (it-vertices) representing the potential trees that interconnect components of different islands and that do not cross any island. The set of it-vertices is denoted G_{IT} $(G_{IT} \subset G_{ISL})$.

Furthermore $G_I \cap G_{IT} = \emptyset$.
If a potential tree has to cross some islands, it is, by the island boundaries, divided into a number of "interconnecting-potential-trees" that do not cross any island area, and a number of leads crossing only island area. Thus in the island interconnection graph, there may be several it-vertices representing the same potential. If some interconnecting-tree reaches (enters) an island, we say that it "contacts" the island. The set of edges U_{ISL} is defined by the following relation on $G_I \times G_{IT}$:

an i-vertex g_i is related to some it-vertex g_j if the island that is associated with g_i is contacted by the interconnecting-tree that is associated with g_j (denoted $[g_i,g_j] \in U_{ISL}$).

Let us now describe the process of constructing a plane representation (G^*_{ISL},U^*_{ISL}) of an island interconnection graph (G_{ISL},U_{ISL}) that is suitable to be used in the island wiring procedure.
Simultaneously to the construction of (G^*_{ISL},U^*_{ISL}) the plane representation of the modified potential graph is extended with the "crossors" that are introduced.
Starting point is the plane representation of the modified potential graph in which each component of the circuit is represented by a c-vertex. Each c-vertex is associated with the island to which the represented component is assigned.

During the procedure two graphs have to be updated:

- the modified potential graph (G_p,U_p), that will be extended with a number of crossors. The set G_p consists of the set of t-vertices G_{pT} and the set of c-vertices G_{pC}, where $G_{pT}\cap G_{pC}=\emptyset$. Its edge relation will be denoted ρ_p and its set of faces W_p.
- the "contraction graph" (G_{CONTR},U_{CONTR}), which is derived from (G_p,U_p) by pairwise identifying c-vertices that represent components of the same island. Its edge-relation will be denoted ρ_c and its set of faces W_{CONTR}. At the end of the procedure described in this section, this graph is identical to the island interconnection graph that will be used in chapter 9.

The graph (G_{ISL},U_{ISL}) will be constructed by exhaustively identifying two vertices of the contraction graph, representing (sets of) components of the same island. Two identified c-vertices yield a new c-vertex, representing the union of the components that were represented by the two original c-vertices. In identifying two c-vertices, a number of t-vertices may have to be split in order to keep the representation of the graph plane. Let us assume that all the (current) c-vertices, representing a set of components of the island I, are incorporated in C^I. Let us want to identify the vertex $c\in C^I$ (the origin) with one of the vertices of $C^I\backslash\{c\}$ (the set of targets). The path optimization algorithm of appendix C is used to determine a curve between the origin and one of the targets, such that the minimum number of t-vertices is crossed:

C= set of cells = G^*_{CONTR}

$S=\{s_c,s_t\}$, where $s_c\in R_+$, $s_t\in R_+$ and $s_c>>s_t$

$(c,s)\in\sigma\leftrightarrow$(c is c-vertex $\wedge$ $s=s_c$)$\vee$(c is t-vertex $\wedge$ $s=s_t$)

$(c_i,c_j)\in\eta\leftrightarrow\exists_{w\in W_{CONTR}}[c_i\in C_w\wedge c_j\in C_w]$.

set of origins = $\{c\}$

set of targets = $C^I\backslash\{c\}$

$(a,s,b)\in\mu\leftrightarrow b=a+s$

Let the target that has been reached, be denoted c', and let the sequence of crossed t-vertices be $t_i,\ldots,t_j,\ldots,t_k$ $(i\leq j\leq k)$. Then for the identification of c and c', the following steps are performed:

1. Split each t-vertex t_j $(i\leq j\leq k)$ in two vertices t_j and t'_j. Also the set of edges incident to t_j is splitted by the curve in two sets:

$$t_j\rho_c = C_{j_1} \cup C_{j_2} \qquad \text{and} \qquad t_j\rho'_c = C_{j_1}$$
$$t'_j\rho'_c = C_{j_2}$$

2. Add for each splitted vertex t_j an auxiliary vertex t''_j and connect this vertex with t_j and t'_j (add the edges $[t''_j,t_j]$ and $[t''_j,t'_j]$).

3. Replace the curve by the sequence of edges
$$[c,t''_i],\ldots,[t''_{j-1},t''_j],\ldots,[t''_k,c']$$

4. Contract all the edges added in step 3.
In the contraction process, parallel edges are only replaced by one edge, in case there are no other vertices embedded in the inner region determined by the parallel edges.

In figure 6.2a the identification of c and c' in the plane representation of the contraction graph is schematically depicted.
If a number of t-vertices has to be split, the modified potential graph is extended with a crossor. This process is schematically shown in figure 6.2b. The t-vertex $\tilde{t}_j$ in (G^*_p,U^*_p) and t-vertex t_j in the contraction graph are associated with the same (part of a) potential tree. The vertices $\tilde{t}_j$, with $i\leq j\leq k$, are each split in two vertices, named $\tilde{t}_j$ and $\tilde{t}'_j$, that both represent the same potential: it becomes available on both sides of the island I to which the crossor is assigned. Furthermore, the sets C_j and $\tilde{C}_j$ in the two graphs represent the same set of circuit components.

Which of the t-vertices are crossed by the curve between two c-vertices that have to be identified, is dependent on which identifications have been performed before: the various curves may block each other and therefore the sequence of identifying the c-vertices has to be considered:

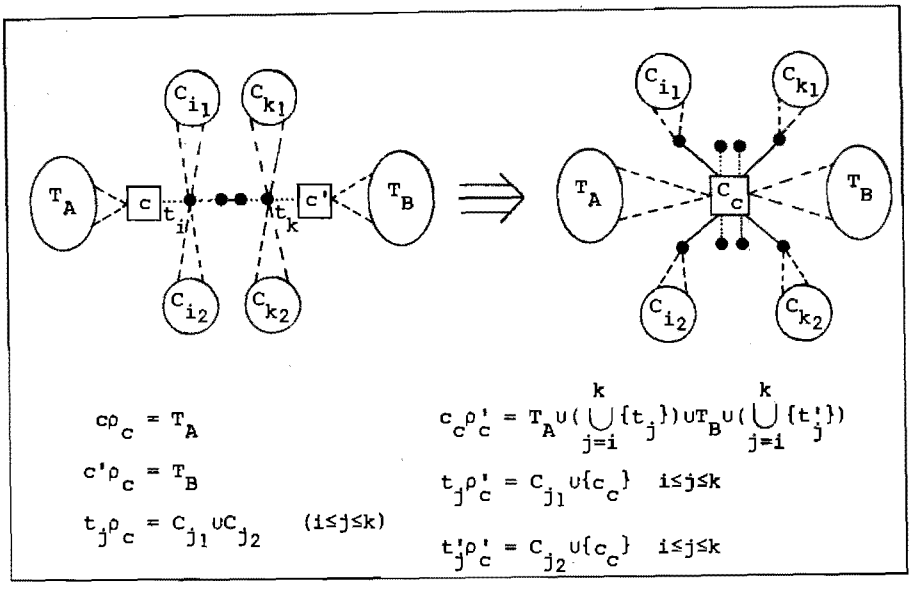


Fig. 6.2a The identification of the vertices c and c' in the contraction graph.

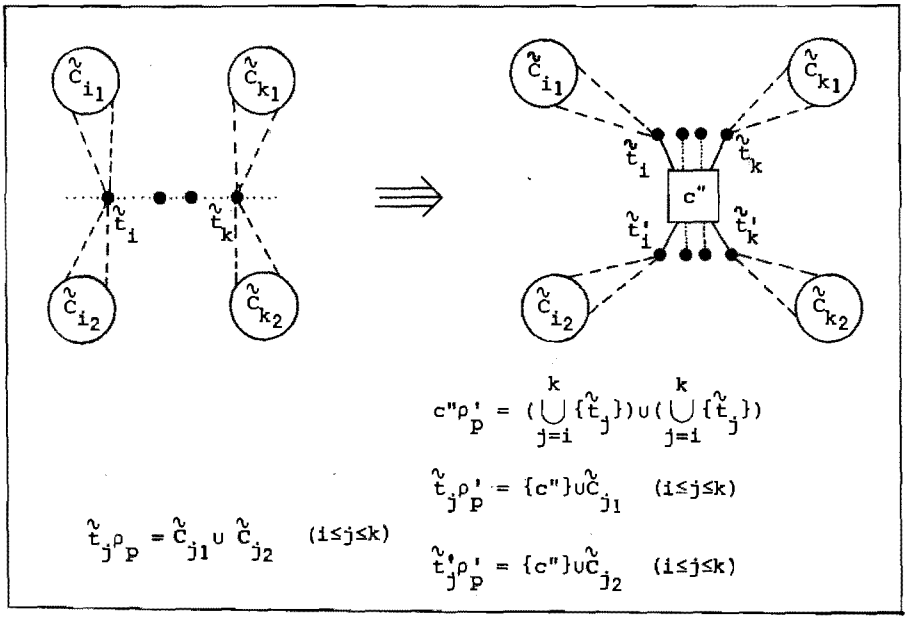


Fig. 6.2b Introduction of a c-vertex c", associated with a crossor, in (G^*_p, U^*_p).

- We give first priority to the identification of two c-vertices that both represent components of the same island, and that are both incident to the same t-vertex. This has been done because it is disadvantageous if some curve determined by the path optimization algorithm, crosses the t-vertex in question between the two c-vertices: the two components in question will be placed in the same island (close to each other), but their interconnecting potential lead has to make a detour and has to cross at least one other island.
- Furthermore the probable area of the crossors is considered. An increase of the number of potentials that has to cross a certain island I, is generally most disadvantageous if the areas A_1^I and A_2^I on both sides of the crossing area are large (see fig. 6.1). Therefore we favour the identification of the c-vertex c having maximal value

$$A^c_{min} = \text{minimum } (A_c, A_{C^I\setminus\{c\}})$$

where

A_c = the total area of the set of components that is represented by c

I = the island assigned to the components that are represented by c

C^I = the set of c-vertices that represent a component of the island I

$A_{C^I\setminus\{c\}}$ = the total area of the components that are represented by the vertices in $C^I\setminus\{c\}$.

Each crossor that is introduced by identifying some vertex $c \in C^I$ with a vertex c' of $C^I\setminus\{c\}$, is assigned to the island I in question, and the area that will be reserved for it is taken proportional to the number of split t-vertices and to the square root of A^c_{min}.

6.4. Subsets of island components that are embedded in rectangular regions.

It already has been mentioned that it is advantageous to create islands of which the geometrical form is (nearly) a square. We are therefore inclined to restrict the layout so as to have isolated regions of rec-

tangular form.However, if an island has to be crossed by some potential lead(s), it can sometimes be advantageous to deviate from the rectangular shape, and admit the geometry of such an island to be built up of a number of rectangular regions that together form a connected area. The length of the crossing potential lead(s), and thus the area that is occupied by them, may become much smaller, if the island can be crossed over a "bulge" (see fig. 6.3).

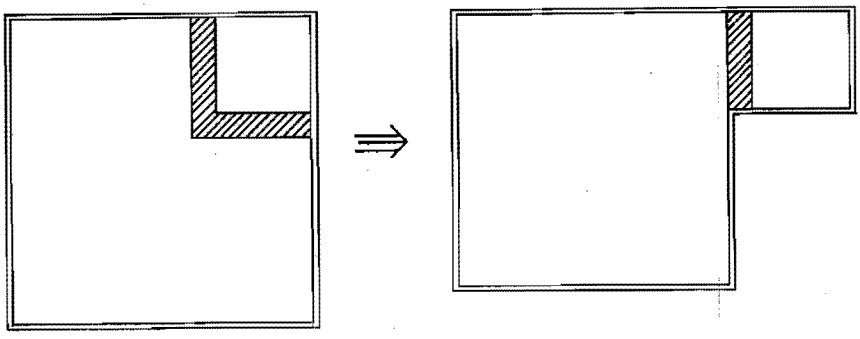


Fig. 6.3. Reducing the "crossing area" by deviating from the rectangular shape of the island.

If we allow an island to consist of a number of rectangular regions, these regions will be considered separately in the procedure that actually determines their geometry (see chapter 7), and measures have to be taken to guarantee that each rectangle of such an island has some common boundary with some other rectangle that is associated with this island (otherwise the epitaxial regions would have to be interconnected, which may cost extra contact area). We have to determine which components will be embedded in the regions, so that it is known what is the minimum area that the region should have.
If an island is divided in many rectangular regions, it may become difficult to satisfy the common-boundary restrictions between them in the procedure that determines their final geometry. Furthermore, there is the disadvantage that each "bulge" of an island increases the area that is occupied by the isolation channel surrounding the island. It is therefore decided to allow an island to consist of at most two

rectangular regions, and the partition has to be "caused" by a crossor that is associated with at least two different potentials.

If an island I contains more than one crossor of which the number of associated crossing potential leads $l_c \geq 2$, then the partition of the set of c-vertices representing the island components, in the two sets C_1^I and C_2^I, will be determined by the crossor having maximum value l_c.

Let the c-vertex c representing the crossor be associated (in clockwise direction in (G_p^*, U_p^*)) with the sequence of t-vertices:

$t_i, \ldots, t_j, \ldots, t_k, t_k', \ldots, t_j', \ldots, t_i'$ $(i \leq j \leq k)$ where t_j and t_j' $(i \leq j \leq k)$ are associated with the same potential.

Let c be situated on the face boundaries of the subset of faces $W^c \subset W_p$ in a sequence (clockwise direction):

$$w_1^c, \ldots, w_x^c, \ldots, w_n^c, w_{n+1}^c, \ldots, w_y^c, \ldots, w_{2n}^c$$

where n=k-i+1, and w_1^c is the face that has both t_i and t_i' on its face boundary.

The set of faces W^c is partitioned in two sets by taking

$$x = \begin{cases} \frac{n}{2} & \text{if n is even} \\ \frac{n+1}{2} & \text{if n is odd} \end{cases}$$

$$y = x+n-1$$

$$W_1^c = \{w_z^c \mid w_z^c \in W^c \wedge x \leq z \leq y\}$$

$$W^c = W_1^c \cup W_2^c$$

$$W_1^c \cap W_2^c = \emptyset$$

The sets C_1^I and C_2^I of c-vertices that are associated with the rectangular regions (where $C_1^I \cup C_2^I = C^I$), are determined by applying the path optimization algorithm of appendix C:

set of cells= $C = C^I \setminus \{c\}$

$S = \{1\}$

$$(c_i, c_j) \in \eta \leftrightarrow (\exists_{t \in G_{pT}^*} [[c_i, t] \in U_p^* \wedge [c_j, t] \in U_p^*]) \vee (\exists_{w \in W_p} [c_i \in C_w \wedge c_j \in C_w])$$

set of origins= $C_o = \{c_i \mid c_i \in C \wedge (\exists_{w \in W_1^c} [c_i \in C_w])\}$

set of targets= $C_t = \emptyset$

$$(a, s, b) \in \mu \leftrightarrow b = a+s$$

If a cell c_i has got a cell mass $c_i v \neq \infty$, it is included in C_1^I.
$C_2^I = C^I \setminus (C_1^I \cup \{c\})$.
The total area of the components in C_1^I (C_2^I) is denoted A_1^I (A_2^I).
If $A_1^I < A_2^I$, the crossor is included in C_1^I, else in C_2^I.

6.5. Example.

Let us continue the design of the operational amplifier μA725 by performing the various steps that distribute the components over the islands, and that construct the island interconnection graph. Using a dc-analysis program we first should calculate the operating states of the circuit in figure 6.4. Only two extreme cases (V=+1 volt and V=-1 volt) have been considered, since we did not dispose of a separate analysis

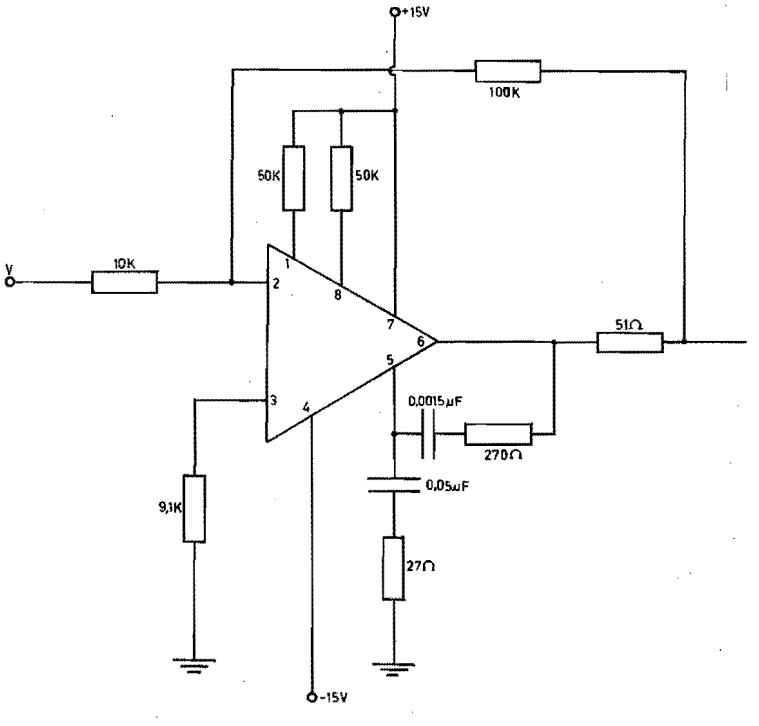


Fig. 6.4. Circuit for dc-analysis.

procedure. In table 6.2 the (N)EP-components are specified that are contained in the various IR-compatibility classes. From this table it

appeared necessary to create the 12 isolated regions IR_i ($1 \leq i \leq 12$). The distribution of the components over these islands is obtained by performing steps -a- up to -e- of section 6.2. The result is visualized in figure 6.5, where for each component the subscript of the isolated region assigned to it is indicated inside the representing c-vertex. In step -a- of the assignment process all the EP-components are assigned

compatibility class	EP-components	NEP-components	epitaxial contact of the EP-component	lowest potential of the epitaxial contact (volt)
1	54	-	-	-
2	55	40,41,42,44,47, 48,49,52,53	9	-0,7
3	56	40,41,42,44,47, 48,49,52,53	11	-14,2
4	57,62,65,67	40,41,42,44,47, 48,49,51,52,53	14	14,5
5	58,61,71,73	35 up to 53	7	15,0
6	59,68	40,41,42,44,47, 48,49,51,52,53	22	13,5
7	60,69	40,41,42,44,47, 48,49,51,52,53	24	13,5
8	63	40,41,42,44,47, 48,49,52,53	23	-14,3
9	64	40,41,42,44,47, 48,49,52,53	5	-13,7
10	66,76	40,41,42,44,47, 48,49,52,53	27	-14,3
11	70,74,75	40,41,42,44,47, 48,49,51,52,53	29	-9,3
12	72,77,78	40,41,42,44,47, 48,49,52,53	32	-10,6

Table 6.2. The IR-compatibility classes of the µA725.

(in fig. 6.5 without accent), together with the NEP-components 35,36,37,38,39,43,45,46,50 (indicated with one accent). After the construction of the various island vertex groups and the group-distances (step -b-), the following assignments are due to the distance-improvement attempts in step -c- (visualized in figure 6.5 with two accents):

52,53 to IR_7

42,44 to IR_6

48,49,51 to IR_{11}.

In step d the rest of the NEP-components is assigned (three accents in fig. 6.5):

40,41,47 to IR_6 .

For the construction of the island interconnection graph we first identify all pairs of directly interconnected c-vertices, representing components of the same island. The result of this step is given in figure 6.6. Crossors have been introduced in island 5 (where t-vertices 7 and 29 are split) and in island 6 (where t-vertex 4 is split in three vertices). Next the other pairs of c-vertices that are associated with the same island are identified in a sequence of decreasing A^c_{min}. During this process the following crossers are introduced:

island:	split potentials:
IR_5	11
IR_{11}	5
IR_7	23,27,4
IR_{12}	5',30
IR_{11}	6
IR_6	21,5",33'
IR_4	7",11',22,7"'

The resulting island interconnection graph is depicted in figure 6.7. The plane representation of the modified potential graph with the introduced crossors is given in figure 6.8.

Reference.

[6.1] D.Ferrari, "On the selection of isolated regions in computer-aided design of integrated circuits", Transactions on Circuit Theory, CT-17, pp. 134-136, 1970.

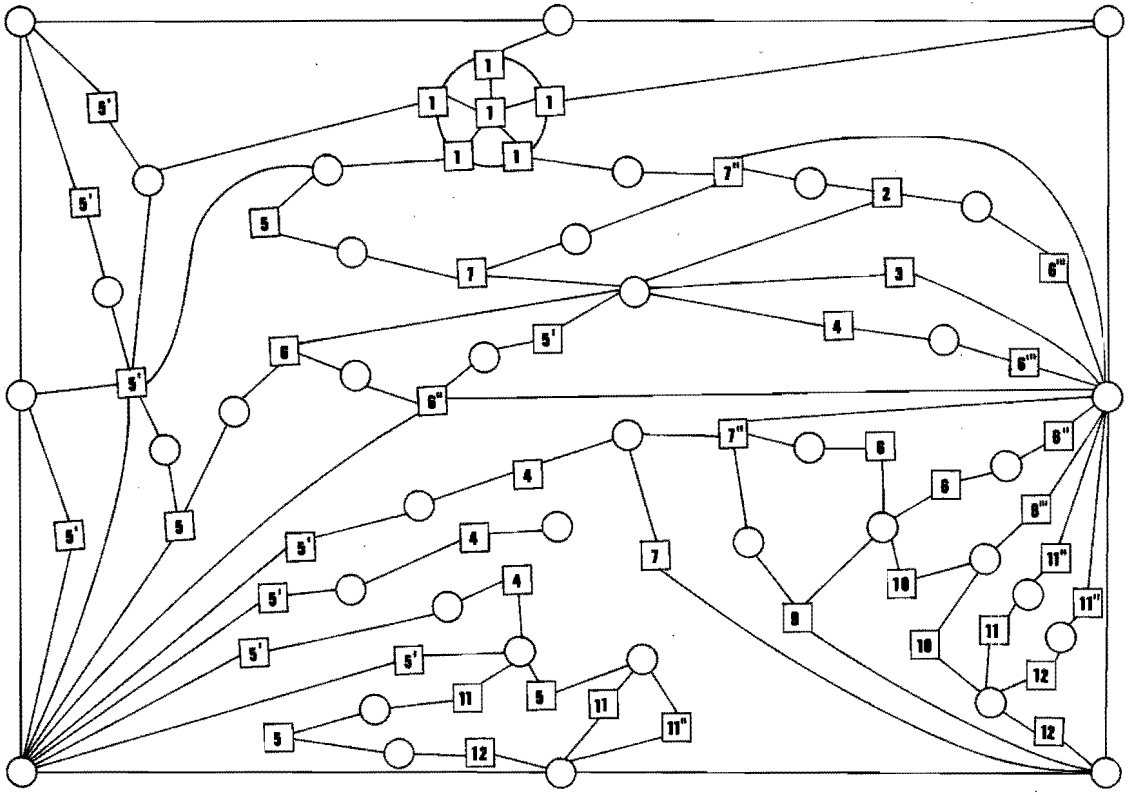


Fig. 6.5. The distribution of the components over the islands (μA725).
(The more accents the island index has, the later the component in question is assigned to this island.)

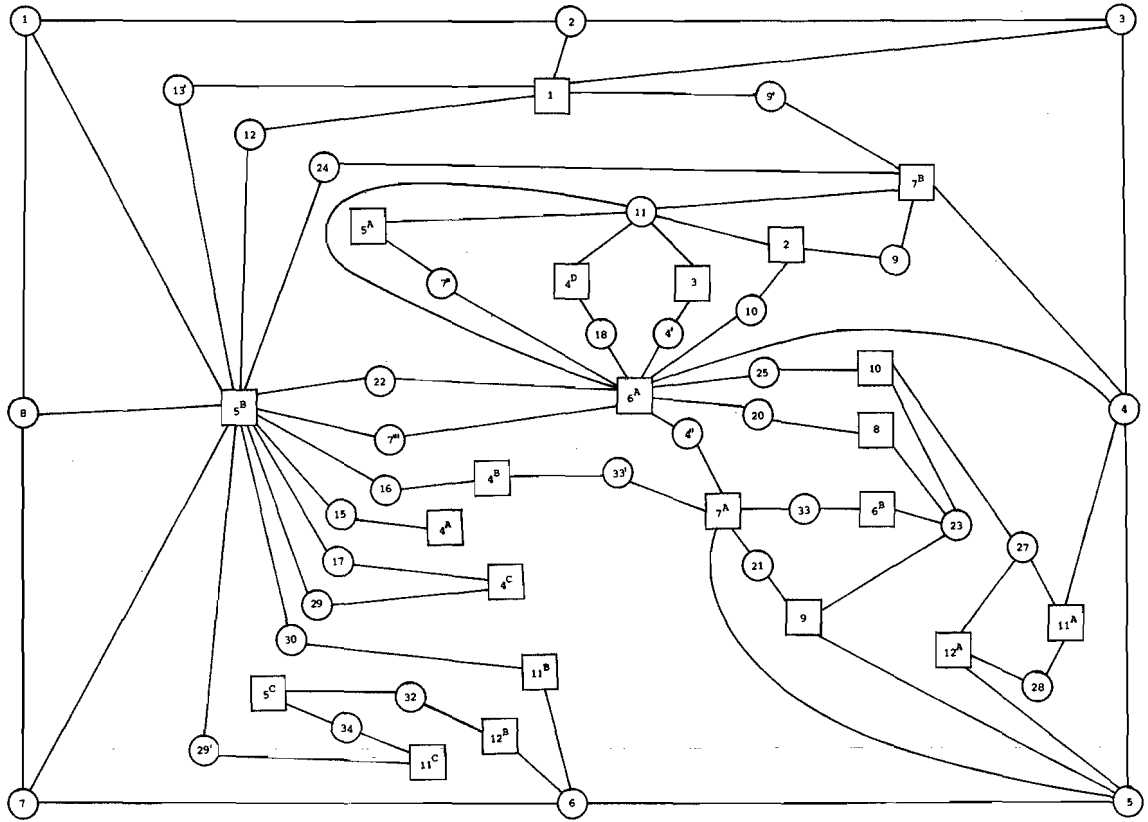


Fig. 6.6. The contraction graph after identifying all pairs of directly interconnected c-vertices, representing components of the same island (μA725).

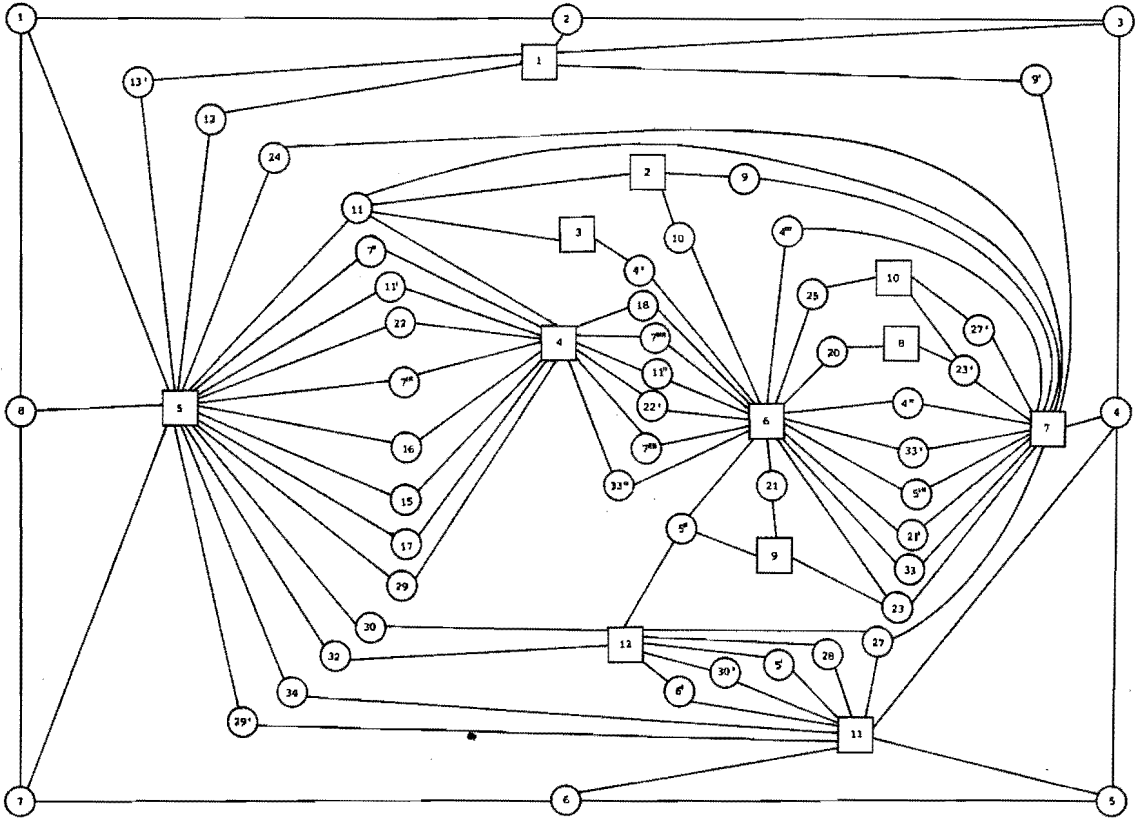


Fig. 6.7. The obtained island interconnection graph (μA725).

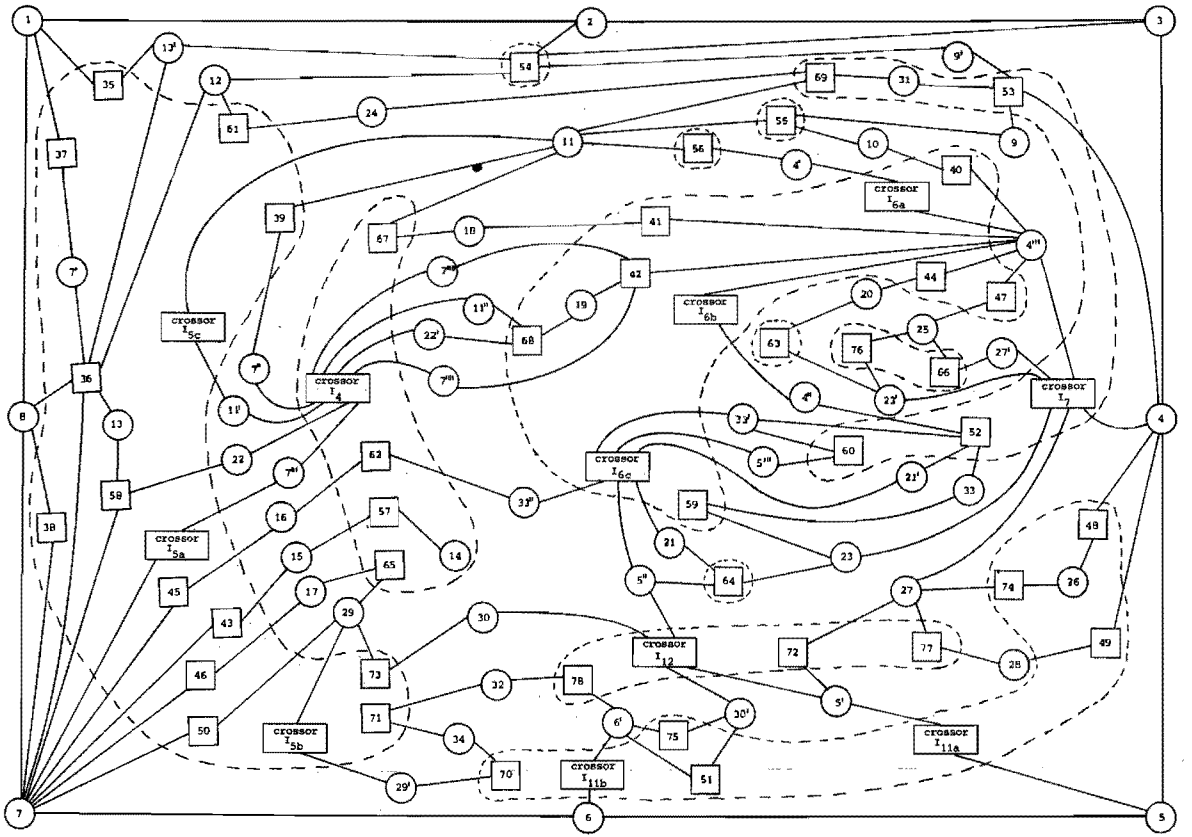


Fig. 6.8. The modified potential graph containing the various crossors. The c-vertices in the interior of a dashed curve represent components of the same island.

7. THE DEEP P-DIFFUSION

7.1. The generation of the neighbour relations.

As pointed out in chapter 3 there is a correspondence between a rectangle partitioned into rectangles and a drain representation. The number of arcs in the drain is equal to the number of rectangles. This number is already fixed,when we enter the procedures that must determine the geometry of the isolated regions. We now describe a method to construct a drain representation with this number of arcs.

Let (G^*,V^*) be a drain representation of a drain (G,V). There is exactly one arc v for which $v \in \Gamma^{-1}$ and $v \in t^d(\Gamma^d)^{-1}$. We call this arc the last arc of (G,V). (Clearly, which arc is the last arc depends on the representation we started from!). We start our construction with a vertex. At each stage in the construction we add an arc not yet contained in the drain constructed so far, in such a way that this arc is the last arc of the new drain. We call these extensions, admissible extensions. The construction stops when the drain contains the required number of arcs. Every drain representation (G^*,V^*) can be constructed by admissible extensions.

> The truth of this assertion can be recognized by reducing its negation to absurdity. Suppose there is a drain representation (G^*,V^*) that cannot be constructed by admissible extensions. Remove its last arc in an appropriate way, i.e. either by a drain-preserving contraction or by a drain-preserving deletion, and repeat this operation as often as possible. This results in a succession of drains ending with a digraph consisting of one vertex. The retrograde process, however, gives exactly a construction of the drain (G,V) by admissible extensions. It must be remarked, that one must choose the obvious representations of the drains in the sequence.

We further need a criterion to decide whether a drain representation is acceptable or not. If possible, this criterion must be such that it can be seen early in the construction, whether the final drain repre-

sentations can be acceptable. Before we make a first attempt to formulate tractable requirements for the drain representations to be constructed, we define some additional entities.

Since it is known at this stage of the layout procedure which components belong to which rectangle, we can make a first estimate of how much area the concerned rectangle needs, for example the sum of the areas of its components (E1). Further, there was some advantage in making endeavour to approach a square form for the isolated regions (E4). Since many of the rectangles to be found contain all components of an isolated region, we try to fit these rectangles together as squares. The chip should not become oblong (E8), which means that we must keep an eye on the dimensions of the chip during the construction of the drain representation.

We therefore define the length of an arc as the square root of a minimum area its associated rectangle is supposed to get. The length of a chain is the sum of the lengths of the arcs contained in that chain. We also define a pseudo-distance between arcs. The pseudo-distance between two distinct arcs, v_1 and v_2, in a drain (G,V) with a drain representation is defined as the length of the longest chain in the set

$$\{(C,C[x_1,y_2>) \mid (C,C[x_1,y_2>) \lhd (G,V) \wedge v_1=[x_1,y_1> \in C[x_1,y_2> \wedge$$

$$\wedge v_2=[x_2,y_2> \in C[x_1,y_2>\} \cup \{(C,C[x_1,y_2>) \mid (C,C[x_1,y_2>) \lhd (G^d,V^d) \wedge$$

$$\wedge v_1^*\tau=[x_1^*,y_1^*> \in C^*[x_1^*,y_2^*> \wedge v_2^*\tau=[x_2^*,y_2^*> \in C^*[x_1^*,y_2^*>\}$$

decreased with the lengths of v_1 and v_2. Notation: $(v_1,v_2)\delta$.

The question is whether this definition always produces a unique number for this pseudo-distance, i.e. whether for every pair of distinct arcs, v_1 and v_2, there exists a chain in the drain containing v_1 and v_2 or there exists a chain in the dual containing $v_1^*\tau$ and $v_2^*\tau$. That this is really the case can be seen from the following argument:

Suppose the statement is true for every drain representation with less than k arcs. Let (G^*,V^*) be a drain representation with k arcs and $[x^*,t^*>$ its last arc. After removing this arc by a drain-preserv-

ing contraction or deletion, we have a drain satisfying the hypothesis. So we only have to prove that $[x^*,t^*>$ satisfies the statement with any other arc. We distinguish two cases

a. $x^*\Gamma_{V^*} \neq \{t^*\}$

Let a^* be either s^* or the last articulation point on a path from s^* to t^*. (G_o^*,V_o^*) is the drain consisting of all chains from s^* to a^*. The subdigraph (G_1^*,V_1^*) formed by all chains from a^* to x^* is enclosed by two chains from a^* to x^*. By theorem D.8 (G_1^*,V_1^*) is a drain and by theorem D.3 we know that $[x^*,t^*>$ is in a chain of $(G_o^* \cup G_1^*, V_1^* \cup V_o^*)$ with any arc in $V_1^* \cup V_o^*$. The arcs in $V^* \setminus (V_1^* \cup V_o^*)$ having a vertex in G_1^* are all going out from this vertex (otherwise (G_1^*,V_1^*) does not contain all chains from a^* to x^*). They correspond with arcs in (G^d,V^d) which form together a chain from s^d to the first vertex of $[x^*,t^*>\tau$, just as the arcs corresponding with the arcs in $\{v^* | v^* = [y^*,t^*>\} \setminus {}^o\{[x^*,t^*>\}$. All the other arcs of $(V^* \setminus V_o^* \cup V_1^*)\tau$ except $[x^*,y^*>\tau$ are in the interior of the Jordan curve formed by these two chains. Again by the theorems D.8 and D.3 we conclude that any arc in $(V^* \setminus (V_1^* \cup V_o^* \cup \{[x^*,t^*>\}))\tau$ is in a chain of (G^d,V^d) with $[x^*,t^*>\tau$

b. $x^*\Gamma_{V^*} = \{t^*\}$

This is exactly the same case as the above if we interchange the roles of (G^*,V^*) and its dual (theorem D.7).

In the drain representation (G^*,V^*) of a rectangle partition we can see which rectangles abut on the same horizontal or vertical line segments. The distance between their corresponding arcs is equal to zero, though the rectangles are not necessarily adjoining. Neverteless we will use this pseudo distance together with a number, measuring the predilection for having the corresponding rectangles adjoining, to calculate a kind of "quality" of the constructed drain. We call this other number the attraction between v_1 and v_2. Notation: $(v_1,v_2)\alpha$. This number may follow from several of the desires listed in section 3.3 (E2,E3,E4,E5). Further, the whole rectangle can be restrained from deviating too much from the square form by keeping the longest chain in the drain as well as in its dual within a bound proportional to the square root of the sum of the minimum areas of the rectangles (E8).

Let us now formalize the ideas of the preceding paragraphs in order to come to a first method to generate a drain representation. Let ω be a mapping that assigns to every arc a length which is equal to the square root of the estimated value for the area its associated rectangle needs. We call a drain representation acceptable, if the length of its longest chain and the length of the longest chain in its dual do not exceed $L.\sqrt{\sum_{v\in V} v\omega}$, where L is a real number greater than 1 and if $\sum_{(v_1,v_2)\in V\times V\setminus\iota} (v_1,v_2)\alpha.(v_1,v_2)\delta<2F$ where F is a positive real number.

It is important to notice that if a certain drain representation (G^*,V^*) is acceptable, any of the drain representations occurring in the construction of (G^*,V^*) by admissible extensions are also acceptable. In other words the length of the longest chain, neither in the original nor in the dual drain, and the value of the "quality function" are never decreased by admissible extensions. This observation evokes the idea of backtracking [7.1].

The construction of the drain representation can be performed by a program that chiefly consists of a recursive procedure. In this procedure a thus-far constructed drain is extended in an admissible way. Initially this drain consists of one vertex and no arc. The arcs that finally must form the set V are initially all "free". As soon as an arc is added to the drain it is made "non-free". A "free" arc is tried in every admissible way as a possible extension. After each attempt the newly formed digraph is tested upon its acceptability. If it is acceptable we may have reached a solution, namely when there is no "free" arc left. If there are still "free" arcs, we take this new drain as the starting point for another application of this procedure. If, however, the obtained drain is not acceptable, another admissible extension must be tried, if possible with the same arc; else this last arc is made "free" again and the next "free" arc is tried.

A more precise description in pseudo-algol is:

```
'begin'
'procedure' EXTEND;
'begin'
'repeat' take the next free arc;
        'repeat' add this arc by the next admissible extension;
                'if' acceptable 'then'
                'begin' 'if' no free arcs left
                        'then' print the constructed drain
                        'else' EXTEND
                'end'
        'until' admissible extensions exhausted;
        make the arc free;
'until' all free arcs tried
'end' of extend;
initialize the drain ({x},∅);
EXTEND
'end' of construction;
```

A possible visualization of the result of such a construction is given in fig. 7.1..

7.2. The simplex-tableau.

Having obtained an acceptable drain representation, i.e. the information about which rectangles abut on the same line segments, we still have to determine the dimensions of these rectangles. We have already seen how we can derive the Kirchhoff relations from the drain, and we know how much area a rectangle at least must get (depending on the components it is going to contain and the crossings it must allow). But we have other desires to take into account: constraints on the lengths and widths of the rectangles, also of the outer rectangle, and we want the total area as small as possible.

Let (G^*,V^*) be the constructed drain representation and W its set of faces. Further we write x_i for the length and y_i for the width of the rectangle corresponding with the arc $v_i^* \epsilon V^*$. x_o and y_o are the length

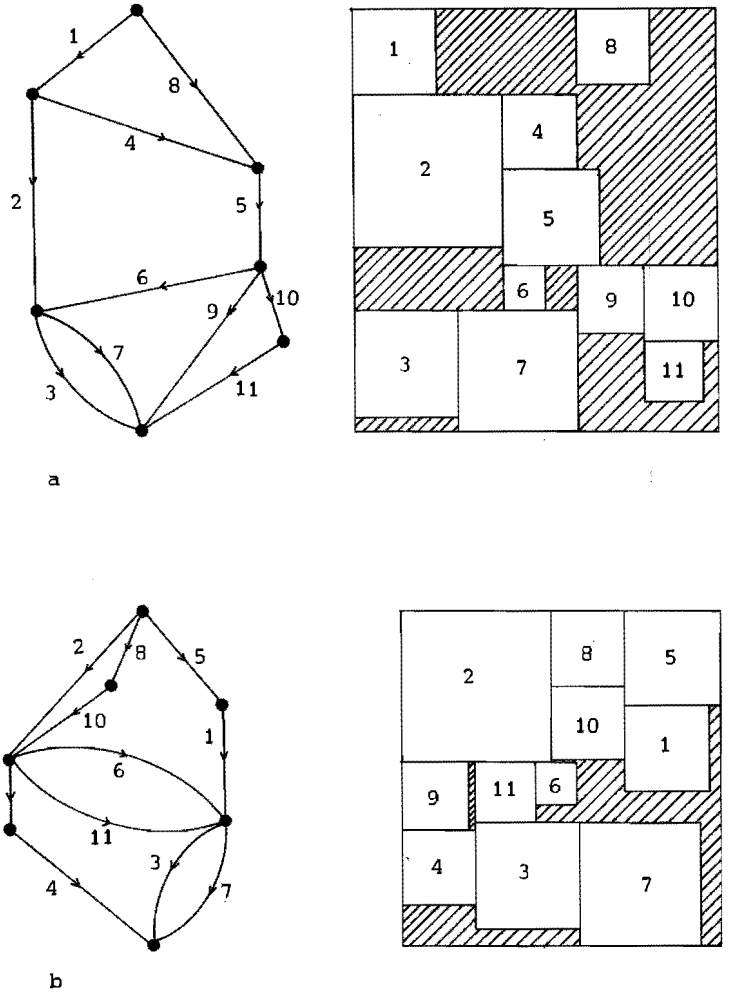
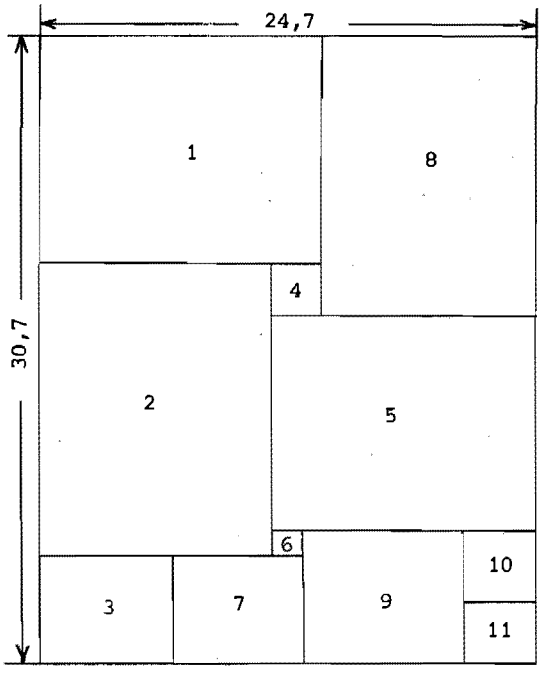
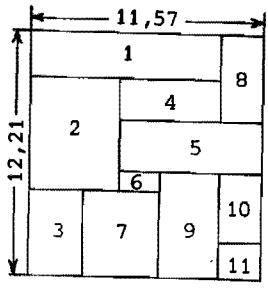


Fig. 7.1. Two possible drains for the same set of rectangles. The squares in the rectangles illustrating the results have the minimal area and are shifted as much as possible upwards and to the left hand side. The longest chains are 13.85 and 11.





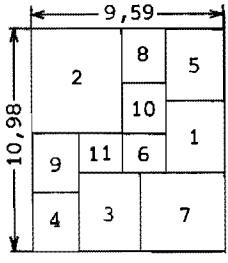


Fig. 7.2. Two results of linear optimization, starting from the drain of fig. 7.1a, both with constraints on the length-width ratio of the rectangles, namely 0.8 in case a and 0.2 in case b. The total areas are 758 and 141. The other drain (fig. 7.1b) leads to the partition of c under a ratio constraint of 0.8. The total area is 105. The minimal areas sum up to 100.

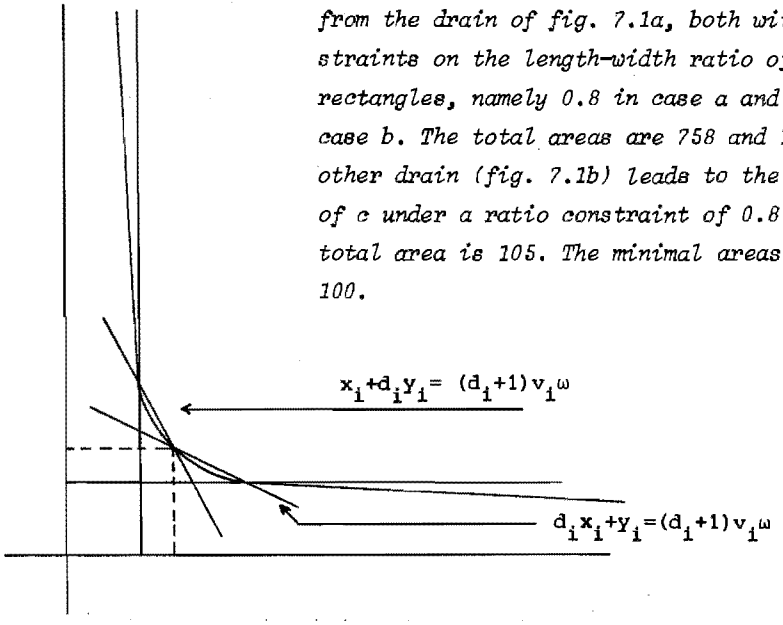


Fig. 7.3. Linearization of the area requirements.

and width respectively of the outer rectangle. s^* is the source of the drain and t^* is its sink; w_o is the outer face.
$V_g^{*+} = \{v^* | v^* = [g^*, h^*> \epsilon V^*\}^o$ and $V_g^{*-} = \{v^* | v^* = [h^*, g^*> \epsilon V^*\}^o$.

$$\forall_{w \epsilon W \setminus \{w_o\}} \left[\sum_{v_i^* \epsilon C_w^+} x_i = \sum_{v_i^* \epsilon C_w^-} x_i \right] \quad , \quad \sum_{v_i^* \epsilon C_{w_o}^+} x_i = x_o$$

$$\forall_{g^* \epsilon G^* \setminus \{s^*, t^*\}} \left[\sum_{v_i^* \epsilon V_g^{*+}} y_i = \sum_{v_i^* \epsilon V_g^{*-}} y_i \right] \quad , \quad \sum_{v_i^* \epsilon V_s^{*+}} y_i = y_o$$

The requirement to keep the length-width ratio of the chip within certain bounds (E8) can be easily implemented by two linear inequalities:

$d_o x_o - y_o \leq 0$

$-x_o + d_o y_o \leq 0$

where d_o is the minimal permissible ratio between the width and the length of the outer rectangle.

The length and the width of the inner rectangles (E6) can be controlled in the same way as the dimensions of the chip, but this can lead to a set of conflicting constraints. And even if the equations are consistent, some discord between the requirements and the drain representation results in too large a rectangle (Fig. 7.2). Since we must obtain a solution in order to keep our procedure fully automatic, we choose for another constraint on the length and width, namely a minimal value:

$x_i \geq d_i$

$y_i \geq d_i$

Up to now all equations and inequalities are linear, but we still have to formulate our area requirements (E1,E7), which are by nature non-linear. However, we expect the rectangles to be almost square, because of the restrictions on the longest chain during the construction of the drain representation. By choosing the values of d_i not too different from the square root of the minimum area the rectangle corresponding with $v_i^* \epsilon V^*$ must get, we can strengthen this effect. Fig. 7.3 shows that linearizing these area requirements cannot have a severe detrimental

influence on the result. This gives two additional inequalities for each rectangle.

$d_i x_i + y_i \geq (d_i+1) v_i \omega$

$x_i + d_i y_i \geq (d_i+1) v_i \omega$

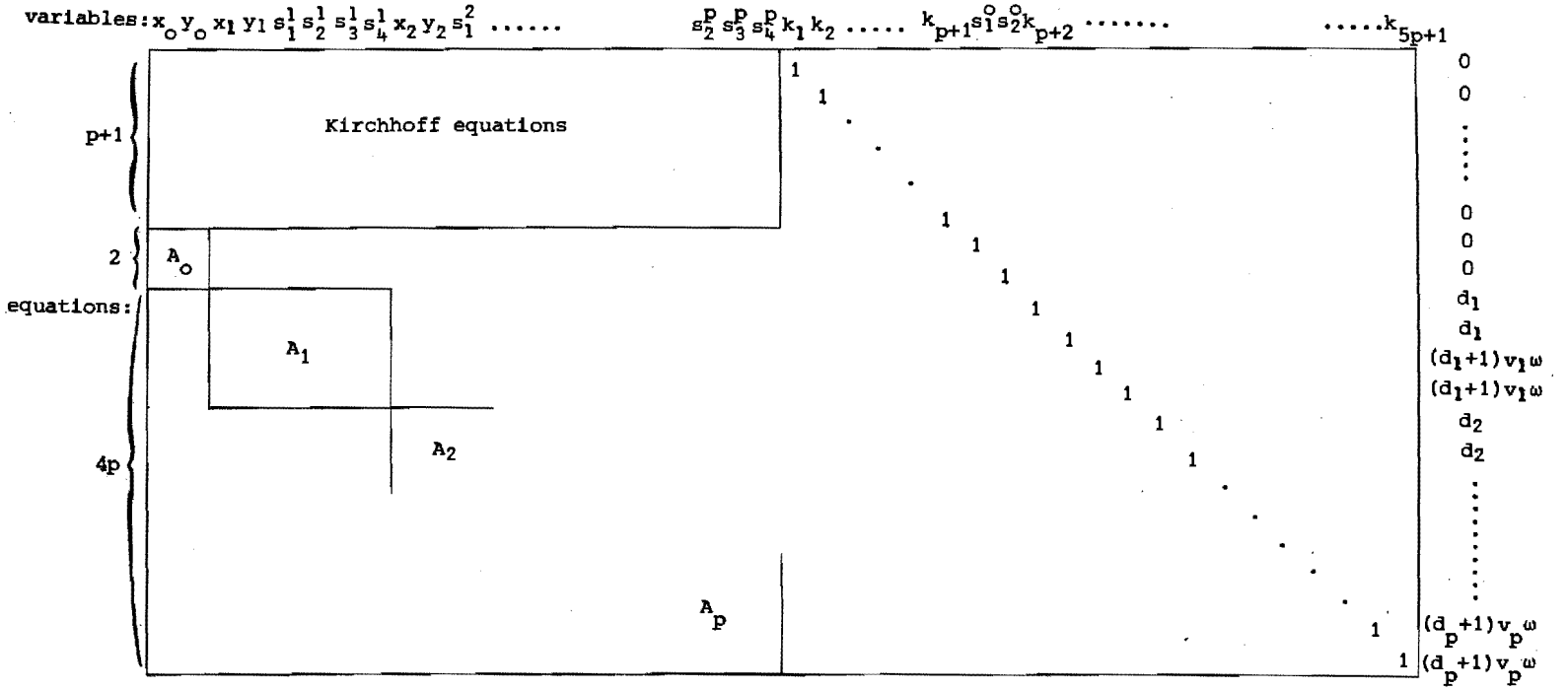
If we now want to minimize the total area (E7), $x_o y_o$ is the appropriate object function. However, this function is not linear. The perimeter of the rectangle is linear, but minimizing x_o+y_o is not the same as minimizing the area, except when the rectangle turns out to be a square. We already included two inequalities in order to keep the length-width ratio close to 1, because we did not want the chip to become too oblong. We can use this again, to recognize the perimeter as a satisfactory substitute for the obvious object function.

The formulational frame work, described in this section and summarized in fig. 7.4, is easily recognized as a problem where a linear criterion function must be minimized in a convex polyhedron, and consequently the methods of linear programming apply to this problem. How to transform the restrictions and object function into a "canonical minimum problem" and thus into a tableau suitable for the simplex algorithm, is described in appendix E. It remains to show that there always is a solution, but this is quite simple.

Starting from squares like those in fig. 7.1 we can obtain rectangles covering the whole outer rectangle, namely by extending them to the right hand side and/or downwards until they reach a rectangle with which they may have a line segment in common. If the length-width ratio of the outer rectangle is not according to the requirements, we enlarge the whole figure in the proper direction. This gives a solution satisfying all constraints, thus a solution with minimal perimeter must also exist.

7.3. Considerations for the implementation.

It goes without saying that the number of possible neighbour relations is very great, and it is not practical to generate them all. The number of acceptable drain representations is controlled by the values of L



object functions: phase 1: $\sum_{\substack{i=1 \\ i\neq p+2,p+3}}^{5p+3}$ equations; phase 2: x_o+y_o

$$A_o=\begin{pmatrix} d_o & -1 \\ -1 & d_o \end{pmatrix}$$

$$A_i=\begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ d_i & 1 & 0 & 0 & -1 & 0 \\ d_i & d_i & 0 & 0 & 0 & -1 \end{pmatrix}$$

Fig. 7.4. The simplex tableau.

and F. The choice of these values is very important, for if we choose a threshold too low, there may be no acceptable drain representation at all, while taking a very weak threshold results in tremendously many cases to consider and a solution possibly far from being optimal. Since backtracking has the salient feature of being exhaustive, we can find the optimal drain representation if properly defined. For example the "longest-chain requirement" remains of the threshold type and we search for the solution with the lowest value of the "quality function", of course with a dynamic value for its threshold, namely the lowest value obtained up to the actual point. This still leads to excessive processing times, since L must still be chosen such that the existence of a solution is certain. Making both values, L and F, variable gives the problem of deciding what the best combination of longest chain and "quality" is. A simpler solution is to run the program first with rather low, fixed thresholds and to take the first acceptable drain representation that is found. If some time has elapsed without finding an acceptable drain representation, L and F are increased and become variable. The best solution found in a fixed period is taken.

Finding a good solution can also be speeded up by including a bias in the sequence in which the arcs are offered for extension. Since the largest rectangle will almost always be placed in a corner, we choose this rectangle as the first one in the sequence. The next rectangle is determined by calculating the sum of all attractions between a non-placed rectangle and the rectangles already represented in the drain. The one with the highest result of such an addition is chosen.

Another improvement can be attained by keeping control over the area that cannot be used any more, when we maintain the drain so far constructed as a subdrain of the final drain. Acception of a partial solution can be made dependent on whether it is possible to place a non-placed rectangle in these interstices.

In section 7.1 we mentioned the disturbing fact that the pseudo-distance can be zero though the corresponding rectangles are not adjoining. To come to a more realistic value of the pseudo-distance we define the "tail depth" and the "point depth" of an arc v=[x,y>. The "tail depth"

$v\delta_t$ is the length of the longest chain from the source to the vertex x. The "point depth" $v\delta_p$ is the length of the longest chain from the source to the vertex y. The "horizontal distance" $(v_1,v_2)\delta_h$ of two arcs is defined as follows:

$$(v_1,v_2)\delta_h = \begin{cases} v_1\delta_t - v_2\delta_p & \text{if} \quad v_1\delta_t > v_2\delta_p \\ v_2\delta_t - v_1\delta_p & \text{if} \quad v_2\delta_t > v_1\delta_p \\ 0 & \text{otherwise} \end{cases}$$

The "vertical distance" $(v_1,v_2)\delta_v$ is defined in exactly the same way, but for the dual drain. A suitable number to replace the pseudo distance as defined in section 7.1 is the sum of the vertical and horizontal distance.

After the determination of the partition by the simplex procedure, we have to place the bonding pads and to cover the whole with a grid structure in order to be able to perform the wiring between the islands. The bonding pads are placed along the longer sides of the rectangle. Their sequence is mostly prescribed, so we have to split this sequence in two rows and to fix the orientation (clockwise or counter-clockwise). Preferences in these decisions are derived from the number of interconnections between the islands and the actual bonding pad.

Finally the dimensions of the rectangles are rounded off to a multiple of the grid constant and between them we leave a space ("isolation channels") with a width of one grid constant. Of course, we could have solved the problem as an integer linear programming problem [7.2], but this would consume a lot of computer time, much more than the method proposed here, which also delivers satisfactory solutions.

7.5. Example.

The data needed for determining the island pattern for the µA725 are given in table 7.1..The result of the drain construction is depicted in fig. 7.5..The main contribution to the frustration is due to the pseudo-distance between rectangle 7 and rectangle 6. The sum of the minimal areas is 2230. The longest chain is 66.95. After the linear optimization (fig. 7.6) the total area occupied by the islands and the deep p-diffusion in between is 2907 (2672+235).

v_i ↓	$(v_i w)^2$ ↓	d_i ↓	v_j →	$(v_i, v_j)\alpha$ 12	11	10	9	8	7	6	5	4	3	2
1	256	16							1		2			
2	32	5							2	1		1	1	
3	32	5							1	1		1		
4	144	9								6	8			
5	815	8		2	2				1					
6	309	11		1		1	2	1	5					
7	230	10		1	1	2	1	1						
8	32	5				1								
9	32	5		1										
10	48	5												
11	179	7		4										
12	121	7												

$d_o = 0.7$

Table 7.1. The data for constructing the drain and performing the optimization.

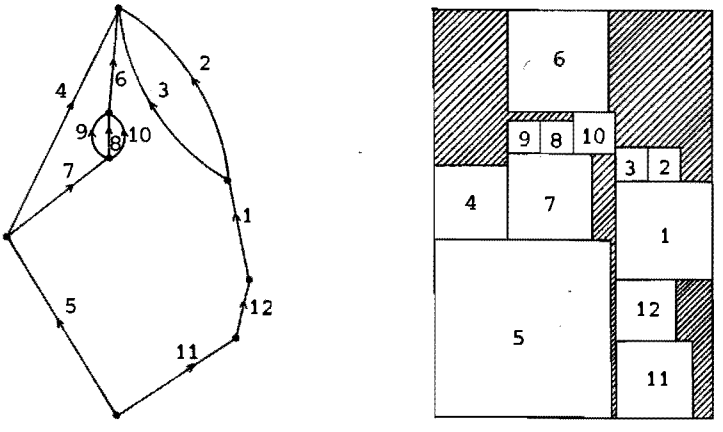


Fig. 7.5. The drain representation constructed from the data of table 7.1..

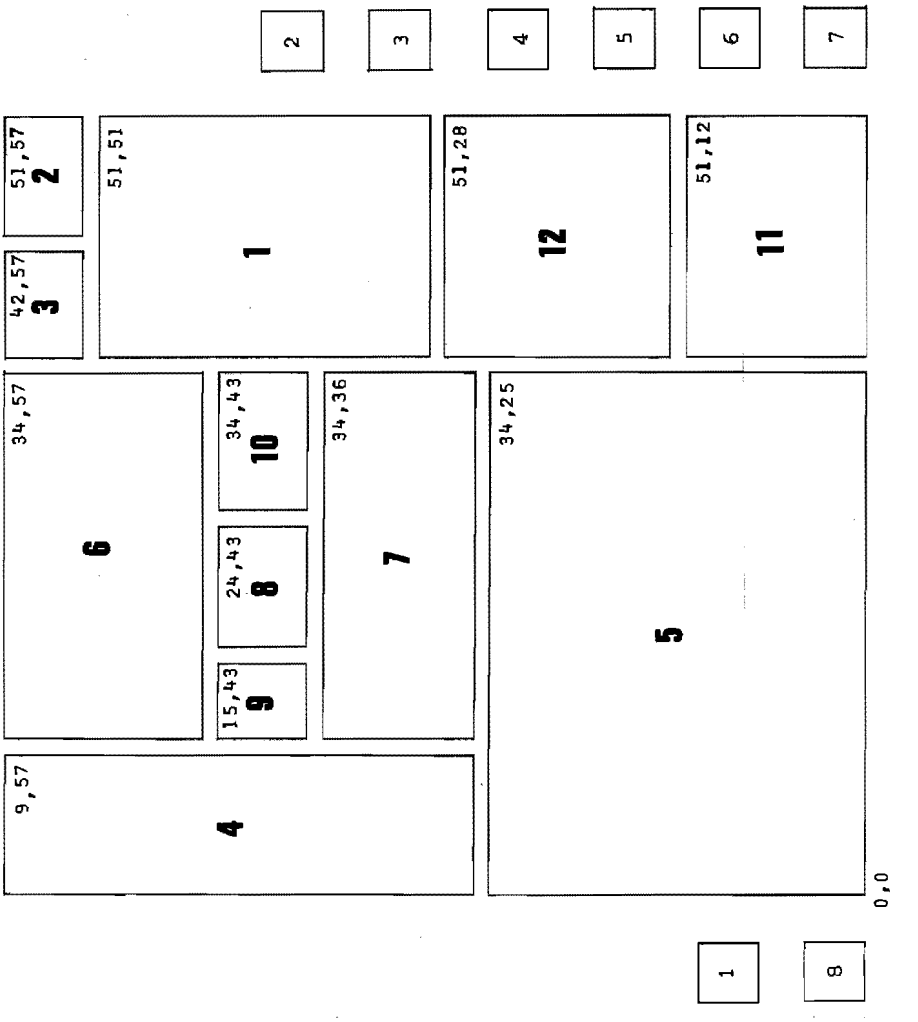


Fig. 7.6. The partition for the μA725 (bonding pads not automatically placed).

References.

[7.1] S.W.Golomb, L.D.Baumert, "Backtrack Programming", Journal of the ACM, Vol. 12, No. 4, pp. 516-524, October 1965.

[7.2] R.E.Gomory, "Essentials of an algorithm for integer solutions to linear programs", Bull. Amer. Math. Soc., Vol. 64, 1958.

8. THE COMPONENTS

8.1. Preliminary remark.

The scope of any layout design is seriously narrowed by the customary serial decomposition of the problem into a placement and wiring routine. The only way of avoiding this drawback is to introduce more interaction between the two by nature differing problems. In the extreme of this line of thought we have the principle of "wire as soon as possible", i.e. after placing a component,immediately make the necessary interconnections to the contacts of the components already placed, and proceed by locating the next component so that its interconnections can be laid out according to the current optimality criterion. Of paramount importance hereby is the order in which the components are to be placed. In the next chapter we will elaborate a sequence based on a rule which is the keystone of the procedures to be treated there (and of the planarization method of appendix F as well): keep all entities of which not all the relevant data concerning placement and interconnections are determined, reachable from one and the same point. It is clear that if we are capable of satisfying this rule throughout the placement-wiring procedure and there are no severe restrictions as to where a component must be placed, the algorithm will never get stuck. These ideas are in essence sound, but there are restriction on the placement. The components of an isolated region must be placed whenever prescribed by the sequence. It is difficult to take into account the positions of the components to be placed in the same island, and a rearrangement at the moment a component cannot be placed would be even more arduous. A better solution is to create more space by expanding the grid with one or more rows or columns, at the cost of a corresponding increase of area. How these expansions can be performed is described in the next chapter. For the same reason the final result will be a larger chip than necessary. This can be countered by applying a "compaction"-technique which is broadly the inverse of the expansion. [8.1]

However, between the two extremes of "wiring after complete placement"

and "wiring as soon as possible" we can search for a compromise in that parts of the placement alternate with parts of the routing. Considering the mentioned objections the following solution seems suitable: first design the wiring between the islands and bonding pads and then the placement and wiring in the islands by either method. We will now describe how to place components when knowing how the interconnections approach the island. The actual wiring procedure is described in the next chapter.

8.2. Placement of components.

The number of components in an island is relatively small, and for most islands this number does not exceed three. The greater islands have of course much more components, but many of them do not have fixed layouts.Because of these small numbers even branch-and-bound methods are applicable, and thus an optimum solution can be obtained. It is, however, difficult to find a criterion upon which one can decide that a certain placement is optimal, for a measure of convenience in wiring is not easily determined a priori. More important for islands having only components with fixed layouts is that the components must fit in the island. Of course, the area of such an island is sufficiently large, but the dimensions of the components cannot be changed, and thus it is still possible that a choice for a position will cause difficulties in placing other components. Here, complete enumeration helps (in placement as well as in the determination of minimal dimensions of the rectangles in the procedure of chapter 7). [8.2]

For the islands with components with flexible geometry, we turn to methods based on heuristic rationales, and constructive placement techniques in particular. This means that we have to determine an order in component selection and rules for fixing the position of the selected component.
Consider the part of the chip available for the positioning of the components in such an island. Initially this is the whole island minus the area occupied by interconnections between islands. Along the periphery of the available area we have the potentials of the interconnections that will enter or cross the island in a certain sequence S.

We determine the first component c_1 to be placed by elimination based on the following rules:

1. if present, it must be a component with a fixed layout;
2. among the remaining components we choose the ones with the greatest number of consecutive potentials in S (i.e. we prefer a component having the greatest number of interconnections to other islands that enter the actual island next to each other);
3. among the selected components the ones with the greatest number of interconnections to other islands.

If the selection is not unique, we take an arbitrary component that is not yet eliminated.

The interconnections between components in the island determine a set system over these components. Two components belong to the same block of this set system if they are directly interconnected (i.e. have contacts at the same potential). The finest partition with the property that any block of the mentioned set system is a subset of a block of this partition, has a block B_o containing c_1. B_o is called the candidate set; it consists of the components we will place before the components of other blocks are placed. Let *SS* be the smallest subsequence of S containing all the potentials of the interconnections the components in B_o have with other islands.

We first place components with a fixed layout. Each time the component with the potential in *SS* closest to the potentials in *SS* of already placed components, is selected. The position is chosen close to the potentials that caused its selection and preferably adjacent to the periphery. However, the placement routine must always take the following requirements into account:

1. two sides of the component must be adjacent to non-available area;
2. there must be at least one grid constant spacing between components;
3. components adjacent to the periphery of the part of the island that is available for components must be placed such that there is space for
   a. the potentials of SS that are passed over (because they have no contact with a component with a fixed layout), and
   b. the interconnections that have to pass between the component

and the periphery in order to reach the spot to enter the islands.

When all the components with a fixed layout that have interconnections with other islands are placed, we select the other components with a fixed layout simply on decreasing number of interconnections with the components already placed. Finally the components with flexible layouts are placed, and their flexibility is used to minimize corners in the remaining available area.

After the placement of all the components of the candidate set, we have arrived in one of three situations.

1. All components of the island are placed, in which case we are ready.
2. A crossor has to separate the components that are placed, from the components that still have to be placed. In this case we place this crossor.
3. There is no crossor to follow immediately.

In the cases 2 and 3 we repeat the whole procedure for the remaining components of the island.

8.3. The shape of the resistor diffusions.

In the preceding section we have reserved certain regions for the realization of the resistors. However, the actual shapes of the resistor diffusions still have to be determined. The shape of a large resistor is very flexible, and that was the reason to give priority to the placement of the components having a fixed layout. The consequence of this is that the resistor region may be the union of a number of rectangular regions. The area of the region is amply large to embed the resistor diffusion in question, having such a length that the desired resistance value is obtained (the highest possible number of bends in the meander has been taken into account).

The boundary of the resistor region coincides with the auxiliary grid structure covering the chip. The grid constant has been chosen such that the required spacing between adjacent parts of the resistor diffusion will always be ensured. We decided to embed the resistor diffusion on the left and/or lower side of the cells, and to position the aluminium interconnection leads on the right and/or upper side of

the cells, in order to reduce parasitic effects.
The contacts of the resistor have already been placed on the boundary of the region. During the planarization of the potential graph it may have been decided to cross the resistor region (between its contacts) with some interconnection lead(s). As already mentioned in preceding chapters we have to prevent possible detrimental effects due to surface inversion. It has to be avoided that some interconnection leads cross a certain resistor diffusion more than once. We take care of this by the following two measures:

- If a certain resistor has to be crossed by several leads, then these leads are forced to cross the resistor region parallel to each other.
- The resistor diffusion is, in the crossing region, meandered in a direction that is parallel to the crossing interconnection lead(s) (e.g. the lead is crossed by the diffusion only one time). Thus, for the rectangular regions that are crossed by leads, the direction of meandering is fixed.

A consequence of these measures is that the meandering of the resistors will have to take place after the execution of the wiring procedure, since we have to know on which place the leads cross the resistor region.

Let us assume that we have determined a sequence of adjacent cells (the so-called "meander sequence") in the resistor region, the first cell of this sequence being a cell of one of the contacts of the resistor, and the last cell being a cell of the other contact. Such a sequence of cells corresponds with a certain configuration of the resistor diffusion, according to the agreement that the diffusion is always placed on the left and/or lower side of the cells (see fig. 8.1).

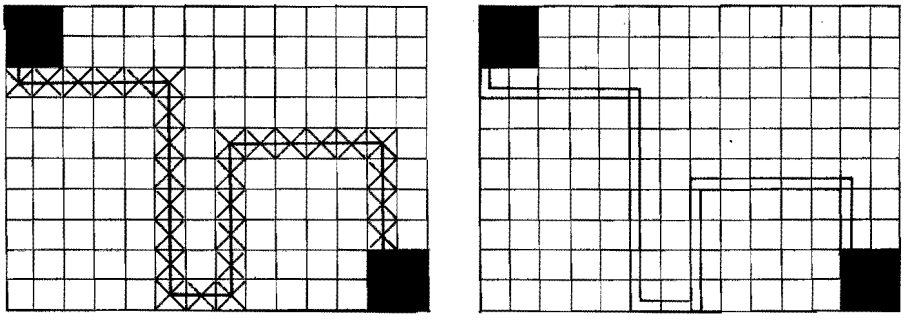


Fig. 8.1. Correspondence between a sequence of cells and the resistor diffusion.

The resistance of the already determined diffusion configuration will be currently updated during the meandering procedure. If in a certain stage of the procedure the desired resistance value is exceeded a meander reduction process is started:

Bends in the meander are (repeatedly) cut until the value of the resistance is within the tolerance of one cell. The exact value can be obtained by shifting one of the contacts, or by enlarging the area of a contact (hole) and thus shortcircuiting a part of the diffusion in the end cell of the meander sequence. If it is not possible to reduce the resistance value satisfactorily, one can bring the contacts close to each other (for example: shift along the boundary of the resistor region, and adapt the wiring) and restart the meandering process. Of course, this situation should be avoided by placing the contacts not too far from each other.

In the other case, namely if the desired resistance value is not yet realized by the current meander sequence, we continue with a meander extension process:

Search for a so-called "free region". Such a region has to be of rectangular shape, all its cells have to be "free" (i.e. not yet included in the meander sequence), and it is at least with one of its sides completely adjoining the cells of the current meander sequence. This side is called "linking side", and is required to be at least two cells long.

One of the following two basic cases (depicted in figure 8.2) will occur: there exist (case -a-), or there do not exist (case -b-) two adjacent non-free cells along the linking side of the free region, that are subsequent in the meander sequence.

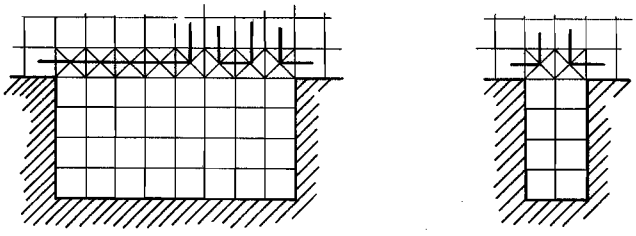


Fig. 8.2. The two basic cases for non-free cells along the linking side of a free region.

For case -a- it is obvious that the current meander sequence can easily be extended by "opening" it between two neighbour cells that are along the linking side of the free region, and adding a meander sequence in the free region. The problem of finding a suitable meander in a rectangular (free) region, such that start and end cell are along the same side, can be easily solved (see figures 8.3 and 8.4). In case the free region is deeper than one cell, then maximally one cell cannot be incorporated in the meander sequence.

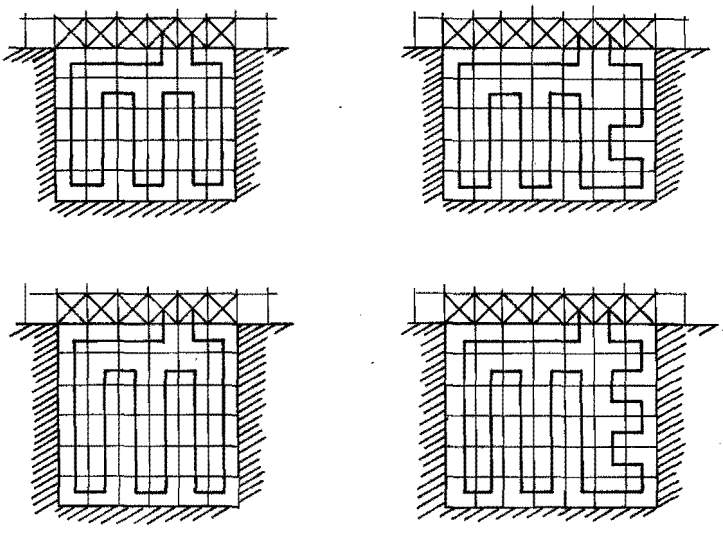


Fig. 8.3. Meandering a free region of more than two cells deep (start and end cell being neighbours, and being situated along the same side).

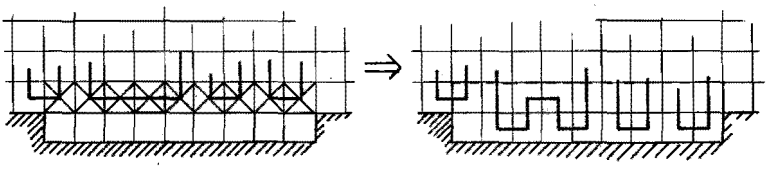


Fig. 8.4. Meandering a free region of only one cell deep.

In the case -b- we are faced with the problem that the current meander sequence cannot be opened along the linking side of the free region. In this case we first try to extend the free region into the adjacent non-free region by cutting one of the adjacent meander bends (see figure

8.5a). If this is not possible we "shortcircuit" the current meander sequence along the linking side, and delete the part of it that is not connected anymore (see figure 8.5b). We now have two neighbour cells along the linking side of the free region, and the cells of this region can be incorporated in the current meander sequence. However, it is

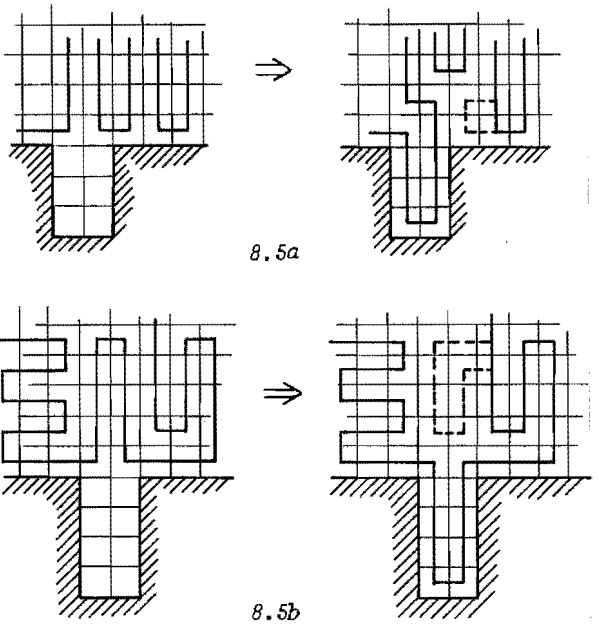


Fig. 8.5. "Extending" the meander sequence in the case -b- (future extensions are dashed).

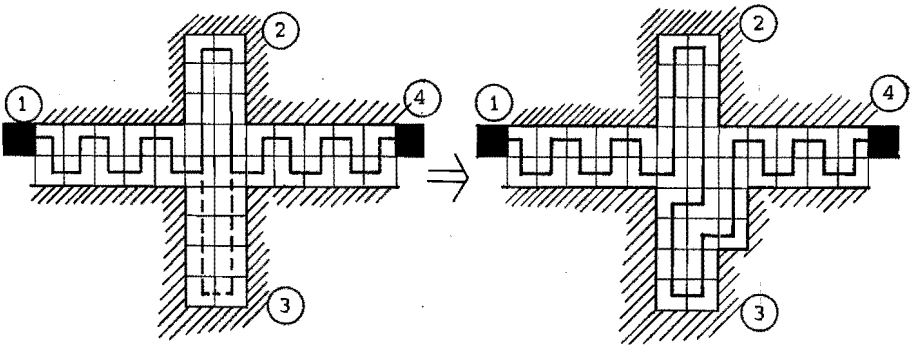


Fig. 8.6a. The passage from 1 to 4 is too narrow to use both the regions 2 and 3 .

Fig. 8.6b. Extending the resistor region with two cells.

possible that this process does not converge (see for example figure 8.6a: the passage from 1 to 4 is too narrow to use both the regions 2 and 3). If the meander has been "shortcircuited" several times (on the same place) the passage in question can be widened by extending the resistor region with two cells as depicted in figure 8.6b. These two cells do not belong to some other component region, since we kept one grid constant spacing between the various component regions. We already mentioned that sometimes a cell of some free region will not be used in extending the meander sequence. In the rare case that the required resistance value has not yet been realized, but no free region can be found anymore, one can use the same grid expansion process as applied during the wiring process (see chapter 9).

We still have to make some remarks about the construction of a meander sequence, suitable to be used as the start sequence for the described meander extension process.
First of all a preferred direction of meandering is determined (for example the direction of crossing interconnection leads that are present, or otherwise the direction for which the average width of the resistor region between the contacts is maximum). Next, a sequence of rectangles is constructed, such that each two consecutive rectangles are adjoining, and the two contacts of the resistor form the extreme rectangles of the sequence (a contact covers a rectangle of cells, for example 2×2). Such a sequence of rectangles can always be found, since the contacts are placed in the same (connected) region. The resistor region can be partitioned into rectangles by drawing, in the preferred meandering direction, lines through each corner of the resistor region. Consider the created rectangles of the resistor region as the vertices of a graph, and introduce an edge between two vertices if the two corresponding regions have a line segment in common. Consider one of the two vertices that represent the regions containing a contact as, the origin, and the other vertex as the target. By searching for a path between origin and target the sequence of rectangles is fixed. Next, in the preferred direction the meander sequences are constructed in these rectangles, such that the end cell of each rectangle is adjacent to the start cell of the next rectangle. The start and end cell of the rectangle(s) that are crossed by an interconnection lead, are placed

such that this rectangle contains the maximum number of cells in the meander sequence. This rectangle will not be used for a possible meander extension.
Of course, there is strived after simple configurations of the various resistor regions. Often the start meander will already include a great part of the resistor region. In case the contacts are placed close to each other, this may be strengthened by first bringing the start and end cell far from each other (e.g. meander from both contacts, along the boundary, to opposite sides of the region) and considering the problem of meandering the rest of the region.

References.

[8.1] S.B.Akers, J.M.Geyer, D.L.Roberts, "IC mask layout with a single conductor layer", Proc. 7th Design Automation Workshop, pp. 7-16, San Francisco, June 1970.

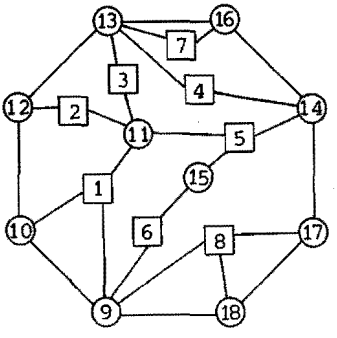
[8.2] P.C.Gilmore, "Optimal and suboptimal algorithms for the quadratic assignment problem", J. Soc. Indust. Appl. Math., Vol. 10, No. 2, pp. 305-313, June 1962.

9. THE ALUMINIUM MASK

9.1. The wiring procedure.

The pattern of the interconnections of the chip is determined in two phases. First the interconnections outside the isolated regions are determined. Starting point for this part of the procedure is a plane graph, namely the plane representation of the island interconnection graph of which the construction is described in section 6.3.. In the second phase the interconnections between the contacts in an island are determined. Again we have a plane graph available, this time the concerned part of the modified potential graph extended with the "crossors" defined in section 6.3. and consecutivity edges to fix the sequence in which the interconnections leave the islands. We call these graphs the island graphs. Let us first give a sketch to reveal the main ideas.

Thus, in either phase we have a plane graph to start from. The potentials that are between the considered part of the circuit and its "outside world" are represented in the boundary of this graph. In the case of the island interconnection graph these vertices correspond with the bonding pads. In the case of an island graph the vertices corresponding with the potentials of the interconnections with other islands and bonding pads are meant. The Jordan curve consisting of the consecutivity edges encloses the rest and is the first of a series of "limit circuits". In general "limit faces" are defined as the faces in the interior of the limit circuit and of which the boundary has at least one edge in common with the limit circuit. We presume for the moment that the simple graph of the structure under consideration (i.e. the limit circuit and the part of the graph in its interior) is 3-connected. This implies that the intersection of the boundary of any limit face and the limit circuit is a path containing all the vertices the two circuits have in common (theorem 10 of appendix A). The generalization to other graphs will be given later. We now define an admissible sequence for routing the interconnections represented in the graph:
A sequence for routing the interconnections represented in the considered graph is said to be admissible if it can be constructed by the



startv = 9

9-8, 8-18

8-17

9-6, 6-(15)-5, 5-14

16-7, 7-13

13-3, 3-(11)-2, 2-12

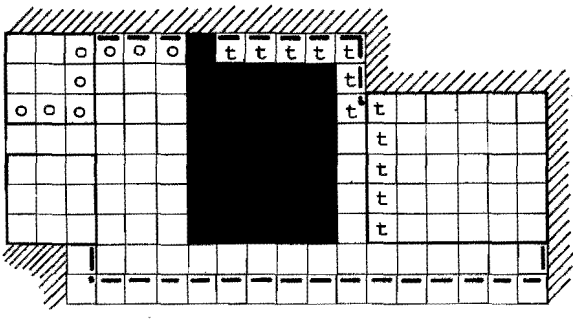
11-1, 1-10

1-9

11-5

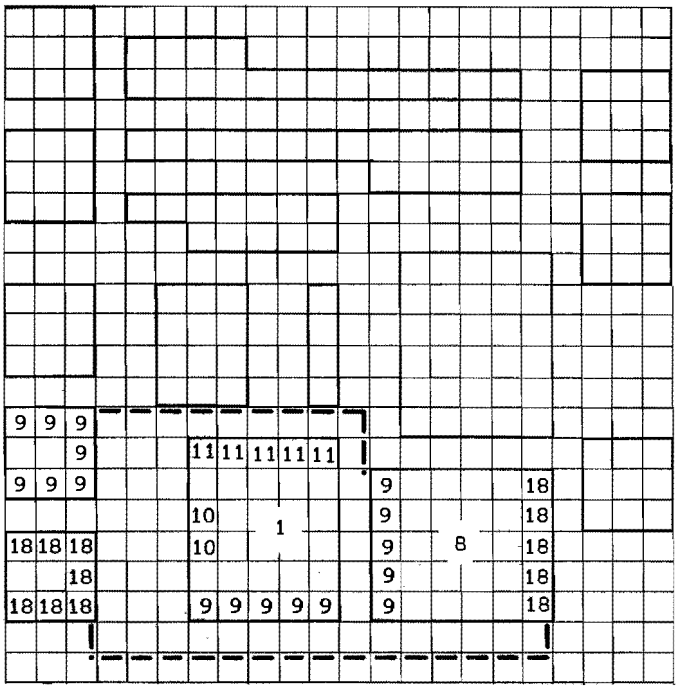
13-4, 4-14

a. A plane graph and an admissible sequence.

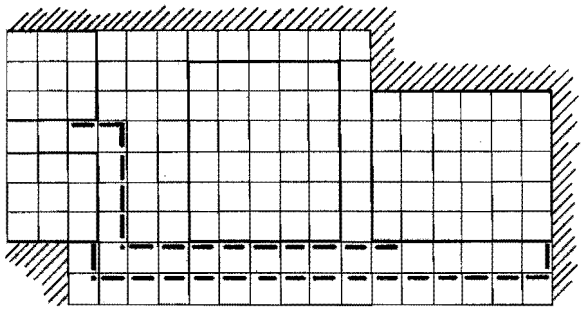


c. The auxiliary route is given by the black cells. The new route must be from one of the cells with an "o" to one of the cells with a "t".

Fig. 9.1. Squeezing out enclosed contacts.



b. Access to the contacts of component 1 is cut off.



d. The contacts of component 1 are squeezed out.

following procedure:

0. start with an empty interconnection sequence IS and the first limit circuit $(C_1, C_1[\,])$;
1. choose a limit face w;
2. add the interconnections symbolized by the edges of $C_w[] \setminus C_1[]$ to IS;
3. replace the edges of $C_w[] \cap C_1[]$ by the edges of $C_w[] \setminus C_1[]$ in order to form the next limit circuit:

$$(C_1, C_1[\,]) := (G,U)^{\lhd}_{U=(C_1[] \cup C_w[]) \setminus (C_w[] \cap C_1[])}(C_w \cup C_1, C_w[] \cup C_1[]);$$

4. return to step 1.

When routing in an admissible sequence, any grid can be wired if the interconnection leads may be chosen arbitrarily thin. For each time we start working at the next limit face all potentials (interconnections and contacts) that still have to be interconnected, are in a region enclosed by the interconnections and components represented in the limit circuit. This region is partitioned into two parts by the first tentative routing of the interconnections represented in the boundary of the limit face, but not in the limit circuit. By scanning the interior of the part that corresponds with the limit face we can find all potentials that are no longer accessible from the other part. If such a potential is found, it has to be squeezed out of that region. This can be achieved by searching an auxiliary route from this contact to one of the newly routed interconnections. This interconnection must be rerouted inside the region avoiding the cells connecting the contact with the faulty route. The cells of this faulty route are available, except the one from which the auxiliary route was traced (fig. 9.1).

However, aluminium leads have a prescribed minimum width, and thus it is possible that there is no space between some non-available cells and an interconnection or non-available cells represented in the limit circuit c.q. the concerned limit face boundary. Perhaps this deadlock can be abolished by shifting some already existing routes away from the cells to be squeezed out. In case there is no possibility of creating space by moving wire routes, we probably have detected a non-wirable situation and we have to expand the grid structure.

Locating these bottlenecks is always possible, because during the placement we have ensured a spacing of one grid constant between components. Thus, whenever a region is cut off an interconnection represented in the limit circuit or in the boundary of the limit face, is involved. One pass along these interconnections and the components they have a contact on suffices to find a starting point for wire shifting or grid expansion.

After outlining the main points of the wiring procedure, we now have to work out the details. In the next section we, therefore, pay attention to the cases, where space is required, because interconnections cannot pass between obstacles, since the passage way a thinner lead would choose is too narrow for an aluminium lead with the prescribed width. Of course inserting extra cells between these obstacles and thus expanding the grid structure solves this problem, but costs area. Beside this final remedy, it is expedient to include some procedures that try to avoid this grid expansion. Two of them are also described in the next section. In the third section we give a procedure to determine an admissible sequence in general.

9.2. Grid expansion, wire shifting and region shrinking.

Whenever application of space-creating procedures appears to be necessary, the location of the bottleneck can be determined, and from the shape of the non-available part that is too close to the interconnection, one can derive in which direction space is required: horizontal, vertical or both. We only describe how an extra row can be generated. (An extra column can be generated by an analogous method).

First a number of horizontal and vertical line segments must be found satisfying the following requirements.

1. these line segments must coincide with the lines of the grid (i.e. they are not allowed to cross a cell);
2. arbitrary vertical straight lines have at most two points in common with the horizontal line segments;
3. after cutting the grid over these line segments it must fall apart into two pieces;

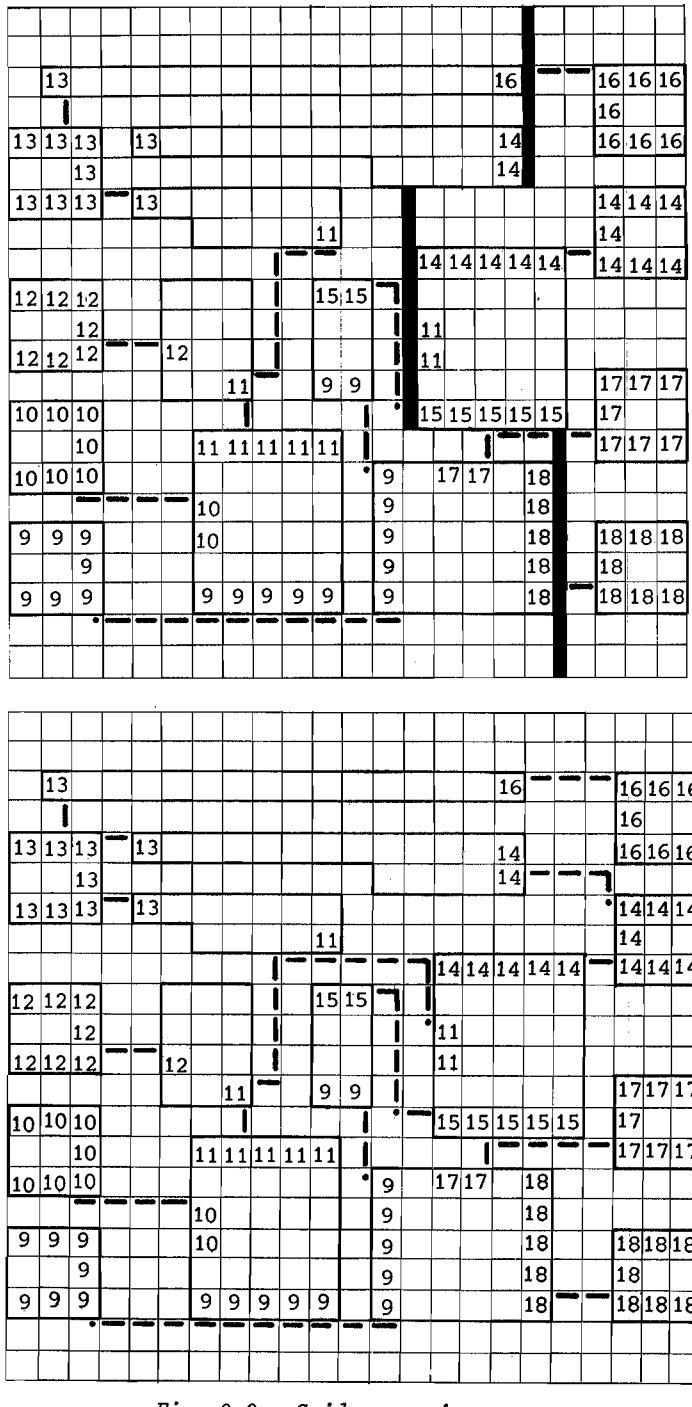


Fig. 9.2. Grid expansion.

4. it is not allowed to cut components with fixed layouts;
5. if possible, breaking of horizontal interconnections by vertical line segments must be avoided.

Between the horizontal edges of the cut new cells are inserted. Interconnections broken by the horizontal segments must be restored by taking new cells into the routes (fig. 9.2).

ad. 5: Whenever the procedure searching for the line segments runs upon a horizontal interconnection, and there are enough "free" cells in the neighbourhood to restore the interconnection, there is no objection against breaking the interconnection (fig. 9.3a). Otherwise the procedure starts following the horizontal interconnection until a point is reached, where it is possible to escape without breaking a horizontal interconnection. If being "caught" in front of a component, the procedure can double upon its steps to see whether there is a way out or it may create one (fig. 9.3b). If none of these attempts is successful, extra columns can be generated. The number of extra columns is equal to the number of adjacent broken interconnections (fig. 9.3c).

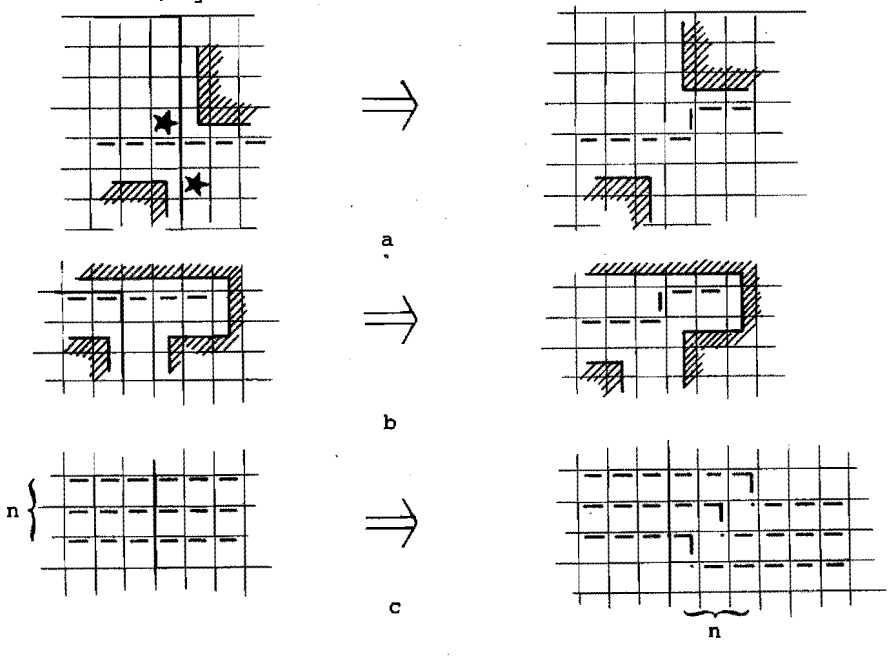


Fig. 9.3. Breaking horizontal interconnections.

The described "expension method" itself must be considered as a last resort, because it results in a considerable increase of chip area, especially for the smaller chips. For the μA 725 each extra row or column costs about 2% increase of area. Thus we discuss the possibilities to move the interconnections already laid out in order to avoid taking recourse to such a rigorous method.

Changing the location of an interconnection is reduced to the application of one of four basic shift operations (fig. 9.4), conditional on the situation that is encountered. To ascertain which situation we

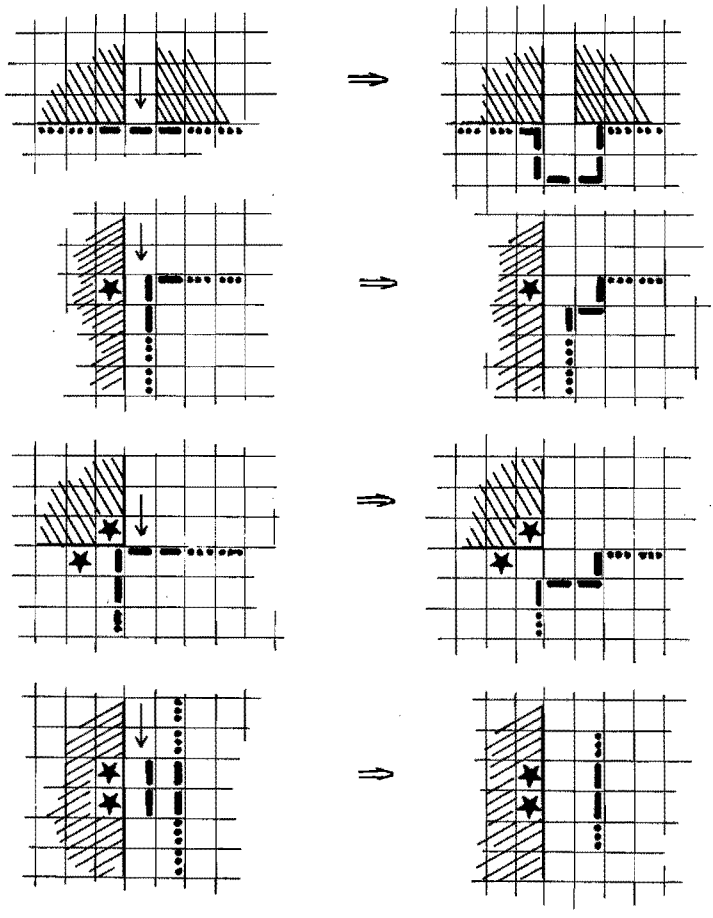


Fig. 9.4. The four basic shift operations.

have arrived in, a square grid of 9 cells is scanned for every neighbour of the reachable cell from which we want to continue our search. A procedure to select the appropriate shift operation is given in fig. 9.5. However, a call for the respective procedures does not mean that the operation is performed, for the cells to which the interconnection passing through b2 is to be moved, have to be examined. This may result in another call for one of the shift operations in order to move one or more of these cells. Furthermore, one has to check, whether the cells, containing a "star" in fig. 9.4, have state b2ϕ. The interconnection with that cell and the potential of the wire that was in b2 may not be broken or the interconnection must be restored. Upon its return into the main procedure (SELECT), a boolean must be set such that the procedure is continued from the label "quarter turn".

The above contrivances may also be used during the tentative routings, when a difficulty is encountered, though there is a greater chance that the grid is wirable. Before turning to wire shifting or even grid expansion, in such a case it is expedient to try another sequence of routing the interconnections represented in the boundary of the limit face, but not in the limit circuit. (This still is an admissible sequence!). We therefore check whether the jammed search wave has reached cells adjacent to one of the newly established tentative routes. By removing this route, and continuing the search that was not successful before the removal of the interconnection, we probably can establish both routes without applying one of the more time consuming methods described in the above.

In order to keep the available space for future wires as large as possible, a shrinking of the regions no longer accessible brings some gain. This is done by moving the straight pieces of the wiring into the region if the cells immediately adjacent to this piece of wiring are free from obstacles. This action is repeated up to three times maximally; otherwise it is probable that the interconnections become unnecessarily long without yielding useful area. Also during the trace-back part of the path optimization the choice among the cells with equal distance from an origin, is based on the principle of keeping the available space as large as possible.

```
'procedure' SELECT;
'for' each direction 'do'
'begin''if' b2φ≠potᴧb2φ=interconnection 'then'
   'begin''if' a2φ=b2φ 'then'
      'begin''if' c2φ=b2φ 'then'
         'begin''if' a3φ=b2φᴧa1φ≠b2φᴧa1φ=non-available 'then' SHIFT3 'else'
            'if' c3φ=b2φᴧc1φ≠b2φᴧc1φ=non-available 'then' SHIFT3 'else'
            'if' (a1φ=non-availableᵛc1φ=non-available)ᴧ
               a3φ≠b2φᴧb3φ≠b2φᴧc3φ≠b2φ 'then' SHIFT1
         'end''else'
         'if' b3φ=b2φ 'then'
         'begin''if' a3φ=b2φ 'then' SHIFT4 'else'
            'if' c2φ≠b2φᴧc2φ=non-available 'then' SHIFT2
         'end'
      'end''else'
      'if' c2φ=b2φ 'then'
      'begin''if' b3φ=b2φ 'then'
         'begin''if' c3φ=b2φ 'then' SHIFT4 'else'
            'if' a2φ≠b2φᴧa2φ=non-available 'then' SHIFT2
         'end'
      'end'
   'end';
quarterturn:
'end'
```

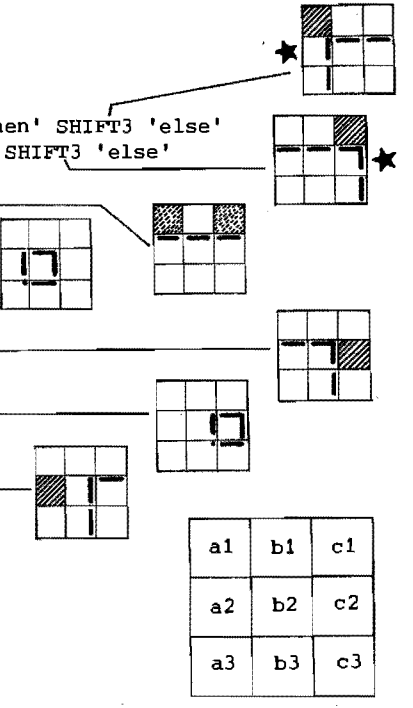


Fig. 9.5. Procedure SELECT.

φ is a mapping assigning to the cells several "states":
pot = the potential searched for is contained in this cell,
interconnection = an aluminium lead is in this cell,
non-available = this cell may not be taken into a route.

9.3. Admissible interconnection sequences.

In the first section we described a construction method for admissible sequences in case the successive limit circuits are such that the structure consisting of this circuit and the part of the graph in its interior is 3-connected. Otherwise it might happen that a limit face boundary contains a separation set of this structure. Treating such limit faces would spoil the principle of the procedure: "keep all entities of which not all the relevant data concerning placement and interconnections are determined, reachable from one and the same point". In order to be capable of handling connected graphs not satisfying the severe requirement upon connectivity, we have to generalize the procedure of section 9.1. Before we do so we introduce some notations such that we come to concise description of the method.

A free edge is an edge that has not been in a limit circuit, and thus the interconnection symbolized by this edge is not yet in the interconnection sequence. F(g) is the set of free edges associated with vertex g. The vertices of C_1 are cyclically ordered by the succession in which they occur at the limit circuit. If g is a vertex of C_1, we denote the next vertex in the "counter clockwise succession" by nextv(g), and the next vertex in the "clockwise succession" by lastv(g). nextu(g,g') is the first free edge that is encountered after [g,g'] when scanning the edges associated with g in a counter clockwise succession around g.

Initially startv is an arbitrary vertex of C_1:

```
'repeat' 'if' F(startv)≠∅ 'then'
   'begin' [x,y]:=nextu (startv, nextv(startv));
           LF:=∅; [x,y]:∈LF;
      'repeat' z:=y; [x,y]:=nextu(y,x);[z,y]:∈LF
      'until' y∈C1;
      'if' y=startv 'then' articulation point (startv)'else'
```

```
'begin' g:=startv;
   'repeat' g:=nextv(g)
      'if' F(g)≠∅ 'then' 'begin' startv:=g; goto L 'end'
   'until' g=lastv(y);
   'if' nextu(y,x)≠[y,lastv(y)] 'then' articulation point (y)
'end'
updating limit circuit; startv:=y
'end' 'else'
startv:=lastv(startv)
L:
'until' no free edges left;
```

A result of this procedure is shown in fig. 9.1a. In the procedure "updating limit circuit" several actions must be executed. First, it must be checked whether the minimal graph of which the set of edges is LF, is a path. In that case the interconnections symbolized by the edges in LF are added to the interconnection sequence. If the graph is not a path, there is at least one articulation point. If an articulation point is discovered, the subgraph that can be separated from the rest of the graph by removing this articulation point and that either consists of edges of LF or consists of edges of LF and the edges enclosed by circuits formed by edges of LF, must be treated first. Mostly this subgraph is simply a path. If it is not a path, the sequence in which the interconnections are to be routed, must be determined. This can be done by a procedure like the one just described, starting with an arbitrary edge. Limit faces are now in the exterior of the limit circuit (initially an edge!).

9.4. The interconnections between islands.

When routing the interconnections between islands, we can avoid the difficulties for which we have given solutions in section 9.2. Initially only the grid cells covering isolation channels are available for the interconnections. Each time the routes for the interconnections represented in the actual limit face boundary are established, the cells adjacent to these routes and outside the region to which access is going to be cut off by these routes, are made available. As a consequence of this the interconnections that pass between two islands use

the isolation channel and a part of only one of the two islands. It is therefore expedient to move the interconnections later, such that their locations are more symmetric with respect to the isolation channel. Sometimes grid expansion is necessary, for example when the aluminium interconnections do not leave enough area for the components.

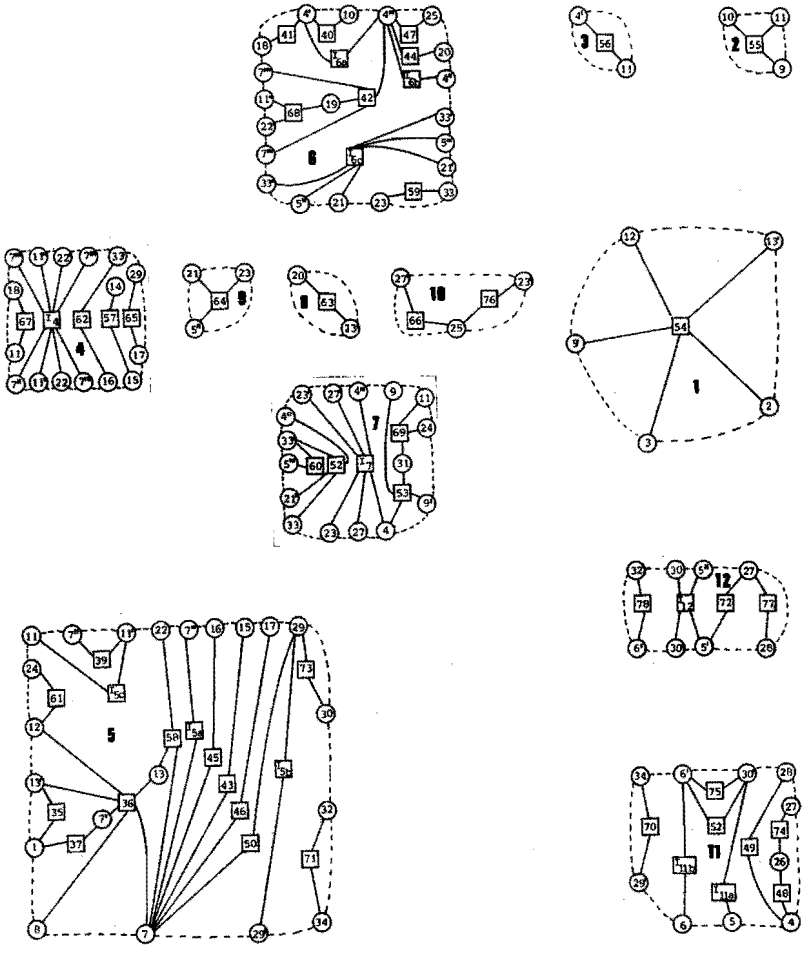


Fig. 9.6. The island graphs of the μA725.

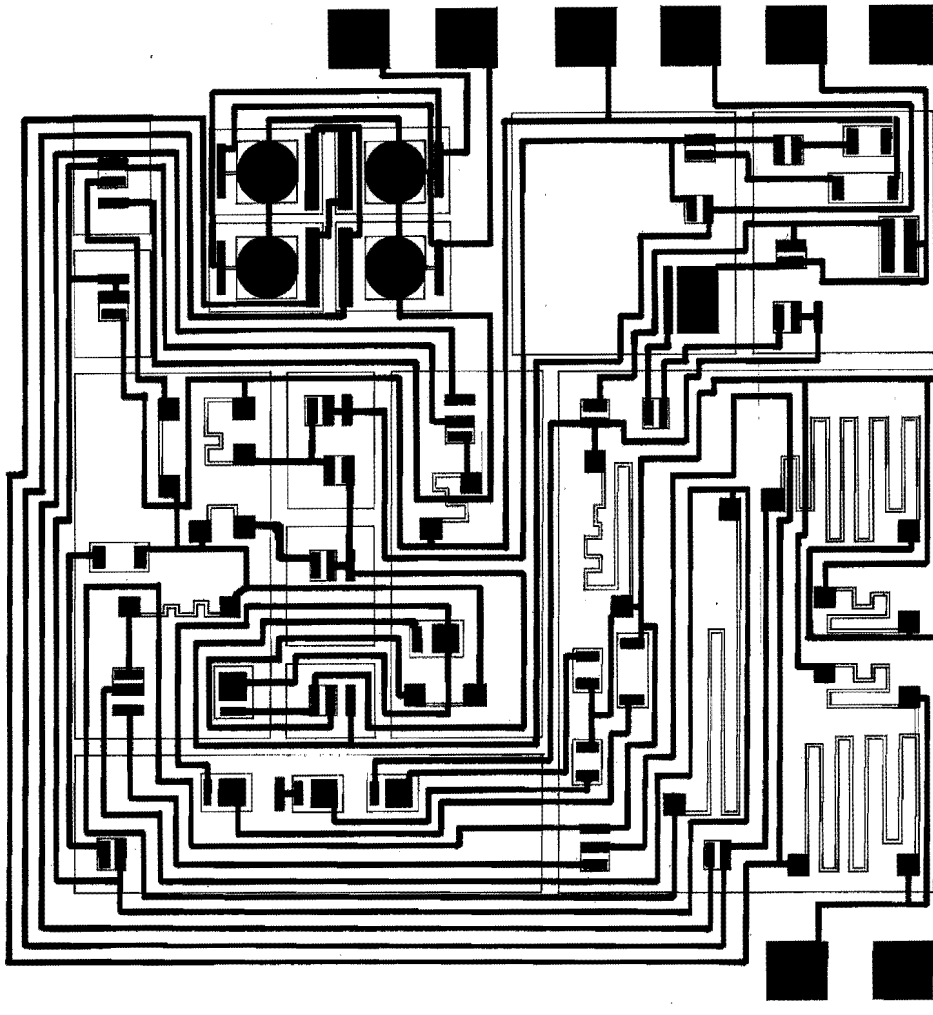


Fig. 9.7. A layout for the μA725.

10. CONCLUSIONS AND FINAL REMARKS.

10.1. Conclusions.

A number of procedures have been described that together form a design system of the layout for bipolar integrated circuits. The design process has been decomposed serially into a number of separate subtasks. Furthermore we tried to utilize the various freedoms in the design as early as possible in the design process. Then the consequences of each decision can be taken into account during all the subsequent steps, and thus the procedures at the end of the design process will be facilitated, and the number of detrimental effects of these procedures can be kept low.
The employed partitioning of the complete design and the information flow between the various blocks is illustrated by the flow diagram of figure 10.1. For several parts we have only implemented one method and chosen for an approach that seemed to be most suitable. Shortage of time prevented us from developing other approaches as well. However, the results obtained from the proposed design process have been very satisfactory. Of course a number of refinements can probably improve these results and enlarge the scope of employing the algorithms. In the next section we will mention refinements that deserve consideration. Frequent application of the design system will be necessary to find out the shortcomings for practical use. However, all basic problems have been solved and all methods are suitable for implementation on the computer. With the described procedures it is possible to design layouts for bipolar integrated circuits in a fully automatic way.

10.2. Final remarks.

The graph-theoretical model of chapter 3 appeared to be a suitable starting point for fixing the basic technological decisions that have to be taken for constructing an acceptable layout of the circuit. In chapter 4 we described a number of methods for fully automatic planarization of the potential graph. The results of these methods compete with results obtained in literature (by methods that are not fully automatic) and in practice. Beside the development of these

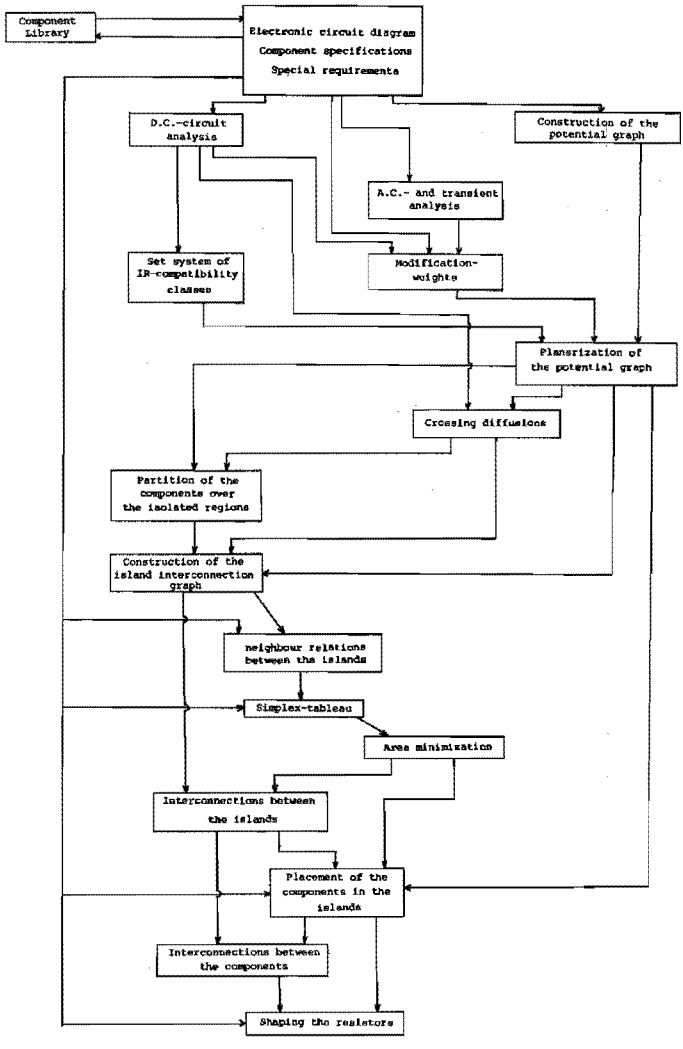


Fig. 10.1. The various parts of the layout design process.

procedures, a very efficient method was developed for drawing a representation of a planar graph (see appendix F). Even interactive planarization using this very efficient drawing procedure turned out to be at least as expensive (qua computation time) as the fully automatic approach, without giving better results. It goes without saying that an interactive approach has additional costs and other disadvantages due to the need of an experienced circuit layout designer and a graphics display.

In many parts of the design process the Lee-algorithm (appendix C) has been used. It is a fundamental algorithm easy to implement and efficiently operating in case the number of neighbour cells is low. We did apply this algorithm in its more general forms. For example in determining the crossing potential leads over the resistor diffusions (see chapter 5) we incorporated effects that can influence the electrical behaviour of the circuit (we favour the leads having a high potential). However, as can be recognized from the results of our example (the operational amplifier μA725), these desires derived from considerations on the electrical performance can be disadvantageous for the cost of the chips (due to detours in the wire routes the total chip area will increase). In the example of section 5.2. the metal lead of potential 7 (+15 volt) was chosen to cross resistor 42. The consequence of this decision is that the distance of c-vertex 39 to the set of other c-vertices of island 5 and the distance of c-vertex 67 to the set of c-vertices associated with island 4, is greater than necessary. The final consequence is that a crossor has to be introduced in the islands 4 and 5 (see chapter 6). It may be worth-while to weigh the advantages against these disadvantages, and consider methods that try to take account of the distances between groups of EP-components of the same island as soon as the resistors are inserted.

Area consuming detours of certain wire routes can also be a consequence of an unfavourable neighbour relation of the islands. In particular two islands are preferably situated near to each other on the chip if there are many interconnection leads between these islands. However, this does not help us very much in case the "channel" of interconnecting leads has to make a detour due to the fact that some other island

has been "pushed" between such pair, thus pushing away the whole interconnection channel in front of it. It may therefore be expedient to add some measures to prevent the occurrence of these situations. In determining the neighbourship relations of the islands we also keep track of the probable length and width of the chip (acting as if all the islands have a square form). For certain islands the shape can be prescribed (e.g. for supercomponents) and thus it may be advantageous to incorporate the fixed island shapes in this procedure.

As mentioned in chapter 8 and 9 the wiring process has been partitioned into two parts. By first determining the wiring between the islands we obtain information that can be used with advantage during the placement procedure. Later on the wiring between the components in the same islands is determined. However, due to these separate applications of the wiring procedure, unnecessary detours can occur for certain wire routes. In a kind of "patching-up" procedure one may delete these unnecessary detours.
Furthermore, one may shrink the islands as much as possible in order to reduce parasitics: if a part of the border of the island does not contain components (for example due to passing metal leads), this part can be incorporated in the deep-p-diffusion.

During the determination of the wiring pattern an expansion procedure may have to be applied in case a certain wire route is blocked (see chapter 9). The number of times that this procedure is called for execution,depends of course on the areas that are reserved for the various islands. The minimal area that is demanded for a certain island (area minimization in chapter 7), has to be estimated, since the configurations of the islands and some components, and also the wiring pattern are not yet determined. The tighter the estimates of island areas, the greater the chance of need for application of the expansion procedure. If the estimates are taken too large, a compaction procedure has to be applied. Which strategy of estimation (tight or rather large) is more suitable still has to be investigated.

Due to the various adjustments of certain areas in order to fit them into the grid structure (the sizes of island and component areas are

rounded off to multiples of the grid constant) a certain amount of chip area will not be "used". This loss of area can be decreased by decreasing the grid constant. However, in that case several procedures have to be adjusted appropriately (e.g. searching paths having a width of more than one cell) and the storage requirements will increase considerably.

Since up to now not all implementations of the various algorithms have been completed, no final computation times of a practical example can be given. However, for the μA725 the computation times of all the (partially completed) procedures are in the order of seconds or minutes (on the B6700). It is expected that the complete layout design of the μA725 costs less than 30 minutes computation time. For larger examples the computation times and storage requirements may grow considerably due to the fact that many of the algorithms are not linearly dependent on the number of edges and/or vertices of the various graphs, or on the number of cells of the grid structure. Acceptable computation times are expected for circuits up to a few hundred edges, unless one is satisfied with lower levels of optimization in the various algorithms. However, the very large circuits can be partitioned (as is also done in designing by hand or interactively) in a number of subcircuits, and the layout of these subcircuits designed first. The obtained layouts then can be incorporated in a final layout design process by using the concept of supercomponent for the various subcircuits.

APPENDICES

A. A précis of graph theory.

Basic notions

A graph consists of two finite sets, a set of vertices and a set of unordered pairs of distinct vertices. These pairs are called edges. Notation: (G,U) where G is the set of vertices and U is the set of edges. [x,y] is the edge associated with $x\epsilon G$ and $y\epsilon G$. $|G|=n$ and $|U|=m$. A graph (G,U) determines a binary, symmetric, antireflexive relation ρ on G:

$$\forall_{(x,y)\epsilon G\times G}[x\rho y\leftrightarrow[x,y]\epsilon U].$$

If $\rho=G\times G\setminus\iota$, ι being the identity relation, the graph (G,U) is said to be complete. Notation: $(G,U)\epsilon CM$. A graph (G,U) for which $\exists_{x\epsilon G}[x\bar{\rho}=G]$, where $\bar{\rho}$ is the transitive closure of ρ, is called connected. Notation: $(G,U)\epsilon K_1$. Two graphs (G_1,U_1) and (G_2,U_2) are isomorphic if there exists a bijective mapping ϕ for which $\rho_2\phi=\phi\rho_1$. The degree of a vertex $x\epsilon G$ is the number $|x\rho|$. Notation: $x\gamma_U$.

A graph (G,U) for which $\forall_{x\epsilon G}[x\gamma_U\neq 2]$ and $\neg\exists_{[x,y]\epsilon U}[x\gamma_U=1\wedge y\gamma_U\neq 1]$ is called simple. Contraction of an edge $[a,b]\epsilon U$ is the deletion of all edges contained in $\{[a,b]\}\cup\{[a,c]\,|\,c\epsilon\{d\,|\,[a,d]\epsilon U\wedge[b,d]\epsilon U\}\}$ and identification of a and b. The graph obtained from (G,U) by deletion of all edges associated with exactly one vertex of degree 1 and this vertex, and by contraction (one by one) of edges that are associated with a vertex of degree 2, is called the simple graph of (G,U). Notation: $(\overline{G},\overline{U})$.

A graph (G',U') is a subgraph of (G,U) if $G'\subseteq G$ and $U'\subseteq U$. Notation: $(G',U')\lhd(G,U)$. $(G'',U'')\lhd(G',U')\wedge(G',U')\lhd(G,U)\rightarrow(G'',U'')\lhd(G,U)$. (G',U') is called a maximal subgraph of (G,U) with respect to a property p if (G',U') has p, $(G',U')\lhd(G,U)$ and $\neg\exists_{(G'',U'')\lhd(G,U)}[(G',U')\lhd(G'',U'')\wedge(G'',U'')$ has p]. Notation: $(G',U')\lhd^{p}(G,U)$.

(G',U') is called a minimal subgraph of (G,U) with respect to a property p if (G',U') has p, $(G',U')\lhd(G,U)$ and $\neg\exists_{(G'',U'')\lhd(G',U')}[(G'',U'')$ has p]. Notation: $(G',U')\lhd_{p}(G,U)$.

A path in (G,U) is a connected subgraph of (G,U) with vertices only of degree less than 3 and of which two vertices have degree 1. Notation: (P,P[x,y]), x and y being the (only) vertices of degree 1 in the path, the so-called end vertices. A circuit in (G,U) is a connected subgraph of (G,U) with vertices only of degree 2. Notation: (C,C[ ]). The length of a path of a circuit is the number of its edges. The distance between two vertices $x \in G$ and $y \in G$ in (G,U) is the length of the shortest path in (G,U) with x and y as end vertices. Notation: $|x,y|_U$. Two paths are said to be independent of each other if they have no vertices in common except possibly their end vertices.

A connected graph without circuits is called a tree.

Theorem 1: A graph (G,U) is a tree fif for every pair of vertices of G there exists exactly one path between these vertices in (G,U) fif $(G,U) \in K_1$ and m=n-1.

Proof: (G,U) is connected, so there is a path between any pair of vertices, and not more than one, since two paths between the same two vertices imply one or more circuits.

For m=1 the statement is certainly correct. Suppose it is correct for every graph (G',U') with m=k-1. We have to prove it for (G,U) with m=k. There is at least one $x \in G$. We follow a path from x through other vertices never meeting a vertex twice (if this happens there exist two paths between some pair of vertices). We can continue this path, unless the reached vertex y is of degree 1. This must happen sooner or later, since G is finite. Delete y and its incident edge. $(G\setminus\{y\}, U\setminus\{[y,z] \mid [y,z] \in U\})$ is a graph with exactly one path between each pair of vertices, and with k-1 edges, thus $|U\setminus\{[y,z] \mid [y,z] \in U\}| = |G\setminus\{y\}| - 1$. Consequently, $|U| = |G| - 1$. Since there is a path between any pair of vertices (G,U) must be connected.

Suppose m=n-1 and (G,U) does contain circuits.
$U' = \{u \mid \exists_{(C,C[\ ]) \triangleleft (G,U)} [u \in C[\ ]]\}$, $(G_1,U_1)_{U_1=U'}(G,U)$

and $(G_2,U_2) \lhd_{U_2=U\setminus U'} (G,U)$. Now (G_2,U_2) consists of a number (p) of trees and $|U_1| = \frac{1}{2} \sum_{x\in G_1} \chi_{U_1} \geq |G_1|$, thus $m=|U|=|U_1|+|U_2| \geq |G_1|+(|G_2|-p)=|G_1|+(|G|-|G_1|+p-p)=|G|= n$, which contradicts our supposition.

A graph of which all circuits have even length, is called bipartite. A tree is a bipartite graph.

Theorem 2: A graph (G,U) is bipartite fif there is no edge between two vertices having an equal distance from some other vertex fif $\exists_{G_1\subset G}[\rho \subseteq (G_1\times G\setminus G_1)\cup(G\setminus G_1\times G_1)]$.

Proof: Suppose there is an $a\in G$ with equal distance from $b\in G$ and $c\in G$ and let $[b,c]\in U$. $(P_1,P_1[a,b])$ be a shortest path between a and b and $(P_2,P_2[a,c])$ be a shortest path between a and c. $\forall_{x\in P_1\cap P_2}[|x,b|_U=|x,c|_U]$, since the two paths are the shortest of their kind. Let $y\in P_1\cap P_2$ be the vertex for which $|y,b|_U$ is minimal and $(P_3,P_3[y,b])$ and $(P_4,P_4[y,c])$ the subgraphs of $(P_1,P_1[a,b])$ and $(P_2,P_2[a,c])$ respectively. Since the circuit $(P_3\cup P_4,P_3[y,b]\cup P_4[y,c]\cup\{[b,c]\})$ is of odd length, the graph (G,U) cannot be bipartite.

$(G',U') \lhd^{(G',U')\in K_1} (G,U)$. Take an arbitrary vertex $x\in G'$ and assign all vertices with even distance from x to G_1. Repeat this for every other maximal, connected subgraph of (G,U). There cannot be an edge between two vertices of G_1 (or $G\setminus G_1$), since there is no edge between vertices with equal distance to x and between vertices of which the distance from x differs more than 1.

Since every circuit must meet vertices of G_1 and $G\setminus G_1$ alternatingly, it must be of even length.

The union of two graphs (G_1,U_1) and (G_2,U_2) is the graph $(G_1\cup G_2,U_1\cup U_2)$, and their intersection is the graph (G',U') for which $(G',U') \lhd_{U'=U_1\cap U_2} (G_1\cup G_2,U_1\cup U_2)$. They are disjoint if $G_1\cap G_2=\emptyset$ and

edge-disjoint, if $U_1 \cap U_2 = \emptyset$. The complement of a subgraph (G',U') in (G,U) is the graph (G'',U'') for which $(G'',U'') \lhd_{U''=U\setminus U'} (G,U)$. Notation: $(G \urcorner G', U\setminus U')$. The set $G' \cap (G \urcorner G')$ is called the attachment set, its elements the vertices of attachment and its cardinality the attachment number, each of (G'',U'') in (G,U). A subgraph of $(G \urcorner G', U\setminus U')$ with at least one edge is (G',U')-bounded if its vertices of attachment in (G,U) are all in G'. If $(G_1,U_1) \lhd_{(G',U')\text{-bounded}} (G \urcorner G', U\setminus U'), (G_1,U_1)$ is called a bridge of (G',U') in (G,U). $(G \urcorner G', U\setminus U')$, the complement of any (G',U')-bounded subgraph in $(G \urcorner G', U\setminus U')$, and the intersection of (G',U')-bounded subgraphs, are (G',U')-bounded. The intersection of all (G',U')-bounded subgraphs containing $u \in U\setminus U'$ is the bridge containing u. (Bridges are edge-disjoint!). Thus the union of all bridges of (G',U') is the complement of (G',U') in (G,U). A bridge is connected. The set of bridges of (G',U') in (G,U) is denoted by $BR_{U'}$.

Two distinct bridges, (G_1,U_1) and (G_2,U_2), of a circuit $(C,C[\,]) \lhd (G,U)$ are said to be $C[\,]$-alternating if there is no path in $(C,C[\,])$ containing all the vertices of attachment of (G_1,U_1) and no vertices of attachment of (G_2,U_2), except possibly its end vertices. Notation: $(G_1,U_1) @_{C[\,]} (G_2,U_2)$. The relation $@_{C[\,]}$ is symmetric and antireflexive. Thus, $(BR_{C[\,]}, \{[x,y] \mid x \in BR_{C[\,]} \wedge y \in BR_{C[\,]} \wedge x @_{C[\,]} y\})$ is a graph, the so-called alternation graph of $(C,C[\,])$ in (G,U). Notation: $(BR_{C[\,]}, A_{C[\,]})$.

Connectivity

The connection number of $(a,b) \in G \times G$ in (G,U) is the maximum number of mutually independent paths between a and b in (G,U). Notation: $(a,b)\omega_{(G,U)}$. T is a separation set of $a \in G$ and $b \in G$ in (G,U) with $[a,b] \notin U$ if $T \subseteq G\setminus\{a,b\}$ and every path between a and b contains at least one vertex of T. The separation number of $(a,b) \in G \times G$ in (G,U) with $[a,b] \notin U$ is the number of vertices in a separation set of a and b with the minimum number of vertices. Notation: $(a,b)\tau_{(G,U)}$.

If $(a,b)\tau_{(G,U)} = 1$, then there is a vertex $c \in G$, $c \neq a$ and $c \neq b$, such that $\forall_{(P,P[a,b]) \lhd (G,U)} [c \in P]$.

$(G,U)\in K_1 \leftrightarrow \forall_{(a,b)\in G\times G}[(a,b)\omega_{(G,U)}\geq 1]$;

Theorem 3: $\forall_{(a,b)\in G\times G\setminus(\rho\cup 1)}[(a,b)\omega_{(G,U)} = (a,b)\tau_{(G,U)}]$
(Menger's theorem)

Proof: Clearly, $\forall_{(a,b)\in G\times G\setminus(\rho\cup 1)}[((a,b)\omega_{(G,U)}=0 \leftrightarrow (a,b)\tau_{(G,U)}=0)\wedge$
$\wedge((a,b)\omega_{(G,U)}=1 \leftrightarrow (a,b)\tau_{(G,U)}=1)\wedge$
$\wedge((a,b)\omega_{(G,U)}\leq(a,b)\tau_{(G,U)})]$

Suppose $h>1$ is the smallest separation number for which the theorem is false and (G,U) is a graph with the minimum number of vertices for which
$\exists_{(a,b)\in G\times G\setminus(\rho\cup 1)}[(a,b)\tau_{(G,U)} = h\wedge(a,b)\omega_{(G,U)}<h\wedge$
$\wedge\forall_{u\in U}[(a,b)\tau_{(G,U\setminus\{u\})}<h]]$, then (G,U) has the following three properties:

(1) $\forall_{u=[x,y]\in U}\exists_{T(u)\subset G}[|T(u)|= h-1\wedge$
$\wedge\forall_{(P,P[a,b])\lhd(G,U\setminus\{u\})}[T(u)\cap P\neq\emptyset]\wedge x\notin T(u)\wedge y\notin T(u)\wedge$
$\wedge(x\neq a \wedge x\neq b \rightarrow \forall_{(P,P[a,b])\lhd(G,U)}[P\cap(T(u)\cup\{x\})\neq\emptyset])$
for $T(u)$ cannot separate a and b in (G,U); so $(G',U')\lhd^{G'=G\setminus T(u)}(G,U)$ must contain a path between a and b and any such path contains $[x,y]$.

(2) $\forall_{c\in G}[[a,c]\notin U \vee [b,c]\notin U]$, because $(\{a,c,b\},\{[a,c],[b,c]\})$ would be a path in (G,U) between a and b and not in $(G',U')\lhd^{G'=G\setminus\{c\}}(G,U)$ for which $(a,b)\tau_{(G',U')}=$
$=(a,b)\omega_{(G'U')}= h-1$.

(3) If T is a separation set of a and b in (G,U) with $|T|= h$, then $\{[a,t]\,|\,t\in T\}\subset U \vee \{[b,t]\,|\,t\in T\}\subset U$. To prove this property we define $(G^a,U^a)\lhd_p(G,U)$, where p means "containing all paths in (G,U) between a and a vertex of T, which is the only vertex of T in this path". Likewise (G^b,U^b). Every path in (G,U) between a and b contains a vertex of T, and thus a minimal subpath between a and a vertex of T and a minimal subpath between b and a vertex

of T. Further $G^a \cap G^b = T$, otherwise there would be a path in (G,U) between a and b, not containing any vertex of T or there exists a separation set with less than h vertices. If $G^a \setminus T \neq \{a\}$ and $G^b \setminus T \neq \{b\}$, then we consider the graph $(G',U') = (G^b \cup \{a\}, U^b \cup \{[a,t] \mid t \in T\})$. Since $|G'| < |G|$ and any separation set of a and b in (G',U') is a separation set of a and b in (G,U), we have $(a,b)\omega_{(G',U')} = h$. Hence, (G^b,U^b) contains h paths having only b in common, and of which the other end vertex is in T. Similarly, (G^a,U^a) contains h independent paths between a and the vertices of T. Together, these paths form h independent paths between a and b in (G,U). Thus $(a,b)\omega_{(G,U)} = h$, which contradicts the definition of (G,U).

Let (P,P[a,b]) be a shortest path in (G,U) between a and b, $\{[a,d],[d,e]\} \subset P[a,b]$. From (2) we have $[d,b] \notin U$ and $e \neq b$, and thus by (3) and (1) $\{[a,t] \mid t \in T([d,e]) \vee t=d\} \subseteq U$. Again by (2) $\neg \exists_{t \in T([d,e])} [[t,b] \in U]$. Now by (1) we conclude that $T([d,e]) \cup \{e\}$ is a separation set of a and b in (G,U) and by (3) we obtain $[a,e] \in U$, which contradicts the definition of (P,P[a,b]).

A graph (G,U) with $|G| \geq 2$ and $\forall_{(a,b) \in G \times G \setminus \iota} [(a,b)\omega_{(G,U)} \geq p]$ is called p-connected. Notation: $(G,U) \in K_p$. The connection number $((G,U))\omega$ of (G,U) is defined as the number p for which $(G,U) \in K_p \wedge (G,U) \notin K_{p+1}$. We adjoin to K_1 the graph consisting of 1 vertex .

Theorem 4: $(\forall_{(a,b) \in G \times G \setminus (\rho \cup \iota)} [(a,b)\omega_{(G,U)} \geq p] \wedge (G,U) \notin CM) \rightarrow (G,U) \in K_p$
(Whitney's theorem)

Proof: The theorem is correct for $p \leq 1$ as is clear from inspection. In case $p > 1$, we have to prove $\forall_{(a,b) \in \rho} [(a,b)\omega_{(G,U \setminus \{[a,b]\})} \geq p-1]$. Suppose this is not so; then there exists a separation set T of a and b in $(G,U \setminus \{[a,b]\})$ with $|T| \leq p-2$. Let us define $(G_1,U_1) \lhd^{G_1 = G \setminus (T \cup \{a\})} (G,U)$ and $(G_2,U_2) \lhd^{G_2 = G \setminus (T \cup \{b\})} (G,U)$. Since $|T \cup \{a\}| = |T \cup \{b\}| \leq p-1$, (G_1,U_1) and (G_2,U_2) must be

connected.

Further, $\forall_{(x,y)\in G_1\times G_2}\ \forall_{(P,P[x,y])\lhd(G,U\setminus\{[a,b]\})}[P\cup T\neq\emptyset]$, otherwise there is a path between a and b, not containing any vertex of T. Thus $G_1\cap G_2=\emptyset$ and T is a separation set in $(G,U\setminus\{[a,b]\})$ for all pairs in $G_1\times G_2$. Suppose $\exists_{c\in G_1}[c\neq b]$, then $[a,c]\not\subset U$ and $T\cup\{b\}$ separates c and a in (G,U), which means that $(a,c)\tau_{(G,U)}\leq p-1$. By theorem 3 $(a,c)\omega_{(G,U)}\leq p-1$, which is in contradiction with our first hypothesis. Thus $G_1=\{b\}$. Similarly, $G_2=\{a\}$ and we must conclude that $|G|\leq p$. But then (G,U) cannot satisfy the conditions of the theorem.

A graph (G,U) with $|G|\geq 2$ is called p-separable if these exists a separation set T of some pair $(x,y)\in G\setminus T\times G\setminus T$ in (G,U) with $|T|=p$. Such a set is called a separation set of (G,U). The separation number $((G,U))\tau$ of (G,U) is the number of vertices in the smallest separation set of (G,U). If {t} is a separationset of (G,U), t is called an articulation vertex. From theorem 3 and 4 follows that $((G,U))\omega = ((G,U))\tau$.
Clearly, $(G,U)\in K_p \rightarrow \forall_{x\in G}[x\gamma_U\geq p]$.

Topological interlude

π denotes a plane including the point ∞. $\Omega_r(x)$ is a neighbourhood of $x\in\pi$ in π with radius r. A set $X\subseteq\pi_1$ and a set $Y\subseteq\pi_2$ are called topologically equivalent if there exists a bijective mapping of X onto Y which is continuous and of which the inverse is also continuous. An open Jordan curve is a subset of $\pi\setminus\{\infty\}$ which is topologically equivalent to the closed interval of the real numbers between 0 and 1. A closed Jordan curve is a subset of $\pi\setminus\{\infty\}$ which is topologically equivalent to the set of complex numbers with modulus 1.
Theorem of Jordan-Schönflies: Every topological equivalence of a closed Jordan curve in π_1 onto a closed Jordan curve in π_2 can be extended to a topological equivalence of π_1 onto π_2.
Theorem of Jordan: A closed Jordan curve in π divides π into three disjoint subsets: the exterior containing ∞, the interior which is bounded, and the Jordan curve itself, which is the boundary of the two open subsets. Such an open subset together with the Jordan curve is topologically equivalent with a "closed circular disk". Corollary:

Every Jordan curve that has a point in common with the interior and the exterior of a closed Jordan curve, also has a point in common with this closed Jordan curve.

A plane graph (G^*,U^*) is a graph for which

(1) $g^*\epsilon G^* \rightarrow (g^*\epsilon\pi\wedge\exists_{\Omega_r(g^*)}\forall_{g_1^*\epsilon G^*\setminus\{g^*\}}[g_1^*\notin\Omega_r(g^*)])$

(2) $u^*\epsilon U^* \rightarrow u^*$ is an open Jordan curve in π with its associated vertices as images of 0 and 1

(3) beside its associated vertices u^* has no point in common with any other edge.

The subset of π consisting of the points in $G^*\cup\{x|\exists_{u^*\epsilon U^*}[x\epsilon u^*]\}$ is denoted by $S_{(G^*,U^*)}$. By induction on the number of edges it follows from Jordan's theorem that $S_{(G^*,U^*)}$ divides π in a finite number of regions and $S_{(G^*,U^*)}$. Exactly one of these regions contains ∞. These regions are called faces. The set of faces is denoted by W; $|W| = f$. The face containing ∞ is called the outer face and its boundary is the boundary of the graph. The points of $X\subset S_{(G^*,U^*)}$ are on the same face boundary fif there exists a point $x\epsilon\pi\setminus S_{(G^*,U^*)}$ and a set of open Jordan curves J for which $\{y|\exists_{j\epsilon J}[y\epsilon j]\}\cap S_{(G^*,U^*)} = X$ and $\bigcup_{j\epsilon J} j = x$. Clearly, the point x is contained in the concerned face.

Two plane graphs, (G_1^*,U_1^*) in π_1 and (G_2^*,U_2^*) in π_2, are said to be equivalent, if there exists a topological equivalence ϕ of π_1 onto π_2 mapping $S_{(G_1^*,U_1^*)}$ onto $S_{(G_2^*,U_2^*)}$ and the restriction of ϕ to G_1^* is an isomorphism of G_1^* onto G_2^*. Furthermore, ϕ maps face boundaries onto face boundaries; corresponding face boundaries are equivalent. Suppose we have a plane graph (G_1^*,U_1^*) in π_1 and a point p in the face w: there always exists a topological equivalence mapping π_1 onto π_2 in such a way that p is mapped onto ∞_2. Thus, with every plane graph there exists an equivalent plane graph with a chosen face boundary mapped onto the boundary of the latter graph. The face boundaries of the faces in a plane 2-connected graph are circuits, and a circuit in a plane graph is a closed Jordan curve. We write for the boundary of a face w in such a case: $(C_w^*,C_w^*[\ ])$.

P l a n a r i t y

A graph (G,U) is called a planar graph, if there exists a plane graph (G^*,U^*) which is isomorphic to (G,U). Such a plane graph is called a plane representation of (G,U). PL denotes the set of planar graphs.

$(G,U)\in PL$ fif $\forall_{(G',U')\lhd(G,U)}[(G',U')\in PL]$ fif $(\overline{G},\overline{U})\in PL$ fif

$\forall_{(G'',U'')\lhd(G'',U'')\in K_1(G,U)}[(G'',U'')\in PL]$ fif

$\forall_{(G''',U''')\lhd(G''',U''')\in K_2(G,U)}[(G''',U''')\in PL]$

Theorem 5: $|\{(G_1^*,U_1^*)\,|\,(G_1^*,U_1^*)\lhd(G_1^*,U_1^*)\in K_1(G^*,U^*)\}|=|G^*|-|U^*|+|W|-1$
(G^*,U^*) being a plane graph. (Euler's theorem)

Proof: The number of maximal, connected subgraphs be c. The theorem is true for m=1. Suppose it is true for all plane graphs with $m\le k$. (G^*,U^*) be a graph with $|U^*|= k+1$, and (G_1^*,U_1^*) is the graph $(G^*,U^*\setminus\{[x^*,y^*]\})$.

(1) Neither x^* nor y^* is an articulation vertex:
$f_1= f-1$, $n_1= n$, $m_1= m-1$, $c_1= c$.

(2) Either x^* or y^* is an articulation vertex:
$f_1= f-1$, $n_1= n$, $m_1= m-1$, $c_1= c$.

(3) Both x^* and y^* are articulation vertices:
$f_1= f$, $n_1= n$, $m_1= m-1$, $c_1= c+1$.

In all three cases follows from $c_1=n_1-m_1+f_1-1$, that $c= n-m+f-1$. Further, deletion of an isolated vertex $x(x\gamma_u=0)$ has no influence on the validity of the equality: at both sides the number is reduced by 1.

Theorem 6: a) $(\{x_i\,|\,1\le i\le 5\},\{[x_i,x_j]\,|\,1\le i<j\le 5\})\notin PL$

b) $(\{x_i\,|\,1\le i\le 6\},\{[x_i,x_j]\,|\,1\le i<4\le j\le 6\})\notin PL$

These two graphs are called the Kuratowski graphs, respectively KUR5 and KUR6.

Proof: a) Suppose the graph is planar. So there exists a plane representation (G^*,U^*) of it. For every plane graph in K_2 we have $\forall_{w\in W}[|c_w^*[\ ]|\ge 3]$ and $\sum_{w\in W}|c_w^*[\ ]|=2|U^*|$. Thus, $2m\ge 3f$.

Well, n=5, m=10, thus f must be 7, which means 2m=20 and 3f=21.

b) This graph is bipartite, which means that all its circuits have even lengths: $\forall_{w\in W}[|C^*_w[\ ]|\geq 4]$. Further, n=6, m=9, thus f must be 5; however, in that case 2m=18 and 4f=20.

Theorem 7: $(G,U)\in PL$. $(C,C[\ ])\lhd(G,U)$. There exists a plane representation of (G,U) such that the circuit corresponding with $(C,C[\ ])$ is a face boundary fif $A_{C[\ ]}=\emptyset$.

Proof: We prove the theorem for $(G,U)\in K_2$; extension to the other cases is trivial.

Suppose $A_{C[\ ]}\neq\emptyset$, i.e.
$\exists_{(G_1,U_1)\in BR_{C[\ ]}}\exists_{(G_2,U_2)\in BR_{C[\ ]}}[(G_1,U_1)@_{C[\ ]}(G_2,U_2)]$. Let $(P_1,P_1[a,b])\lhd_p(C,C[\ ])$, where p stands for "containing all the vertices of attachment of (G_1,U_1)". Likewise, $(P_2,P_2[c,d])$. $P_1[a,b]\cap P_2[c,d]\neq\emptyset$ and $C[\ ]\setminus(P_1[a,b]\cup P_2[c,d])\neq\emptyset$.
Let $[x,y]\in C[\ ]\setminus(P_1[a,b]\cup P_2[c,d])$. $z\notin G$
$(G',U')=(G\cup\{z\},(U\setminus\{[x,y]\})\cup\{[x,z],[z,y]\})$;
$(C',C'[\ ])=(C\cup\{z\},(C[\ ]\setminus\cup\{[x,y]\})\cup\{[x,z],[z,y]\})$
(Note: $(\overline{G}',\overline{U}')$ is isomorphic to $(\overline{G},\overline{U})$). We distinguish three cases

(1) a,b,c,d are four distinct vertices and $b\in P_2$, $c\in P_1$
Both $(P_1',P_1'[a,b])\lhd(G_1,U_1)$ and $(P_2',P_2'[c,d])\lhd(G_2,U_2)$ with $P_1'\cap C'=\{a,b\}$ and $P_2'\cap C'=\{c,d\}$ must exist.
$(G'',U'')=(C'\cup P_1'\cup P_2'\cup\{x'\},C'[\ ]\cup P_1'[a,b]\cup P_2'[c,d]\cup\cup\{[x',z],[x',b],[x',c]\})\lhd(G''',U''')=(G'\cup\{x'\},\ U'\cup\{[x',z],[x'b],[x',c]\})$. Since (G'',U'') contains a subgraph of which the simple graph is isomorphic to KUR6, (G''',U''') cannot be planar, and thus the desired plane representation does not exist. (Theorem 6)

(2) d=b and $c\in P_1$.
Since $(G_1,U_1)@_{C[\ ]}(G_2,U_2)$, there must be a vertex of

attachment of (G_1,U_1) in $P_2\backslash\{c,d\}$. This vertex adopts the role of b in case 1.

(3) a=c and d=b.

Since $(G_1,U_1)@_{C[\ ]}(G_2,U_2)$ there must be a vertex of attachment b' of (G_1,U_1) and a vertex of attachment c' of (G_2,U_2) in $P_1\backslash\{a,d\}$. In case $b'\neq c'$, let a,b',c',d play the roles of a,b,c,d in case 1. Remains the case b'=c': there must exist a vertex x_1 in $G_1\backslash\{a,d,b'\}$, such that there are three independent paths in $(G_1,U_1):(P_3,P_3[a,x_1]),(P_4,P_4[a,x_1])$ and $(P_5,P_5[b',x_1])$. Likewise, there is a vertex x_2 in $G_2\backslash\{a,d,b'\}$, such that there are three independent paths, $(P_6,P_6[a,x_2]),(P_7,P_7[d,x_2])$ and $(P_8,P_8[b',x_2])$. The union of these six paths, with $(C';C'[\ ]\cup\{z,b'\})$ is not planar, since it contains a subgraph of which the simple graph is isomorphic to KUR6. Again, the desired plane representation does not exist.

The other half of the theorem follows from an induction on the number of bridges k.For k=0 the result is obvious. Suppose it is true for circuits with fewer than k bridges, $|BR_{C[\ ]}|=k$, and (G_0^*,U_0^*) is a plane representation in π of $(G\rceil G_1,U\backslash U_1)$, (G_1,U_1) being a bridge of $(C,C[\ ])$, for which $(C,C[\ ])$ corresponds with a face boundary $(C^*_{w_1},C^*_{w_1}[\ ])$ in (G_0^*,U_0^*). Since (G_1,U_1) does not alternate with any other bridge, there is a path $(P,P[a,b])\lhd(C,C[\ ])$ containing no vertices of other bridges except possibly a and b and all vertices of attachment of (G_1,U_1). Thus $(P^*,P^*[a,b])$ is a part of another face boundary $(C^*_{w_2},C^*_{w_2}[\ ])$ beside $(C^*_{w_1},C^*_{w_1}[\ ])$. From the Jordan-Schönflies theorem it follows that a plane representation of $(G_1\cup C,\ U_1\cup C[\ ])$ can be mapped in $w_2\cup S_{(C^*_{w_2},C^*_{w_2}[\ ])}$ such that the corresponding path coincides with $(P^*,P^*[a,b])$.

Theorem 8: $(G,U)\in PL$ fif $\exists_{(C,C[\ ])\lhd(G,U)}[\forall_{(G_1,U_1)\in BR_{C[\ ]}}[(C\cup G_1,C[\ ]\cup U_1)\in PL \wedge (BR_{C[\ ]},A_{C[\ ]})$ is bipartite] fif

$\forall_{(C,C[\ ])\lhd(G,U)}[(BR_{C[\ ]},A_{C[\ ]})$ is bipartite] fif

$\neg\exists_{(G',U')\lhd(G,U)}[(\overline{G}',\overline{U}')$ is isomorphic to KUR5 or KUR6].

(Kuratowski's theorem)

Proof: $\exists_{(C,C[\ ])\lhd(G,U)}[\forall_{(G_1,U_1)\in BR_{C[\ ]}}[(CUG_1,C[\ ]UU_1)\in PL]\wedge$

$\wedge(BR_{C[\ ]},A_{C[\ ]})$ is bipartite] $\rightarrow$ $(G,U)\in PL$. Notation: $(\tilde{C},\tilde{C}[\ ])$. This will be proved by induction on the minimum number of bridges of a $(\tilde{C},\tilde{C}[])$. If (G,U) contains a circuit with one bridge (G_1,U_1), $(CUG_1,C[\ ]UU_1)=(G,U)$ and thus (G,U) must be planar. Suppose that any graph satisfying the condition and containing a $(\tilde{C},\tilde{C}[])$ with fewer than k bridges is planar. (G,U) be a graph satisfying the condition and containing such a $(\tilde{C},\tilde{C}[\ ])$ with k bridges and no $(\tilde{C},\tilde{C}[])$ with fewer than k bridges. Divide $BR_{\tilde{C}[\ ]}$ into two blocks EX and IN, such that no two bridges of the same block alternate with each other and no block is empty. By our hypothesis the union of $(\tilde{C},\tilde{C}[\ ])$ and the bridges in EX is planar and by theorem 7 there is a plane representation of this graph such that $(\tilde{C},\tilde{C}[\ ])$ is represented as a face boundary. Similarly, there is a representation of the union of $(\tilde{C},\tilde{C}[\ ])$ with the bridges of IN in the plane, such that $(\tilde{C},\tilde{C}[\ ])$ is mapped onto a face boundary. With the Jordan-Schönflies theorem we find a plane representation of (G,U), thus (G,U) is planar.

$\forall_{(C,C[\ ])\lhd(G,U)}[(BR_{C[\ ]},A_{C[\ ]})$ is bipartite] $\rightarrow$

$\rightarrow\exists_{(C,C[\ ])\lhd(G,U)}[\forall_{(G_1,U_1)\in BR_{C[\ ]}}[(CUG_1,C[\ ]UU_1)\in PL]]$. Suppose the conclusion holds for all concerned graphs with less than k edges. (G,U) be a graph with only bipartite alternation graphs and with k edges. Two cases:

(1) $\forall_{(C,C[\ ])\lhd(G,U)}[|BR_{C[\ ]}|=1]$. Suppose

$\neg\exists_{(C,C[\ ])\lhd(G,U)}[(G\setminus C,U\setminus C[\ ])$ is a tree], for otherwise the theorem is clearly true. Further, suppose $\{a,b\}\subseteq(G\setminus C)\cap C$, $a\neq b$ and $(P_1,P_1[a,b])\lhd(C,C[\ ])$ contains all vertices of attachment. Since bridges are connected there is a $(P_2,P_2[a,b])$ with $P_2\cap C=\{a,b\}$. The circuit

$(P_1 \cup P_2, P_1[a,b] \cup P_2[a,b])$ must have at least two bridges From this we conclude that $|Cn(G \setminus C)| = 1$. $(G \setminus C, U \setminus C[\ ])$ contains at least one circuit less and satisfies the condition of the theorem, so continuing in this way we must end with a circuit. Conclusion: (G,U) is planar and thus all its subgraphs are planar.

(2) $\exists_{(C,C[\ ]) \lhd (G,U)} [|BR_{C[\ ]}| \geq 2]$. If $(G_1,U_1) \in BR_{C[\ ]}$, then $(C \cup G_1, C[\ ] \cup U_1)$ has less than k edges and satisfies the condition of the theorem. Thus $(C \cup G_1, C[\ ] \cup U_1) \in PL$

$\neg \exists_{(G',U') \lhd (G,U)} [(\overline{G}', \overline{U}')$ is isomorphic to KUR5 or KUR6] $\rightarrow$

$\rightarrow \forall_{(C,C[\ ]) \lhd (G,U)} [(BR_{C[\ ]}, A_{C[\ ]})$ is bipartite]. As appeared from the proof of theorem 7 alternations can be of two kinds

1) $(G_1,U_1) @_{C[\ ]} (G_2,U_2)$ and $G_1 \cap C = G_2 \cap C$ and $|G_1 \cap C| = 3$

2) $(G_1,U_1) @_{C[\ ]} (G_2,U_2)$ and

$$\exists_{\{x_1,y_1\} \subseteq G_1 \cap C} \exists_{\{x_2,y_2\} \subseteq G_2 \cap C} [\{x_1,y_1\} \cap \{x_2,y_2\} = \emptyset \wedge$$

$$\wedge \neg \exists_{(P,P[x_1,y_1]) \lhd (C,C[\ ])} [\{x_2,y_2\} \cap P = \emptyset]]$$

Suppose $\exists_{(C,C[\ ]) \lhd (G,U)} [(BR_{C[\ ]}, A_{C[\ ]})$ is not bipartite], thus there is a circuit of odd length in $(BR_{C[\ ]}, A_{C[\ ]})$. One of three cases must be true (fig. A.1).

(1) $(BR_{C[\ ]}, A_{C[\ ]})$ contains a circuit of length 3 of which the edges stand for alternations of the first kind: there is a subgraph of (G,U) of which the simple graph is isomorphic to KUR6

(2) $(BR_{C[\ ]}, A_{C[\ ]})$ contains a circuit of length 3 of which the edges stand for an alternation of the first kind and two alternations of the second kind: again there is a subgraph of (G,U) whose simple graph is isomorphic to KUR6

(3) $(BR_{C[\ ]}, A_{C[\ ]})$ contains a circuit $(B_1, B_1[\ ])$ of odd length of which the edges stand for alternations of the second kind. Of course there is in such a case also a circuit $(B_2, B_2[\ ])$ of minimum odd length with $B_2 \subseteq B_1$. Consider the subgraph of (G,U) which is the union of

(C,C[ ]) and the bridges corresponding with the vertices of B_2: $(G',U') = (C \cup G_1 \cup G_2 \cup \ldots \cup G_k, C[\ ] \cup U_1 \cup U_2 \cup \ldots \cup U_k)$ with k odd. Suppose further

$(G_1,U_1)@_{C[\ ]}(G_2,U_2)@_{C[\ ]}\cdots@_{C[\ ]}(G_k,U_k)@_{C[\ ]}(G_1,U_1)$;

since $(B_2,B_2[\ ])$ is of minimum odd length with $B_2 \subseteq B_1$, these are the only alternations with respect to (C,C[ ]) in (G',U'). If $k \geq 5$, then let $(P,P[x_k,y_k]) \lhd (C,C[\ ])$ contain the vertices y_{k-1} of G_{k-1} and x_1 of G_1 and all the vertices of attachment of (G_k,U_k). $(P_k,P_k[x_k,y_k])$ is a path in (G_k,U_k) containing no other vertices of attachment than x_k and y_k. The circuit $(C',C'[\ ]) = ((C \setminus P) \cup P_k,\ C[\ ] \setminus P[x_k,y_k] \cup P_k[x_k,y_k])$ has an alternation graph $(BR_{C'[\ ]},A_{C'[\ ]})$ with a circuit of k-2 alternations in (G',U'). Thus there is always a subgraph (G",U") of (G,U) containing a circuit (C",C"[ ]) of which the alternation graph $(BR_{C"[\ ]},A_{C"[\ ]})$ with respect to (G",U") contains a circuit of length 3; all alternations in this circuit are of the second kind. Thus (G",U") consists of a circuit (C",C"[ ]) and three bridges: $(G_1",U_1")$, $(G_2",U_2")$ and $(G_3",U_3")$. $(P_1",P_1"[x_1,y_1])$ contains all the vertices of attachment in $G_1"$ and $(P_2",P_2"[x_2,y_2])$ all the vertices of attachment of $G_2"$. Further, $y_1 \in P_2"$ and $x_2 \in P_1"$. The circuit (C",C"[ ]) consists of four edge disjoint paths $(P_1,P_1[x_1,x_2])$, $(P_2,P_2[x_2,y_1])$, $(P_3,P_3[y_1,y_2])$ and $(P_4,P_4[y_2,x_1])$. In $(G_1",U_1")$ there is a path $(P_5,P_5[x_1,y_1])$ with $P_5 \cap C" = \{x_1,y_1\}$ and in $(G_2",U_2")$ there is a path $(P_6,P_6[x_2,y_2]$ with $P_6 \cap C" = \{x_2,y_2\}$. (K,V) be the graph $(\bigcup_{i \in I} P_i, \bigcup_{i \in I} P_i[.,.])$ with I a set of indices to be specified. (Vertices with different names are distinct). One of the following cases must apply (except for trivial exchanges of names of vertices):

(a) $\exists_{(P_7,P_7[a,b]) \lhd (G_3",U_3")} [a \in P_1 \wedge b \in P_3 \wedge P_7 \cap C" = \{a,b\}]$:

(K,V) with $I = \{1,2,3,4,5,6,7\}$ is isomorphic to KUR6

(b) $\exists_{(P_7,P_7[a,b]) \lhd (G_3",U_3")} \exists_{(P_8,P_8[c,y_1]) \lhd (G_3",U_3")}$
$[a \in P_1 \wedge b \in P_4 \wedge c \in P_7 \wedge P_7 \cap C" = \{a,b\} \wedge P_8 \cap C" = \{y_1\}]$:

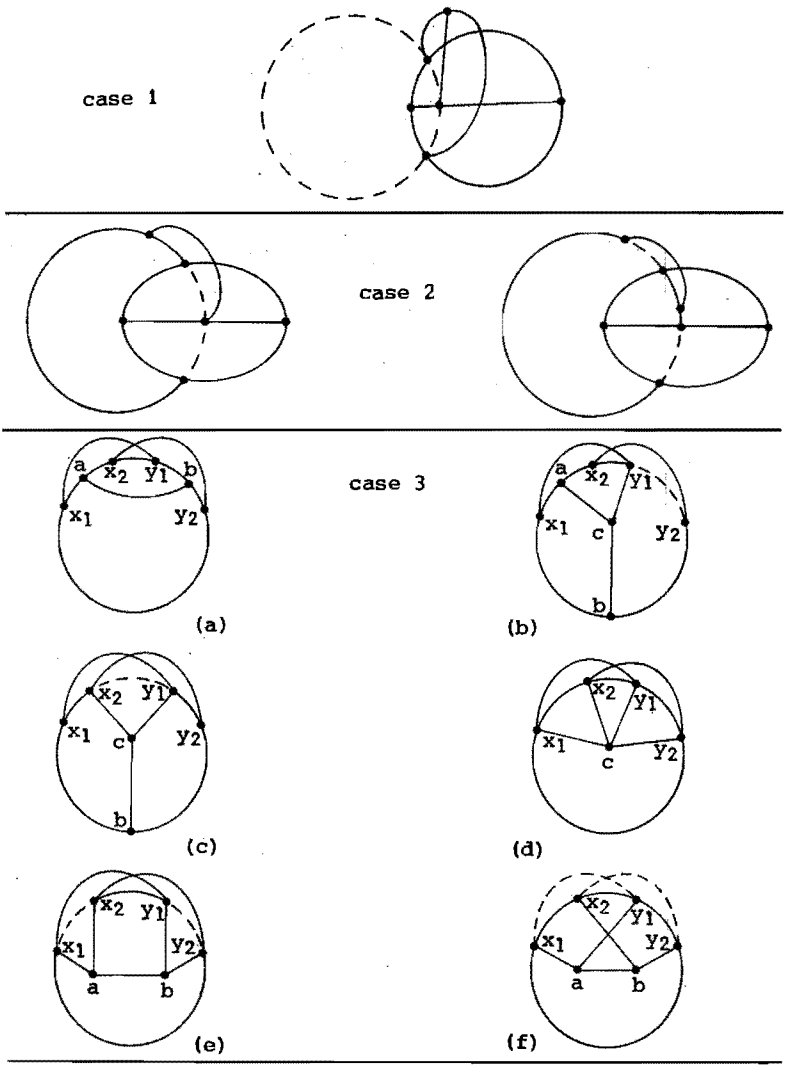


Fig. A.1. Theorem 8.

(K,V) with I={1,2,4,5,6,7,8} is isomorphic to KUR6

(c) $\exists_{(P_7,P_7[x_2,b])\lhd(G_3'',U_3'')}\exists_{(P_8,P_8[c,y_1]\lhd(G_3'',U_3'')}$
$[b\in P_4\wedge c\in P_7\wedge P_7\cap C''=\{x_2,b\}\wedge P_8\cap C''=\{y_1\}]$:
(K,V) with I={1,3,4,5,6,7,8} is isomorphic to KUR6

(d) $\exists_{(P_7,P_7[x_1,y_1])\lhd(G_3'',U_3'')}\exists_{(P_8,P_8[x_2,y_2])\lhd(G_3'',U_3'')}$
$[P_7\cap P_8=\{c\}\wedge P_7\cap C''=\{x_1,y_1\}\wedge P_8\cap C''=\{x_2,y_2\}]$:
(K,V) with I={1,2,3,4,5,6,7,8} is isomorphic to KUR5

(e) $\exists_{(P_7,P_7[x_1,x_2])\lhd(G_3'',U_3'')}\exists_{(P_8,P_8[y_1,y_2])\lhd(G_3'',U_3'')}$
$\exists_{(P_9,P_9[a,b])\lhd(G_3'',U_3'')}[a\in P_7\wedge b\in P_8\wedge P_7\cap C''=\{x_1,x_2\}\wedge$
$\wedge P_8\cap C''=\{y_1,y_2\}\wedge P_9\cap(C''\cup P_8\cup P_7)=\{a,b\}]$:
(K,V) with I={2,4,5,6,7,8,9} is isomorphic to KUR6

(f) $\exists_{(P_7,P_7[x_1,y_1])\lhd(G_3'',U_3'')}\exists_{(P_8,P_8[x_2,y_2])\lhd(G_3'',U_3'')}$
$\exists_{(P_9,P_9[a,b])\lhd(G_3'',U_3'')}[a\in P_7\wedge b\in P_8\wedge P_7\cap C''=\{x_1,y_1\}\wedge$
$\wedge P_8\cap C''=\{x_2,y_2\}\wedge P_9\cap(C''\cup P_7\cup P_8)=\{a,b\}]$:
(K,V) with I={1,2,3,4,7,8,9} is isomorphic to KUR6.

$(G,U)\in PL \rightarrow \neg\exists_{(G',U')\lhd(G,U)}[(\bar{G}',\bar{U}')$ is isomorphic to KUR5 or KUR6]. This follows trivially from theorem 6 and the fact that every subgraph of a planar graph is planar.

Theorem 9: A graph (G',U') obtained by contraction of edges of a planar graph (G,U) is planar.

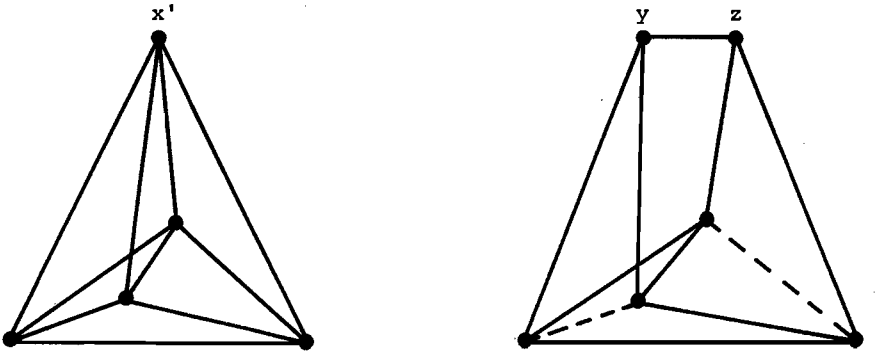


Fig. A.2. Theorem 9.

Proof: $(G',U') \notin PL \rightarrow \exists_{(H',V') \lhd (G',U')} [(\overline{H}',\overline{V}')$ is isomorphic to KUR5 or KUR6]. (H,V) is the corresponding subgraph in (G,U) with the not yet contracted edge $[y,z] \in V$; in (G',U') y and z are identified with $x' \in H'$. Thus $y\gamma_V + z\gamma_V = x'\gamma_{V'} + 2$. $(\overline{H},\overline{V})$ is not isomorphic to (H',V') only if $y\gamma_V > 2$ and $z\gamma_V > 2$; thus $(\overline{H}',\overline{V}')$ must be isomorphic to KUR5 and $y\gamma_V = 3$ and $z\gamma_V = 3$. But then $(\overline{H},\overline{V})$ contains a subgraph isomorphic to KUR6 which contradicts with the fact that (G,U) is planar (fig. A.2).

Theorem 10: Two face boundaries of a 3-connected plane graph have at most one edge in common.

Proof: Suppose that they have two edges in common: an end vertex of one of these edges and an end vertex of the other are clearly a separation set of the graph, for the Jordan curves connecting these vertices in the two faces enclose a subgraph of which the complement can only be attached to these two vertices.

Theorem 11: A planar simple graph (G,U) is not 3-connected fif in every plane representation of (G,U) there is at least one face boundary with more than one bridge.

Proof: Suppose that $\exists_{w \in W} [|BR_{C_w[\ ]}| \geq 2]$. Two bridges of $(C_w, C_w[\ ])$ cannot alternate. If (G_1,U_1) is such a bridge, then there is a path $(P,P[x,y])$ containing all vertices of attachment of (G_1,U_1), and - except possibly x and y -, no vertex of the other bridges. Clearly, {x,y} is a separator of (G,U).

If {x,y} is a separation set of (G,U), then $(G',U') = (G, U \cup \{[x,y]\})$ is still planar and separated by {x,y}. In a plane representation of (G',U') x^* and y^* are connected by Jordan curves with a point on the Jordan curve representing [x,y]. Thus, after deletion of $[x^*,y^*]$ we have a plane representation of (G,U) with x^* and y^* at the same face boundary. Since (G,U) is simple this face boundary has at least two bridges.

Corollary 1: (G,U) is a simple graph with a separation set $\{x,y\}$, separating (G,U) in (G_1,U_1) and (G_2,U_2).
$(G,U)\in PL \leftrightarrow (G_1,U_1\cup\{[x,y]\})\in PL \wedge (G_2,U_2\cup\{[x,y]\})\in PL$.

A graph (G,U) is said to have a unique plane representation if the set of subgraphs of (G,U) that are mapped onto the face boundaries is the same for every plane representation of (G,U).

Theorem 12: A 2-connected graph (G,U) has a unique plane representation fif $(\overline{G},\overline{U})$ is 3-connected.

Proof: $(\overline{G},\overline{U})$ is not 3-connected; then there is a $(C_w,C_w[\ ])$ in (G^*,U^*) with more than one bridge. By transferring one of them into the face w, we have obtained another plane representation in which the face boundaries represent other circuits of (G,U).

Suppose that $(C,C[\ ])\lhd(G,U)$ is mapped onto the face boundary of w in (G^*,U^*) and it does not correspond with a face boundary in (G^{**},U^{**}), another plane representation of (G,U). There is at least one bridge in the closure of the interior of $S_{(C^{**},C^{**}[\ ])}$ and one bridge in the closure of its exterior; thus $(C,C[\ ])$ has two bridges in (G,U) and (G^*,U^*) is a plane representation with a face boundary with two bridges. By theorem 11 $(\overline{G},\overline{U})$ is not 3-connected.

In a plane graph (G^*,U^*) a symmetric relation on $x^*\rho^*$ can be recognized: two vertices, y_1^* and y_2^*, in $x^*\rho^*$ are called wheel-consecutive if there exists a closed Jordan curve C containing all the vertices of $x^*\rho^*$ and no other point of $S_{(G^*,U^*)}$ and there is an open Jordan curve in C containing y_1^* and y_2^* and no other point of $x^*\rho^*$. (G^*,U^*) is called H^*-periphere with $H^*\subset G^*$ if the vertices of H^* are all on the face boundary of some face w of (G^*,U^*). If (G^*,U^*) is 2-connected one can also recognize a symmetric relation on H^*; two vertices, h_1^* and h_2^*, of H^* are called face-consecutive if there exists a closed Jordan curve C containing all points of H^* and no other point of $S_{(G^*,U^*)}$ and there is an open Jordan curve in C containing h_1^* and h_2^* and no other point

of H^*.

Remark: If the plane graph is not 2-connected, face consecutivity is not necessarily unique. Nevertheless we speak of face consecutivity if we can add some Jordan curves to the graph such that the concerned relation becomes unique. If a graph has a unique plane representation all wheel consecutivities and face consecutivities are fixed of course.

A graph (G,U) is called H-accessible, $H \subseteq G$ if it has an H^*-periphere plane representation.

B. Some planarity testing algorithms.

In section 4.3. we have given a short review on methods that test the planarity of a graph. Some important algorithms are in a stepwise manner described in this appendix. The algorithms have to be applied on a 2-connected graph (G,U) unless mentioned otherwise.

The "pseudo-hamilton"-planarity testing algorithm (Auslander and Parter [B.1]).

This method is based on theorem 8 of appendix A. The algorithm consists of the following steps:

1. $i:=k:=0$; $G_o:=G$; $U_o:=U$.
2. Search for a circuit $(C_i,C_i[])$ in (G_i,U_i) .
3. Construct the set of bridges of $(G_i \setminus C_i, U_i \setminus C_i[])$, denoted $BR^i_{C_i[]}$.
4. Test whether the alternation graph $(BR^i_{C_i[]},A_{C_i[]})$ is bipartite. If this is not the case then goto step 9.
5. For each bridge $(G_b,U_b) \in BR^i_{C_i[]}$ that is not a path, we store a new graph, and a circuit of this graph:

   $k:=k+1$; $G_k:=C_i \cup G_b$; $U_k:=C_i[] \cup U_b$

   The circuit $(C_k,C_k[]) \lhd (G_k,U_k)$ is not taken identical to $(C_i,C_i[])$. It is obtained by replacing a path of $(C_i,C_i[])$ between two subsequent attachment vertices x and y by a path $(P,P[x,y])$ in (G_b,U_b). It can be advantageous for the convergence of the algorithm to take the path $(P,P[x,y])$ as long as possible.
6. $i:=i+1$.
7. if $i \leq k$ then goto step 3.
8. The graph (G,U) is planar, and the algorithm terminates.
9. The graph (G,U) is not planar, and the algorithm terminates.

The test on bipartiteness of a graph (G_a,U_a) is performed by the steps:

1. $G_I=G_{II}=\emptyset$.
2. Take one arbitrary vertex of the graph $(g \in G_a)$, and assign it to G_I: $G_I:=\{g\}$.

3. $\forall_{g\in G_I}[[g,g']\in U_a \wedge g'\notin G_I \cup G_{II} \rightarrow g':\in G_{II}]$

4. $\forall_{g\in G_{II}}[[g,g']\in U_a \wedge g'\notin G_I \cup G_{II} \rightarrow g':\in G_I]$

5. if $\exists_{g\in G_a}[g\notin G_I \cup G_{II}]$ then goto step 3

6. if $\exists_{[g,g']\in U_a}[(g\in G_I \wedge g'\in G_I)\vee(g\in G_{II}\wedge g'\in G_{II})]$ then "(G_a,U_a) is not bipartite" else "(G_a,U_a) is bipartite".

The planarity algorithm of Yoshida and Ohta [B.2].

This algorithm tests the planarity of a so-called "whirl". A whirl consists of:
- a set of vertices
- a cycle of arcs (the outer cycle) which has to form the boundary of the outer face
- a set of edges that all have to be embedded in the inside region of the outer cycle.

The whirls are partitioned in subwhirls until only digraphs are left that are a cycle. The circuits that can be obtained from these cycles. (by replacing each arc by an edge) form the face boundaries of the faces of the plane representation that is constructed.
Let us assume that in the graph (G,U) there is a circuit $(C_1,C_1[])$, and that its complement $(G_1,U_1)=(G\urcorner C_1, U\backslash C_1[])$ has to be embedded in a planar way in the inner region determined by the representation of this circuit.
The initial whirl is constructed by orienting the edges of the circuit.
The algorithm consists of the following steps:

1. $k:=i:=1$.

2. Find a path $(P_i,P_i[x,y])$ in (G_i,U_i) between two vertices x and y of the cycle $(C_i,C_i[>)$.
   The cycle $(C_i,C_i[>)$ is divided by x and y in two chains $(C_{i_1},C_{i_1}[x,y>)$ and $(C_{i_2},C_{i_2}[y,x>)$.

3. Determine the set of bridges of $((G_i\cup C_i)\urcorner P_i,(U_i\cup C_i[])\backslash P_i[x,y])$, denoted $BR^i_{P_i[x,y]}=\{B^i_j \mid 1\leq j\leq p\}$, where $B^i_j=(G^i_j,U^i_j)$.

If $(G_i,U_i)=(P_i,P_i[x,y])$ then store the two circuits (face boundaries): $(C_{i_1} \cup P_i,\ C_{i_1}[x,y] \cup P_i[x,y])$ and $(C_{i_2} \cup P_i, C_{i_2}[x,y] \cup P_i[x,y])$.

4. If there is some bridge that contains vertices of both $C_{i_1} \setminus \{x,y\}$ and $C_{i_2} \setminus \{x,y\}$ then goto step 10.

5. Try to construct two sets BR_I^i and BR_{II}^i of mutually non-alternating bridges such that the bridge $B_1^i \in BR_I^i$, with $(C_{i_1}, C_{i_1}[x,y]) \lhd B_1^i$ and furthermore the bridge $B_2^i \in BR_{II}^i$ with $(C_{i_2}, C_{i_2}[x,y]) \lhd B_2^i$. This is accomplished by testing the extended alternation graph $(BR^i_{P_i}[x,y], A^i_{P_i}[x,y] \cup \{[B_1^i, B_2^i]\})$ on bipartiteness. If this graph is not bipartite, then goto step 10.

6. Construct two new whirls:

   a) replace all edges of $P_i[x,y]$ by two parallel arcs, having opposite orientations

   b) now, two cycles have been created:

   $(C_{k+1}, C_{k+1}[>) = (C_{i_1} \cup P_i, C_{i_1}[x,y> \cup P_i[y,x>)$ and

   $(C_{k+2}, C_{k+2}[>) = (C_{i_2} \cup P_i, C_{i_2}[y,x> \cup P_i[x,y>)$

   c) the graphs (G_{k+1}, U_{k+1}) and (G_{k+2}, U_{k+2}) are defined by:

$$G_{k+1} = \{g \mid g \in G_j^i \wedge (G_j^i, U_j^i) \in BR_I^i\}$$
$$U_{k+1} = \{u \mid u \in U_j^i \wedge (G_j^i, U_j^i) \in BR_I^i\}$$
$$G_{k+2} = \{g \mid g \in G_j^i \wedge (G_j^i, U_j^i) \in BR_{II}^i\}$$
$$U_{k+2} = \{u \mid u \in U_j^i \wedge (G_j^i, U_j^i) \in BR_{II}^i\}$$

   d) k:=k+2.

7. i:=i+1.

8. If i≤k then return to step 2.

9. The graph (G_1,U_1) can be planely embedded in the inner region determined by $(C_1,C_1[])$. The faces of a plane representation are stored in step 3. The algorithm terminates.

10. The graph (G_1,U_1) cannot be planely embedded in the inner region determined by $(C_1,C_1[])$. The algorithm terminates.

Remarks:

1. If the whirl is planar, the number of partitions of a whirl in subwhirls is equal to the number of edges minus the number of vertices.
2. If one wants to test the planarity of a graph instead of a whirl, one has to start with replacing an arbitrary edge by two parallel arcs with opposite orientations.

The planarity testing algorithm of Goldstein [B.3].

The following algorithm constructs a plane representation of (G,U), (if existing). In each $(i+1)$-th step of the algorithm, the currently constructed plane subrepresentation (G_i^*,U_i^*) is extended: $(G_i,U_i) \lhd (G_{i+1},U_{i+1})$. The set of faces of (G_i^*,U_i^*) is denoted W_i. The following steps are distinguished:

1. $i:=1$;
   Search for a circuit $(C,C[])$ in (G,U), and embed this circuit in the plane. Thus, we have created $(G_1^*,U_1^*)=(C^*,C^*[])$.
2. Construct the set of bridges of $(G \rceil G_i, U \backslash U_i)$, denoted BR_{U_i}
   If $BR_{U_i}=\emptyset$ then goto step 10.
3. A bridge $(G_b,U_b) \in BR_{U_i}$ is "embeddable" if for its set of attachment vertices holds:
   $$\exists_{w \in W_i} [G_b \cap (G \rceil G_b) \subset C_w]$$
   Determine for each bridge its "embedding number", i.e. the number of faces in which it is embeddable.
4. If there is a bridge $BR_j^i \in BR_{U_i}$ with embedding number 0 then goto step 11.
5. If there is a bridge $BR_j^i \in BR_{U_i}$ with embedding number 1 then goto step 7.
6. Select some arbitrary bridge BR_j^i (having embedding number ≥ 2).
7. Choose an arbitrary path $(P_i,P_i[x,y])$ in the bridge BR_j^i (where x and y are two attachment vertices of BR_j^i), and embed this path in a face w in which BR_j^i is embeddable.
   $G_{i+1}:=G_i \cup P_i$; $U_{i+1}:=U_i \cup P_i[x,y]$.

8. i:=i+1.
9. Return to step 2.
10. The graph (G,U) is planar and the algorithm terminates. (G_i^*,U_i^*) is a plane representation of (G,U).
11. The graph (G,U) is not planar. The algorithm terminates.

If the extensions of the representation are performed with first priority for the bridges that consist of a single edge, the number of searches in step 7 tends to be reduced [B.4].

The planarity testing algorithm of Hotz [B.5].

This algorithm is only suitable to test the planarity of a 3-connected graph (G,U).

It consists of the following steps:

1. i:=0
2. Construct the plane representation (G_1^*,U_1^*) of the initial graph (G_1,U_1), which is the simplest 3-connected graph, namely the wheel having three vertices on the rim, and one vertex as hub. Let $G_1 \subset G$.
3. i:=i+1
4. Let the set of faces of (G_i^*,U_i^*) be denoted W_i

   If $\exists_{x\in G_i}\exists_{y\in G_i}[x\neq y \wedge \exists!_{w\in W_i}[x\in C_w \wedge y\in C_w]\wedge$

   $$\wedge \exists_{(P,P[x,y])\lhd(G,U)}[P\subset(G\setminus G_i)\cup\{x,y\}]]$$

   then $(G_{i+1},U_{i+1}):=(G_i\cup P, U_i\cup P[x,y])$ and return to step 3.

   (The plane representation (G_{i+1}^*,U_{i+1}^*) is obtained by embedding $(P,P[x,y])$ in the face w of (G_i^*,U_i^*)).
5. If $\exists_{[x,y]\in U_i}\exists_{(P,P[x,y])\lhd(G,U)}[P\subset(G\setminus G_i)\cup\{x,y\}\wedge P[x,y]\neq\{[x,y]\}]$

   then $(G_{i+1},U_{i+1}):=(G_i\cup P,(U_i\cup P[x,y])\setminus\{[x,y]\})$ and return to step 3.

   (The plane representation (G_{i+1}^*,U_{i+1}^*) is obtained by replacing $[x,y]$ in (G_i^*,U_i^*) by $(P,P[x,y])$).
6. If $U\setminus U_i\neq\emptyset$ then goto step 8.
7. The graph (G,U) is planar, and the algorithm terminates. (G_i^*,U_i^*) is the plane representation of (G,U).
8. The graph (G,U) is not planar, and the algorithm terminates.

The planarity testing algorithm of Klemm [B.6].

If the graph (G,U) is planar, the following algorithm constructs a plane representation of it, in which the edge $[s,t] \in U$ is on the face boundary of the outer face.

1. Construct a drain of the graph $(G, U \setminus \{[s,t]\})$ with vertex s being the source and vertex t being the sink. This drain is denoted (G,V).
2. Determine for each vertex v the "chain length" $v\delta$, i.e. the length of the longest chain $(C, C[s,v>)$ in (G,V).
3. The "chain length difference" of an arc $[x,y>$ of the drain, is defined as $y\delta - x\delta$. For each arc $[x,y>$ having chain length difference d, add d-1 new vertices $v_1, \ldots, v_{d-1}$, and replace $[x,y>$ by the chain containing the arcs $[x,v_1>$, $[v_1,v_2>, \ldots, [v_{d-1},y>$
   By doing this, it is achieved that all chains between s and an arbitrary vertex v have equal length. The created extended drain is denoted (G_d, V_d).
4. Partition the set of vertices of (G_d, V_d) in subsets of vertices that have equal chain length (these subsets are called "classes"):
   $G_d = G_o \cup G_1 \cup \ldots\ldots \cup G_k$ with $k = |t\delta|$ and
   $G_i = \{g \mid g \in G_d \wedge g\delta = i\}$ $\quad (0 \leq i \leq k)$
   The drain is partitioned in k "segments" (S_i, V_i), where $(1 \leq i \leq k)$ and $S_i = G_{i-1} \cup G_i$ and $V_i = \{[x,y> \mid [x,y> \in V_d \wedge x \in G_{i-1} \wedge y \in G_i\}$.
5. i:=0; embed $G_o = \{s\}$ on the straight line α_o.
6. i:=i+1.
7. Determine the "sequence formula" for G_i. This formula is built up in the following way:
   To every vertex in G_i a "start label" is assigned ("colours" in [B.6]). This label is transported to the next class via the arcs. To the vertices of this next class we assign the "product" of the labels that are transported to it. These labels are called the "factors" of the product. The product forms a new label and is put between brackets. The "rank" of a label (product) that is associated with a vertex $v \in G_j$ is equal to j.
   The transportation process is continued until the sink has been reached. The label (formula) that is associated with the sink is reduced by the following three rules:
   a) The factors of a product can be permutated.

b) Identical factors Φ_i having equal rank and being neighbours in a product, can be contracted to one factor Φ_i.

c) Contraction of identical factors Φ_i that are in different products Φ_I and Φ_{II} of equal rank, and that have only brackets in between, restrict the commutativity of the created products. The obligatory neighbourships are denoted by the relation N:

$$\underbrace{(\Phi_a \cdots \Phi_b\ \Phi_i)}_{\Phi_I}\underbrace{(\Phi_i\ \Phi_x \cdots \Phi_y)}_{\Phi_{II}} \rightarrow (\Phi_a \cdots \Phi_b) N \Phi_i N (\Phi_x \cdots \Phi_y)$$

If the formula cannot be reduced until it contains only factors that exist of the original start labels (assigned to the vertices of G_i), then goto step 11. Otherwise a plane representation of the i-th segment can be obtained from the obligatory neighbourships in the reduced formula, and the already determined neighbourships of the vertices in G_{i-1}.
The vertices of G_i are embedded on the straight line α_i, which is positioned in the right half plane of α_{i-1}. The edges are represented by straight lines.

8. If i<k then return to step 6.
9. Embed edge [s,t> in the outer face.
10. The graph (G,U) is embedded in a planar way, and the algorithm terminates.
11. The graph (G,U) cannot be embedded in a planar way, and the algorithm terminates.

References.

[B.1] L.Auslander and S.V.Parter, "On embedding graphs in the sphere", J. of Math. & Mech., Vol. 10, nr. 3, pp. 517-523, 1961.

[B.2] K.Yoshida and T.Ohta, "Topological layout of a monolithic IC", Electronics and Communications in Japan, Vol. 52-C, No. 12, pp. 173-178, 1969.

[B.3] A.J.Goldstein, "An efficient and constructive algorithm for testing whether a graph can be imbedded in a plane", Graph and Combinatorics Conf., Office of Naval Res. Logistics Proj., Contract Nr. NDNR 1858-(21), Dept. of Math., Princeton Univ., 1963.

[B.4] R.W.Shirey, "Implementation and analysis of efficient planarity testing algorithms", Ph.D.-dissertation, Computer Science Department, Univ. of Wisconsin, 1969.

[B.5] G.Hotz, "Einbettung von Streckenkomplexen in die Ebene", Math. Annalen, Vol. 167, pp. 460-472, 1966.

[B.6] U.Klemm, "Ein algorithmisches Planaritätskriterium", Computing 3, pp. 194-204 and 245-257, 1968.

C. Searching optimal sequences.

MODEL

R_+ := the set of non-negative real numbers

C := a finite set, the set of cells

S := a set, the cell-alphabet

η := the neighbour relation

$$\eta \subset C \times C \wedge \forall_{(c,c') \in \eta} [c \neq c' \wedge (c',c) \in \eta]$$

σ := labeling relation

$$\sigma \subset C \times S \wedge \forall_{c \in C} \exists!_{s \in S} [(c,s) \in \sigma]$$

μ := weighing relation

$$\mu \subset R_+ \times S \times R_+ \wedge$$

$$\forall_{(a,s) \in R_+ \times S} \exists!_{b \in R_+} [(a,s,b) \in \mu] \wedge$$

$$\forall_{(a,s,b) \in \mu} [a \leq b]$$

C_o := set of origins

$$C_o \subset C$$

C_t := set of targets

$$C_t \subset C \setminus C_o$$

$$P := \{(c_1,c_2,\ldots,c_k) \mid (c_1,c_2,\ldots,c_k) \in \bigcup_{i=1}^{|C|} C^i \wedge$$

$$\forall_{1 \leq i < k} [(c_i,c_{i+1}) \in \eta \wedge \forall_{1 < j \leq k} [i \neq j \leftrightarrow c_i \neq c_j]]\}$$

ϕ :$\subset P \times R_+$

$$\forall_{c \in C} [((c),a) \in \phi \rightarrow a = 0]$$

$$\forall_{(c_1,c_2,\ldots,c_k) \in P} [((c_1,c_2,\ldots,c_k),b) \in \phi \leftrightarrow$$

$$\leftrightarrow ((c_1,c_2,\ldots,c_{k-1})\phi, c_k\sigma, b) \in \mu]$$

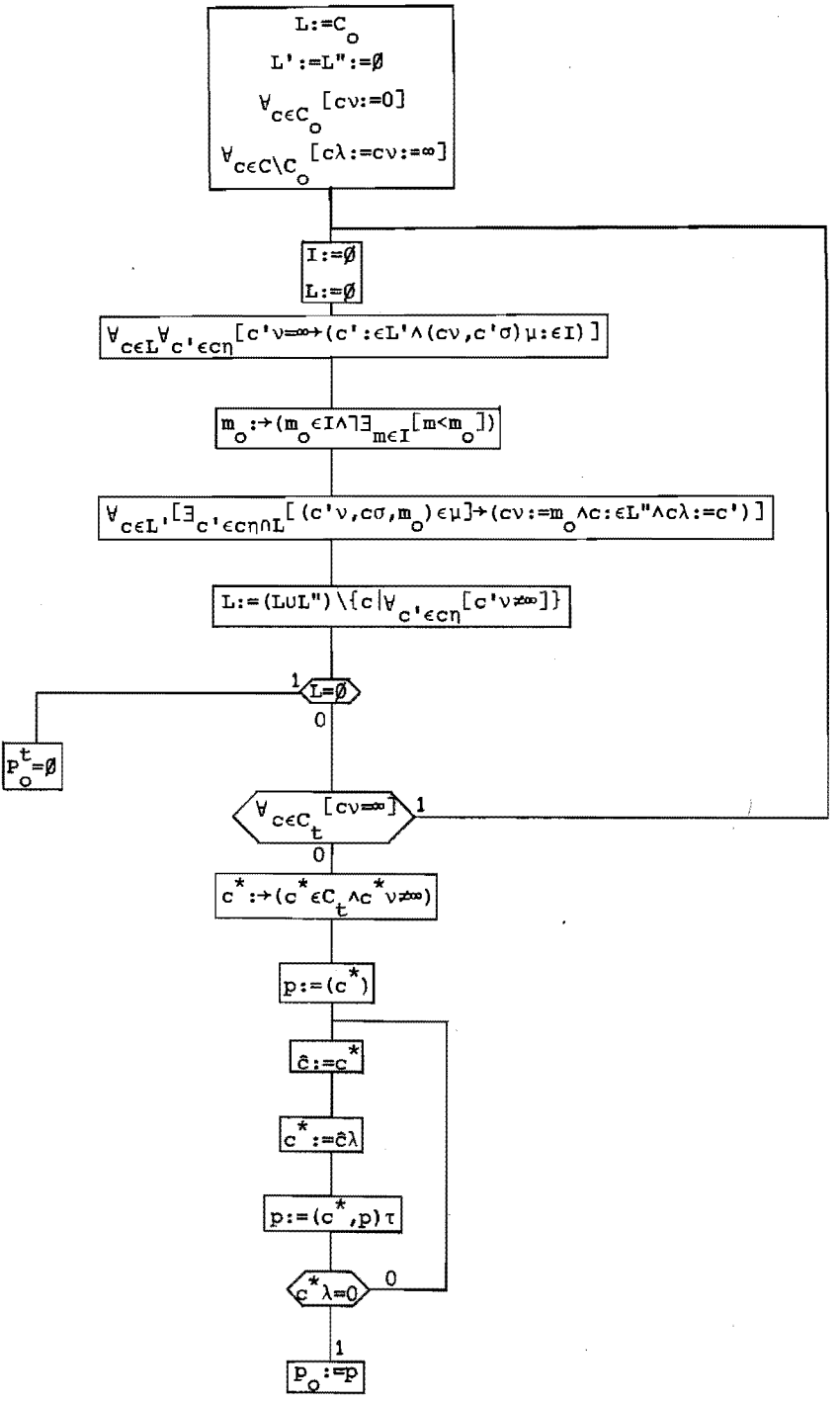
$$P_o^t := \{p \mid p \in P \cap C^k \wedge p\pi_1 \in C_o \wedge p\pi_k \in C_t\}$$

τ :$\subset C \times P \times P$

$$\forall_{(c,p,p') \in \tau} [p = (c_1,c_2,\ldots,c_k) \rightarrow$$

$$\rightarrow (c \in c_1\eta \setminus \{c_i \mid 1 \leq i \leq k\} \wedge p' = (c,c_1,c_2,\ldots,c_k))]$$

$$p_o :\rightarrow (p_o \in P_o^t \wedge \forall_{p \in P_o^t} [p_o\phi \leq p\phi])$$


L:=C_o
L':=L":=∅
∀_{c∈C_o} [cν:=0]
∀_{c∈C\C_o} [cλ:=cν:=∞]
I:=∅
L:=∅
∀_{c∈L}∀_{c'∈cη} [c'ν=∞→(c':∈L'∧(cν,c'σ)μ:∈I)]
m_o:→(m_o∈I∧∃_{m∈I}[m<m_o])
∀_{c∈L'}[∃_{c'∈cη∩L}[(c'ν,cσ,m_o)∈μ]→(cν:=m_o∧c:∈L"∧cλ:=c')]
L:=(L∪L")\{c|∀_{c'∈cη}[c'ν≠∞]}
L=∅
1
0
p_o^t=∅
∀_{c∈C_t} [cν=∞]
1
0
c*:→(c*∈C_t∧c*ν≠∞)
p:=(c*)
ĉ:=c*
c*:=ĉλ
p:=(c*,p)τ
c*λ=0
0
1
p_o:=p

D. A theory of draingraphs.

A multigraph consists of a finite set of vertices and a finite family of unordered pairs of distinct vertices, the edges. Notation: (G,F), where G is the set of vertices and F the family of edges. If two edges are associated with the same pair of vertices we call them parallel edges. The simple graph $(\overline{G},\overline{F})$ of a multigraph (G,F) is the simple graph of the graph obtained by identifying parallel edges.

A digraph consists of a finite set of vertices and a finite family of ordered pairs of distinct vertices, the arcs. Notation: (G,V), where G is the set of vertices and V the family of arcs. An arc from x to y is denoted by $[x,y\rangle$. The multigraph of a digraph is the multigraph obtained by replacing each arc $[x,y\rangle$ by an edge $[x,y]$.

Remark: By introducing families, denoting an edge or an arc by its vertices is no longer unique; since there may be distinct edges or arcs associated with the same pair of vertices. Furthermore, set-theoretical notations are no longer unambiguous. Nevertheless, we use them without explicit definitions, but we add an $^\circ$ to indicate that one should remember that we are dealing with families.

A binary multirelation over a set G is a family of which the elements are in the cartesian product G×G. Both, a multigraph and a digraph define an antireflexive multirelation Γ over the set of their vertices. In case of a multigraph Γ is symmetric. The degree of a vertex x is defined as $|x\Gamma|^\circ$. Notation: $x\gamma_F$. For a digraph the outdegree of a vertex x is defined as $|x\Gamma|^\circ$ and the indegree as $|x\Gamma^{-1}|^\circ$, denoted as $x\gamma_V^+$ and $x\gamma_V^-$ respectively. A vertex x with $x\gamma^-=0$ is called a source and a vertex x with $x\gamma^+=0$ is called a sink.

Remark: Many notions defined in appendix A for graphs are transferred, sometimes in an adapted form, into this appendix, often without an explicit definition, for example subdigraph, submultigraph, path and circuit in a multigraph, etc..

A chain in a digraph (G,V) is a subdigraph consisting of a sequence of arcs $[x_1,x_2>,[x_2,x_3>\ldots[x_{k-1},x_k>$ and the vertices $x_1,x_2,\ldots,x_k$ which are distinct. Notation: $(C,C[x_1,x_k>)$. A cycle in a digraph (G,V) is a subdigraph $(C,C[x,y>\cup\{[y,x>\})$, denoted by $(C,C[>)$. A digraph is acyclic if it contains no cycles. An acyclic digraph with exactly one source s and one sink t is called a drain.

Suppose we have a multigraph (G,F) with $|G|=n$. Of course we can define a bijective mapping ϕ from G onto the set of positive integers smaller than n+1. In such a case the ϕ-oriented digraph (G,V) of (G,F) is defined by $\forall_{x\in G}\forall_{y\in G}[([x,y]\in{}^{\circ}F\wedge x\phi<y\phi)\leftrightarrow[x,y>\in{}^{\circ}V]$. ϕ is called a drain function of (G,F) if $\forall_{x\in G}[(x\phi\neq 1\wedge x\phi\neq n)\leftrightarrow\exists_{[x,y]\in{}^{\circ}F}\exists_{[x,z]\in{}^{\circ}F}[y\phi>x\phi>z\phi]]$. Clearly, if ϕ is a drain function of (G,F), the ϕ-oriented digraph of (G,F) is a drain.

Theorem 1: A multigraph (G,F) has a drain function with a given source s and a given sink t fif $(\neg\exists_{x\in G\setminus\{s,t\}}[x\gamma_F=1]\wedge$
$\wedge(G,U)=(\overline{G},\overline{F\cup{}^{\circ}\{[s,t]\}})\in K_2)\vee\ |\overline{F}|=1$.

Proof: If we have a drain function for (G,U), it is easy to obtain a drain function for (G,F).

Suppose (G,U) is not 2-connected, thus (G,U) has an articulation vertex a. We distinguish two cases:

a) $a=s$: $G_1=\{x|\forall_{(P,P[x,t])\triangleleft(G,U)}[a\in P]\}\rightarrow\exists!_{y\in G_1}\forall_{x\in G_1}[y\phi\geq x\phi]$
Clearly, $\forall_{b\in G}[[b,y]\in U\rightarrow b\in G_1]$, thus $\neg\exists_{z\in G}[[y,z]\in U\wedge z\phi>y\phi]$, which means that ϕ is not a drain function of (G,U).

b) $a\neq s$: $G_1=\{x|\forall_{(P,P[x,s])\triangleleft(G,U)}[a\in P]\}\rightarrow$
$\rightarrow\exists!_{x\in G_1}\exists!_{y\in G_1}\forall_{w\in G_1}[x\phi\leq w\phi\wedge y\phi\geq w\phi]$. Now if $x=a$ then $\neg\exists_{z\in G}[[y,z]\in U\wedge z\phi>y\phi]$, and if $x\neq a$, then $\neg\exists_{z\in G}[[x,z]\in U\wedge$
$\wedge x\phi>z\phi]$, which also means that ϕ is not a drain function of (G,U).

The graph $(\{s,t\},[s,t])$ is correctly numbered by $s\phi=1$ and $t\phi=2$. Suppose $(G',U')\triangleleft(G,U)$ is correctly numbered (one source,

one sink and of course no cycles). Suppose $x \in G'$ and $y \in G \setminus G'$. If (G,U) is 2-connected, there is a circuit (C,C[ ]) containing x and y. There is a path (P,P[a,b]) in (C,C[ ]) with all its edges in $U \setminus U'$ and only its end vertices a and b in G'. $a\phi' < b\phi'$. The subgraph $(G' \cup P, U' \cup P[a,b])$ can be correctly numbered by maintaining all numbers less than $b\phi'$, number the vertices of $P \setminus \{a,b\}$ with $b\phi'$ up to $b\phi' + |P| - 3$. And increase the other numbers by $|P| - 2$.

Theorem 2: If the digraph (G,V) is acyclic, then $\hat{\Gamma}$ is a partial ordering of G, $\hat{\Gamma}$ being the transitive closure of Γ ($\hat{\Gamma} = \Gamma \cup \Gamma^2 \cup \Gamma^3 \cup \ldots \cup \Gamma^n$).

Proof: Since (G,V) is acyclic, $\hat{\Gamma}$ is antireflexive: $(x,x) \notin \hat{\Gamma}$. Further, $(x,y) \in \hat{\Gamma} \rightarrow \exists_{1 \le i \le n} [(x,y) \in \Gamma^i]$ and $(y,z) \in \hat{\Gamma} \rightarrow$ $\rightarrow \exists_{1 \le j \le n} [(y,z) \in \Gamma^j]$, then $(x,z) \in \Gamma^{i+j}$. This means that $(x,z) \in \hat{\Gamma}$. In other words $\hat{\Gamma}$ is transitive, and antireflexive, and thus $\hat{\Gamma}$ is a partial ordering.

Theorem 3: If the digraph (G,V) is a drain, then G has exactly one first element and exactly one last element by $\hat{\Gamma}$.

Proof: Take an arbitrary $x \in G$. x is contained in at least one chain, and thus there exists a maximal chain $(C, C[a,b>)$ containing x. From the maximality it follows that $\neg\exists_{y \in G}[[y,a> \in V]$, and $\neg\exists_{z \in G}[[b,z> \in V]$, thus a must be the only source s of (G,V) and b the only sink t in (G,V) and these are the first and the last element of G by $\hat{\Gamma}$ respectively.

Suppose we have a drain (G,V). (G,F) is the multigraph of (G,V). A drain function ϕ of (G,F) is called a natural drain function of (G,V) if the ϕ-oriented digraph of (G,F) is isomorphic to (G,V).

Theorem 4: Every drain has at least one natural drain function.

Proof: We describe a procedure to find a natural drain function. Assign the number "1" to the source of the drain (G,V). After deletion of s and the arcs associated with s, we have

a digraph. From theorem 2 and Zorn's lemma it follows that there is at least one source in this digraph. One of the sources gets the next number, is deleted with its arcs and the remaining digraph is treated in the same way. Finally, we end up with the digraph $(\{t\},\emptyset)$. We assign the number $|G|$ to t.
Clearly, the obtained numbering is a natural drain function of (G,V).

A plane representation of a planar drain (G,V) is called a drain representation if the source and the sink are on the same face boundary. Not every planar drain has a drain representation!

Theorem 5: The face boundaries of a drain representation (G^*,V^*) correspond with drains in the original drain (G,V). If the multigraph of (G,V) contains no articulation points the face boundaries consist of exactly two independent chains.

Proof: The source and the sink can be connected by an open Jordan curve which we consider to be the arc $[s,t>$. The new digraph (G',V') has a multigraph of which the simple graph is 2-connected, which means that every face boundary in the multigraph of (G',V') is a circuit.
Every face boundary must have a source and a sink, since a drain is acyclic. If there are more sources and sinks on a face boundary, they have to alternate. Let us take a sequence of two source-sink pairs, (x_1^*,y_1^*) and (x_2^*,y_2^*). From theorem 3 we know that there are chains from s^* to x_1^* and x_2^* and there are chains from y_1^* and y_2^* to t^*. The chain from s^* to x_2^* and the one from y_1^* to t^* must have a vertex in common. Thus, these chains together with the face boundary must contain a cycle. Consequently the digraph (G,V) must contain a cycle. However, (G,V) was acyclic.

In a drain representation (G^*,V^*), W is the set of faces, w_0 the outerface. $(C_w+,C_w+[>)$ denotes the chains of the face boundary of $w\in W\setminus\{w_0\}$ of which the directions of the arcs coincide with a clockwise orien-

tation along the face boundary (C_w^-,C_w^-[>) denotes the other chain. For w_0 it is just the other way around. From a drain representation (G^*,V^*) we can obtain another drain (G^d,V^d), which also has a drain representation: σ be an injective mapping from $W\backslash\{w_0\}$ into G^d, and τ be a bijective mapping from V^* onto V^d, such that:

$$G^d=\{s^d,t^d\}\cup(W\backslash\{w_0\})\sigma$$

$$v^*\in C_{w_i}^-[>\cap C_{w_j}^+[> \rightarrow v^*\tau=[w_i\sigma,w_j\sigma>$$

$$v^*\in C_{w_0}^-[>\cap C_{w_j}^+[> \rightarrow v^*\tau=[s^d,w_j\sigma>$$

$$v^*\in C_{w_i}^-[>\cap C_{w_0}^+[> \rightarrow v^*\tau=[w_i\sigma,t^d>$$

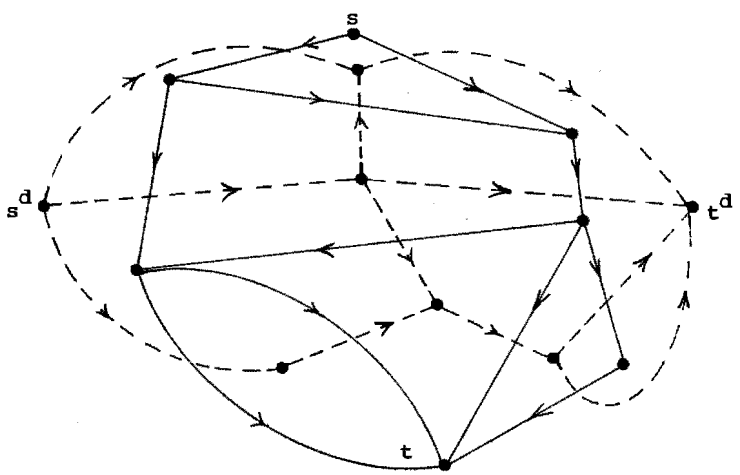


Fig. D.1. A drain representation with its dual.

Theorem 6: (G^*,V^*) be a drain representation. The digraph (G^d,V^d) obtained from (G^*,V^*) is a drain with a drain representation.

Proof: s^d clearly is the only source, and t^d the only sink, for any arc is contained in exactly two face boundaries. The planarity follows from the construction: place a vertex in every face of $W\backslash\{w_o\}$ and place s^d and t^d in the outer face; connect two vertices by an arc only if an arc is found that the two corresponding face boundaries have in common (the new arc is only allowed to cross the arc that the face boundaries have in common). s^d and t^d are clearly in the same face boundary of the constructed representation of

(G^d,V^d). Now we still have to prove that (G^d,V^d) is acyclic. Since the orientation of the arc $v^*\tau$ is always directed from the vertex corresponding with the face w_i for which $v^*\epsilon C_{w_i}-[>$, to the vertex corresponding with the face w_j for which $v^*\epsilon C_{w_j}+[>$, the arcs of an eventual cycle in (G^d,V^d) are all pointing to the interior or all pointing to the exterior of the Jordan curve formed by this cycle. In the former case there must be a sink in the interior, in the latter a source.

The drain (G^d,V^d) is called the dual drain of (G,V). The dual drain of (G^d,V^d) is not the original drain (G,V)! It is the drain that one obtains by reversing all directions in (G,V).

Contraction of an arc in a drain is called drain-preserving if the resulting digraph is a drain. Deletion of an arc in a drain is called drain-preserving if the resulting digraph is a drain. The reverse operations are called extensions.

Theorem 7: (G,V) be a chain, which has a dual (G^d,V^d). The drain obtained by a drain-preserving contraction of the arc v in (G,V) has a dual that can be obtained from (G^d,V^d) by a drain-preserving deletion of $v^*\tau$. The drain obtained by a drain-preserving deletion of the arc v in (G,V) has a dual that can be obtained from (G^d,V^d) by a drain-preserving contraction of $v^*\tau$.

Proof: Trivial, after the observation that a deletion of [x,y> is drain-preserving only if $x\gamma^+>1$ and $y\gamma^-<1$ and a contraction of [x,y> is drain-preserving only if $\neg\exists_{w\in W}[C_w+[>=\{[x,y>\}\vee vC_w-[>=\{[x,y>\}]$.

Theorem 8: The subdigraph (G',V') in the interior of a closed Jordan curve formed by two independent chains $(C_1^*,C_1^*[x^*,y^*>)$ and $(C_2^*,C_2^*[x^*,y^*>)$ in a drain representation of (G^*,V^*) together with these two chains is a drain representation itself.

Proof: Clearly, this subdigraph (G',V') is acyclic, and only x can be a source and only y can be a sink. What we have to prove is that x is a source and y is a sink.
Suppose: $x\Gamma_{V'}^{-1} \neq \emptyset$ and $x_1 \in x\ \Gamma_{V'}^{-1}$. If $x_1 \in C_1 \cup C_2$ then we have found a cycle in (G,V) which is impossible. If $x_1 \notin C_1 \cup C_2$, then $x_1\Gamma_{V'}^{-1} \neq \emptyset$ and there is an $x_2 \in x_1\Gamma_{V'}^{-1}$. If $x_2 \in C_1 \cup C_2$, then again we would have a cycle, else we can continue with x_2 as we did with x and x_1. Sooner or later we must find a vertex of $C_1 \cup C_2$, and thus a cycle. But (G,V) is acyclic. Thus we conclude that x^* is a source. Analogously, we prove that y is a sink of (G',V'). The subrepresentation of (G',V') in (G^*,V^*) is a drain representation, since x^* and y^* are on the outer face boundary formed by $(C_1^*,C_1^*[x^*,y^*>)$ and $(C_2^*,C_2^*[x^*,y^*>)$.

E. Simplex method.

The simplex algorithm is a method to solve linear programming problems such as the one described in section 7.2. [E.1]
Given a set of equations: $A\underline{x}=\underline{b}$, $\underline{x}\in R^n$, $\underline{b}\in R^m$, A is an nxm-matrix of rank m. A solution of this set is called feasible if it has no negative co-ordinate. Further, there is an objective function: $\underline{p}^T.\ \underline{x}$, $\underline{p}\in R^n$. The algorithm has to find a feasible solution for which the objective function is minimal.
It is required that initially one finds a particular feasible solution, with n-m co-ordinates equal to zero, and such that the matrix of the others is non-singular. After an eventual renumbering it can be supposed that all non-zero co-ordinates are in the first m co-ordinates of the solution vector. Such a solution is called a basic solution. Partition the system in the following way:

$$\underline{x}_1^T=(x_1,x_2,\ldots,x_m) \qquad \underline{x}_2^T=(x_{m+1},\ldots,x_n) \qquad \underline{x}^T=(\underline{x}_1^T \mid \underline{x}_2^T)$$

$$\underline{p}_1^T=(p_1,p_2,\ldots,p_m) \qquad \underline{p}_2^T=(p_{m+1},\ldots,p_n) \qquad \underline{p}^T=(\underline{p}_1^T \mid \underline{p}_2^T)$$

$$A=(A_1 \mid A_2)$$

thus

$$A_1\underline{x}_1+A_2\underline{x}_2=\underline{b} \quad \text{with} \quad \underline{x}_1=A_1^{-1}\underline{b}-A_1^{-1}A_2\underline{x}_2$$

At this stage the co-ordinates of $\underline{x}_1$ are called the basic variables, and the initial solution has $\underline{x}_1=A_1^{-1}\underline{b}\geq\underline{0}$ and $\underline{x}_2=\underline{0}$
The objective function can be rewritten as a function of $\underline{x}_2$ only:

$$\underline{p}_1^TA_1^{-1}\underline{b}+(\underline{p}_2^T-\underline{p}_1^TA_1^{-1}A_2)\underline{x}_2$$

If $\underline{x}^*=(x_1^*,x_2^*,\ldots,x_m^*,0,0,\ldots,0)$ is a feasible solution of the system $A\underline{x}=\underline{b}$ and $\underline{p}_2^T-\underline{p}_1^TA_1^{-1}A_2\geq\underline{0}^T$, then $\underline{x}^*$ is an optimal solution vector.
However, if the actual vector is not optimal the rewritten form of the objective function must have a negative coefficient. By allowing the corresponding co-ordinate of $\underline{x}_2$ to increase, while holding the other co-ordinates of $\underline{x}_2$ at zero, the objective function can be decreased, but, in general, some co-ordinate of $\underline{x}_1=A_1^{-1}\underline{b}-A_1^{-1}.A_2\underline{x}_2$ will vanish. At the point, where the first element of $\underline{x}_1$ has vanished, the non-basic variable which has been allowed to increase is adjoined to the set of basic variables, and the basic variable which has vanished

is adjoined to the non-basic ones, and the process is repeated. After a finite number of repetitions an optimal solution is found.

What remains, however, is to find a basic solution to depart from. Before we describe a method, we set up the tableau for our problem of section 7.2: The restrictions were of three kinds: equalities with a zero at the right-hand-side of the "equality"-sign, inequalities with a zero at the right-hand-side of the "less-or equal"-sign, and inequalities with a positive real number at the right-hand-side of the "greater-or-equal"-sign. We transform them by adding so-called slack variables:

$$\begin{aligned}
&\underline{a}_j^T.\underline{x}=0 \quad &:& \quad k_j \quad +\underline{a}_j^T.\underline{x}=0\\
&\underline{a}_j^T.\underline{x}\leq 0 \quad &:& \quad k_j+s_j+\underline{a}_j^T.\underline{x}=0\\
&\underline{a}_j^T.\underline{x}\geq b_j \quad &:& \quad k_j-s_j+\underline{a}_j^T.\underline{x}=b_j
\end{aligned}$$

Any solution of this new system with all $k_j=0$, all $s_j\geq 0$ and $\underline{x}\geq\underline{0}$, gives us a vector $\underline{x}$ satisfying the original equations and inequalities. In this case it is easy to find a basic solution of the second system, namely by giving the k_j (the "artificial slack variables") the value of the corrosponding right-hand member, and all the other variables zero. The only problem left is, how to come from this first solution to the starting solution of the original system. In the so-called "two-phase-method" this is done by minimizing another objective function, namely $\sum k_j$.[E.2]. For the problem of section 7.2. the complete tableau is given in fig. 7.4.

The algorithm now works as follows: With the simplex algorithm a basic solution of the system without artificial slack variables is found by minimizing the first-phase objective function, restricted by the tableau with artificial slack variable. During the first-phase the second-phase objective function is not used for pivot selection, but we take care of keeping it expressed in non-basic variables. When none of the artificial slack variables is a basic variable, we continue with the tableau obtained, but without the artificial slack variables, and now we use the second-phase objective function for pivot selections. Since

we are sure of the existence of a solution, the program will finish both phases and end up with an optimum solution.

There are, of course, many implementations possible for the simplex-algorithm. We have chosen for the so-called "symmetric-revised-simplex method". [E.3]

References.

[E.1] G.B.Dantzig, "Linear programming and extensions", Princeton University Press, Princeton, New Jersey, 1963.

[E.2] H.P.Künzi, "Die Simplexmethode zur Bestimmung einer Ausgangslösung bei bestimmten linearen Programmen", Unternehmensforschung, Vol. 2, pp. 60-69, 1958.

[E.3] H.Müller-Merbach, "Die symmetrische revidierte Simplex-Methode der linearen Planungsrechnung", Elektronische Datenverarbeitung, 7. Jahrgang, Heft 3, pp. 105-113, 1963.

F. Representation of planar graphs.

In chapter 4 a method is described that completely automatically planarizes the potential graph. Many considerations concerning the predilection for certain modifications over others are worked into that procedure in order to come to an acceptable layout. They emanated from integration principles, experiences of layout specialists and design desires. In the opinion of many people such a large range of diverse considerations cannot be taken into account by an automatic procedure and so they chose for interactive design procedures. Often their starting point is also a graph derived from the circuit diagram, that must be planarized. Here the need for surveyable and fastly generated representations of planar graphs became urgent.

A plane representation in which the edges are arbitrary open Jordan curves will not satisfy the requirement of surveyability, since it is difficult to notice connections in an eye glance when these Jordan curves meander from one vertex to the other. Besides, these programs are mostly interactive up to a very high degree, which makes them slow and hardly better than a "pencil-and-paper" procedure [F.1]. However, since 1936 it is known that every planar graph has a plane representation in which every edge is a straight line segment between its vertices [F.2]. Both interactive and fully automatic programs delivering such a straight-line representation exist, but all of them display clusters of vertices, which means that the picture still is not surveyable, and that limits depending on the resolution are reached very soon [F.3], [F.4]. Moreover, even the automatic drawing procedure is too slow for interactive planarization. Attempts to avoid these clusters were not successful, and it is quite possible that for many planar graphs clusters in their straight-line representations cannot be avoided.

We therefore leave the idea of representing planar graphs by their plane representation, and propose the "horvert" representation. In this representation the vertices are horizontal line segments and the edges are vertical line segments having only points in common with

their vertices. This horvert representation has the following salient features:

1. it can be generated in a very simple and fast way without any interaction;
2. provided that a sequence in "vertex levels" is observed any distribution of these levels is allowed;
3. the same statement is true for the edges;
4. parallel edges present no difficulty as they do in straight-line representations.

We assume the graph to be biconnected (this is not a restriction), and assign a drain function λ to this graph such that the resulting drain has a drain representation (G^*,V^*). This can be achieved by choosing the source and the sink such that there is an arc between them.

Horvert representation:

η assigns to each vertex a y-co-ordinate such that
$\forall_{a\in G} \forall_{b\in G} [a\lambda>b\lambda \rightarrow a\eta>b\eta]$

μ is a natural drain function of (G^d,V^d), the dual derived from (G^*,V^*)

χ assigns to each face an x-co-ordinate such that
$\forall_{w_1\in W\setminus\{w_o\}} \forall_{w_2\in W\setminus\{w_o\}} [w_1\sigma\mu>w_2\sigma\mu \rightarrow w_1\chi>w_2\chi]$

$S_{(g^*)}$ be the horizontal line segment representing $g\in G$:
$\forall_{g\in G} [g\in C_{w_1}^- \cap C_{w_2}^+ \rightarrow \forall_{(x,y)\in S_{(g^*)}} [w_1\chi<x<w_2\chi]]$

$\forall_{g_1\in G} \forall_{g_2\in G} [[g_1,g_2>\in V \rightarrow \exists_{x\in R} [(x,g_1\eta)\in S_{(g_1^*)} \wedge (x,g_2\eta)\in S_{(g_2^*)}]]$

An x-co-ordinate satisfying the last predicate, can be chosen as the x-co-ordinate of the arc $[g_1,g_2>$, which must be drawn between $S_{(g_1^*)}$ and $S_{(g_2^*)}$.

The various co-ordinates can be determined by linear programming, but in the proof of theorem 1 a more efficient construction is described.

Theorem 1: A graph (G,U) is planar fif it has a horvert representation.

Proof: Suppose (G,U) has a horvert representation. One may consider this representation as a plane representation of another graph (G',U') with only horizontal and vertical edges. After contraction of all horizontal edges the resulting graph is isomorphic to (G,U). By theorem A.9 we know that planarity is preserved under contraction. So (G,U) must be planar.

The other half of the statement is proved by giving the construction of a horvert representation from the biconnected, planar graph (G,U).

1) Choose an arbitrary edge $[s,t] \in U$.
2) Assign the integers 1 up to $|G|$ to the vertices by means of a drain function λ such that $s\lambda=1$ and $t\lambda=|G|$ (Theorem D.1).
3) Choose a drain representation (G^*,V^*) with $C_{w_o}+[\ >=\{[s,t>\}$ (Appendix A).
4) Construct a dual drain (G^d,V^d) from (G^*,V^*) (Theorem D.6)
5) Determine a natural drain function μ for (G^d,V^d) (Theorem D.4).
6) Assign to each vertex a y-co-ordinate as prescribed in the definition of η.
7) Assign to each face an x-co-ordinate as prescribed in the definition of χ.
8) Draw the edge $[s,t]$ (which forms the whole $C_{w_i}-[>$ with $i=|W|$) at an abscis greater than $w_i\chi$.
9) Draw the edges in $C_{w_i}+[\ >$ at an abscis between $w_{i-1}\chi$ and $w_i\chi$ $(w_o\chi<w_1\chi)$ (the whole $C_{w_{i-1}}-\ [\ >$ is now drawn; this follows from the drain properties of (G^d,V^d)).
10) If $i\neq 1$, then $i:=i-1$ and goto step 9.
11) Draw the vertices taking into account the constraints given in the definition.

In order to planarize an arbitrary graph interactively, using the horvert representation we need a suitable planarization algorithm. The algorithm that lends itself outstandingly to this purpose is the "CEL-algorithm", which will now be concisely described [F.5].

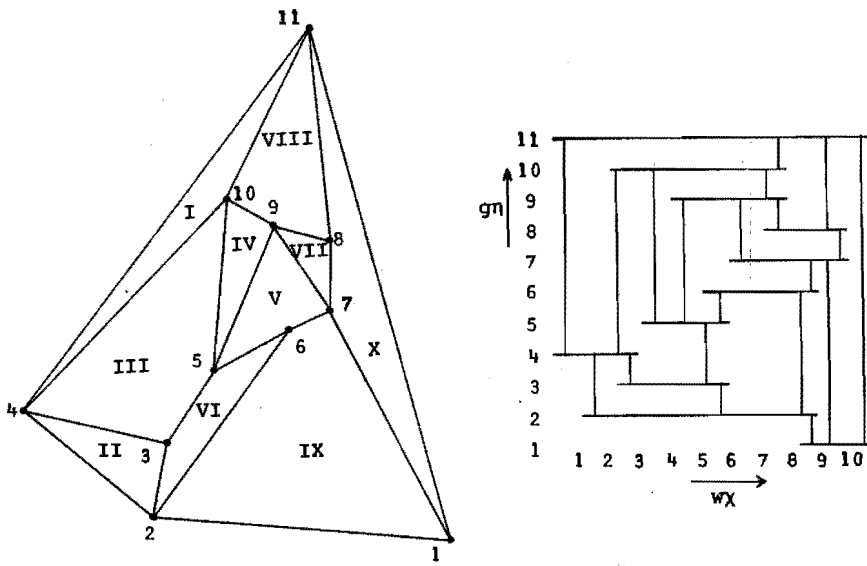


Fig. F.1. A straight-line representation and a horvert representation of a graph.

Let A be a finite set. The elements of A are called labels and A is called the label alphabet. $|A|=n$. ζ is a linear ordering over A (i.e. transitive, antireflexive and antisymmetric).

ϕ is called a formula if ϕ is a sequence of symbols from the set $A \cup \{*, \circ, [,]\}$ and satisfies one of the following requirements:

1. ϕ is an element of A;
2. ϕ is a product of factors, i.e. $\phi=[\phi_1]*[\phi_2]*\ldots*[\phi_p]$, where $\phi_1,\phi_2,\ldots,\phi_p$ are formulas;
3. ϕ is a concatenation of components, i.e. $\phi=[\phi_1]\circ[\phi_2]\circ\ldots\circ[\phi_q]$, where $\phi_1,\phi_2,\ldots,\phi_q$ are formulas;

while the following transformations are performed if possible:

1. $[[\phi]] := [\phi]$,
2. $[\phi] := \phi$, if $\phi \in A$
3. $[\phi] := \phi$, if ϕ is a concatenation and $[\phi]$ is enclosed by * and *, or * and], or [and *, or the beginning of the formula and *, or * and the end of the formula.

The set of labels occurring in a certain formula ϕ is denoted by A_ϕ. One of its elements is called the index label il, defined by $\forall_{x\in A_\phi}[x\neq il \rightarrow (il,x)\in\zeta]$.

We allow two kinds of formula manipulation: a permutation of the factors of a product and a reflection of a concatenation (i.e. a reversal of the order of the components of this concatenation). A formula is called normal if there are no other labels than index labels between index labels. A formula ϕ is normalizable if it can be transformed into a normal formula η_ϕ by a succession of permutations and reflections. The set of normalizable formulas is denoted with NR. The contraction of a normal formula η is the replacement of the minimal sequence of symbols in η containing all the index labels of η by one index label.

The symbols [and] are called a pair if they enclose as many symbols of the type [as of the type]. Such a pair is weak if one of its two symbols is placed between two index labels without other labels between these index labels. A weak pair is dead if it is not enclosed by a weak pair.

Reduction of a normal formula η is the application of the following two rules as many times as possible:

1. if ϕ is a product directly enclosed by a dead pair, and ϕ' is the first or the last factor of that product and ϕ' contains an index label, then $[\phi]$ is replaced in the following way:
   $[.....*\phi']:= [.....]\circ\phi'$
   $[\phi'*.....]:= \phi'\circ[.....]$,
2. if ϕ is a concatenation containing an index label and directly enclosed by a dead pair, then $[\phi]$ is replaced by ϕ:
   $[\phi]:= \phi$,

followed by a contraction. The result is denoted by ρ_η.

Suppose $A=\{l_1,l_2,\ldots,l_n\}$ such that $\forall_{1\leq i<n}[(l_i,l_{i+1})\in\zeta]$. Further, we have n-1 basic formulas: $\beta_1,\beta_2,\ldots,\beta_{n-1}$, with $\beta_k=[l_{x_1}*l_{x_2}*\ldots*l_{x_m}](m\geq 1)$ and $\forall_{1\leq i<m}[((l_{x_i},l_{x_{i+1}})\in\zeta\vee l_{x_i}=l_{x_{i+1}})\wedge(l_k,l_{x_1})\in\zeta]$. If in a reduced

formula ρ the index label is replaced by a basic formula β_k, the result is denoted by $\beta_k \downarrow \rho$.

CEL-algorithm

'begin' p:=1;ϕ:=β_1;
  'for' k:=2, k+1 'while'(k<n)$\wedge$(p=1)'do'
  'if' $\phi \in NR$'then' $\phi := \beta_k \downarrow \rho_{\eta_\phi}$ 'else'p:=0
'end' of CEL;

With a given set of basic formulas the CEL-algorithm gives p always the same value. This is not a trivial statement, for normalization is not a unique process.

After defining the CEL-algorithm we still have to explain its relation with graphs and planarization in particular. We therefore build a similar framework for a special kind of digraphs as we did for formulas.

A plane representation of a digraph is called a delta (D ,V) if

1. (D,V) is acyclic
2. $\exists!_{s_\Delta \in D}[s_\Delta \Gamma^{-1}=\emptyset]$
3. $\forall_{t \in D}[t\Gamma=\emptyset \leftrightarrow |t\Gamma^{-1}|=1]$
4. (D^*,V^*) is in π; in π we have a straight line α dividing π into two half planes; $S_{(D^*,V^*)}$ is completely in one of these half planes, and if $(\alpha,x^*)\delta$ is the distance between $x^* \in D^*$ and α, then
   $\forall_{x \in D}\ \forall_{y \in D}[[x,y> \in V \rightarrow (\alpha,x^*)\delta < (\alpha,y^*)\delta]$
   $\forall_{t \in D}\ \forall_{z \in D}[t\Gamma=\emptyset \rightarrow (\alpha,t^*)\delta \geq (\alpha,z^*)\delta]$

All sinks of a delta can be located on a straight line β parallel to α. β is called the gutter.

A sector is a maximal subdigraph (D',V') of (D,V) containing the source s_Δ, but such that s_Δ is not an articulation point of (D',V'). The maximal 2-connected subdigraph of a sector (D',V') that contains s_Δ, is called the kernel of (D',V'). The maximal connected subdigraphs of a sector (D',V'), obtained by removing all the arcs of the kernel and the formed isolated vertices, are called shells.

If we assign an orientation to the gutter β, we can recognize an order in the sectors and shells of a delta. Two deltas are called similar if they are isomorphic and if the corresponding subdeltas occur in the same order. A permutation of a delta is a transformation yielding an isomorphic delta with similar sectors. A reflection of a sector is a non-trivial transformation yielding an isomorphic sector with similar shells.

A be a label alphabet with linear ordering ζ over A and a (not necessarily injective) mapping from A onto $\{t|t\Gamma=\emptyset\}$. Two deltas are called Λ-isomorphic if they are isomorphic and the corresponding vertices have the same labels. The vertices in $\{x|x\Gamma=\emptyset\wedge\forall_{t\in\{y|y\Gamma=\emptyset\}}[t\lambda=x\lambda\vee v(x\lambda,t\lambda)\in\zeta]\}$ are called index sinks. A delta is normal if all its index sinks succeed each other at the gutter without another vertex in between. A delta is normalizable if it has a Λ-isomorphic delta which is normal.

Reduction of a normal delta (D^*,V^*) is

1. identification of all index sinks in a new vertex r^* located at the same side of β as the source and in the same face as the other sinks,
2. addition of a new arc $[r^*,t^*>$; t^* is a new vertex on β with such a location that vertices that were not neighbours on the gutter before the reduction are not neighbours after the addition of the new arc,
3. r^* and t^* obtain the same label as the index sinks had.

Finally we assign to every delta a formula in the following way:

1. $(\{x^*,y^*\},\{[x^*,y^*>\})$ obtains $y\lambda$ as a formula
2. A delta consisting of p sectors obtains $[\phi_1]*[\phi_2]*\ldots*[\phi_p]$ as a formula, where ϕ_i is the formula of i-th sector
3. A sector with q shells obtains $[\phi_1]\circ[\phi_2]\circ\ldots\circ[\phi_2]$, where ϕ_i is the formula of the i-th shell.

The by names already suggested correspondences between the various concepts for formulas and deltas are more explicitely expressed in the following statements. A reflection of a sector has a reversal in the sequence of shells as a consequence, thus the formula of the resulting sector is the reflected formula of the original one. A permutation of sectors of a delta corresponds with a permutation in the respective

formulas. A normalizable delta can be normalized by successive applications of permutations and reflections. A delta is normalizable fif its formula is normalizable. The formula of a delta is normal fif the delta is normal. The formula of a reduced delta of (D^*,V^*) is a reduced formula of (D^*,V^*).

We are now ready to explain how the CEL-algorithm can be used to test the planarity of a biconnected graph (G,U). Assign to (G,U) a drain function λ such that the source and the sink in the λ-oriented digraph are connected by an arc. The basic set of the λ-oriented digraph of (G,U) is the set of basic formulas $\{\beta_i \mid 1 \leq i < |G|\}$ where $A_{\beta_i} = i\lambda^{-1}\Gamma$. The connection between the CEL-algorithm and the planarity of graphs is expressed in the theorem: A biconnected graph (G,U) is planar fif the CEL-algorithm gives p the value 1 when operating upon the basic set of the λ-oriented digraph of (G,U) (Fig. F.2).

The application of the CEL-algorithm in interactive planarization is straightforward. We start from the potential graph and assign a drain function to it in which source and sink are chosen such that they are connected in the digraph. Next its basic set is derived, and the CEL-algorithm continues its operation until p gets the value 0. At that moment the procedure for generating horvert representations is called, and delivers a picture of the last non-normalizable delta by identifying all sinks with one vertex which is not drawn. In the menu of the screen several kinds of information can be displayed (for example which component or potential is represented by a certain vertex, with which entities it is connected). This can be helpful in deciding which modification can be best applied in the actual situation. Depending on the consequences the decision has for the graph the CEL-algorithm is restarted from the point it was stopped at or the whole procedure is repeated from the search for a drain function for the modified graph.

Remark: In order to speed up the generation of the picture it is expedient to keep the sequences of vertices on the face boundaries and their orientation during the operations on the formulas. This is not very complicated, since any new sequence can be derived from the preceding outer face. Further, we implemented

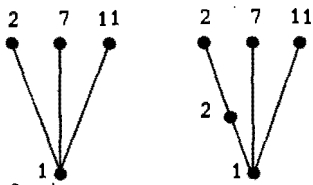
β_1 = [2*7*11]
β_2 = [3*4*6]
β_3 = [4*5]
β_4 = [10*11]
β_5 = [6*9*10]
β_6 = [7]
β_7 = [8*9]
β_8 = [9*11]
β_9 = [10]
β_{10}= [11]

$\beta_1 = \phi_2 = \eta_2 = \rho_2$ = 2*7*11
$\phi_3 = \eta_3 = \rho_3$ = [3*4*6]*7*11
ϕ_4 = [[4*5]*4*6]*7*11
η_4 = [[5*4]*4*6]*7*11
ρ_4 = [5∘4*6]*7*11
$\phi_5 = \eta_5 = \rho_5$ = [5∘[10*11]*6]*7*11
ϕ_6 = [[6*9*10]∘[10*11]*6]*7*11
η_6 = [[10*11]∘[10*9*6]*6]*7*11
ρ_6 = [10*11]∘[10*9]∘6*7*11
$\phi_7 = \eta_7$ = [10*11]∘[10*9]∘7*7*11
ρ_7 = [10*11]∘[10*9]∘7*11
$\phi_8 = \eta_8 = \rho_8$ = [10*11]∘[10*9]∘[8*9]*11
ϕ_9 = [10*11]∘[10*9]∘[[9*11]*9]*11
η_9 = [10*11]∘[10*9]∘[9*[9*11]]*11
ρ_9 = [10*11]∘10∘9∘11*11
ϕ_{10}= [10*11]∘10∘10∘11*11
η_{10}= [11*10]∘10∘10∘11*11
ρ_{10}=11∘10∘11*11
$\phi_{11}=\eta_{11}$=11∘11∘11*11
ρ_{11}=11

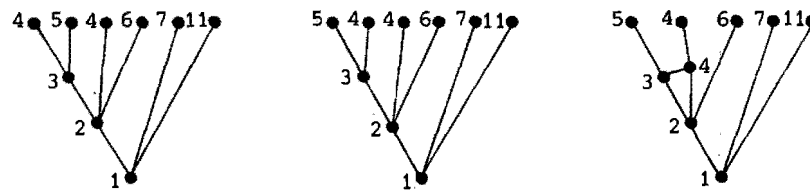
β_1 = [2*3*6]
β_2 = [4*5]
β_3 = [4*5]
β_4 = [6]
β_5 = [6]

$\beta_1 = \phi_2 = \eta_2 = \rho_2$ = 2*3*6
$\phi_3 = \eta_3 = \rho_3$ = [4*5]*3*6
ϕ_4 = [4*5]*[4*5]*6
η_4 = [5*4]*[4*5]*6
ρ_4 =5∘4∘5*6
ϕ_5 =5∘6∘5*6

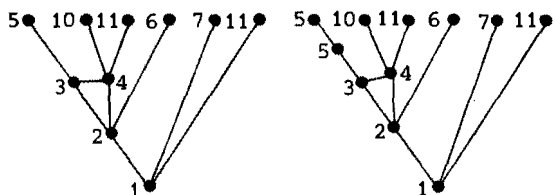
Fig. F.2. The CEL-algorithm applied to the graph of fig. F.1 and KUR6. The operation is illustrated by the delta at the next pages.



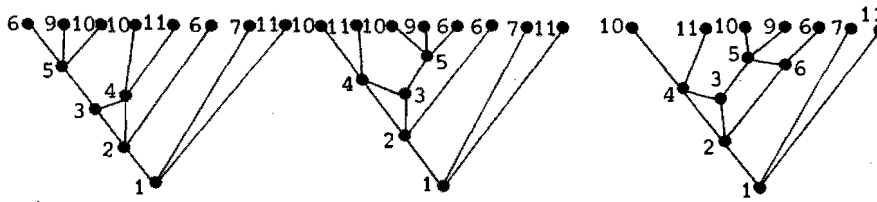
$\beta_1=\phi_2=\eta_2=\rho_2$ $\phi_3=\eta_3=\rho_3$



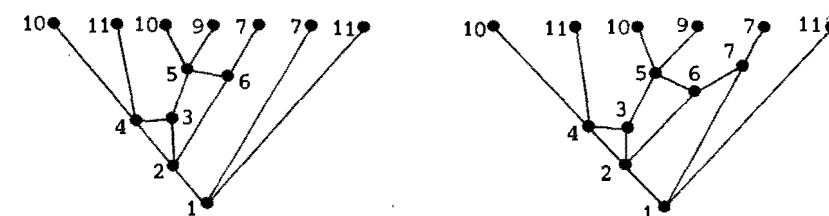
ϕ_4 η_4 ρ_4



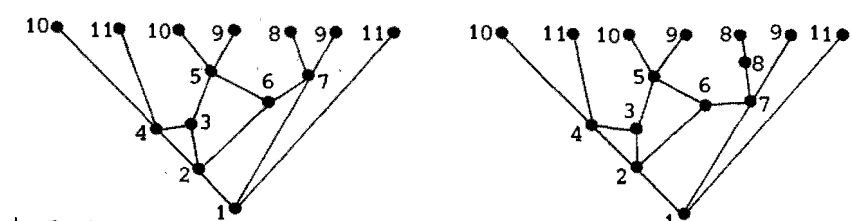
$\phi_5=\eta_5=\rho_5$



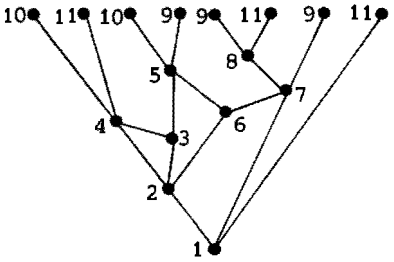
ϕ_6 η_6 ρ_6



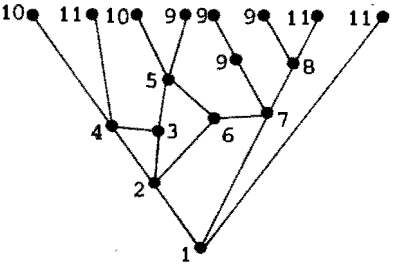
$\phi_7=\eta_7$ ρ_7



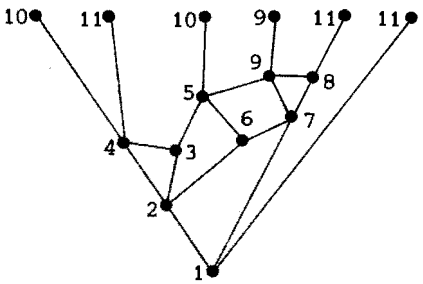
$\phi_8=\eta_8=\rho_8$



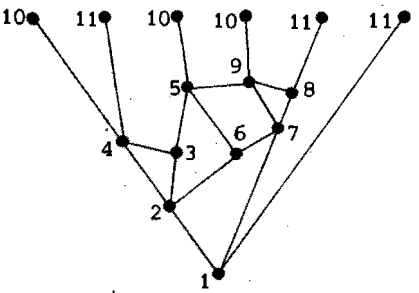
ϕ_9



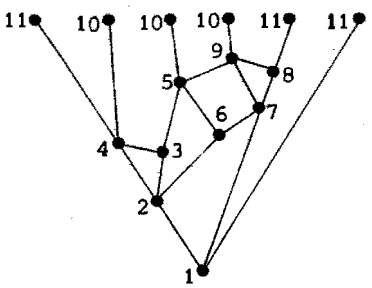
η_9



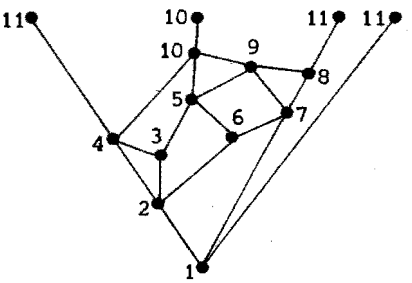
ρ_9



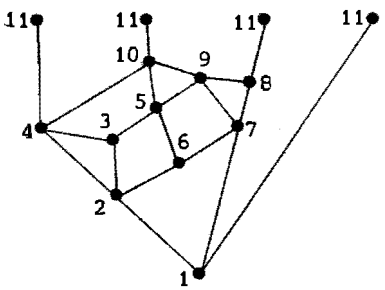
ϕ_{10}



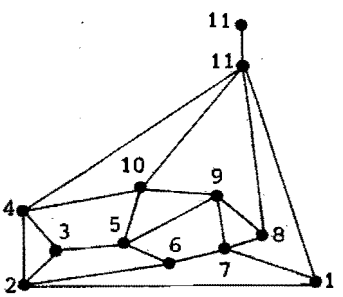
η_{10}



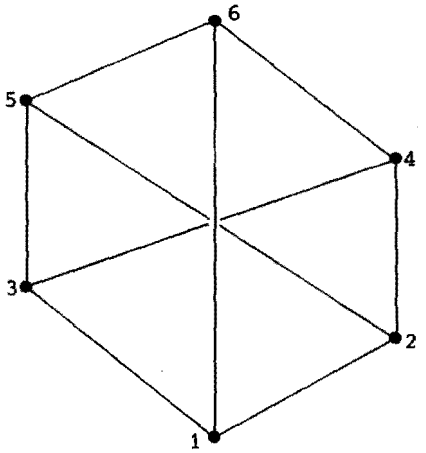
ρ_{10}

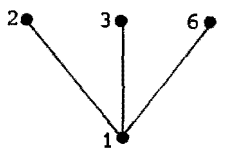


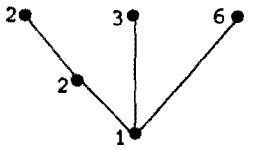
$\phi_{11}=\eta_{11}$



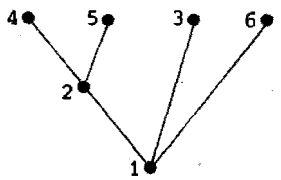
ρ_{11}

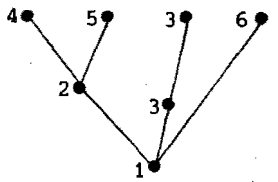




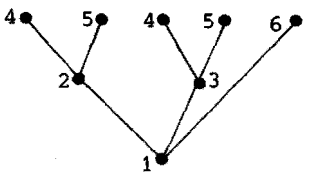


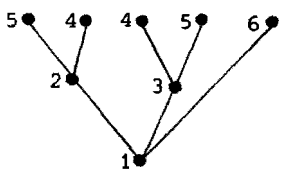
$\beta_1=\phi_2$





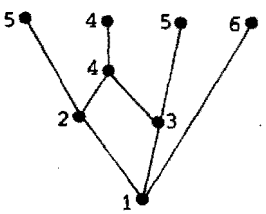
$\phi_3=\eta_3=\rho_3$

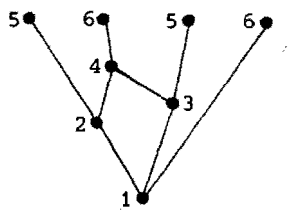




ϕ_4

η_4





ρ_4

ϕ_5

CEL by turning to a prefix notation with indication of the range of parenthesis expression. This enabled us to examine the formula level by level, (two expressions are of the same level if they are enclosed by the same number of parentheses). In case of storing formulas in infix form this is much more difficult.

References.

[F.1] I.Fary, "On straight line representation of planar graphs", Acta Sci. Math. 11, pp. 229-233, 1948.

[F.2] A.K.Hope, "A planar graph drawing program", Software-Practice and Experience, Vol. 1, pp. 83-91, 1971.

[F.3] Lin Woo, "An algorithm for straight-line representation of simple planar graphs", Journal of the Franklin Institute, Vol. 287, 3, pp. 197-208, March 1969.

[F.4] R.Koppe, "Automatische Abbildung eines planaren Graphen in einen ebenen Streckengraphen", Computing 10, pp. 317-333, 1972.

[F.5] A.Lempel, S.Even, J.Cederbaum, "An algorithm for planarity testing of graphs", Theory of Graphs, Intern. Symposium, pp. 215-232, Rome 1966.

G. The component library.

The component library contains a number of files with the data for the standard components, and procedures to construct the geometrical data for the components having a standard configuration that is only dependent on the component value ("variable standard components" such as low-valued resistors). Of course procedures are incorporated for reflection, rotation and translation of the various layout-configurations of the components.

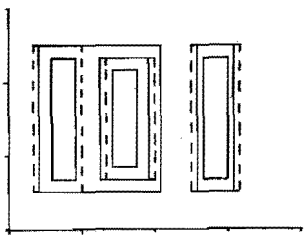
The data for each standard component is partitioned into different parts.

- First of all the library contains the geometrical data for the various masks (SP,SN,CO,IN) in sequences of 5-tupels of numbers. Each 5-tupel fixes a line segment or circle arc.

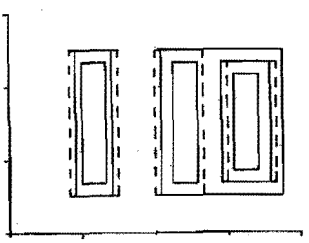
  line segment: <line code><x_1><y_1><x_2><y_2>
  where (x_1,y_1) and (x_2,y_2) are the start and end point of the line segment

  circle arc : <α_1><α_2><r><x_m><y_m>
  where (x_m,y_m) $\hat{=}$is the centre of the circle in question
  r $\hat{=}$the radius of the circle in question
  α_1 $\hat{=}$the angle between the X-direction and the line determined by (x_m,y_m) and the start point.
  α_2 $\hat{=}$idem for the end point.

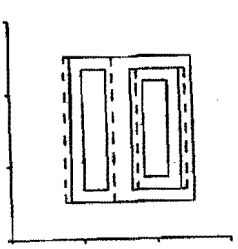
The boundary of the region that is "occupied" by a standard component coincides with the grid structure over the chip (see chapters 7,8,9), i.e. the length and width of the regions are a multiple of the grid constant. The coordinates that fix the line segments have been given with respect to the rectangular XY-coordinate system having the origin in the left and lower corner of the region (axes parallel to the boundary segments of the region).
In figure G.1 we have depicted the layouts of some commonly used standard components (the origin of the coordinate system, and the grid units are in the left corner; the aluminium mask configuration is given in dashed lines).



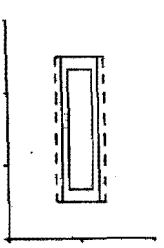
1. *npn-transistor (sequence of contacts base-emitter-collector: b-e-c).*



2. *npn-transistor (c-b-e).*

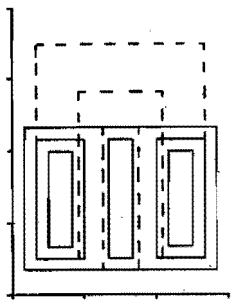


3. *base-emitter part of an npn-transistor (b-e).*

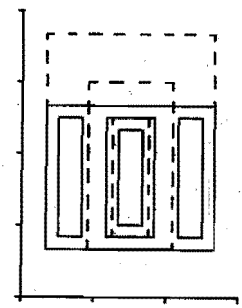


4. *island potential contact.*

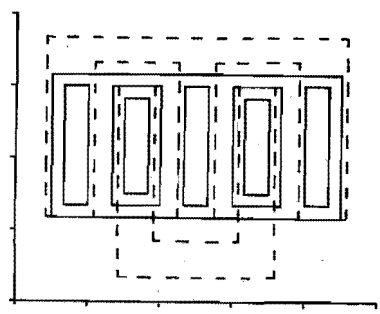
Fig. G.1. Some standard components.



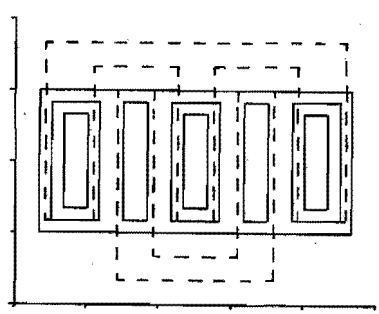
5. *base-emitter part of an npn-transistor (2 emitter contacts).*



6. *base-emitter part of an npn-transistor (2 base contacts).*



7. *base-emitter part of an npn-transistor (2 emitter contacts; 3 base contacts).*



8. *base-emitter part of an npn-transistor (3 emitter contacts; 2 base contacts).*

- Furthermore the library contains data fixing the (grid) cells of the region that is occupied by a certain contact. For each contact of the component we store the cells along the boundary of the component region, that contain (a part of) the aluminium pattern over the contact in question. These groups of cells are used during the wiring procedure as origins and targets for searching the wire routes. As an example we give this data for standard component #10:

  emitter contact : the cells (3,1),(4,1),(5,1),(5,2),(3,3), (4,3) and (5,3)

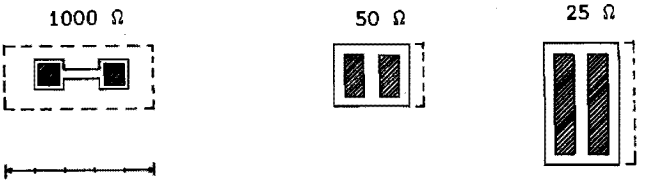
  base contact : the cells (1,1),(1,2) and (1,3)

  collector contact: the cells (2,1) and (2,3)

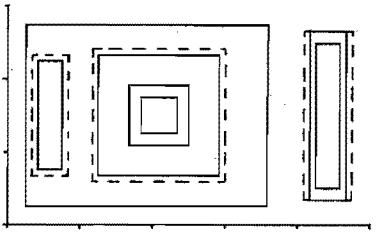
For the variable standard components (e.g. the low-valued resistors) we store some configurations of parts that are non-variable (the contacts). The configuration of the actual resistor diffusion is determined by a procedure in which a number of standard rules are incorporated, such as:

1. The width of the resistor region is taken minimally two grid constants (since the contacts require a width of two grid cells).
2. The length of the actual resistor diffusion is taken minimally half a grid constant.
3. The width of the resistor diffusion is taken as small as possible.
4. If the width of the resistor diffusion exceeds one grid constant, the width of the contacts are adjusted to this width.

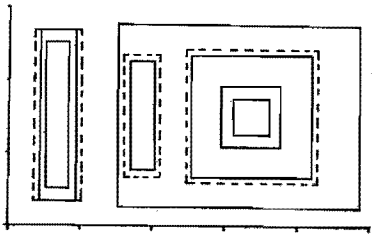
As an example we give the configurations of some low-valued resistors:



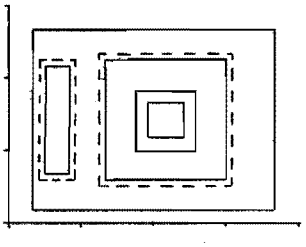
The component region (the part of the grid structure that is occupied) is given with dashed lines.



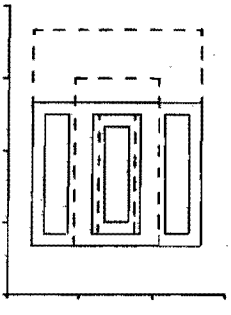
9. lateral pnp-transistor (c-e-b).



10. lateral pnp-transistor (b-c-e).



11. emitter-collector part of a lateral pnp-transistor (c-e).



12. substrate pnp-transistor (2 emitter contacts; no collector contact).

STELLINGEN

bij het proefschrift van

M.C.van Lier

17 december 1976

Technische Hogeschool Eindhoven

I

Een daadwerkelijke bescherming van natuur en milieu vereist in de verschillende takken van onderwijs een intensivering van de natuur- en milieu-edukatie.

II

Het huidige aantal mogelijkheden om een passende werkkring met een gedeeltelijke dagtaak te verkrijgen is volstrekt onvoldoende. Een gevolg hiervan is niet alleen een oneerlijke verdeling van arbeid en ontplooiingskansen, maar ook van arbeidsvoldoening en persoonlijk geluk.

III

De vergoeding voor een geneeskundige behandeling dient afhankelijk te zijn van de ervaring van de behandelende arts.

IV

Het is niet logisch om een voetgangersstrook van een spoorwegovergang aan slechts één zijde te voorzien van een slagboom. De overgang behoort aan beide zijden voldoende beveiligd te zijn.

V

Het recht op arbeid wordt in bepaalde arbeidsvoorwaarden ten onrechte afhankelijk gesteld van de huwelijkse staat en het kostwinnersschap van de werknemers.

VI

Een door een voetbalscheidsrechter opgelegde straf wordt vaak "te licht" of "te zwaar" bevonden. Ter verkrijging van een rechtvaardigere bestraffing van overtredingen, en ter verlichting van de taak van de scheidsrechter verdient het aanbeveling het aantal oplegbare straffen aanzienlijk uit te breiden. Mogelijkheden hiertoe zijn bijvoorbeeld het toekennen van strafschoppen ook voor overtredingen buiten het strafschopgebied en invoering van de strafbank.

STELLINGEN

bij het proefschrift van

R.H.J.M. Otten

17 december 1976

Technische Hogeschool Eindhoven

Het laatste deel van de "Messa da Requiem" van Verdi is zeker niet identiek aan het "Libera Me" gecomponeerd voor de herdenkingsmis van Rossini en er is nauwelijks reden om aan te nemen dat gedeelten van de "Rossini-mis" gebruikt zijn.

G.Cesari, A.Luzio: "I Copialettere di Giuseppe Verdi" Milano, 1913 (Lettera CCX)

D.Hussey: "Verdi" New York, 1940 (Chapter 14).

II

Het ontbreken van een nevenschikkend voegwoord dat de volledige verwisselbaarheid van een aantal beweringen uitdrukt, is niet alleen nadelig bij het aanleren van exacte denkwijzen, maar ook in het dagelijks taalgebruik. In die gevallen waarin het alsnog invoeren van een dergelijk woord wenselijk lijkt, zou ik willen pleiten voor het woord "slals" in plaats van de in de wiskunde-literatuur gebruikelijke verdubbeling van een medeklinker, teneinde het nieuwe woord ook fonetisch van het oorspronkelijke te kunnen onderscheiden.

E.W.Beth: "Geformaliseerde talen en normaal taalgebruik". Algemeen Nederlands tijdschrift voor Wijsbegeerte en Psychologie, 50, pp. 265-276, 1957/58.

III

Bij alle aandacht die de inkomensverdeling over de individuen tegenwoordig krijgt, is het opmerkelijk dat de inkomensverdeling naar leeftijd nauwelijks ter discussie gesteld wordt, terwijl juist in de perioden waarin men meestal het meeste geld nodig heeft, het inkomen lager is, namelijk bij het opbouwen van een zelfstandig bestaan en wanneer men de meeste vrije tijd heeft.

De dubbelzinnigheid in het tijdschema van "Othello" is geen bewijs voor Shakespeare's genialiteit als dramaturg. De aanvaarding van het tegendeel door vrijwel alle Shakespeare-deskundigen van deze eeuw duidt niet op een objectieve benadering van Shakespeare's werken.

J.Wilson ("Christopher North"), Blackwood's Magazine, Nov. 1849, Apr. 1850, May 1850.

V

Het is jammer dat bij de propagering van veel voedingssystemen pseudo-wetenschappelijke en onzinnige argumenten overheersen.

VI

Teneinde de bestudering van de praktische toepassingen van de moderne algebra door aanstaande elektrotechnische ingenieurs te stimuleren, is het nodig dat de betreffende docent - zolang er in het kandidaats-programma geen plaats is voor een degelijke basiscursus "Moderne algebra" - de termen uit dit gebied van de wiskunde correct gebruikt, begrippen niet essentieel voor de opbouw van de theorie achterwege laat en voor een meer intuïtieve en op de praktijk gerichte benadering kiest.