



**HAL**  
open science

# Problèmes d'implantation, le programme DESMAG

Michel J. Sarret

► **To cite this version:**

Michel J. Sarret. Problèmes d'implantation, le programme DESMAG. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1969. Français. NNT: . tel-00281651

**HAL Id: tel-00281651**

**<https://theses.hal.science/tel-00281651>**

Submitted on 23 May 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre

TU 324

# THESE

présentée à

LA FACULTE DES SCIENCES DE L'UNIVERSITE DE GRENOBLE

pour obtenir

LE GRADE DE DOCTEUR DE TROISIEME CYCLE

" Mathématiques Appliquées "

par

**Michel J. SARRET**

licencié ès-sciences



## **Problèmes d'implantation Le programme DESMAG**



Thèse soutenue le 27 juin 1969 devant la commission d'examen :

Monsieur J. KUNTZMANN Président

Messieurs B. VAUQUOIS Examineur

C. BENZAKEN Examineur

J. LACOUR Invité





PROFESSEURS TITULAIRES (suite)

MM.	KLEIN J.	Mathématiques
	VAILLANT F.	Zoologie et Hydrobiologie
	ARNAUD Paul	Chaire de Chimie
	SENGEL P.	Chaire de Zoologie
	BARNOUD F.	Chaire de Biozyntèse de la Cellulose
	BRISSONNEAU P.	Physique
	GAGNAIRE	Chaire de Chimie Physique
Mme	KÖFLER L.	Botanique
	DEGRANGE Charles	Zoologie
	PEBAY-PEROULA J.C.	Physique
	RASSAT A.	Chaire de Chimie Systématique
	DUCROS P.	Vhaire de Cristallographie Physique
	DODU Jacques	Chaire de Mécanique Appliquée I.U.T.
	ANGLES D'AURIAC P.	Mécanique des Fluides
	LACAZE A.	Thermodynamique

PROFESSEURS SANS CHAIRE

MM.	GIDON P.	Géologie et Mnéralogie
	GIRAUD P.	Géologie
	PERRET R.	Servomécanisme
Mme	BARBIER M.J.	Electrochimie
Mme	SOUTIF J.	Physique
	CÖHEN J.	Electrotechnique
	DEPASSEL R.	Mécanique des Fluides
	GASTINEL N.	Mathématiques Appliquées
	GLENAT R.	Chimie
	BARRA J.R.	Mathématiques Appliquées
	COUMES A.	Electronique
	PERRIAUX J.	Géologie et Minéralogie
	ROBERT A.	Chimie Papetière
	BIAREZ J.P.	Mécanique Physique
	BONNET G.	Electronique
	CAUQUIS G.	Chimie Générale
	BONNETAIN L.	Chimie Minérale
	DEPOMMIER P.	Etude Nucléaire et Génie Atomique
	HACQUES Gérard	Calcul Numérique
	POLOUJADOFF M.	Electrotechnique

PROFESSEURS ASSOCIES

MM.	NAPP-ZINN	Botanique
	RÖDRIGUES Alexandre	Mathématiques Pures
	STANDING Kenneth	Physique Nucléaire

MAITRES DE CONFERENCES

MM.	LANCIA Roland	Physique Atomique
Mme	KAHANE J.	Physique
	DEPORTES C.	Chimie
Mme	BOUCHE L.	Mathématiques
	SARROT-REYNAUD	Géologie Propédeutique

MAITRES DE CONFERENCES (suite)

Mme	BONNIER M.J.	Chimie
MM.	KAHANE A.	Physique Générale
	DOLIQUE J.M.	Electronique
	BRIERE G.	Physique M.P.C.
	DESRE G.	Chimie S.P.C.N.
	LAJZROWICZ J.	Physique M.P.C.
	VALENTIN P.	Physique M.P.C.
	BERTRANDIAS J.P.	Mathématiques Appliquées T.M.P.
	LAURENT P.J.	Mathématiques Appliquées T.M.P.
	CAUBET J.P.	Mathématiques Pures
	PAVAN J.J.	Mathématiques
Mme	BERTRANDIAS F.	Mathématiques Pures M.P.C.
	LONGEGUEUE J.P.	Physique
	NIVAT M.	Mathématiques Appliquées
	SÖHM J.C.	Electrochimie
	ZADWORNÝ F.	Electronique
	DURAND F.	Chimie Physique
	CARLER G.	Biologie Végétale
	AUBERT G.	Physique M.P.C.
	DELPUECH J.J.	Chimie Organique
	PFISTER J.C.	Physique C.P.E.M.
	CHIBON P.	Biologie Animale
	IDELMAN S.	Physiologie Animale
	BOUVARD Maurice	Hydrologie
	RICHARD Lucien	Botanique
	PELMONT Jean	Physiologie Animale
	BLOCH D.	Electrotechnique I.P.
	BOUSSARD J. Claude	Mathématiques Appliquées I.P.
	MOREAU René	Hydraulique I.P.
	BRUGEL L.	Energétique I.U.T.
	SIBILLE R.	Construction Mécanique I.U.T.
	ARMAND Yves	Chimie I.U.T.
	BOLLINET Louis	Informatique I.U.T.
	KUHN Gérard	Energétique I.U.T.
	GERMAIN J.P.	Construction Mécanique I.U.T.
	CONTE René	Thermodynamique
	JOLY Jean René	Mathématiques Pures
Mme	PIERY Yvette	Biologie Animale
	BENZAKEN Claude	Mathématiques Appliquées

MAITRE DE CONFERENCES ASSOCIES

MM.	SAWCZUK A.	Mécanique des Fluides
	CHEEKE J.	Thermodynamique
	YAMADA O.	Physique du Solide
	NATR Lubomír	B.M.P.V.
	NAVLOK Arch	Physique Industrielle
	SILBER Léo	Radioélectricité
	NOZAKI Akihiro	Mathématiques Appliquées
	RUTLEDGE Joseph	Mathématiques Appliquées
	DONOHÓ Paul	Physique Générale
	EGGER Kurt	B.M.P.V.



Le présent travail a été effectué dans le cadre de contrats passés entre, la Délégation Générale à la Recherche Scientifique et Technique, la Direction des Recherches et Moyens d'Essais, d'une part, et le Service de Mathématiques Appliquées de l'Université de Grenoble d'autre part.



*Je tiens à exprimer ici toute ma reconnaissance à,*

*Monsieur le Professeur J. KUNTZMANN, Directeur du Service de Mathématiques Appliquées de l'Université de Grenoble, qui a dirigé ce travail et qui, par son aide et ses précieux conseils, m'a permis de le mener à bien.*

*Je suis particulièrement sensible à l'honneur qu'il m'a fait en acceptant de présider le Jury.*

*Je tiens à remercier*

*Monsieur le Professeur B. VAUQUOIS, Directeur du Centre d'Etudes pour la Traduction Automatique,*

*Monsieur C. BENZAKEN, Maître de Conférences à la Faculté des Sciences de Grenoble,*

*Monsieur J. LACOUR, Chef du Service de Microélectronique du L.E.T.I., qui ont bien voulu accepter de faire partie du Jury.*

*Je voudrais également remercier Monsieur J.C. SAILLARD, dont la collaboration active et amicale me fut très précieuse, ainsi que les personnels des Laboratoires d'Electronique du C.E.N.G. et de la C.O.S.E.M., qui ont permis, par leurs conseils éclairés, de donner à ce travail un aspect concret.*

*Je ne saurais enfin oublier dans mes remerciements, les personnels des Services Administratifs et Techniques de l'I.M.A.G. qui ont assuré la réalisation matérielle de cette thèse.*



# TABLE DES MATIERES

## INTRODUCTION

PREMIERE PARTIE : Etude de la planéarité pour les circuits intégrés.

CHAPITRE I : Méthodes relatives à la planéarité des graphes.

1. Rappels sur les réseaux de noeuds et d'étoiles.	5
1.1. Définitions.	5
1.1.1. Réseau d'articulations.	5
1.1.2. Réseau de noeuds et d'étoiles.	6
1.1.3. Graphe.	7
1.2. Planéarité d'un graphe.	7
1.2.1. Théorème de KURATOWSKI.	8
1.2.2. Graphes planaires.	9
1.2.3. Simplifications.	9
1.3. Faces d'une représentation plane topologique.	9
1.3.1. Ensemble de sommets extérieurs.	9
1.3.2. Ensemble de sommets extérieurs dans un certain ordre.	12
1.3.3. Ensemble de sommets extérieurs, formant deux sous-ensembles disjoints, les sommets d'un seul sous-ensemble étant dans un certain ordre.	13
1.4. Planéarité d'un réseau.	16
1.4.1. Condition d'orientation dans un réseau planaire.	16
1.4.2. Théorème.	16
2. Graphes non planaires.	17
2.1. Introduction - Position du problème.	17
2.1.1. Définition.	17
2.1.2. Une condition suffisante de non planéarité.	17
2.1.3. Coût d'une représentation d'un graphe non planaire.	17
2.1.4. Vecteur booléen pour une figure de KURATOWSKI.	18
2.1.5. Ensembles d'arêtes à supprimer d'un graphe G, en vue de le rendre planaire.	19

2.2. Recherche des figures de KURATOWSKI.	20
2.2.1. Simplifications.	20
2.2.2. Définitions.	21
2.2.3. Figures de KURATOWSKI de type 2(K2).	23
2.2.3.1. Algorithme d'obtention d'une nouvelle figure K2, à partir d'une figure déjà répertoriée.	23
2.2.3.2. Algorithme général d'obtention des figures de type K2.	27
2.2.3.3. Figures de K2 non atteintes par l'algorithme général.	30
2.2.4. Figures de KURATOWSKI de type 1(K1).	33
2.2.4.1. Un élément rajouté à une figure de K1.	33
2.2.4.2. Deux éléments rajoutés à une figure de K1.	34
 <b><u>CHAPITRE II</u> : Application à la réalisation des circuits intégrés.</b>	
1. Introduction.	37
2. Les différentes étapes de l'étude.	37
3. Planéarité des réseaux électroniques.	38
3.1. Passage du schéma électronique à un réseau de noeuds et d'étoiles.	38
3.2. Passage du réseau de noeuds et d'étoiles à un graphe.	41
3.3. Coût de la réalisation d'un réseau électronique.	41
4. Possibilités offertes par la réalisation du transistor.	43
4.1. Du point de vue topologique.	43
4.2. Du point de vue électronique.	46
4.3. Comment tenir compte de ces possibilités.	46
 <b><u>CHAPITRE III</u> : Programmes - Résultats.</b>	
1. Ensembles d'arêtes à supprimer d'un graphe G afin de le rendre planaire.	48
1.1. Organigramme pour la recherche des figures de type K2.	49
1.2. Programmes.	52
1.2.1. Langage de programmation.	52
1.2.2. Choix d'un codage.	52
1.2.3. Sous-programmes standards.	53
1.2.4. Procédures NEWKURA 2.	53
1.2.5. Programmes généraux.	62

1.3. Résultats.	62
1.4. Conclusions.	66

DEUXIEME PARTIE : Implantation des éléments électroniques.

<u>INTRODUCTION</u>	69
---------------------	----

CHAPITRE I : La dispersion barycentrique.

1. Position du problème.	70
2. Déroulement de l'algorithme.	70
3. Méthode de résolution par relaxation.	71
3.1. Généralités.	71
3.2. Principes de la méthode.	71
4. Méthodes directes de résolution d'un système linéaire.	71
4.1. Le système linéaire.	71
4.2. Les méthodes de résolution utilisées.	73

CHAPITRE II : Non chevauchement.

1. Définitions.	75
2. Déroulement de l'algorithme.	75
3. Algorithme de non chevauchement.	76

CHAPITRE III : Programmes - Résultats.

1. Dispersion barycentrique.	77
1.1. Parties communes aux trois méthodes.	77
1.1.1. Les entrées.	77
1.1.2. La procédure de sortie.	79
1.1.3. Les sous-programmes de traitement.	80
1.2. Méthode de relaxation.	82
1.3. Méthode de GAUSS-SEIDEL.	82
1.3.1. Sous-programme de traitement.	82
1.3.2. Le programme.	83
1.3.3. Résultats.	83

1.4. Méthode de CHOLESKI.	83
1.4.1. Sous-programme de traitement.	83
1.4.2. Le programme.	85
1.4.3. Résultats.	85
1.5. Conclusions.	85
2. Non chevauchement.	85
2.1. Les entrées.	85
2.2. Procédures de sortie.	87
2.3. Sous-programme de traitement.	89
2.4. Résultats.	91
2.5. Conclusions.	91
<u>TROISIEME PARTIE</u> : Réalisation des interconnexions.	
<u>INTRODUCTION</u>	93
<u>CHAPITRE I</u> : Tracé automatique d'équipotentielles.	
1. Position du problème.	94
2. Algorithme de tracé d'équipotentielles.	94
2.1. Simplifications éventuelles.	95
2.2. Algorithme général.	96
3. Programme - Résultats.	97
3.1. Sous-programmes standards.	98
3.2. Procédures de sortie.	99
3.3. Sous-programmes de traitement.	99
3.4. Le programme général.	101
3.5. Résultats - Conclusions.	101
<u>CHAPITRE II</u> : Traçage guidé des connexions par évitement d'obstacles.	
1. Position du problème.	102
2. Algorithme de tracé.	102
3. Programme - Résultats.	103

3.1. Les données.	103
3.2. Le programme.	104
3.3. Résultats.	106

QUATRIEME PARTIE : Le programme DESMAG.

<u>INTRODUCTION</u>	109
---------------------	-----

CHAPITRE I : Description des divers modules.

1. Planéarité.	111
2. Dispersion barycentrique.	111
3. Non chevauchement.	112
4. Dessin sur ordinateur.	112
5. Tracé automatique d'équipotentiellles.	113
6. Traçage guidé des connexions par évitement d'obstacles.	113
7. Gestion de la liste des noeuds.	113
8. Conformité au schéma de principe.	114

CHAPITRE II : Quelques réalisations.

1. Double porte, type CENG.	115
1.1. Etude de la planéarité.	115
1.2. Dispersion barycentrique des éléments.	117
1.3. Non chevauchement.	118
1.4. Tracé automatique d'équipotentiellles.	120
1.5. Traçage guidé des connexions.	120
1.6. Circuit intégré avant ventilation des masques.	122
2. Bascule J.K. Maître-esclave-TTL normal, type COSEM.	123
2.1. Etude de la planéarité.	123
2.2. Dispersion barycentrique des éléments.	125
2.3. Non chevauchement.	126
2.4. Traçage guidé des connexions - Résultat final.	127

CONCLUSION	130
------------	-----

BIBLIOGRAPHIE	131
---------------	-----



I N T R O D U C T I O N



Le présent travail a pour but de fournir, aux concepteurs de circuits intégrés, un ensemble de programmes, dont l'utilisation, aux différents stades de la conception d'un circuit, permet de résoudre rapidement certains problèmes, qui, jusqu'alors, n'étaient traités que de façon empirique.

Un circuit intégré est un réseau formé d'éléments électroniques interconnectés et associés définitivement sur une lamelle de matériau semi-conducteur, appelé substrat. Nous nous intéresserons particulièrement aux circuits intégrés monolithiques (à semi-conducteur) : ses éléments sont réalisés sur la même pastille par dopages successifs à l'intérieur de celle-ci, les interconnexions des éléments sont faites par dépôt d'aluminium sur la surface de la pastille.

Nous nous sommes attachés, dans la mise au point du programme DESMAG, à l'aspect conversationnel, afin de permettre, pendant le déroulement du programme, qu'un dialogue homme-machine puisse s'établir. DESMAG, de ce fait, s'inscrit dans l'optique de la Conception Assistée par Ordinateur (C.A.O.).

Pour présenter cette étude, dont le départ est le schéma électronique du circuit à réaliser et dont le point final est la topologie complète du circuit intégré, nous avons fractionné l'exposé en quatre parties :

- La 1<sup>ère</sup> partie est une étude du problème posé par les croisements des connexions. Nous avons schématisé la réalité en assimilant le circuit électronique à un réseau de noeuds et d'étoiles dans lequel les divers éléments sont assimilés à des points. Nous nous sommes efforcés de résoudre les deux problèmes suivants :

a) reconnaître si un graphe est planaire (la solution est classique).

b) si un réseau n'est pas planaire, chercher à le rendre planaire en minimisant le nombre des croisements.

- La 2<sup>ième</sup> partie donne une solution au problème de la disposition des éléments électroniques sur une surface plane, de façon à obtenir la meilleure occupation possible de cette dernière.

- Dans la 3<sup>ième</sup> partie nous nous intéressons à la réalisation des interconnexions entre éléments.

- La 4<sup>ième</sup> partie est la présentation de DESMAG, résultat de l'étude. Nous y indiquons l'enchaînement des divers programmes le composant, l'interaction de ses différentes parties et nous y présentons **deux exemples de réalisation**.

Le programme DESMAG est le résultat d'un travail d'équipe où MM. FAYOLLE et PEYRACHON, lors d'un stage à l'I.M.A.G., ont commencé à étudier la partie concernant l'implantation des éléments.

Par la suite, avec M. SAILLARD, nous avons poursuivi l'étude et élaboré l'ensemble des programmes qui constituent DESMAG.

Pour ma part j'ai fait l'étude théorique et mis au point les programmes de la première partie et du tracé des connexions (3<sup>ième</sup> partie) et j'ai participé à la mise au point, lors du traitement d'exemples, des autres programmes.

# P R E M I E R E   P A R T I E

## ETUDE DE LA PLANEARITE POUR LES CIRCUITS INTEGRES

Chapitre I - Méthodes relatives à la planéarité des graphes.

Chapitre II - Application à la réalisation des circuits intégrés.

Chapitre III - Programmes - Résultats.



## CHAPITRE I

### METHODES RELATIVES A LA PLANEARITE DES GRAPHES

#### 1 - RAPPELS SUR LES RESEAUX DE NOEUDS ET D'ETOILES

##### 1.1. Définitions.

##### 1.1.1. Réseau d'articulations.

L'articulation est l'élément de base des réseaux. Sa structure interne n'est pas détaillée et elle peut communiquer avec l'extérieur par un nombre fini de connexions.

Degré d'une articulation = nombre de ses connexions.

##### Réseau d'articulations.

Constitué par deux ensembles :

- un ensemble d'articulations  $A = \{a_1, a_2, \dots, a_n\}$  en nombre fini
- un exemple  $C$  de couples de connexions tel que : toute connexion fait partie d'un couple au plus.

Réseau fermé : si toute connexion appartient à un couple et un seul.

Relation de jointivité :  $J(a_i, a_j)$  du réseau.

Relation sur  $A \times A$  définie par :

Deux articulations,  $a_i$  et  $a_j$ , seront dites jointives s'il existe au moins un couple de connexions  $(C_k, C_l) \in C$ ,  $C_k$  et  $C_l$  appartenant respectivement à  $a_i$  et  $a_j$ .

Constituants connexes du réseau : classes d'équivalence définies dans A par la fermeture transitive forte de J.

Réseau connexe : n'a qu'un constituant connexe.

k Fracture d'un réseau connexe d'articulations : Un ensemble k minimal d'articulations, qui supprimées du réseau, ainsi que leurs connexions, rendent le réseau ainsi obtenu non connexe.

Charnière : fracture composée d'une seule articulation.

Chaîne : séquence d'articulations

$(a_1, a_2, \dots, a_i, \dots, a_n)$ ;  $a_i$  jointive à  $a_{i+1}$  pour

$\forall_i \neq n$ .

Cycle : chaîne où  $a_1 = a_n$ ;

Chaîne ou cycle élémentaire : toutes les articulations  $a_i$  sont distinctes.

### 1.1.2. Réseau de noeuds et d'étoiles.

C'est un réseau d'articulations où ces dernières sont partagées en deux classes disjointes : les étoiles et les noeuds de telle façon que si deux articulations sont jointives, l'une est un noeud, l'autre est une étoile.

La notion de réseau qui a l'avantage d'être assez prêt de la réalité physique peut se ramener à celle de graphe. Il suffit de nommer :

- sommet un noeud ou une étoile
- branche un couple de connexions jointives.

1.1.3. Graphe.

Défini par  $\left[ \begin{array}{l} \text{un ensemble de sommets } X = \{x_1, x_2, \dots, x_n\} \\ \text{une application multivoque } \Gamma \text{ de } X \text{ dans } X, \text{ c'est-à-dire} \end{array} \right.$   
 une loi qui fait correspondre à tout élément  $x \in X$ , un sous-ensemble bien défini de  $X$ , que l'on note  $\Gamma x$ ; on peut avoir  $\Gamma x = \emptyset$ .

L'ensemble  $U$  des arêtes du graphe est défini par :

$u_{ij} \in U$  si, étant donné le couple des sommets  $x_i$  et  $x_j$ ,  $x_i \in \Gamma(x_j)$  et  $x_j \in \Gamma(x_i)$ .

On peut noter un graphe  $G(X, \Gamma)$ ;

Sous-graphe d'un graphe  $G(X, \Gamma)$ ;

C'est la restriction du graphe à certains sommets, définie par :

$$(X', \Gamma_{X'}) \text{ tel que } \left| \begin{array}{l} X' \subset X \\ \Gamma_{X'}(x_i) = \Gamma(x_i) \cap X' \end{array} \right.$$

Graphe partiel d'un graphe  $G(X, \Gamma)$  : restriction du graphe à certaines arêtes, définie par :

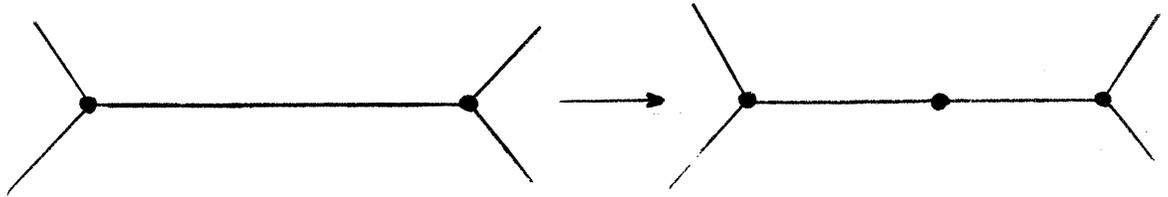
$$(X, \Delta) \text{ tel que } \Delta(x_i) \subset \Gamma(x_i), \forall_i$$

Sous-graphe partiel d'un graphe  $G(X, \Gamma)$  : défini par :  $(X', \Delta_{X'})$   
 tel que  $\left| \begin{array}{l} X' \subset X \\ \Delta_{X'}(x_i) \subset \Gamma(x_i) \cap X' \end{array} \right.$

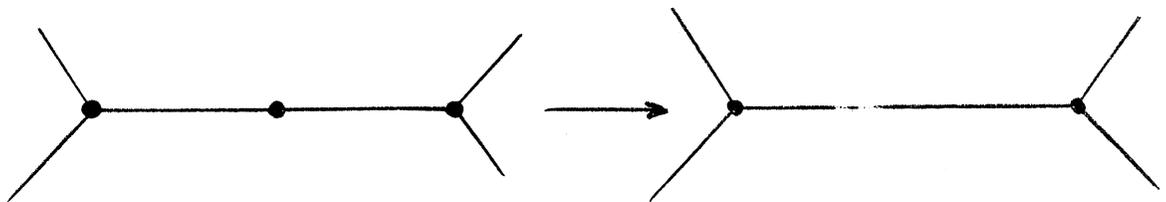
1.2. Planéarité d'un graphe.

La planéarité ou la non planéarité d'un graphe n'est pas affectée par les deux opérations suivantes :

- une arête est divisée en 2 arêtes par adjonction d'un sommet de degré 2.



- deux arêtes incidentes à un même sommet de degré 2 sont transformées en 1 arête, par suppression de ce sommet.



Deux graphes  $G_1$  et  $G_2$ , que l'on peut rendre isomorphes, par applications répétées des deux opérations ci-dessus, seront dits : conformes .

### 1.2.1. Théorème de KURATOWSKI.

Une condition nécessaire et suffisante pour qu'un graphe  $G$  soit planaire est qu'il n'admette pas de sous-graphes partiels, conformes aux deux figures ci-dessous.

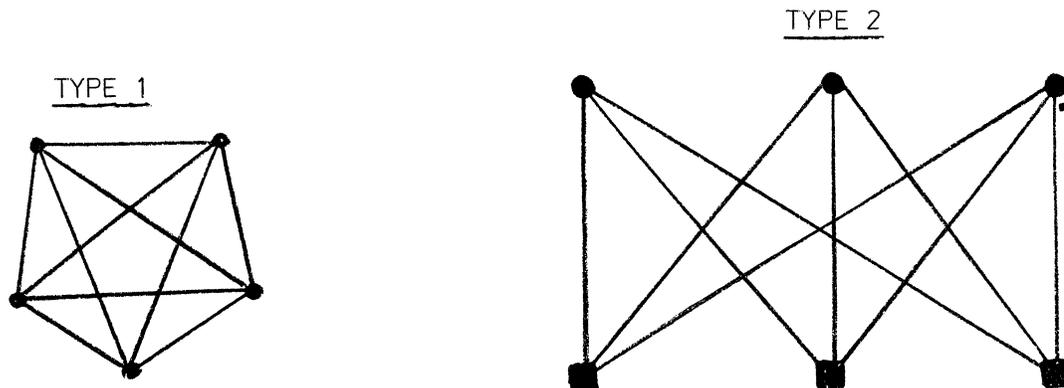


Fig. 1

### 1.2.2. Graphes planaires.

Problèmes de reconnaissance et de construction des graphes planaires topologiques résolus et exposés dans la littérature [10], [12], [17].

### 1.2.3. Simplifications.

Lorsque l'on étudie la planéarité d'un graphe on peut

- supprimer les sommets pendants et l'arête incidente à chacun de ces sommets.
- supprimer les sommets de degré 2 en remplaçant les 2 arêtes incidentes à chacun de ces sommets par une seule arête.
- supprimer les arêtes parallèles, c'est-à-dire incidentes à deux sommets identiques, sauf une.

## 1.3. Faces d'une représentation plane topologique.

### 1.3.1. Ensemble de sommets extérieurs.

#### Lemme 1.

Soit  $D$ , une représentation plane topologique d'un graphe  $G$ . On conserve une représentation plane topologique,  $D'$ , de  $G$  si, par rapport à un cycle  $C$  du graphe, on met tous les sous-graphes partiels de  $G$  intérieurs au cycle  $C$ , à l'extérieur et réciproquement.

Ceci est évident par projection stéréographique.

#### Lemme 2.

Une condition nécessaire et suffisante pour qu'un cycle  $C$  d'un graphe  $G$  plane, soit une face dans une certaine représentation plane topologique est que le graphe  $G'$ , construit à partir de  $G$  de la façon suivante :

- on ajoute le sommet  $x_0$  et on le relie à tous les sommets du cycle  $C$ , soit planaire.

Démonstration.

La condition est nécessaire, en effet  $G$  étant planaire et  $C$  une face dans une représentation planaire topologique, il est alors toujours possible de relier un point  $x_0$ , intérieur à  $C$ , à tous les sommets du cycle  $C$ , sans détruire la planéarité.

La condition est suffisante car,  $G'$  étant planaire, d'après le lemme 1, les seuls éléments intérieurs à  $C$  sont  $x_0$  et des sous-graphes partiels  $g_i$  reliés à deux sommets consécutifs, par rapport au cycle  $C$ .

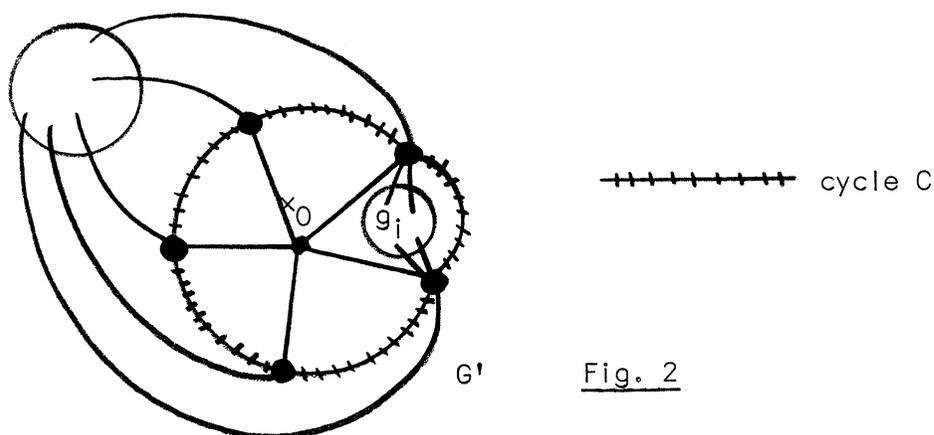


Fig. 2

Il suffit alors de supprimer  $x_0$  et ses arêtes incidentes, et de mettre les sous-graphes partiels  $g_i$  à l'extérieur de  $C$  pour que ce dernier soit face de  $G$ .

Lemme 3.

Toute face d'une représentation planaire topologique  $D$  d'un graphe planaire  $G$  peut être face extérieure (frontière de la région infinie).

La démonstration est évidente.

Théorème 1.

Une condition nécessaire et suffisante, pour qu'un ensemble de sommets d'un graphe  $G$  planaire, soit sur la face extérieure d'une certaine représentation planaire topologique est que le graphe  $G'$ , construit à partir de  $G$ , de la façon suivante :

- on ajoute le sommet  $x_0$  et on le relie à tous les sommets de l'ensemble considéré  $\{x_1, x_2, \dots, x_n\}$ , soit planaire.

On notera  $G'$  :

$$G \cup x_0 \cup \{ \text{arêtes } [x_0 x_i], \forall i \text{ de } 1 \text{ à } n \}.$$

Démonstration.

Si  $G'$  est planaire, on peut toujours joindre les  $x_i$ , deux à deux, de façon à obtenir un cycle passant par les seuls  $x_i$  et ne détruisant pas la planéarité; l'ordre des  $x_i$  sur le cycle est alors donné par l'ordre d'adjacence des arêtes  $[x_0 x_i]$  autour du sommet  $x_0$ .

D'après les lemmes 2 et 3 la démonstration est achevée.

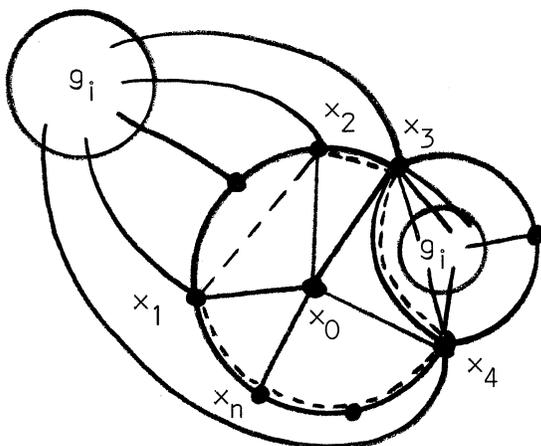


Fig. 3

1.3.2. Ensemble de sommets extérieurs dans un certain ordre.

Le résultat obtenu dans le paragraphe ci-dessous, s'applique aux graphes biconnexes, c'est-à-dire non séparables.

Théorème 2.

Une condition nécessaire et suffisante pour qu'un ensemble de sommets  $\{x_1, x_2, \dots, x_n\}$ , avec  $n > 3$ , d'un graphe planaire  $G$ , biconnexe, soit sur la face extérieure d'une certaine représentation planaire topologique de  $G$ , et dans cet ordre, est que le graphe  $G'$  :

$G' = G \cup x_0 \cup \{arêtes [x_0, x_i], \forall i \text{ de } 1 \text{ à } n\} \cup \text{cycle } (x_1, x_2, \dots, x_n)$  dans cet ordre, soit planaire.

Démonstration.

Soit  $G'$  planaire, considérons une représentation planaire topologique de  $G'$  dans laquelle  $x_0$  est extérieur (il en existe toujours au moins une). Les sommets  $x_i$  sont dans l'ordre  $\{1, 2, \dots, n\}$ .

On obtient les faces  $(x_0, x_i, x_{i+1}) / \forall i \text{ de } 1 \text{ à } n-1$  et la face  $(x_0, x_n, x_1)$ .

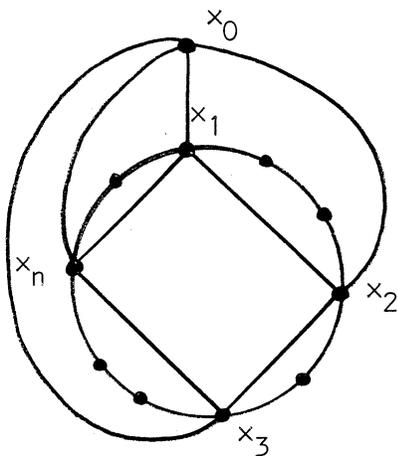


Fig. 4

Nous allons montrer qu'une inversion entre deux  $x_i$  entraîne la non planéarité de  $G'$ .

Supposons l'ordre  $\{2, 1, 3, \dots, n\}$ .

Il y a des chaînes  $(x_1, x_2)$ ,  $(x_1, x_3)$ ,  $(x_1, x_n)$  n'ayant aucun sommet commun à part  $x_1$ .

De même pour  $(x_2, x_3)$  et  $(x_2, x_n)$  et  $(x_3, x_n)$ .

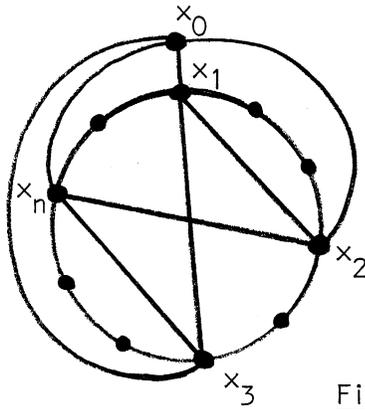


Fig. 5

Comme  $x_0$  est adjacent à  $x_1, x_2, x_3, x_n$ , on voit que l'on peut extraire de  $G'$  un pentagone complet ayant pour sommets  $x_0, x_1, x_2, x_3, x_n$ . Ceci entraîne que  $G'$  est non planaire.

Remarque 1.

Si  $n = 3$  on a toujours l'ordre voulu, éventuellement en retournant la représentation du graphe.

Remarque 2.

Il est bien évident que l'ordre est déterminé à un retournement de représentation, soit dans l'autre sens, près. Ceci apparaît sur la figure 4 où l'on voit que l'ordre des sommets suivant  $\{2, 1, 4, 3\}$  convient et entraîne  $G'$  planaire.

1.3.3. Ensemble de sommets extérieurs, formant deux sous-ensembles disjoints, les sommets d'un seul sous-ensemble étant dans un certain ordre.

Théorème 3.

Une condition nécessaire et suffisante pour qu'un ensemble de sommets  $\{x_1, x_2, \dots, x_k, \dots, x_n\}$  avec  $n > 3$ , d'un graphe planaire  $G$ , biconnexe, soit sur la face extérieure d'une certaine représentation planaire topologique de  $G$ , et certains de ces sommets  $\{x_1, x_2, \dots, x_k\}$  avec  $k \leq n$ , dans cet ordre, est que le graphe  $G'$ , construit à partir de  $G$  de la façon suivante :

- on ajoute le sommet  $x_0$  et on le relie à tous les  $x_i$  (on crée les arêtes  $x_0 x_1, \dots, x_0 x_n$ ).
- on ajoute les sommets  $\{x'_1, x'_2, \dots, x'_k\}$  de degré 2, sur les arêtes joignant  $x_0$  à  $x_1, x_2, \dots, x_k$ , respectivement.
- on relie les sommets  $x'_1, x'_2, \dots, x'_k$ , dans cet ordre, par une chaîne d'arêtes joignant  $x'_i$  à  $x'_{i+1}$ , pour  $\forall i$  de 1 à  $k-1$ .

$$G' = G \cup x_0 \cup (x'_i / i = 1 \text{ à } k) \cup \{\text{arêtes } [x_0 x_i] / i = k+1 \text{ à } n\}$$

$$\cup \{\text{arêtes } [x_0 x'_i] / i = 1 \text{ à } k\} \cup \{\text{arêtes } [x'_i x'_j] / i = 1 \text{ à } k\}$$

$$\cup \{\text{arêtes } [x'_i x'_{i+1}] / i = 1 \text{ à } k-1\}.$$

soit planaire.

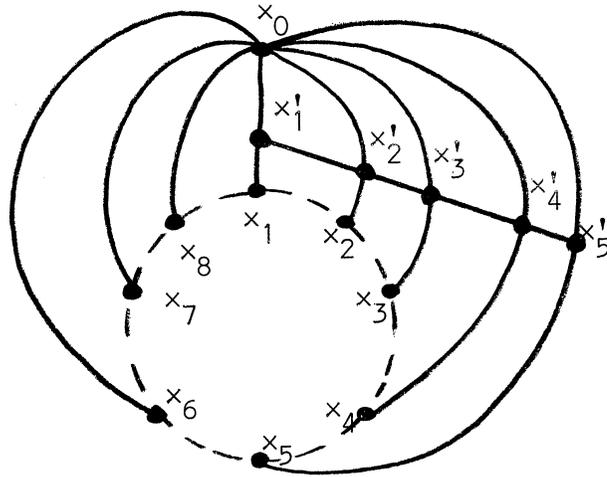


Fig. 6

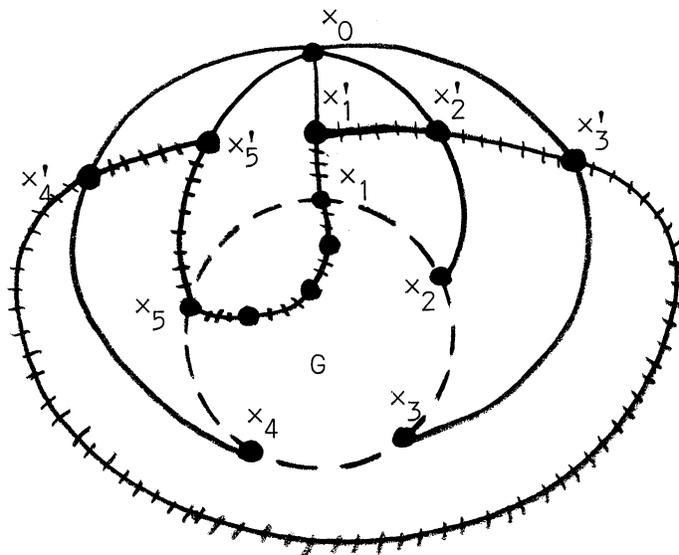
Démonstration.

1<sup>er</sup> cas.  $k = n$ .

$\exists$  un cycle  $x_1, x'_1, x'_2, \dots, x'_n, x_n$  car :

$G$  est non séparable, donc il existe une chaîne dans  $G$  reliant

$x_1$  à  $x_n$ .



cycle arêtes cochées

Fig. 7

D'après le théorème 2, si  $G'$  est planaire alors les  $x_i'$  seront sur la face extérieure d'une certaine représentation planaire topologique de  $(G' - x_0 - \{\text{arêtes } [x_0, x_i] / i\})$  dans un ordre fixé  $\Rightarrow$  les  $x_i$  seront sur la face extérieure d'une certaine représentation planaire topologique de  $G$ , et dans le même ordre que les  $x_i'$ .

La réciproque: les sommets  $x_i$  sont sur la face extérieure d'une certaine représentation planaire topologique de  $G$ , et dans cet ordre

$\Rightarrow G'$  est planaire, est vraie car le théorème 2 est une CNS.

2<sup>ième</sup> cas.  $k < n$ .  $G'$  planaire :

. les arêtes  $[x_0, x_i] / i = k+1 \text{ à } n$ , présente dans  $G'$

$\Rightarrow$  les  $x_i / i = k+1 \text{ à } n$  sont sur la face extérieure d'une certaine représentation planaire topologique de  $G$ , de même que pour les sommets  $x_i / i = 1 \text{ à } k$ .

. De plus ces sommets  $x_i / i = k+1 \text{ à } n$  ne s'intercalent pas, sur une certaine face extérieure, aux sommets  $x_i / i = 1 \text{ à } k$  :  $G$  étant planaire, il n'y a pas de croisement, dans  $G'$ , entre les arêtes  $[x_i', x_{i+1}'] / i = 1 \text{ à } k-1$ , et les arêtes  $[x_0, x_i] / i = k+1 \text{ à } n$ , on pourra alors toujours déplacer, les sommets  $x_i$  sur l'arête  $[x_0, x_i]$  et les arêtes incidentes à  $x_i$ , par déformation continue de celles-ci, de façon à avoir  $x_i$  aussi près que l'on veut de  $x_0$ .

. On pourra alors toujours trouver un cycle passant par  $x_1, x_1', x_2', \dots, x_k', x_k$  et reliant  $x_k$  à  $x_1$  en passant par les sommets  $x_k \dots x_n$ , au besoin en créant les arêtes  $[x_i, x_l] / i$  et  $l \in \{k \dots n\}$ .

Nous sommes alors ramenés au cas précédent.

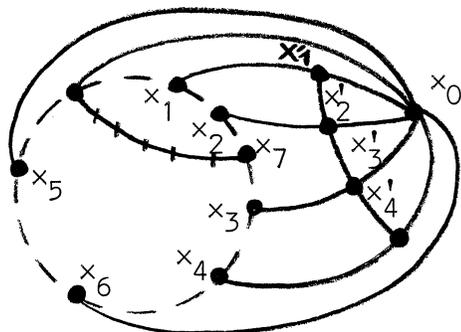


Fig. 8

#### 1.4. Planéarité d'un réseau.

Nous ne citerons ici que des résultats, sans en donner les démonstrations

##### 1.4.1. Condition d'orientation dans un réseau planaire.

Soient deux articulations  $A_1$  et  $A_2$ , et  $n$  chaînes élémentaires, disjointes, les joignant. Le réseau étant planaire, ces chaînes ont la propriété suivante :

On rencontrera autour de  $A_1$  les connexions dans l'ordre direct,  $C_1, C_2, \dots, C_n$  et pour le même ordre, autour de  $A_2$ , on rencontrera  $C_n, C_{n-1}, \dots, C_2, C_1$ . Nous qualifierons cette orientation de cohérente.

##### 1.4.2. Théorème.

Une condition nécessaire et suffisante pour qu'un réseau d'articulations non précisées, soit planaire, est qu'il possède une orientation cohérente, de toutes ses articulations.

## 2 - GRAPHES NON PLANAIRES

### 2.1. Introduction - Position du problème.

#### 2.1.1. Définition.

On dit qu'un graphe est non planaire, s'il est impossible de le représenter sur un plan de telle sorte qu'aucune arête ne rencontre une autre arête en dehors des sommets.

#### 2.1.2. Une condition suffisante de non planéarité [13].

$G$  est un graphe avec  $m$  arêtes,  $n$  sommets. Si  $G$  est tel que :  $m > 3n - 6$ , alors  $G$  est non planaire.

#### 2.1.3. Coût d'une représentation d'un graphe non planaire.

Nous nous intéressons ici, au problème suivant :

Etant donné un graphe  $G$  non planaire, comment en donner une représentation topologique aux "moindres frais".

Nous pouvons donner deux définitions du coût :

- nombre de croisements d'arêtes (2 arêtes au plus par croisement).
- nombre d'arêtes à enlever afin de rendre  $G$  planaire.

Nous choisirons, pour la suite de notre travail, la deuxième définition.

Il est à noter que FISCHER dans la deuxième partie de [12] propose une solution qui consiste à construire une représentation de G à partir d'un cycle et à enlever les arêtes, conduisant à une impossibilité de planéarité, au fur et à mesure de la progression.

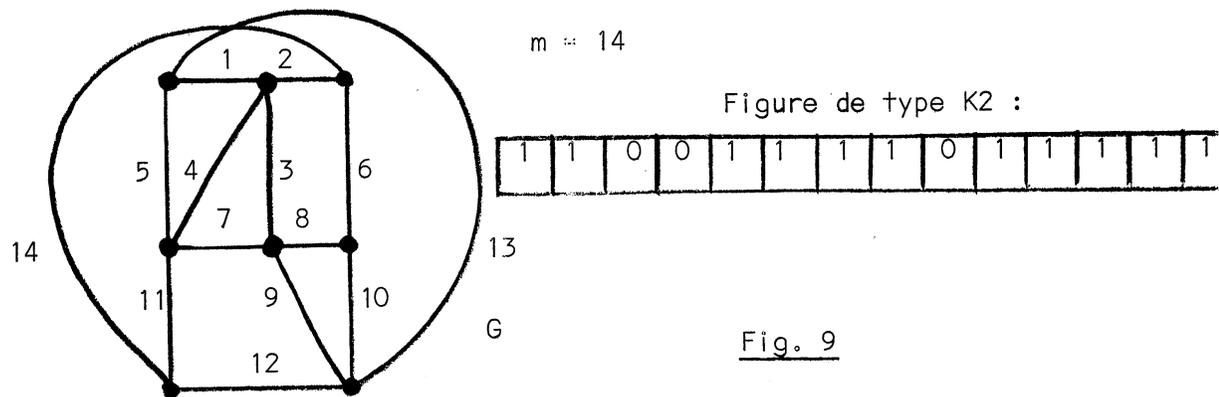
#### 2.1.4. Vecteur booléen pour une figure de KURATOWSKI.

Dans chacune des 2 figures de KURATOWSKI les arêtes ou chaînes d'arêtes, joignant les sommets de degré  $> 2$ , jouent un rôle identique du point de vue de la planéarité.

Nous sommes donc amenés à considérer des ensembles d'arêtes que nous repérerons par un vecteur à m composantes (les arêtes sont numérotées une fois pour toutes). Nous nous placerons sur le corps des entiers modulo deux :

$$E_2 = \begin{cases} 1 \text{ arête présente on lui affecte la valeur } 1 \\ 1 \text{ arête absente on lui affecte la valeur } 0 \end{cases}$$

Exemple.



Une figure de KURATOWSKI peut alors correspondre biunivoquement à un vecteur booléen de l'espace vectoriel  $E_2^m$ .

2.1.5. Ensembles d'arêtes à supprimer d'un graphe G, en vue de le rendre planaire.

Nous allons chercher tous les sous-graphes partiels de G conformes aux deux figures de KURATOWSKI.

A chaque sous-graphe nous ferons correspondre une ligne d'un tableau T à m colonnes.

Pour toute ligne, une colonne représente une arête. Nous noterons 1 la présence de l'arête dans la figure de KURATOWSKI et 0 dans le cas contraire.

Lorsque nous aurons obtenu toutes les figures de KURATOWSKI qu'on peut extraire de G alors :

Pour rendre le graphe G planaire, il faut détruire toutes les figures de KURATOWSKI que contient G.

Une figure de KURATOWSKI peut être détruite par suppression de l'une quelconque de ses arêtes. Ceci revient donc à supprimer une arête présente (un 1) dans toutes les lignes du tableau T.

De plus, une arête supprimée du graphe G, détruit toutes les figures de KURATOWSKI dans lesquelles elle apparaît.

Notre problème revient donc à chercher un ensemble minimal d'arêtes, donc de colonnes de T, qui, par la présence de 1, couvre toutes les lignes du tableau T.

Exemple.

tableau T.

arêtes de G

a	b	c	d	e	f	g	h	i	j	k	l
1	1	1		1	1	1	1	1			1
	1	1		1	1	1	1	1		1	1
1	1		1	1		1	1		1	1	1
1		1	1		1	1	1	1		1	1

sous-graphes partiels de G conformes aux figures de KURATOWSKI

l'ensemble minimal pour cet exemple est = {arête g}.

Remarque.

Ne possédant pas de méthode rigoureuse de recherche des figures de KURATOWSKI, nous appliquerons une méthode heuristique, selon un processus itératif.

2.2. Recherche des figures de KURATOWSKI.

2.2.1. Simplifications.

- Si le graphe possède une 1-fracture, on peut traiter séparément les deux sous-graphes obtenus, à partir de  $G$ , après dédoublement de la charnière, et en introduisant une condition supplémentaire afin que dans une représentation topologique, le point dédoublé apparaisse sur la face extérieure de chaque sous-graphe. (ce qui est toujours possible).

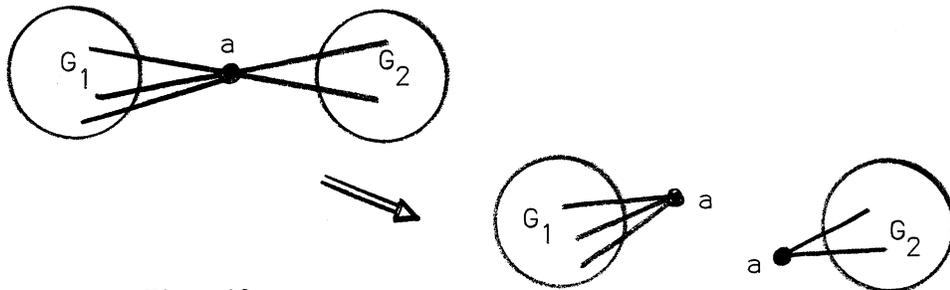


Fig. 10

- On peut généraliser le procédé lorsqu'il y a des 2-fractures de la façon suivante :

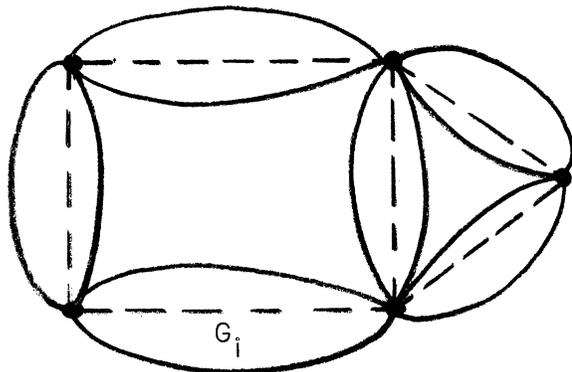


Fig. 11

en pointillé le pseudo-graphe des 2-fractures (analogie avec l'arbre des charnières).

$G_i$  : les ensembles connexes des 2-fractures. On est donc ramené à extraire les figures trouvées en partant du pseudo-graphe et les figures appartenant à chaque ensemble connexe car : le pseudo-graphe doit d'abord être planaire; ensuite chaque ensemble connexe doit être planaire, les deux charnières appartenant à la face extérieure.

### 2.2.2. Définitions.

Soient  $G$  un graphe non planaire et  $G_0$  une figure de KURATOWSKI de type 2 (ce qui suit reste valable pour une figure de type 1),  $G_0$  étant un sous-graphe partiel de  $G$ .

#### Exemple.

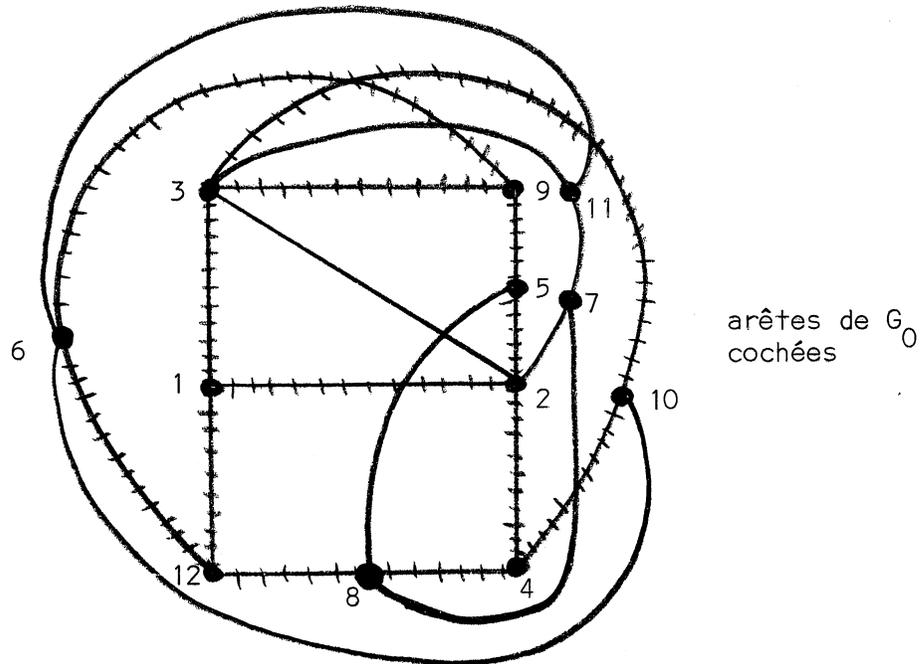


Fig. 12

On appelle pièce de  $G$  relativement à  $G_0$  :

- une arête  $\in G_0$  et dont les 2 sommets extrémité appartiennent à l'ensemble des sommets de  $G_0$ .

- une composante connexe du sous-graphe partiel engendré par  $(G - G_0)$  augmentée des sommets extrémité de  $(G - G_0)$  et appartenant à  $G_0$ .

Exemple de la figure 12.

Soit donc le sous-graphe partiel  $G_0$  suivant :

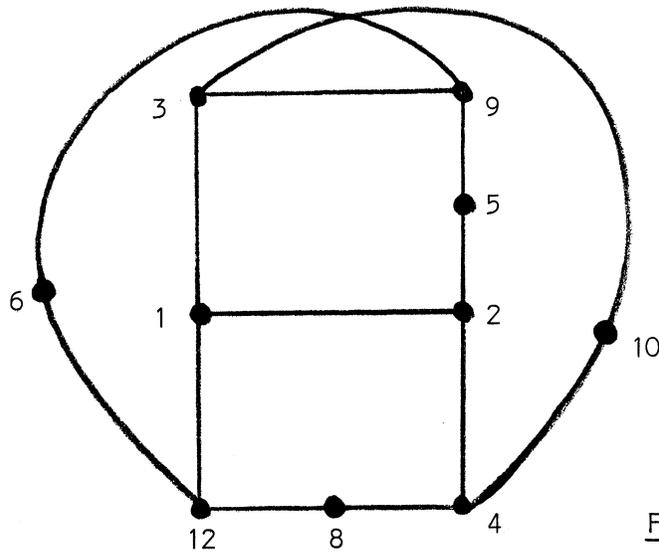
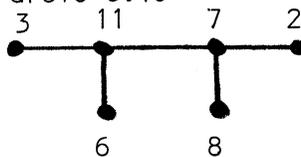


Fig. 13

- On obtient les pièces :
1. arête 2.3
  2. arête 5.8
  3. arête 6.10
  4. 

Lorsque l'on a obtenu les pièces, nous définissons alors les éléments de  $G$  relativement à  $G_0$ , à partir des pièces, de la manière suivante :

<u>Pièce</u>	→	<u>Élément</u>
1 arête	→	1 arête
1 composante connexe	→	Les chaînes élémentaires disjointes obtenues pour $\forall$ couple de sommets $i, j$ tels que :

$$i, j \in \left. \begin{array}{l} \text{composante connexe} \\ G_0 \end{array} \right|$$

Exemple de la figure 12.

<u>Pièces</u>		<u>Eléments</u>
1.		arête 2.3
2.		arête 5.8
3.		arête 6.10
	→	
4.	<u>chaînes</u>	$\left\{ \begin{array}{l} 2.7.8 \\ 2.7.11.6 \\ 2.7.11.3 \\ 8.7.11.6 \\ 8.7.11.3 \\ 6.11.3 \end{array} \right.$

- Figure de KURATOWSKI  $G_0$  totale :

Si  $G_0$  est un graphe partiel de  $G$  contenant tous les sommets de  $G$ . Les éléments de  $G$ , relativement à  $G_0$ , ne sont alors que des arêtes.

2.2.3. Figures de KURATOWSKI de type 2 (K2).

Par un procédé itératif nous allons, à chaque pas, rajouter à toute figure déjà répertoriée ( $G_0$  au 1<sup>er</sup> pas) un des éléments définis sur  $G$  à partir de  $G_0$  choisie, et extraire du graphe ainsi constitué, les nouvelles figures, non encore répertoriées.

2.2.3.1. Algorithme d'obtention d'une nouvelle figure K2, à partir d'une figure déjà répertoriée.

Soit  $G$  le graphe à étudier.  $G_0$  et les éléments de  $G$ , relativement à  $G_0$ , ont été définis une fois pour toutes, comme exposé en (2.2.2.).

Supposons que nous ayons déjà plusieurs figures de K2 répertoriées (comme sous-graphes partiels de G) en plus de  $G_0$ , figures de K2 obtenues par application de l'algorithme.

Définition. Classes de sommets pour une figure de type K2.

Une classe de sommets est constituée :

- soit par 2 sommets de degré 3, si ces sommets sont joints par une arête.
- soit par 2 sommets de degré 3, avec en plus les sommets de degré 2, qui appartiennent à la chaîne d'arêtes joignant les 2 sommets de degré 3.

On obtient donc 9 classes différentes pour une figure de type K2.

Algorithme.

Nous allons ajouter à une figure K2, répertoriée (qui peut être  $G_0$ ), un élément pris dans la liste des éléments de G, relativement à  $G_0$  (l'élément est soit une arête, soit une chaîne d'arêtes).

Cinq cas peuvent se produire : (l'élément rajouté figure en pointillé).

1<sup>er</sup> cas.

L'élément ajouté relie 2 sommets de degré 3, appartenant à des classes différentes. Aucune nouvelle figure ne peut être déduite dans ce cas.

Exemple.

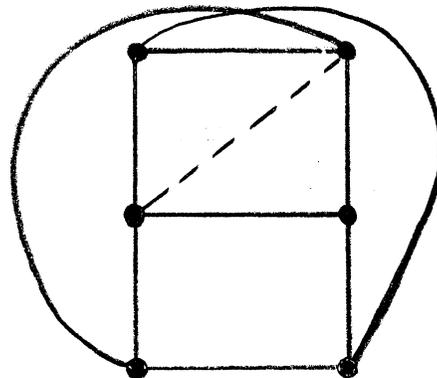


Fig. 14.a

2<sup>ième</sup> cas.

L'élément ajouté relie 2 sommets appartenant à la même classe :  
"arête ou chaîne, parallèles".

Nous obtenons 1 figure nouvelle :

Exemple.

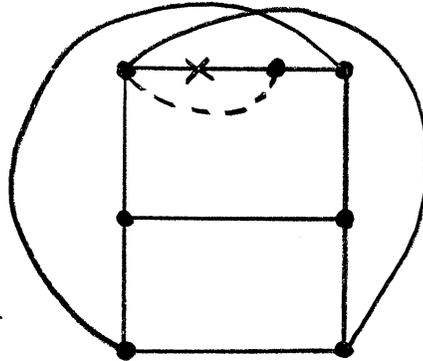


Fig. 14.b

Figure de départ - arête  
cochée + élément.

3<sup>ième</sup> cas.

L'élément ajouté relie 2 sommets, de degrés différents et, appartenant  
à des classes différentes.

Nous obtenons 1 figure nouvelle :

Exemple.

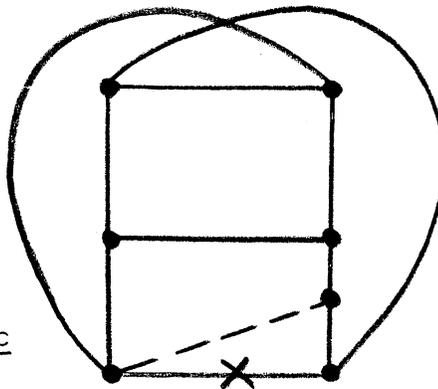


Fig. 14.c

Figure de départ - arête  
cochée + élément.

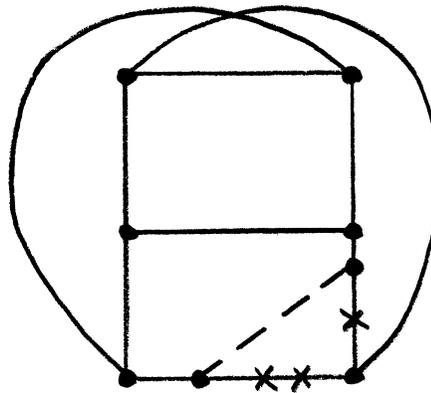
4<sup>ième</sup> cas.

L'élément ajouté relie 2 sommets, dont le degré = 2, appartenant à des classes différentes.

4a. Les 2 classes auxquelles appartiennent les sommets (de l'élément) sur la figure de  $K_2$ , ont un sommet de degré 3 en commun.

Nous obtenons 2 figures nouvelles :

Exemple.



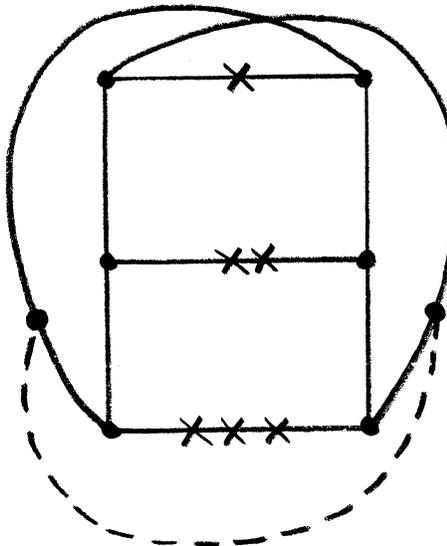
- figure de départ - arête cochée (X) + élément.
- figure de départ - arête cochée (XX) + élément.

Fig. 14.d

4b. Les deux classes n'ont aucun sommet en commun.

Nous obtenons 3 figures nouvelles :

Exemple.



- figure de départ - arête cochée (X) + élément.
- figure de départ - arête cochée (XX) + élément.
- figure de départ - arête cochée (XXX) + élément.

Fig. 14.e

### 2.2.3.2. Algorithme général d'obtention des figures de type K2.

Nous partons de  $G_0$ , sous-graphe partiel de  $G$ , conforme à la figure de KURATOWSKI de type K2, et de la liste des  $N$  éléments de  $G$  relativement à  $G_0$ .

#### 1<sup>er</sup> balayage.

- s'effectue par pas  $i/i = 1$  à  $N$ .

- 1<sup>er</sup> pas. Nous ajoutons à  $G_0$  le 1<sup>er</sup> élément de la liste. Nous obtenons (Algorithme 2.2.3.1.)  $k_1^1$  figures nouvelles.

- Au pas  $i$ , nous ajoutons aux figures déjà trouvées  $(1 + k_1^1 + \dots + k_{i-1}^1)$  le  $i^{\text{ème}}$  élément. 

Nous trouvons  $k_i^1$  figures nouvelles.

#### 2<sup>ième</sup> balayage, puis balayages pairs.

- s'effectue par pas  $i/i = N - 1$  à  $1$ .

- 1<sup>er</sup> pas. Nous ajoutons le  $(N-1)^{\text{ème}}$  élément aux  $k_N^1$  figures obtenues à la suite du  $N^{\text{ème}}$  pas du 1<sup>er</sup> balayage (ou du  $(N-1)^{\text{ème}}$  pas du balayage impair précédent).

Nous obtenons  $k_{N-1}^2$  figures nouvelles.

- Au  $j^{\text{ème}}$  pas du 2<sup>ième</sup> balayage (ou d'un balayage pair) nous ajoutons le  $(N-j)^{\text{ème}}$  élément aux :

$$(k_{N-j+1}^1 + k_{N-j+2}^1 + \dots + k_N^1 + k_{N-1}^2 + \dots + k_{N-j+1}^2)$$

figures répertoriées.

Nous obtenons  $k_{N-j}^2$  figures nouvelles.

$i = N-j;$

3<sup>ième</sup> balayage, puis balayages impairs.

- s'effectue par pas  $i/i = 2$  à  $N$ .

- 1<sup>er</sup> pas. Nous ajoutons le 2<sup>ième</sup> élément aux  $k_1^2$  figures obtenues à la suite du  $(N-1)$ <sup>ième</sup> pas du 2<sup>ième</sup> balayage (ou du balayage pair précédent).

Nous obtenons  $k_2^3$  figures nouvelles.

- Au j<sup>ième</sup> pas du 3<sup>ième</sup> balayage (ou d'un balayage impair) nous ajoutons le  $(j+1)$ <sup>ième</sup> élément aux :

$$(k_j^2 + k_{j+1}^2 + \dots + k_1^2 + k_2^3 + \dots + k_j^3)$$

figures répertoriées.

Nous obtenons  $k_{j+1}^3$  figures nouvelles.

$$i = j+1;$$

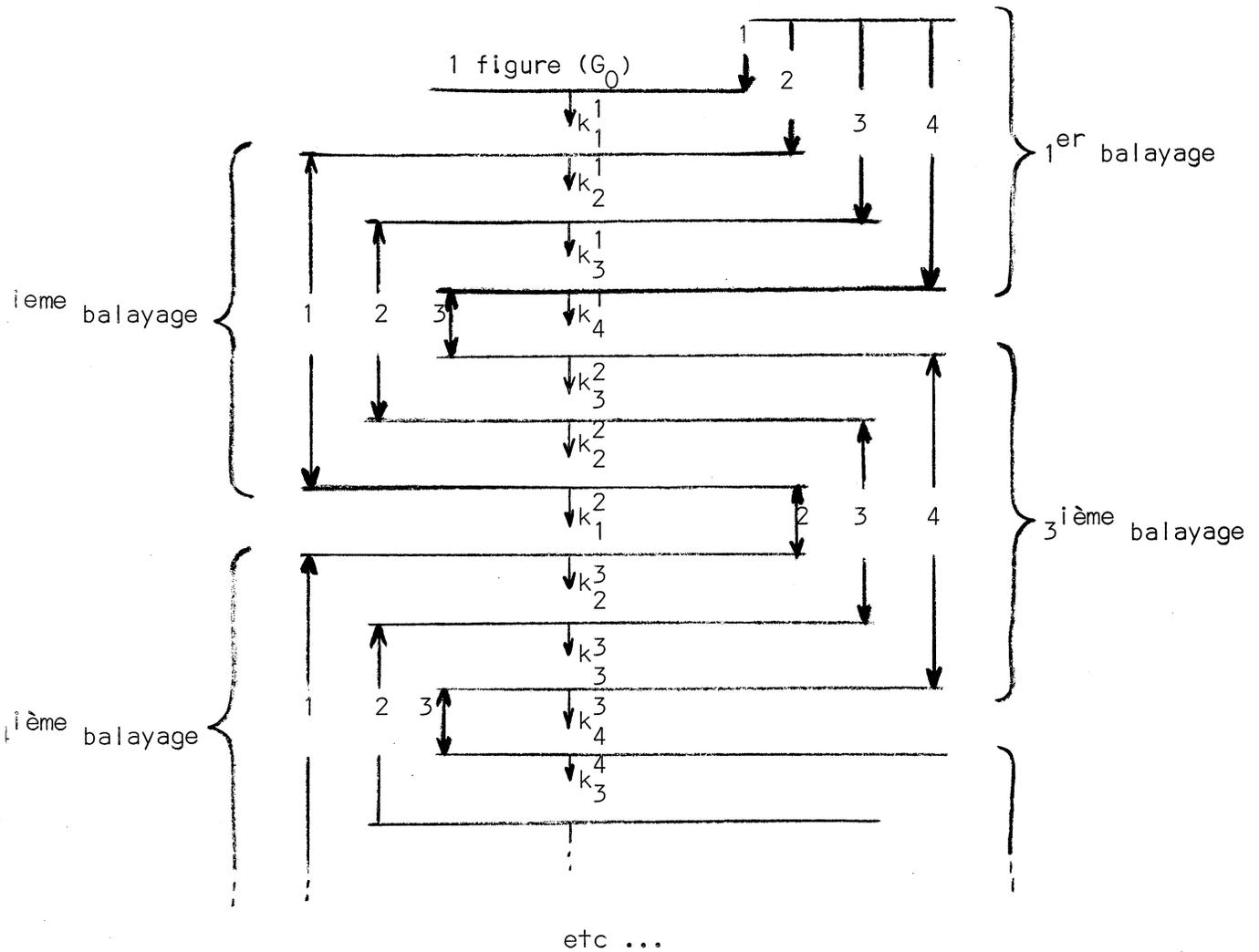
Le procédé continue par enchaînement des balayages pairs et des balayages impairs.

Le processus prend fin lorsque, au cours d'un balayage complet, le nombre total de figures nouvelles, pour l'ensemble du balayage est égal à 0.

Montrons sur un exemple l'enchaînement du processus :

Soit :  $G_0$  donné, 4 éléments de  $G$  relativement à  $G_0$ .

application de  
l'élément n°



Le processus s'arrêtera lorsque au  $l^{\text{ième}}$  balayage :

$$l \text{ pair} : k_1^l + k_2^l + k_3^l = 0$$

$$l \text{ impair} : k_2^l + k_3^l + k_4^l = 0$$

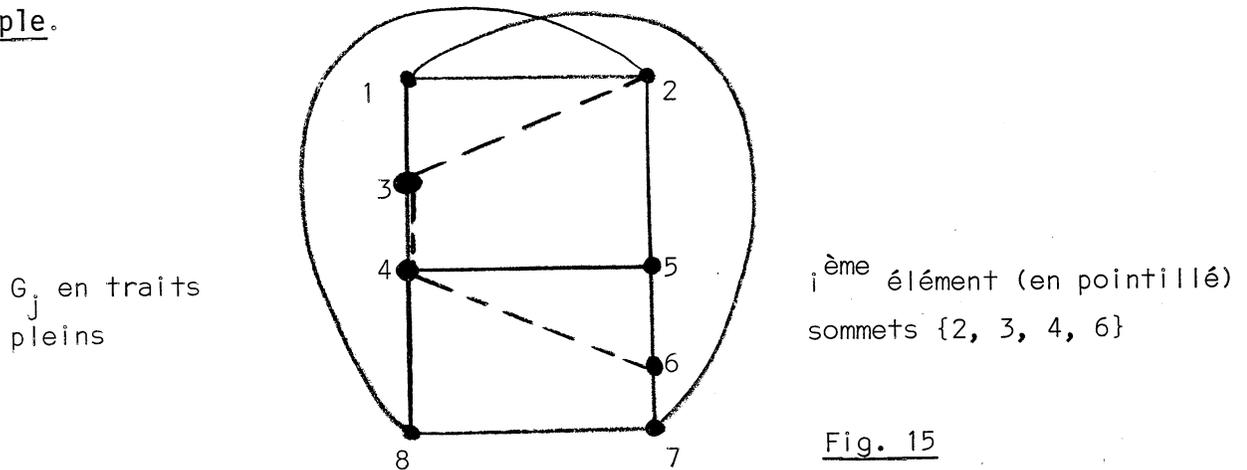
Remarque.

Lorsque l'on ajoute le  $i^{\text{ième}}$  élément à la  $j^{\text{ième}}$  figure de type K2, répertoriée,  $G_j$  :

- si le  $i^{\text{ème}}$  élément est une arête, ou une chaîne d'arêtes dont aucun sommet n'appartient à  $G_j$ , en dehors des 2 sommets extrémité, on applique l'algorithme tel qu'il est décrit en 2.2.3.1..

- si le  $i^{\text{ème}}$  élément est une chaîne d'arêtes dont, au moins 3 sommets (dont les 2 sommets extrémité) appartiennent à  $G_j$ , alors on traite séparément (comme des éléments différents) les différentes arêtes ou chaînes d'arêtes du  $i^{\text{ème}}$  élément, composantes connexes, relativement à  $G_j$ .

Exemple.



Le  $i^{\text{ème}}$  élément, appliqué à  $G_j$  donne en fait lieu à l'application des 2 éléments

$$i_1 = \{2, 3\}$$
$$\text{et } i_2 = \{4, 6\} \text{ sur } G_j.$$

### 2.2.3.3. Figures de K2 non atteintes par l'algorithme général.

1<sup>er</sup> cas.

4 éléments à rajouter relient, respectivement, 2 sommets de degré 3, appartenant à des classes différentes, et rajoutés à une même figure de K2 donnent une nouvelle figure de K2.

Exemple.

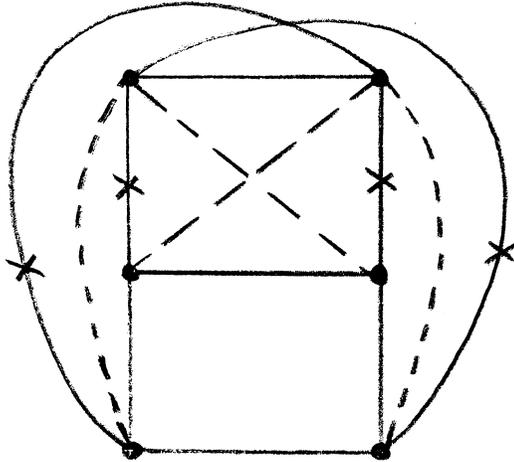


Fig. 16

Nouvelle figure de K2 :

- figure de départ
- + éléments pointillés
- classes cochées

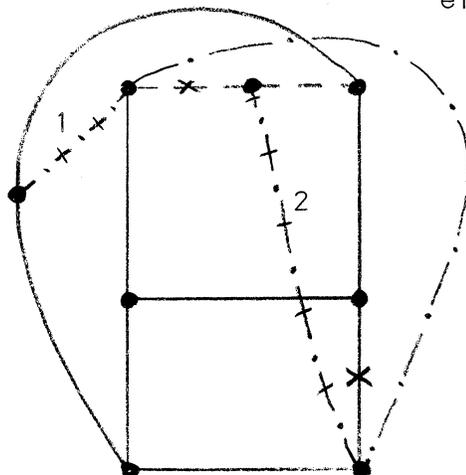
Remarque.

3 éléments du type ci-dessus suffisent d'ailleurs à créer un pentagone complet (figure de type K1).

2<sup>ième</sup> cas.

Des figures de K2 peuvent ne pas être découvertes par l'algorithme général. En effet, prenons le cas de 2 arêtes appartenant à l'ensemble des éléments de  $G$  relativement à  $G_0$ .

Exemple.



éléments notés + . . . +

Fig. 17

Dans le déroulement de l'algorithme général

- lorsque l'on ajoute l'élément (1) à  $G_0$  on obtient la figure de K2 :

$(G_0 \cup (1))$  - chaîne d'arêtes en pointillés ---

- lorsque l'on rajoute à  $G_0$ , l'élément (2) on obtient la figure de K2 :

$(G_0 \cup (2))$  - chaîne d'arêtes . - . - . -

Or il existe une figure K2 :

$(G_0 \cup (1) \cup (2))$  - arêtes cochées

que l'on n'atteindra pas.

### Correspondants d'éléments par rapport à $G_0$ .

Ce sont les arêtes ou chaînes qui sont supprimées lors de l'application de l'algorithme pour un élément considéré et pour une figure de départ  $G_0$ .

On peut généraliser les cas des figures non atteintes de la façon suivante :

(figure de départ  $\cup$  ensemble d'éléments) - Partie (pouvant être vide) des correspondants aux éléments considérés - des arêtes de  $G_0$  n'appartenant pas aux correspondants, (il peut ne pas y en avoir).

### Remarque 1.

Il peut y avoir plusieurs figures de K2 non atteintes par un ensemble d'éléments.

### Remarque 2.

Etant donnée une figure de départ  $G_0$ , nous chercherons les figures non atteintes pour tous les ensembles de 2, 3, ..., N éléments, et nous appliquerons alors l'algorithme général aux figures ainsi découvertes et à  $G_0$ .

Remarque 3.

A titre d'exemple, nous avons dressé un tableau de figures non atteintes pour un ensemble de 2 éléments, les éléments étant seulement des arêtes, donc  $G_0$  de départ étant totale, dans tous les cas possibles et pour tout type d'élément par rapport à  $G_0$ .

nombre de cas = 22

nombre de figures non atteintes par

cas : 1 à 16.

2.2.4. Figures de KURATOWSKI de type 1 (K1).

Ce cas de figure ne nous intéressant pas directement, nous présenterons seulement le problème, sans étudier de méthode de recherche. Nous verrons d'ailleurs que la figure de K1 ne se prête pas à l'élaboration d'algorithmes efficaces.

De la même manière que pour les figures de K2, nous définissons, pour les figures de K1, 10 classes de sommets. Le degré des sommets d'une figure K1 étant de 4 ou de 2 au lieu de 3 ou de 2, respectivement.

2.2.4.1. Un élément rajouté à une figure de K1.

Le seul cas où l'on obtient une figure nouvelle est celui où l'élément est "arête parallèle" de la figure de K1 (les 2 sommets extrémité de l'élément appartiennent à la même classe de sommets de K1).

Exemple.

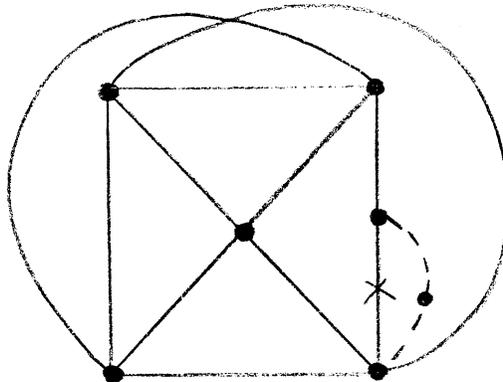


Fig. 18

2.2.4.2. Deux éléments rajoutés à une figure de K1.

1<sup>er</sup> cas.

Les 2 éléments sont "arête parallèle", et les sommets extrémité des deux éléments appartiennent à la même classe de sommets.

Exemple.

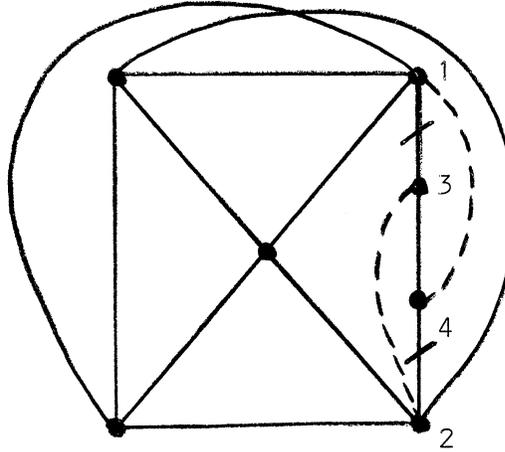


Fig. 19

Une figure non atteinte lorsque le processus général n'utilise qu'un seul élément :

la classe de sommets 1, 3, 4, 2 est remplacée (du point de vue des arêtes) par la classe : 1432;

2<sup>ième</sup> cas.

Les éléments ont un sommet extrémité en commun de degré 2 sur K1.

- les deux autres sommets extrémité sont de degré 4 :

Exemple.

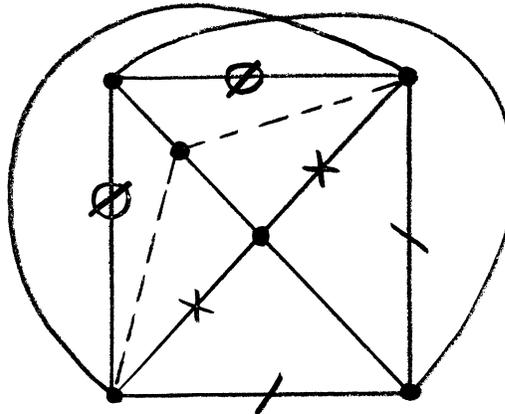


Fig. 20

3 figures obtenues après adjonction des 2 éléments :

1. deux classes  $\emptyset$  supprimées
2. deux classes X supprimées
3. deux classes / supprimées

- 1 sommet extrémité restant est de degré 4, l'autre de degré 2.

Exemple.

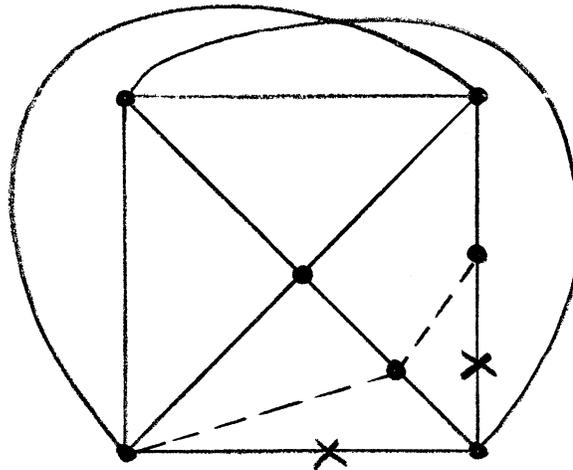


Fig. 21

Une figure obtenue après adjonction des 2 éléments et suppression des arêtes ou chaînes d'arêtes cochées.

- Les deux autres sommets extrémité sont de degré 2.

Exemple.

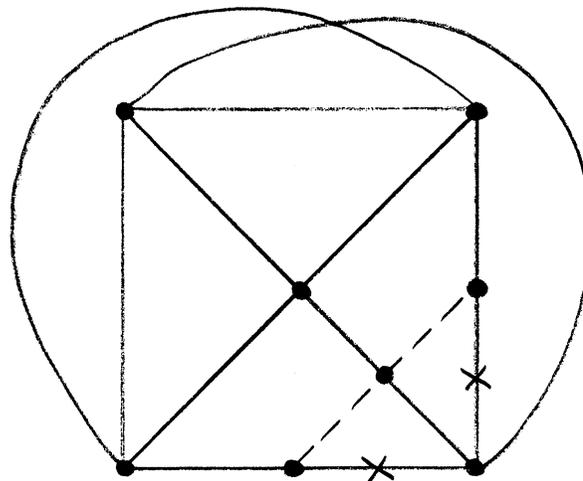


Fig. 22

Une figure obtenue après adjonction des 2 éléments et suppression des arêtes ou chaînes d'arêtes cochées.

Nous n'étudierons pas les cas où le nombre d'éléments d'un ensemble, rajouté à une figure de  $K_1$ , de départ, est supérieur à 2 : la recherche est alors non systématique et le nombre de cas différents, beaucoup trop grand.

## C H A P I T R E II

### APPLICATIONS A LA REALISATION DES CIRCUITS INTEGRES

#### 1 - INTRODUCTION

Nous étudions ici des circuits électroniques où les éléments, à implanter, sont simples, c'est-à-dire des transistors, des diodes, des capacités, des résistances.

On peut remarquer que l'étude faite ici, pour une technique de réalisation des circuits électroniques avancée, peut s'appliquer, en partie, à une technique plus classique, c'est-à-dire à la réalisation des circuits imprimés : certaines contraintes rencontrées lors de la réalisation des circuits intégrés recouvrant celles des circuits imprimés.

#### 2 - LES DIFFERENTES ETAPES DE L'ETUDE

Elles sont données par les différentes contraintes d'implantation que l'on doit respecter :

1°) La planéarité du réseau, c'est-à-dire tracer les interconnexions entre les éléments électroniques de telle façon qu'il n'y ait pas de croisements.

2°) La longueur totale des connexions minimum, ou acceptable.

3°) Les groupements d'éléments, par exemple, les transistors à même tension collecteur seront mis dans le même caisson, les résistances seront, si possible, regroupées dans le même caisson.

4°) La surface occupée par les éléments. (cf. 2<sup>ème</sup> partie).

5°) L'éloignement minimum de deux éléments, imposé pour la dissipation de la chaleur, ou pour réduire l'effet des capacités parasites, etc ... (cf. 2<sup>ème</sup> partie).

Notre problème est donc, en premier lieu, de donner une représentation planaire topologique du réseau électronique, si elle existe, sinon d'en donner une représentation topologique aux moindres frais (cf. ch. 1, 2.1.3.).

En même temps que la planéarité du réseau nous traiterons les groupements d'éléments électroniques.

### 3 - PLANEARITE DES RESEAUX ELECTRONIQUES

Nous devons tenir compte des possibilités données par la réalisation des éléments électroniques et aussi des obligations imposées au circuit terminé :

1°) On peut ignorer les résistances, car une connexion peut passer dessus.

2°) Possibilités données par la réalisation topologique du transistor. (Nous étudierons à part ce cas particulier, et considérerons pour le moment le transistor comme ayant trois points de contact avec l'extérieur.

3°) Obligation pour les points d'entrée-sortie du réseau de ne pas être "enfermés" à l'intérieur du circuit, par une connexion. (Nous considérerons l'ordre de rencontre des entrées-sorties, lors du cheminement le long du périmètre du circuit, comme indifférent, ou comme imposé, selon les vœux du concepteur).

#### 3.1. Passage du schéma électronique à un réseau de noeuds et d'étoiles.

Le point de départ est le schéma électronique donné par le circuit. Le processus suivi apparaîtra mieux sur un exemple.

Soit donc à réaliser la bascule suivante :

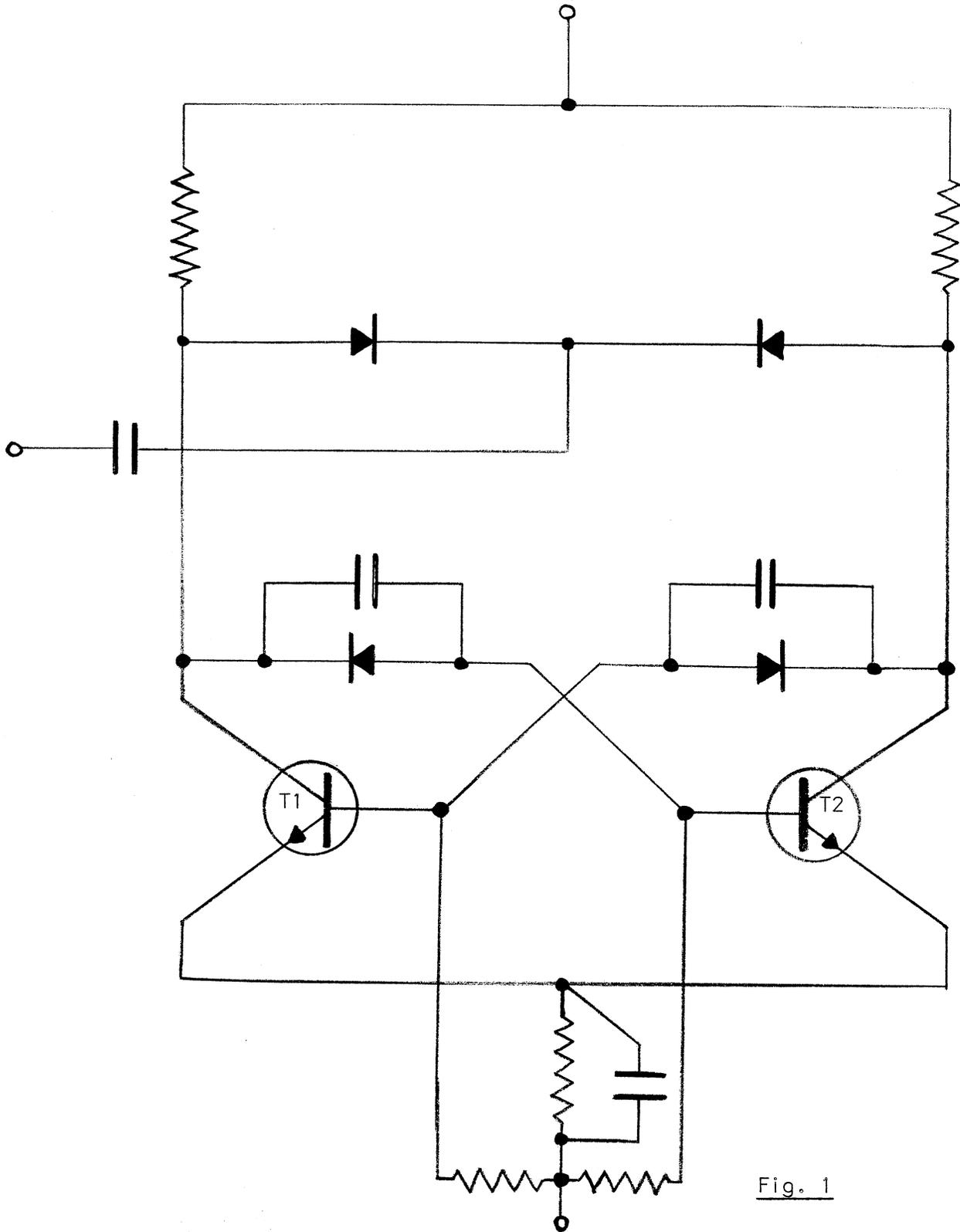


Fig. 1

noeuds  $X_i$  : éléments électroniques.

$$\delta^\circ(X_i) = 2 \text{ ou } 3, \quad \forall_i$$

étoiles  $\bullet_j$  : réunit les connexions au même potentiel.

$$1 \leq \delta^\circ(j) \leq n, \text{ si } n \text{ est le nombre des noeuds du réseau.}$$

Nous éliminerons du schéma électronique les résistances (1<sup>ère</sup> possibilité) et réunirons les entrées-sorties en une seule étoile afin de répondre à la 3<sup>ème</sup> obligation (cf. ch. 1, 1.3.1.).

Nous obtenons donc le réseau suivant :

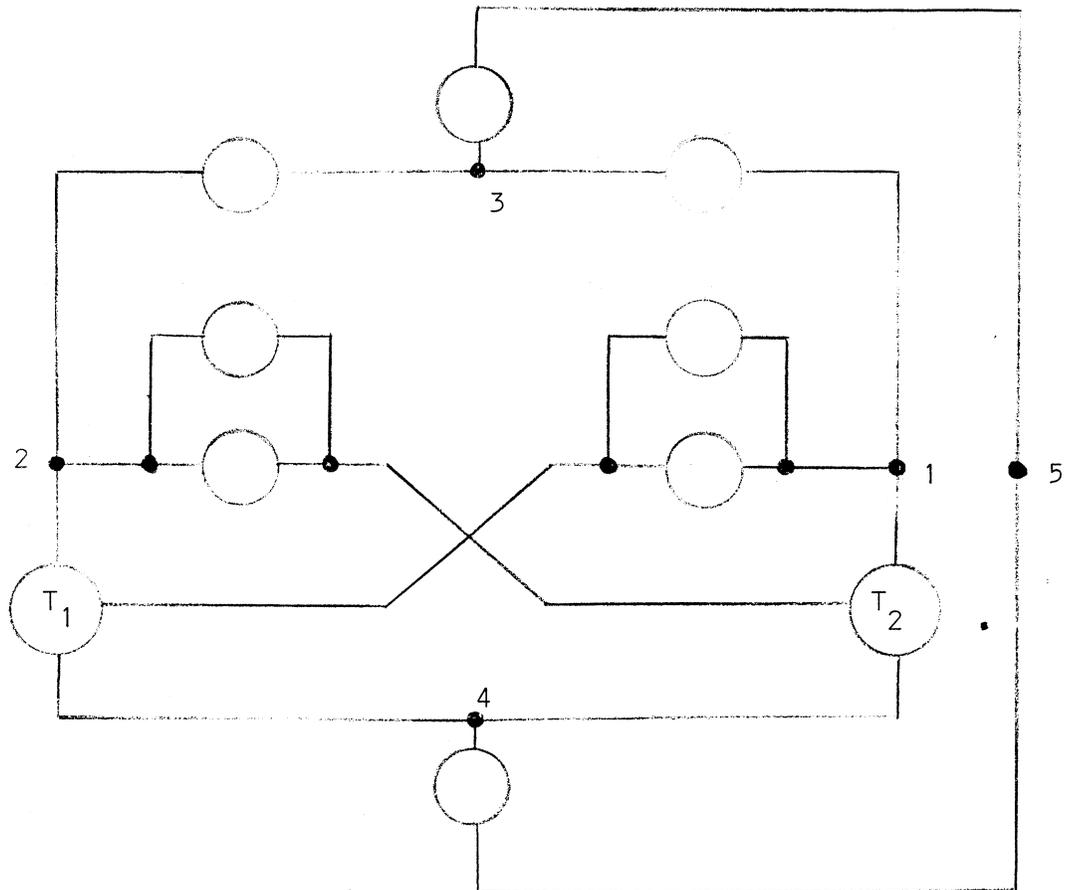


Fig. 2

### 3.2. Passage du réseau de noeuds et d'étoiles à un graphe.

Nous considérerons le transistor comme réduit à un sommet de degré égal à 3.

Nous simplifierons (d'après 1.2.3. ch. 1) pour les arêtes parallèles, et les sommets de degré 2. Nous obtenons pour l'exemple traité :

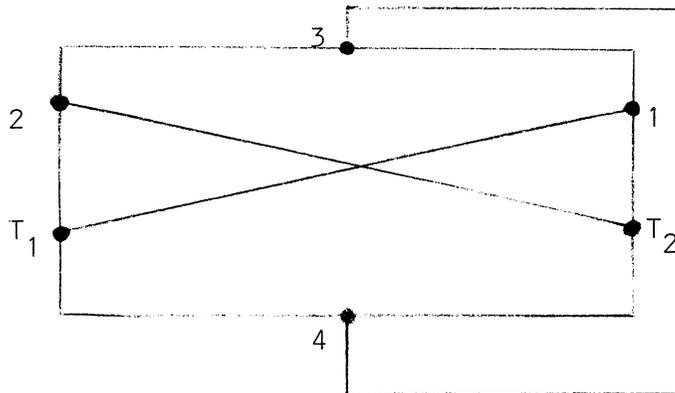


Fig. 3

### 3.3. Coût de la réalisation d'un réseau électronique.

Si le graphe  $G$  obtenu en 3.2. est planaire nous en donnerons une représentation planaire topologique, sinon nous chercherons les ensembles minimaux d'arêtes à enlever de  $G$  afin de le rendre planaire, en utilisant la méthode exposée (§2, ch. 1).

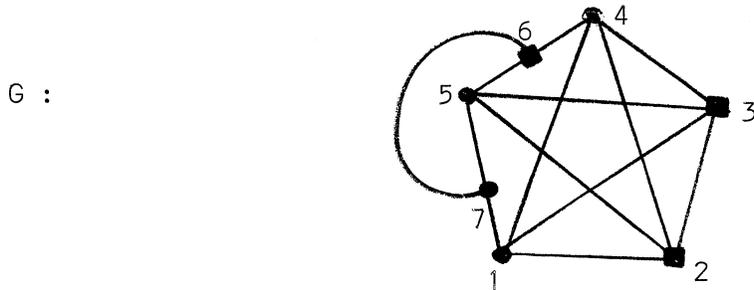
(Pour l'exemple traité nous obtiendrons tous les ensembles formés d'une arête.

#### Remarques.

1 - Dans le cas précis que nous étudions ici, c'est-à-dire l'étude d'un réseau électronique, où de nombreux sommets sont de degré = 3 (transistors), la propriété suivante montre que, la recherche des figures de KURATOWSKI de type K1 présente relativement moins d'intérêt, lorsque l'on a trouvé les figures de KURATOWSKI de type K2.

Propriété.

Soit le graphe suivant :



- $G_1$  le graphe partiel,  $G - (\text{arête } \{6,7\})$ , est une figure  $K_1$ .
- $G_2$  le graphe partiel,  $G - (\text{arêtes } \{1,4 - 2,3 - 5,7\})$ , est une figure  $K_2$ .

Lorsque l'on détruira la figure  $K_2$ , en enlevant une arête quelconque de  $G_2$  (sauf l'arête 6,7), on détruira en même temps la figure de  $K_1$ .

2 - Le transistor étant de degré 3, les seuls sommets ayant un degré supérieur à 3 sont les équipotentielles (étoiles). Si le graphe  $G$  obtenu en 3.2., est non planaire et si le nombre de ses sommets, ayant un degré supérieur à 3, est supérieur à 4, on pourra se ramener à la recherche des seules figures de type  $K_2$  en éclatant les sommets de degré supérieur à 3 et ceci de façon à n'avoir pas plus de 4 sommets dans ce cas, pour  $G$ .

Exemple.

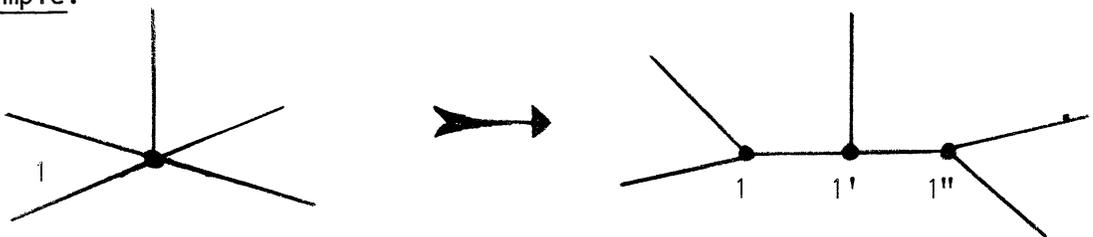
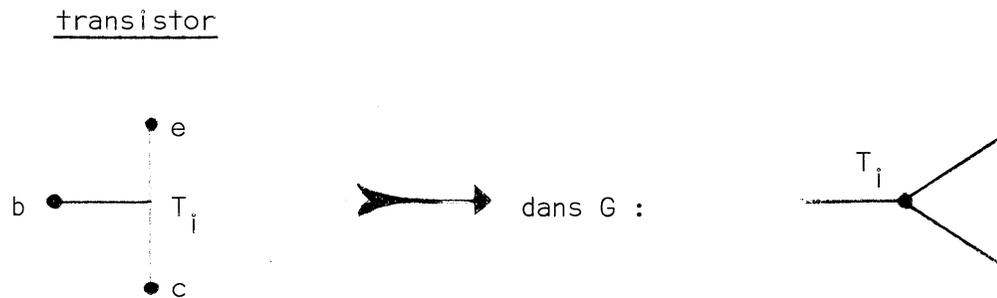


Fig. 4

#### 4 - POSSIBILITES OFFERTES PAR LA REALISATION DU TRANSISTOR

##### 4.1. Du point de vue topologique.

Jusqu'à présent nous avons considéré le transistor comme un point ayant trois arêtes ce qui revient à :



Or les différentes technologies de réalisation du transistor offrent en fait, plusieurs possibilités. Prenons, par exemple, les deux technologies classiques suivantes :

1 -

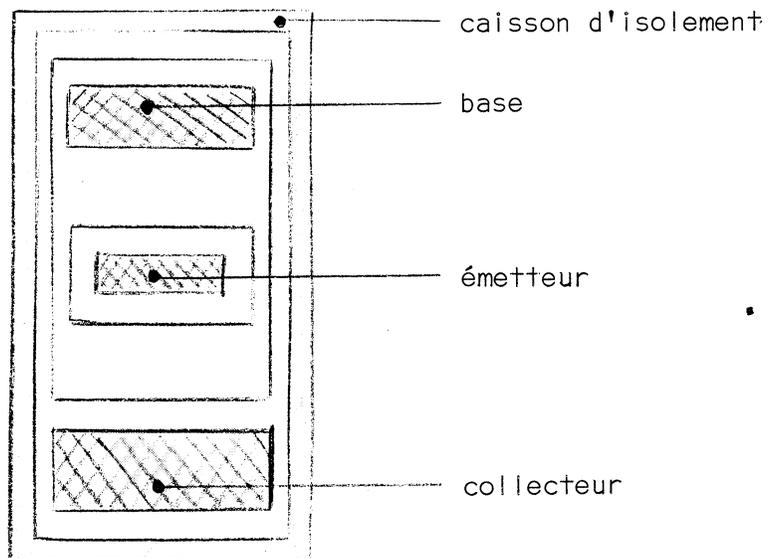


Fig. 6.a

ce qui se traduit dans G par :

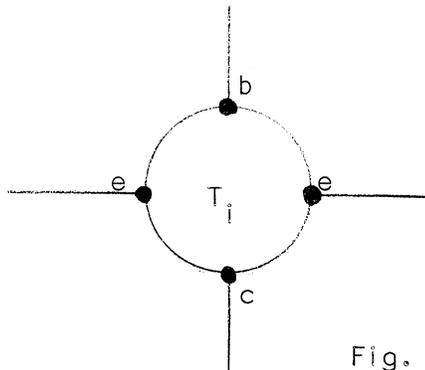


Fig. 6.b

d'où possibilité de répartir les connexions équipotentielles de l'émetteur de ce transistor d'un côté ou de l'autre de la ligne fictive formée par la base et le collecteur.

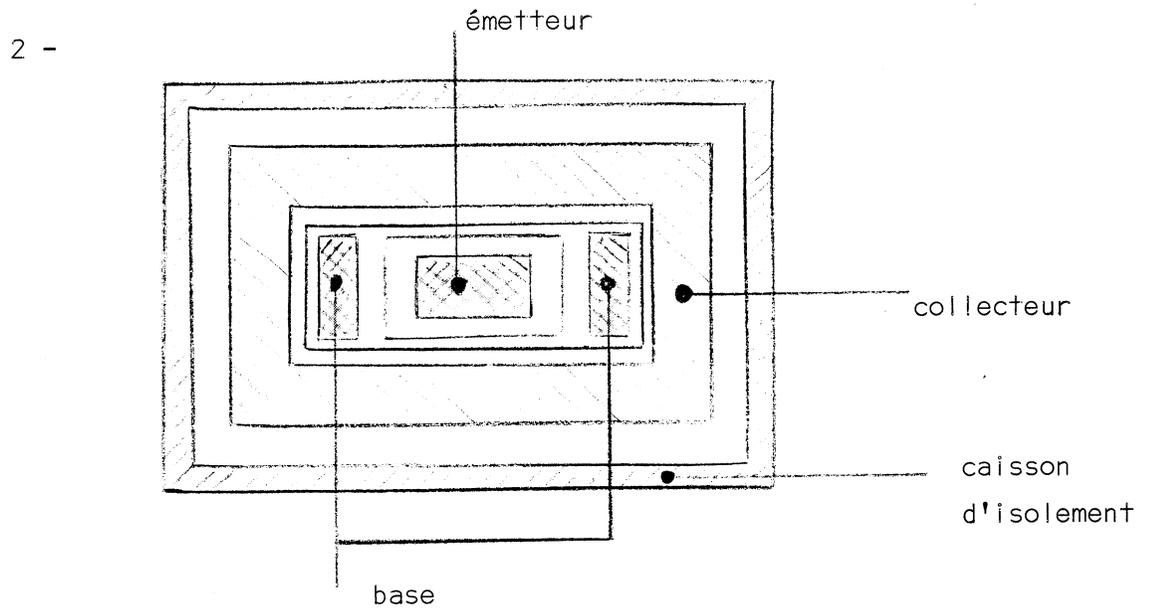


Fig. 7.a

- la sortie collecteur peut être faite en un point quelconque de celui-ci.
- la sortie base est faite en deux points sur la base.

Ceci nous conduit à représenter le transistor dans  $G$  par :

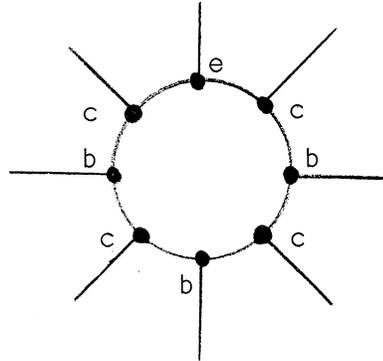


Fig. 7.b

d'où le grand nombre, ici, de répartitions différentes des connexions équipotentielles relatives à la base ou au collecteur.

Si nous reprenons l'exemple traité Fig. 1 de ce chapitre et si nous voulons réaliser le circuit électronique avec le transistor du type 1 nous aurions alors une solution planaire telle que :

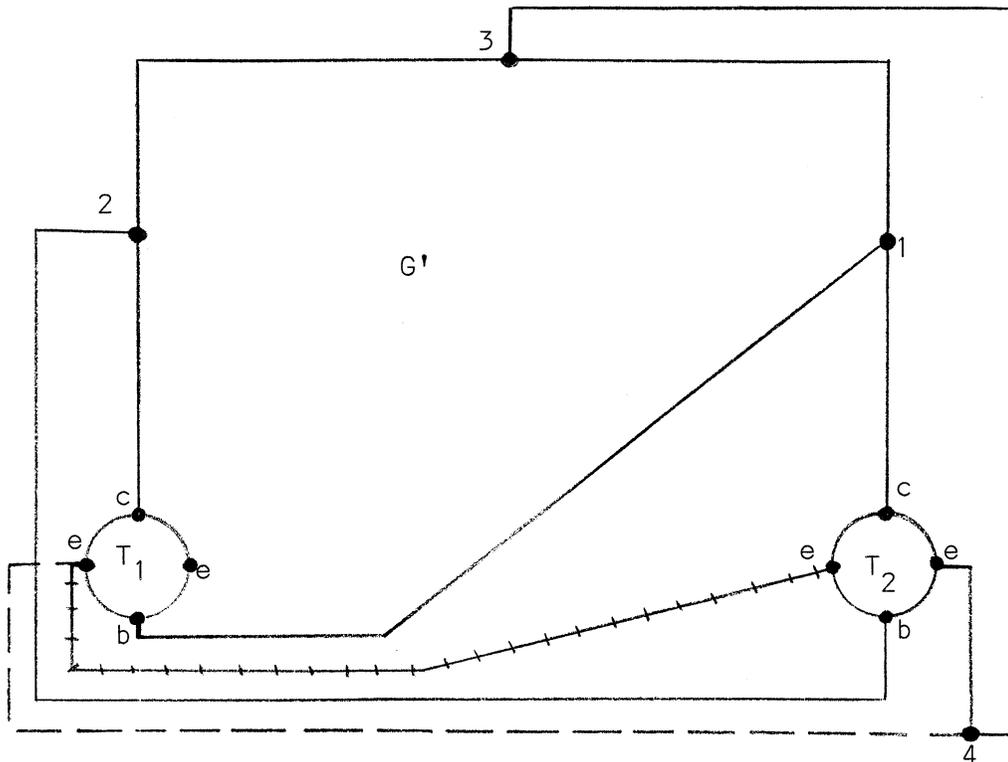


Fig. 8

nous avons rajouté l'arête  $\dashrightarrow$  et supprimé l'arête en pointillé  $\Rightarrow G'$  planaire.

#### 4.2. Du point de vue électrique.

Ceci intéresse surtout le transistor de type 1. En effet, il est possible, dans certains cas, de dégénérer le collecteur d'un transistor et de faire, alors, passer dessus une connexion.

#### Exemple.

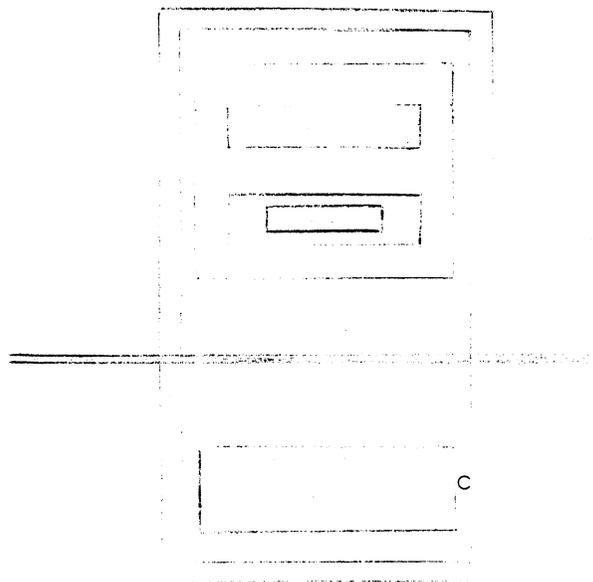


Fig. 9

#### 4.3. Comment tenir compte de ces possibilités.

Plusieurs façons de procéder existent et malgré de nombreux exemples traités, aucune politique de choix ne nous est apparue clairement. On peut :

- traiter le problème sans tenir compte des possibilités et chercher un ensemble d'arêtes, à supprimer de  $G$  afin de le rendre planaire, compatible avec ces possibilités (non forcément minimum).

- tenir compte des possibilités dès le départ et proposer un réseau de noeuds et d'étoiles, à traiter, déjà presque planaire.

- supprimer systématiquement les connexions dont les points équipotentiels peuvent être choisis en plusieurs endroits d'un transistor, traiter le problème ainsi simplifié, et reconstituer le réseau ensuite.

Dans les exemples traités, nous nous sommes servis de toutes ces façons d'aborder le problème.



# CHAPITRE III

## PROGRAMMES - RESULTATS

### 1 - ENSEMBLES D'ARETES A SUPPRIMER D'UN GRAPHE G AFIN DE LE RENDRE PLANAIRE

Nous verrons plus loin, par les exemples traités, tout l'intérêt de partir d'une figure de KURATOWSKI  $G_0$ , si possible totale, ou contenant un nombre maximal de sommets.

Nous chercherons donc dans  $G$  un cycle hamiltonien, s'il existe, ou le cycle le plus long.

Ce résultat obtenu nous chercherons une figure de KURATOWSKI ayant un nombre maximal de sommets, le cycle le plus long pouvant conduire à une figure totale.

Exemple.

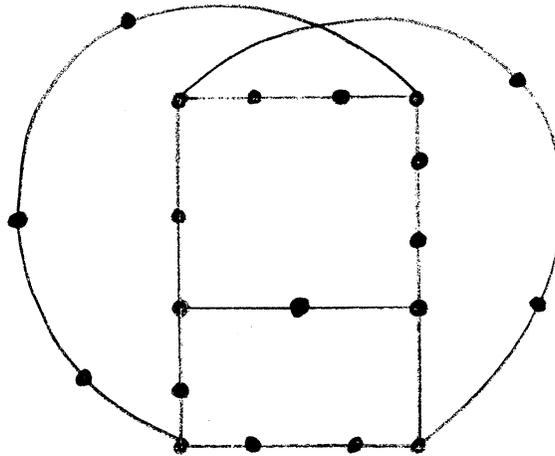


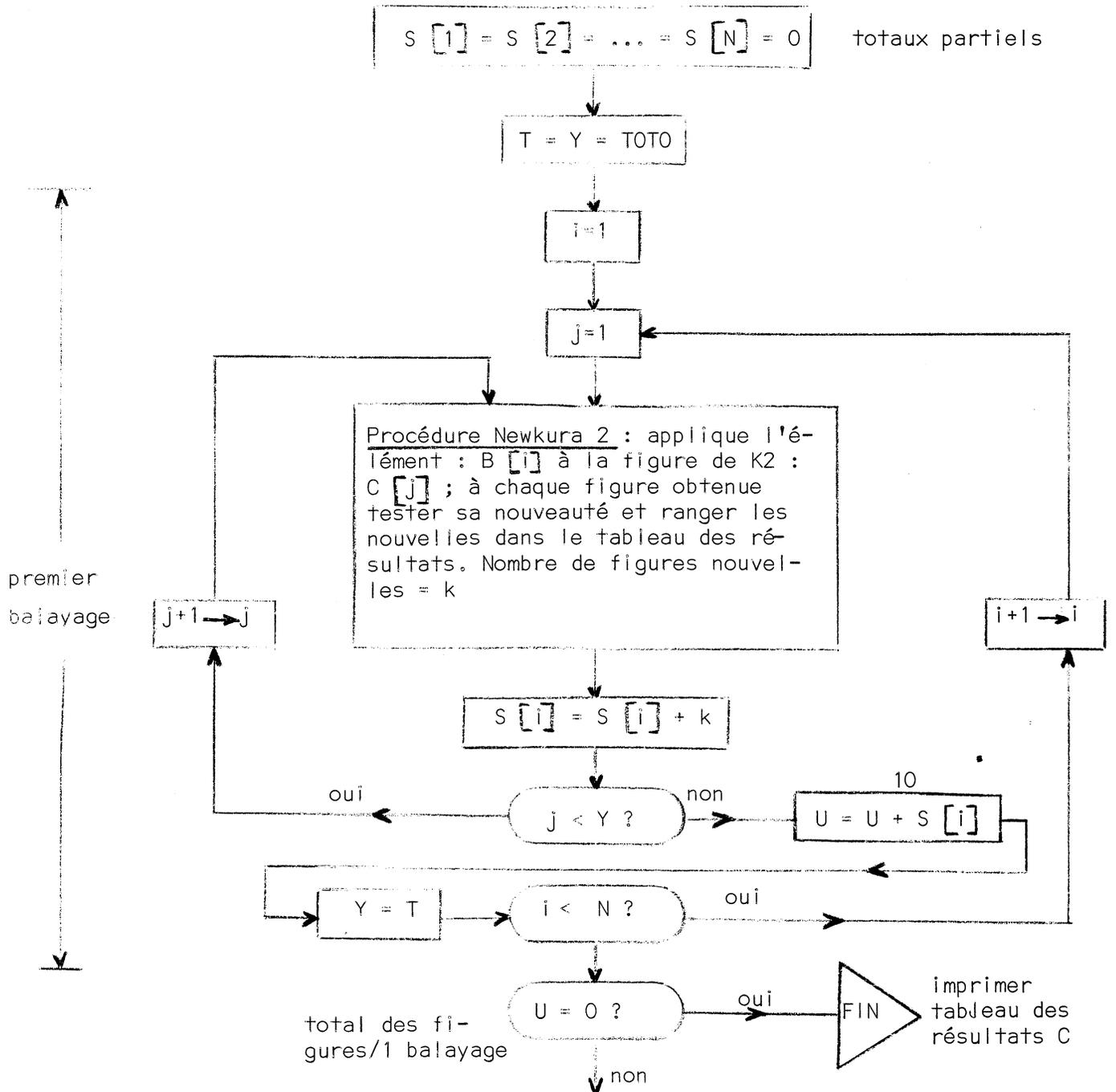
Fig. 1

Ayant alors une figure de départ  $G_0$  totale, ou non, et les éléments :  
de  $G$

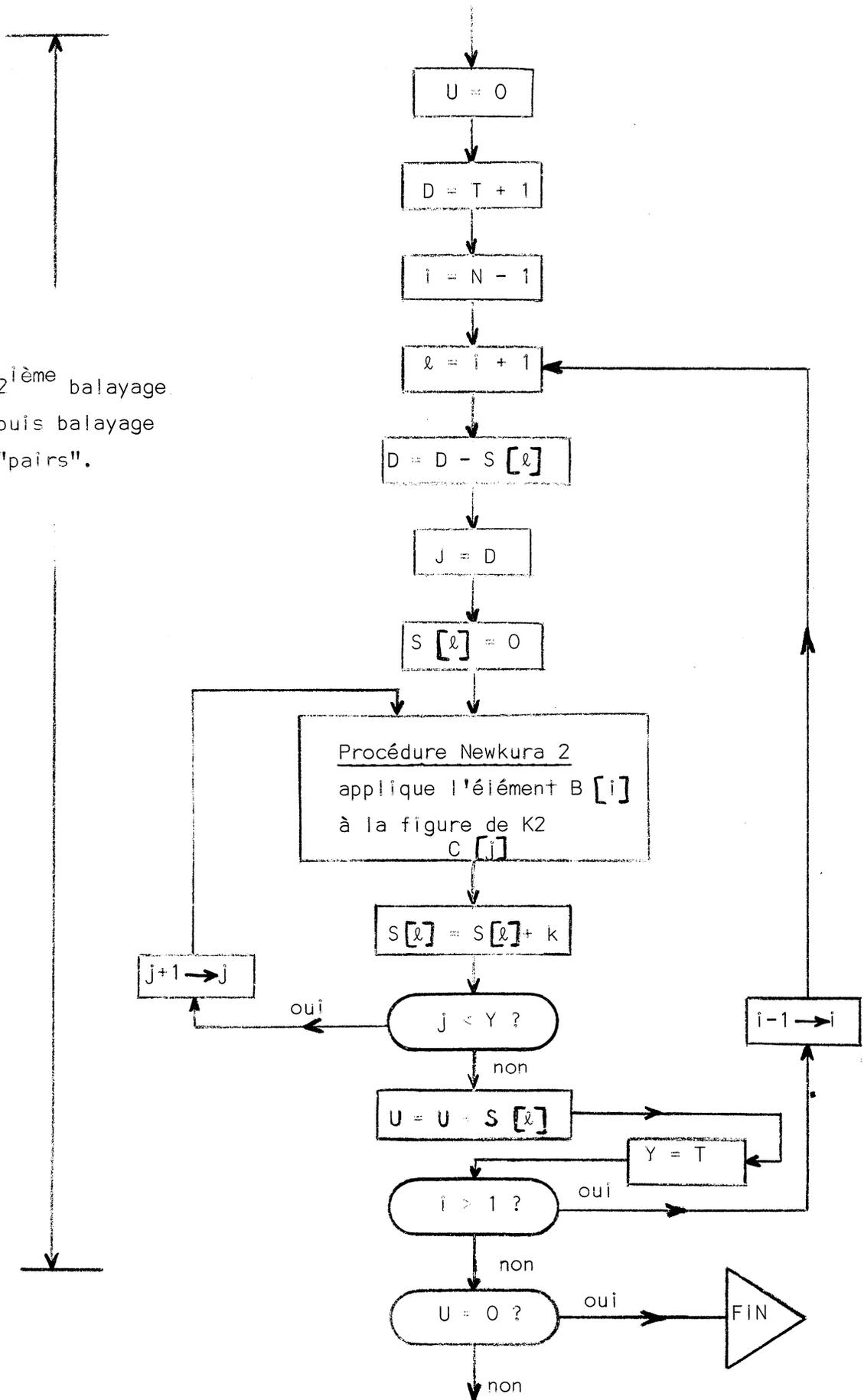
Nous obtiendrons donc comme données :

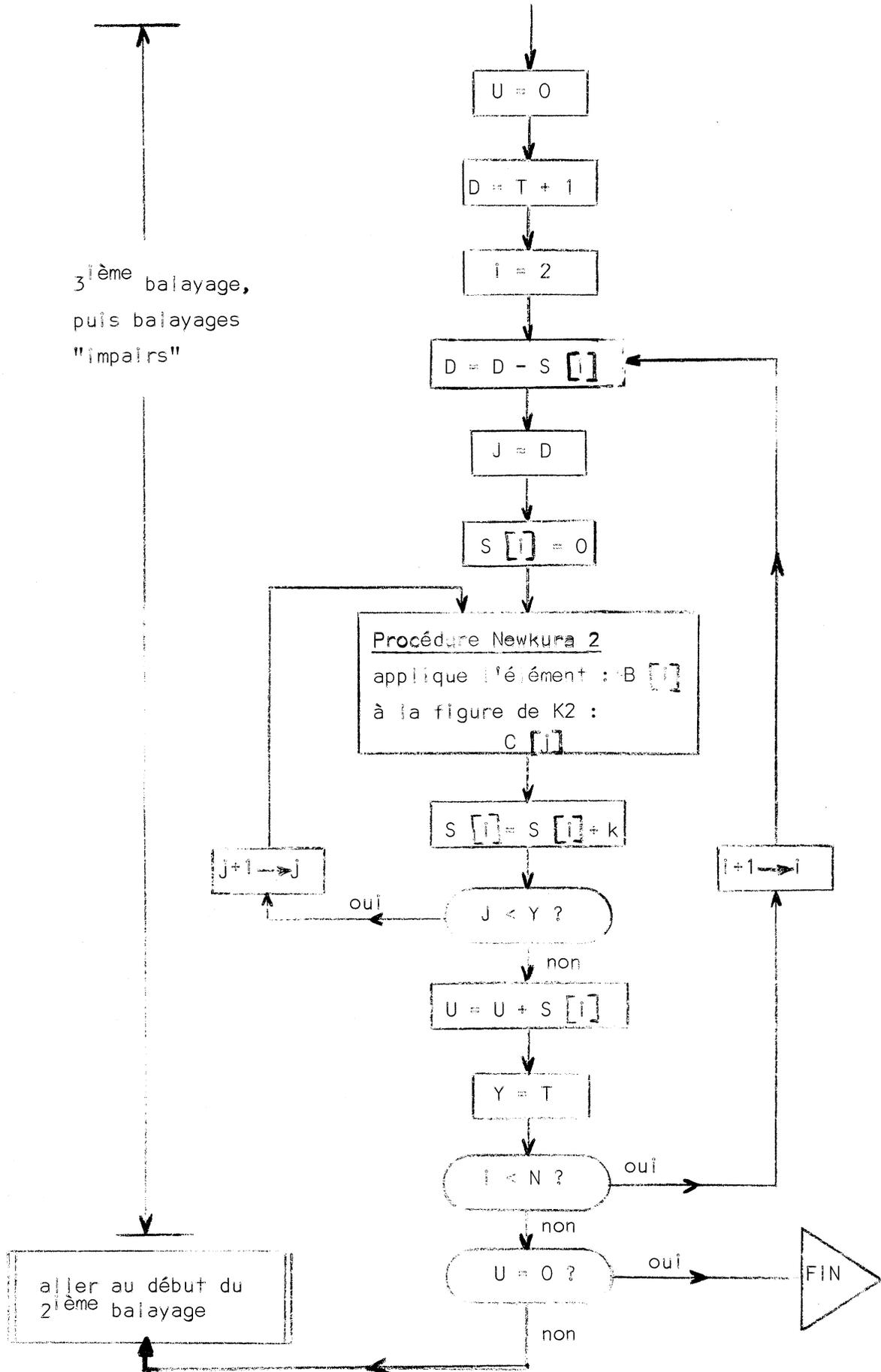
- la liste des N éléments rangées en tête du tableau des résultats C .
- la figure de départ  $G_0$
- les figures non atteintes

1.1. Organigramme pour la recherche des figures de type K2.



2<sup>i</sup>ème balayage  
puis balayage  
"pairs".





## 1.2. Programmes.

### 1.2.1. Langage de programmation.

Désirant avoir des programmes efficaces, compréhensibles de beaucoup de personnes, spécialistes ou non, et surtout très maniables, c'est-à-dire autorisant des modifications éventuelles, nous avons été amenés à choisir le langage Algol 60.

De plus ce langage offre l'avantage de permettre une utilisation locale du langage Machine, au sein de procédures.

### 1.2.2. Choix d'un codage.

Les quantités à coder sont essentiellement des sous-graphes partiels d'un graphe donné, représentées sous forme de vecteurs booléens.

- Le graphe G sera représenté par sa matrice d'incidence ayant pour lignes les sommets et pour colonnes les arêtes.

#### a) Représentation d'un sous-graphe partiel.

Chaque sous-graphe partiel est représenté par une ligne de m bits (si m = nombre d'arêtes de G) donc q mémoires si  $q = (m/37)+1$  ;

$$\begin{array}{l} i^{\text{ème}} \text{ bit de} \\ \text{la ligne} \end{array} \left\{ \begin{array}{l} = 1 \text{ } i^{\text{ème}} \text{ arête de G présente dans le sous-graphe partiel} \\ = 0 \text{ } i^{\text{ème}} \text{ arête de G absente dans le sous-graphe partiel} \end{array} \right.$$

#### b) Opérations sur les sous-graphes partiels.

- La matrice d'incidence du sous-graphe partiel peut être obtenue à partir de celle de G en faisant l'intersection de toutes ses lignes avec le vecteur booléen représentant le sous-graphe partiel.

- Comparaison. Si deux sous-graphes sont identiques, alors leur disjonction donnera le vecteur booléen constant  $\equiv 0$  ;

### 1.2.3. Sous-programmes standards.

Pour parer aux insuffisances du langage Algoi, nous utilisons des procédures écrites en langage machine. Ces procédures sont valables sur IBM 7044 ayant le compilateur de Grenoble.

a) Procédures d'écriture (dues à Monsieur A. COLMERAUER)

procédure RCH;

commentaire retour à la ligne;

procédure LIGNE (A,B); valeur A,B; entier A,B;

commentaire définit deux marges entre lesquelles tous les textes suivants seront écrits.

procédure SORCHANE (I,J,X); valeur I,J;

entier I,J,X;

commentaire écrit un texte de I symboles dont le premier est dans la mémoire X, en position J. Le texte s'imprime à la suite des textes précédents, s'il y a suffisamment de place. Aucun espace n'est prévu.

b) Sous-programme de traitement logique.

entier procédure REUNION(F,M);

valeur M; entier M, entier tableau F;

commentaire cette procédure prend l'union logique des M premières mémoires du tableau F;

### 1.2.4. Procédures NEWKURA 2.

Nous avons été amenés à écrire deux procédures, l'une s'appliquant à une figure de départ totale, l'autre à une figure de départ quelconque.

Nous ne donnerons ici que la procédure s'appliquant au premier cas, le plus simple.

```

'PROCEDURE' NEWKURAZ(A,B,C,MA,I,J,K,P,Q,T,NALTUB) $$
  'VALEUR' A,B,MA,P,C $$
  'ENTIER' I,J,K,P,Q,T $$
  'ENTIER' 'TABLEAU' A,B,C,MA $$
  'PROCEDURE' NALTUB $$
    'COMMENTAIRE' CETTE PROCEDURE AJOUTE L ARETE B.(I). A LA FIGURE
    DE K2 RESUMEE DANS LE TABLEAU C A LA LIGNE J $ C.(J,Q). . LES
    NOUVELLES FIGURES SONT RESUMEEES DANS C A LA SUITE DES T FIGURES
    DEJA EXISTANTES .
    A LA FIN DE LA PROCEDURE ON OBTIENT K$= NOMBRE DE FIGURES
    NOUVELLES DE L ESSAI ET LE NOUVEAU NOMBRE T DE FIGURES (LIGNES)
    DU TABLEAU C $$
  'DEBUT' 'ENTIER' 'TABLEAU' D.(1$P,1$Q).,E.(1$P). $$
  'ENTIER' R,S,U,V,W,X,Y,Z,F1 $$
  'POUR' R$=1 'PAS' 1 'JUSQUA' P 'FAIRE'
  'DEBUT' S$=0 $$
    'POUR' U$=1 'PAS' 1 'JUSQUA' Q 'FAIRE'
    'DEBUT' D.(R,U).$=ET(A.(R,U).,C.(J,U).) $$
      S$=S+POIDS(D.(R,U).) $$
    'FIN' $$
    E.(R).$=S $$
  'FIN' $$
  X$=B.(I,3). $$ Y$=(X$/'27)+1 $$ Z$=X-36*(Y-1) $$
  'SI' BIT(C.(J,Y).,37-Z)=1 'ALORS' 'DEBUT' K$=0 $$ 'ALLERA' FINI $$
    'FIN' $$
  VI=B.(I,1). $$ W$=B.(I,2). $$
  'SI' E.(V).=0 'OU' E.(W).=0 'ALORS' 'DEBUT' K$=0 $$
    'ALLERA' FINI $$
    'FIN' $$
  'DEBUT' 'ENTIER' 'TABLEAU' F.(1$P).,G.(1$P). $$
  'ENTIER' H,L,M,N,KA $$
  NALTUB(F,P) $$ NALTUE(G,P) $$
  F$=0 $$
  L$=1 $$
HCRQ: 'SI' E.(L).=3 'ALORS'
  'DEBUT' M$=L+1 $$
TCTC: 'SI' E.(M).=3 'ALORS'
  'DEBUT' S$=0 $$
    'POUR' R$=1 'PAS' 1 'JUSQUA' Q 'FAIRE'
    'DEBUT' U$=ET(C.(L,R).,D.(M,R).) $$
      S$=S+POIDS(U) $$
    'FIN' $$
    'SI' S=1 'ALORS'
    'DEBUT' F$=H+1 $$ G.(L).$=G.(L).+1 $$
      G.(M).$=G.(M).+1 $$
      F.(L).$=DU(F.(L).,MA.(37-H).) $$
      F.(M).$=DU(F.(M).,MA.(37-H).) $$
    'FIN' $$
  'FIN' $$
  'SI' M 'INFER' P 'ALORS'
  'DEBUT' M$=M+1 $$ 'ALLERA' TCTC $$
  'FIN'

```



```

'SINCN' 'SI' L 'INFER' P-1 'ALCRS'
'DEBLT' L=L+1 $$ 'ALLERA' RORO $$
'FIN'
'SINCN' 'ALLERA' SUITE $$
'FIN'
'SINCN' 'SI' L 'INFER' P-1 'ALCRS'
'DEBLT' L=L+1 $$ 'ALLERA' RORO $$
'FIN'
'SINCN' 'ALLERA' SUITE $$
SUITE$ R$=H+1 $$ F1$=R $$
ITER5$ L$=1 $$
ITER1$ 'SI' E.(L). 'INFEG' 2 'OU' G.(L).=3 'ALCRS'
'DEBLT' L$=L+1 $$ 'ALLERA' ITER1 $$
'FIN' $$
G.(L).$=G.(L).+1 $$ F.(L).$=OU(F.(L).,MA.(37-R).) $$
H$=1 $$
ITER2$ 'SI' E.(H). 'NCNEG' 2 'OU' G.(H). =1 'ALCRS'
'DEBLT' H$=H+1 $$ 'ALLERA' ITER2 $$
'FIN' $$
U$=0 $$
'POUR' S$=1 'PAS' 1 'JUSQUA' G 'FAIRE'
'DEBLT' N$=ET(D.(L,S).,D.(H,S).) $$
U$=U+PCIDS(N) $$
'FIN' $$
'SI' U 'INFEG' 0 'ALCRS'
'DEBLT' H$=H+1 $$ 'ALLERA' ITER2 $$
'FIN' $$
G.(H).$=1 $$ F.(H).$=OU(F.(H).,MA.(37-R).) $$
ITER4$ M$=1 $$
ITER3$ 'SI' G.(M).=3 'OU' E.(M). 'INFEG' 0 'OU' M=L 'OU' G.(M). =1
'ET' E.(M). =2 'ALCRS'
'DEBLT' M$=M+1 $$ 'ALLERA' ITER3 $$
'FIN' $$
U$=0 $$
'POUR' S$=1 'PAS' 1 'JUSQUA' G 'FAIRE'
'DEBLT' N$=OJ(D.(H,S).,D.(M,S).) $$
U$=U+PCIDS(N) $$
'FIN' $$
'SI' U=4 'OU' U=5 'ALCRS'
'DEBLT' M$=M+1 $$ 'ALLERA' ITER3 $$
'FIN'
'SINCN' 'SI' U=2 'ALCRS'
'DEBLT' G.(M).$=1 $$ F.(M).$=OU(F.(M).,MA.(37-R).) $$
L$=H $$
H$=M $$ 'ALLERA' ITER4 $$
'FIN'
'SINCN' C.(M).$=G.(M).+1 $$
F.(M).$=OU(F.(M).,MA.(37-R).) $$
'SI' P 'INFER' 9 'ALCRS'
'DEBLT' R$=P+1 $$ 'ALLERA' ITER5 $$
'FIN' $$
'DEBLT' 'PROCEDURE' ANNULCLASS(F,E,C,MA,P,G,V,W,F1) $$

```



```

'VALEUR' MA,P,C $$ 'ENTIER' 'TABLEAU' F,E,D,MA $$
'ENTIER' P,Q,V,W,F1 $$
'COMMENTAIRE' CETTE PROCEDURE ANNULE TOUTES LES ARETES D UNE
MEME CLASSE S, COMMUNE AIX 2 SCMMETS DE CO 3 $ V ET W $$
'DEUT' 'ENTIER' 'TABLEAU' G.(I$36). $$
'ENTIER' F,L,R,S,U $$
'POUR' R$=1 'PAS' 1 'JUSQUA' 36 'FAIRE'
G.(R).$=NG(MA.(R).) $$
S$=1 $$
ITER$ 'SI' BIT(F.(V).,37-S)=1 'ET' BIT(F.(W).,37-S)=1 'ALORS'
'DEUT' 'SI' S 'INFER' F1 'ALORS'
'POUR' U$=1 'PAS' 1 'JUSQUA' Q 'FAIRE'
'POUR' R$=36 'PAS' -1 'JUSQUA' 1 'FAIRE'
'DEUT' 'SI' BIT(D.(V,U).,R)=1 'ET'
BIT(D.(W,U).,R)=1 'ALORS'
'DEUT' D.(V,U).$=ET(D.(V,U).,G.(R).) $$
D.(W,U).$=ET(D.(W,U).,G.(R).) $$
'ALLERA' PP $$
'FIN' $$
'FIN' $$
'ALLERA' TRAVAIL $$
'FIN' $$
S$=S+1 $$ 'ALLERA' ITER $$
TRAVAIL$ 'POUR' F$=1 'PAS' 1 'JUSQUA' P 'FAIRE'
'DEUT' 'SI' BIT(F.(H).,37-S)=1 'ET' E.(H).=2 'ALORS'
'POUR' U$=1 'PAS' 1 'JUSQUA' Q 'FAIRE'
'POUR' R$=36 'PAS' -1 'JUSQUA' 1 'FAIRE'
'DEUT' 'SI' BIT(D.(F,U).,R)=1 'ALORS'
'POUR' L$=1 'PAS' 1 'JUSQUA' P 'FAIRE'
D.(L,U).$=ET(D.(L,U).,G.(R).) $$
'FIN' $$
'FIN' $$
FF$
'FIN' ANNULCLASS $$
'PROCEDURE' ACHVE(D,MA,C,KA,V,W,Y,Z,P,C,T,REUNION) $$
'VALEUR' MA,P,C $$ 'ENTIER' 'TABLEAU' D,MA,C $$
'ENTIER' KA,V,W,Y,Z,P,Q,T $$
'ENTIER' 'PROCEDURE' REUNION $$
'COMMENTAIRE' CETTE PROCEDURE AJOUTE DANS LE TABLEAU DE PSEUDC
INCIDENCE D , EN POSITION Z DE LA CCLENNE MEMOIRE Y , L ARETE
DEFINIE PAR UN 1 EN V ET W LIGNE . DE PLS LA NOUVELLE K2 EST
MISE EN LIGNE ET SA NOUVEAUTE TESTEE . ( SI NOUVELLE , KA$=1 ) $$
'DEUT' 'ENTIER' 'TABLEAU' G.(I$P). $$
'ENTIER' R,S,U $$
E.(V,Y).$=CU(D.(V,Y).,MA.(37-Z).) $$
E.(W,Y).$=CU(D.(W,Y).,MA.(37-Z).) $$
'POUR' R$=1 'PAS' 1 'JUSQUA' C 'FAIRE'
'DEUT' 'POUR' S$=1 'PAS' 1 'JUSQUA' P 'FAIRE'
G.(S).$=D.(S,R). $$
C.(T+1,R).$=REUNION(G,P) $$
'FIN' $$
'POUR' S$=1 'PAS' 1 'JUSQUA' T 'FAIRE'

```



```

'DEBUT' U:=0 $$
  'POUR' R:=1 'PAS' 1 'JUSQUA' Q 'FAIRE'
  'DEBUT' C.(T+2,R).:=CJ(C.(T+1,R).,C.(S,R).) $$
  L:=U+PCIDS(C.(T+2,R).) $$
  'FIN' $$
  'SI' U'INFEQ'0 'ALORS'
  'DEBUT' K:=0 $$
    'POUR' R:=1 'PAS' 1 'JUSQUA' C 'FAIRE'
    C.(T+1,R).:=C $$
    'ALLERA' TERM $$
  'FIN' $$
'FIN' $$
K:=1 $$
  'POUR' R:=1 'PAS' 1 'JUSQUA' Q 'FAIRE'
  C.(T+2,R).:=C $$
TERM:
'FIN' ACHVEE $$
'PROCEDURE' PARTANNULE(F,E,D,S,P,Q,V,W) $$
'VALEUR' P,Q $$ 'ENTIER' 'TABLEAU' F,E,D $$
'ENTIER' S,P,Q,V,W $$
'COMMENTAIRE' CETTE PROCEDURE ANNULE LES ARETES DE LA CLASSE S,
ALLANT DU POINT DE DC 3,V AU POINT DE DC 2,W $$
'DEBUT' 'ENTIER' 'TABLEAU' G.(1:Q). $$
'ENTIER' F,L,M,P $$
F:=1 $$
ITER1: R:=C $$
'SI' BIT(F.(H).,37-S)=1 'ET' F.(H).=2 'ALORS'
'DEBUT' 'POUR' L:=1 'PAS' 1 'JUSQUA' C 'FAIRE'
'DEBUT'
G.(L).:=CJ(D.(V,L).,E.(H,L).) $$
R:=R+PCIDS(G.(L).) $$
'FIN' $$
'SI' R 'NONEQ' 3 'ALORS'
'DEBUT' F:=H+1 $$ 'ALLERA' ITER1 $$
'FIN'
'SINON' 'POUR' L:=1 'PAS' 1 'JUSQUA' Q 'FAIRE'
'DEBUT'
C.(V,L).:=ET(D.(V,L).,G.(L).) $$
C.(H,L).:=ET(D.(H,L).,G.(L).) $$
'FIN' $$
'ALLERA' SUZY $$
'FIN'
'SINON' H:=F+1 $$ 'ALLERA' ITER1 $$
SUZY: 'SI' H=W 'ALORS' 'ALLERA' TERM $$
SUITE: M:=1 $$
ITER2: R:=0 $$
'SI' BIT(F.(M).,37-S)=1 'ET' E.(M).=2 'ET' M 'NONEQ' H 'ALORS'
'DEBUT' 'POUR' L:=1 'PAS' 1 'JUSQUA' C 'FAIRE'
'DEBUT'
C.(L).:=ET(D.(H,L).,E.(M,L).) $$
R:=R+PCIDS(G.(L).) $$
'FIN' $$

```



```

'SI' R 'NONEG' 1 'ALORS'
  'DEBUT' M$=M+1 $$ 'ALLERA' ITER2 $$
  'FIN'
'SINCN' 'POUR' L$=1 'PAS' 1 'JUSQUA' Q 'FAIRE'
'DEBUT'
C.(L).$=DJ(D.(F,L).,C.(M,L).) $$
E.(F,L).$=ET(D.(F,L).,C.(L).) $$
E.(M,L).$=ET(D.(M,L).,C.(L).) $$
'FIN' $$
'ALLERA' ETC $$
'FIN'
'SINCN' M$=M+1 $$ 'ALLERA' ITER2 $$
ETC$ 'SI' M=W 'ALORS' 'ALLERA' TERM
'SINCN' H$=M $$ 'ALLERA' SUITE $$
TERM$
'FIN' PARTANNULE $$
'SI' E.(V).=2 'ET' E.(W).=2 'ALORS'
'DEBUT' S$=ET(F.(V).,F.(W).) $$
F$=PCIDS(S) $$
'SI' R=1 'ALORS'
'DEBUT' ANNULCLASS(F,E,C,MA,P,Q,V,W,FI) $$
ACHEVE(D,MA,C,KA,V,W,Y,Z,P,Q,T,REUNION) $$
K$=KA $$ T$=T+KA $$
'ALLERA' FINI $$
'FIN'
'SINCN' K$=0 $$ 'ALLERA' FINI $$
'FIN'
'SINCN' 'SI' E.(V).=3 'OU' E.(W).=3 'ALORS'
'DEBUT' 'SI' E.(V).=2 'ALORS'
'DEBUT' L$=V $$ M$=W $$ 'ALLERA' ETC $$
'FIN'
'SINCN' L$=W $$ M$=V $$
ETC$ S$=1 $$
ITI$ 'SI' BIT(F.(L).,37-S)=0 'ALORS'
'DEBUT' S$=S+1 $$ 'ALLERA' ITI $$
'FIN'
'SINCN' 'SI' BIT(F.(M).,37-S)=1 'ALORS'
'ALLERA' BOSSE
'SINCN' 'ALLERA' MAGNE $$
BOSSE$ PARTANNULE(F,E,C,S,P,Q,M,L) $$
ACHEVE(D,MA,C,KA,V,W,Y,Z,P,Q,T,REUNION) $$
K$=KA $$ T$=T+KA $$
'ALLERA' FINI $$
MAGNE$ H$=1 $$ U$=0 $$
IT2$ 'SI' BIT(F.(H).,37-S)=1 'ET' E.(H).=3 'ALORS'
'DEBUT' 'SI' U 'INFEG' 0 'ALORS' 'DEBUT' R$=H $$
L$=U+1 $$ H$=H+1 $$ 'ALLERA' IT2 $$
'FIN'
'SINCN' 'ALLERA' SUI $$
'FIN'
'SINCN' H$=H+1 $$ 'ALLERA' IT2 $$
SLI$ S$=ET(F.(M).,F.(R).) $$

```



```

'SI' PCIDS(S)=1 'ALORS' U$=R 'SINCN' U$=H $$
ANLLCLASS(F,E,C,MA,P,Q,U,M,FI) $$
ACFEVE(D,MA,C,KA,V,W,Y,Z,P,C,T,RELNION) $$
K$=KA $$ T$=T+KA $$
'ALLERA' FINI $$
'FIN'
'SINCN' 'DEBUT' 'ENTIER' 'TABLEAU' PC.(1$2,1$2)..G.(1$Q). $$
S$=1 $$
ITAC1$ 'SI' BIT(F.(V)..,37-S)=C 'ALORS'
'DEBUT' S$=S+1 $$ 'ALLERA' ITAC1 $$
'FIN'
'SINCN' 'SI' BIT(F.(W)..,37-S)=1 'ALORS' 'ALLERA' MAG
'SINCN' R$=1 $$
ITAC2$ 'SI' BIT(F.(W)..,37-R)=C 'ALORS'
'DEBUT' R$=R+1 $$ 'ALLERA' ITAC2 $$
'FIN'
'SINCN' L$=1 $$ M$=1 $$ U$=0 $$ H$=1 $$
ITAC3$ 'SI' BIT(F.(H)..,37-S)=1 'ET' E.(H).=3 'ALORS'
'DEBUT' PC.(L,M).$=H $$
'SI' L 'INFEG' 0 'ALORS' 'DEBUT' U$=U+1 $$
H$=H+1 $$
L$=L+1 $$
'ALLERA' ITAC3 $$
'FIN'
'SINCN' 'ALLERA' SUMC $$
'FIN'
'SINCN' H$=H+1 $$ 'ALLERA' ITAC3 $$
SUMC$ L$=1 $$ M$=2 $$ U$=0 $$ H$=1 $$
ITAC4$ 'SI' BIT(F.(H)..,37-R)=1 'ET' E.(H).=3 'ALORS'
'DEBUT' PC.(L,M).$=H $$
'SI' U 'INFEG' 0 'ALORS' 'DEBUT' U$=U+1 $$
H$=H+1 $$
L$=L+1 $$
'ALLERA' ITAC4
'FIN'
'SINCN' 'ALLERA' ETC $$
'FIN'
'SINCN' H$=H+1 $$ 'ALLERA' ITAC4 $$
ETC$ 'SI' PC.(1,1).=PC.(1,2). 'OU' PO.(1,1).=PO.(2,2). 'ALORS'
'DEBUT' U$=PC.(1,1). $$ 'ALLERA' TRAVAIL $$
'FIN'
'SINCN' 'SI' PD.(2,1).=PC.(1,2). 'OU' PC.(2,1).=PO.(2,2). 'ALORS'
'DEBUT' U$=PC.(2,1). $$ 'ALLERA' TRAVAIL $$
'FIN'
'SINCN' 'ALLERA' BOUSI $$
MAG$ R$=1 $$
ITAC5$ 'SI' BIT(F.(R)..,37-S)=C 'OU' E.(R).=2 'ALORS'
'DEBUT' R$=R+1 $$ 'ALLERA' ITAC5 $$
'FIN'
'SINCN' H$=1 $$
ITAC6$ U$=U $$
'SI' BIT(F.(H)..,37-S)=1 'ET' E.(H).=2 'ALORS'

```



```

'FCUR' L$=1 'PAS' 1 'JUSQUA' Q 'FAIRE'
'DEBUT' G.(L).$=ET(D.(R,L).,D.(F,L).) $$
U$=U+PCIDS(C.(L).) $$
'FIN' $$
'SI' U=1 'ALORS' 'ALLERA' CEDE
'SINON' H$=H+1 $$ 'ALLERA' ITACE $$
CEDE$ H$=0 $$
'SI' H=V 'OU' H=W 'ALORS' 'ALLERA' PIED
'SINON' L$=H $$ H$=1 $$
ITAC7$ U$=0 $$
'SI' BIT(F.(H).,37-S)=1 'ET' E.(H).=2 'ET' H 'NONEG' M 'ALORS'
'FCUR' R$=1 'PAS' 1 'JUSQUA' Q 'FAIRE'
'DEBUT' G.(R).$=ET(D.(L,R).,D.(H,R).) $$
U$=U+PCIDS(G.(R).) $$
'FIN' $$
'SI' U=1 'ALORS' 'ALLERA' FEPE
'SINON' H$=H+1 $$ 'ALLERA' ITAC7 $$
FEPE$ H$=L $$ 'SI' H=V 'OU' H=W 'ALORS' 'ALLERA' PIED
'SINON' L$=H $$ H$=1 $$ 'ALLERA' ITAC7 $$
PIED$ L$=H $$ H$=1 $$
ITAC8$ U$=0 $$
'SI' BIT(F.(H).,37-S)=1 'ET' E.(H).=2 'ET' H 'NONEG' M
'ET' H 'NONEG' L 'ALORS'
'FCUR' R$=1 'PAS' 1 'JUSQUA' Q 'FAIRE'
'DEBUT' G.(R).$=ET(D.(L,R).,D.(H,R).) $$
U$=U+PCIDS(G.(R).) $$
'FIN' $$
'SI' U=1 'ALORS' 'ALLERA' MCMC
'SINON' H$=H+1 $$ 'ALLERA' ITAC8 $$
MCMC$ 'FCUR' R$=1 'PAS' 1 'JUSQUA' Q 'FAIRE'
'DEBUT' G.(R).$=DJ(L.(L,R).,D.(H,R).) $$
D.(L,R).$=ET(D.(L,R).,G.(R).) $$
D.(H,R).$=ET(D.(H,R).,G.(R).) $$
'FIN' $$
M$=L $$
'SI' H=V 'OU' H=W 'ALORS' 'DEBUT' ACHEVE(D,MA,C,KA,V,W,Y,Z,P,Q,T,
RELNCK) $$
T$=T+KA $$ K$=KA $$ 'ALLERA' FINI $$
'FIN'
'SINON' 'ALLERA' PIED $$
TRAVAIL$ 'DEBUT' 'ENTIER' 'TABLEAU' DI.(I$P,IHQ). $$
'FCUR' F$=1 'PAS' 1 'JUSQUA' P 'FAIRE'
'FCUR' L$=1 'PAS' 1 'JUSQUA' Q 'FAIRE'
DI.(F,L).$=E.(F,L). $$
FARTANLLE(F,E,D,S,P,Q,U,W) $$
ACHEVE(D,MA,C,KA,V,W,Y,Z,P,Q,T,REUNION) $$
K$=KA $$ T$=T+KA $$
FARTANLLE(F,E,D,I,R,P,Q,U,W) $$
ACHEVE(DI,MA,C,KA,V,W,Y,Z,P,Q,T,REUNION) $$
K$=K+KA $$ T$=T+KA $$
'ALLERA' FINI $$
'FIN' $$

```



```

BCUSI$ 'DEBUT' 'ENTIER' 'TABLEAU' D1,D2.(1$P,1$Q). $$
      'ENTIER' 'TABLEAU' G.(1 $ 2). $$
      'FCUR' H$=1 'PAS' 1 'JUSQUA' P 'FAIRE'
      'FOUR' L$=1 'PAS' 1 'JUSQUA' Q 'FAIRE'
      D2.(H,L).$=D1.(H,L).$=D.(H,L). $$
      F$=1 $$ U$=C $$ L$=1 $$
EBEBE$ 'SI' E.(H).=3 'ET' H 'NCNEG' PO.(1,1). 'ET' H 'NONEG' PC.(1,2).
      'ET' F 'NCNEG' PO.(2,1). 'ET' H 'NCNEG' PO.(2,2). 'ALORS'
      'DEBUT' G.(L).$=F $$
      'SI' U 'INFEG' U 'ALORS' 'DEBUT' H$=H+1 $$
      U$=U+1 $$
      L$=L+1 $$
      'ALLERA' EBEBE $$
      'FIN'
      'SINCN' 'ALLERA' GOGC $$
      'FIN'
      'SINCN' H$=H+1 $$ 'ALLERA' EBEBE $$
GCCG$ F$=PC.(1,1). $$ L$=PO.(1,2). $$ R$=PO.(2,1). $$
      M$=ET(F.(H).,F.(L).) $$
      L$=PCIDS(M) $$
      'SI' U=1 'ALORS' 'DEBUT' S$=PC.(2,2). $$ 'ALLERA' JIJI $$ 'FIN'
      'SINCN' S$=L $$
      L$=PC.(2,2). $$
JIJI$ ANNULCLASS(F,E,D,MA,P,Q,G.(1).,G.(2).,F1) $$
      ACHEVE(D,MA,C,KA,V,W,Y,Z,P,Q,T,REUNION) $$
      K$=KA $$ T$=T+K $$
      ANNULCLASS(F,E,D1,MA,P,C,H,L,F1) $$
      ACHEVE(D1,MA,C,KA,V,W,Y,Z,P,Q,T,REUNION) $$
      T$=T+KA $$ K$=K+KA $$
      ANNULCLASS(F,E,D2,MA,P,C,R,S,F1) $$
      ACHEVE(D2,MA,C,KA,V,W,Y,Z,P,Q,T,REUNION) $$
      T$=T+KA $$ K$=K+KA $$
      'FIN' $$
      'FIN' $$
      'FIN' $$
      'FIN' $$
FINI$
      'FIN' NEWKURAZ $$

```



1.2.5. Programmes généraux.

1<sup>er</sup> Programme : figure de départ totale.

- 5680 unités syntaxiques

- temps d'assemblage et de compilation : 2'20"

2<sup>ème</sup> Programme : figure de départ quelconque

- 6360 unités syntaxiques

- temps d'assemblage et de compilation : 2'24"

1.3. Résultats.

1<sup>er</sup> Exemple.

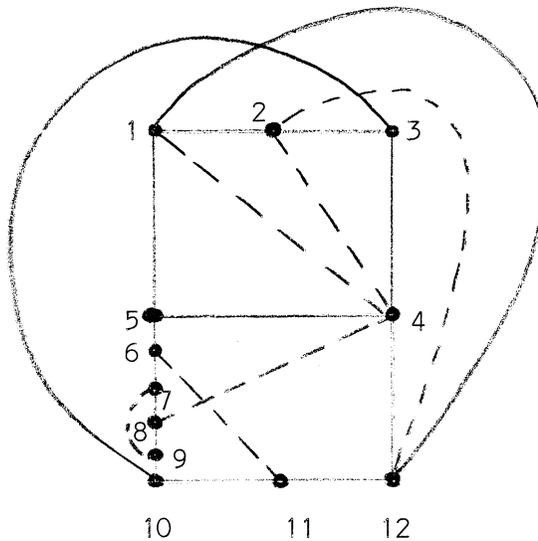


Fig. 2

G { 12 sommets  
21 arêtes  
6 éléments

application du 1<sup>er</sup> algorithme, avec une seule figure de départ (figures non atteintes par l'algorithme général non cherchées).



nombres de figures K2 = 61
temps travail = 47"

nous obtenons comme ensembles minimaux d'arêtes à enlever, les arêtes :

$\{1,2\}$  ,  $\{1,5\}$  ,  $\{2,3\}$  ,  $\{3,6\}$  ,  $\{4,7\}$  ,  $\{5,11\}$  ,  $\{7,12\}$

nous pouvons vérifier que toutes ces solutions sont exactes, donc que toutes les figures de types K2 contiennent les arêtes ci-dessus.

Exemple.

Arête  $[1,2]$  enlevée.

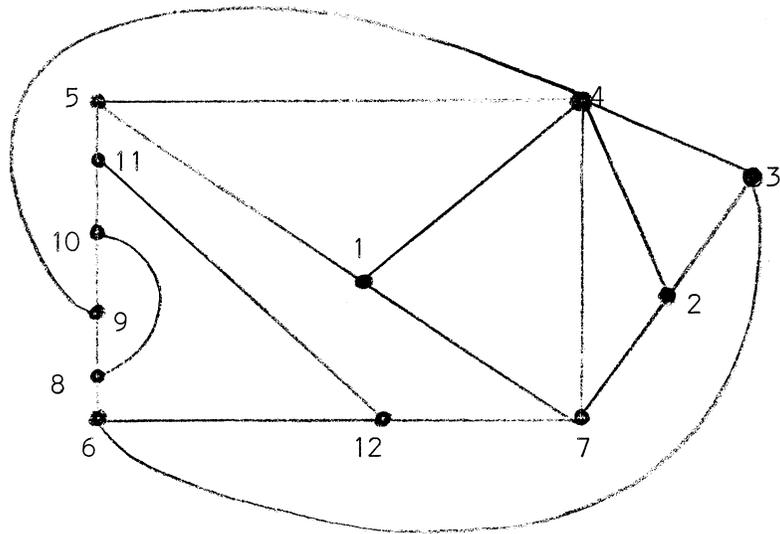


Fig. 3

2<sup>ème</sup> Exemple.

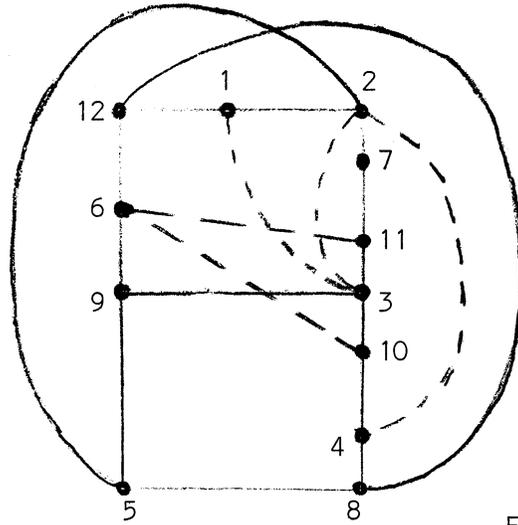


Fig. 4

G { 12 sommets  
20 arêtes  
5 éléments

application du 1<sup>er</sup> programme avec une seule figure de départ

nombre de figures K2 = 101
temps travail = 56"

3<sup>ème</sup> Exemple.

G idem à la Fig. 4 avec, en plus l'arête [7,8] comme élément.

G { 12 sommets  
21 arêtes  
6 éléments

application du 3<sup>er</sup> programme avec une seule figure de départ

nombre de K2 = 336
temps travail = 6'06"

Lorsque l'on applique le 1<sup>er</sup> programme avec 13 figures de départ ( $G_0$  + figures non atteintes) on obtient

nombre de figures K2 = 371
temps travail = 7'17"

4<sup>ème</sup> Exemple.

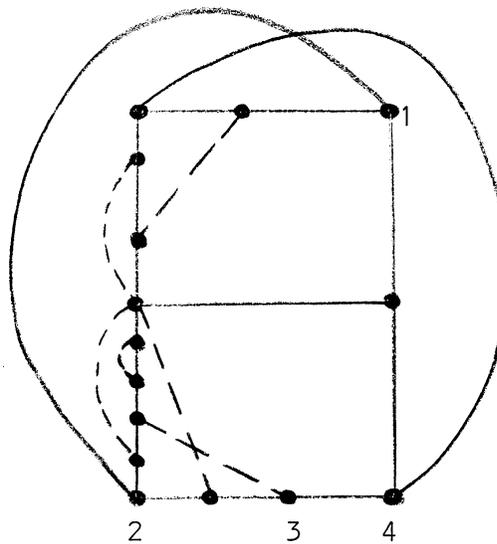


Fig. 5

G { 15 sommets  
24 arêtes  
6 éléments

application du 1<sup>er</sup> programme avec 24 figures de départ

nombre de figures K2 = 11
temps travail = 2'17"

Nous obtenons comme ensembles minimaux d'arêtes à enlever les 2 arêtes :  
{1,2} , {3,4} .

5<sup>ème</sup> Exemple.

G identique à celui du 3<sup>ème</sup> exemple.

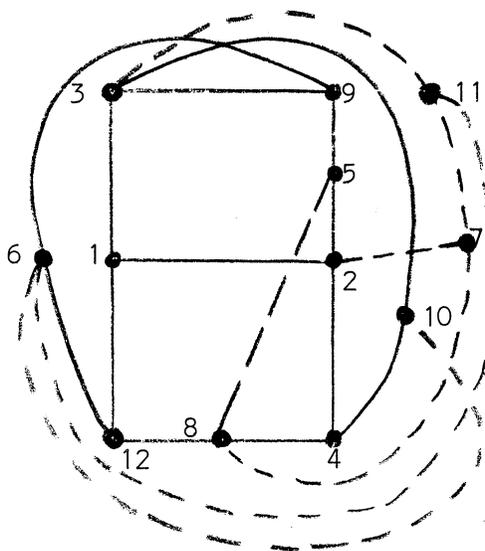


Fig. 6

application du 2<sup>ème</sup> programme avec 1 figure de départ.

nombre de figures K2 = 287
temps travail = 12'26"

1.4. Conclusions.

- Le temps de calcul croît très vite avec le nombre de figures trouvées (sans doute exponentiellement). Pour un élément rajouté à G on passe de 101 à 336 figures et de 56" à 6'06" (2<sup>ème</sup> et 3<sup>ème</sup> exemple).

- Le premier exemple montre qu'il n'est pas obligatoire de chercher les figures non atteintes par l'algorithme général pour obtenir une solution exacte. On peut donc envisager, pour un graphe ayant un grand nombre de figures, d'arrêter le déroulement du programme au bout d'un certain temps, de chercher un ensemble minimal d'arêtes à enlever dans le tableau partiel des figures, et de traiter ensuite le graphe obtenu à partir de G après suppression de cet en-

semble d'arêtes minimal. On peut d'ailleurs itérer le processus jusqu'à obtenir  $G$  planaire.

- On voit que pour un même graphe, le temps de travail augmente beaucoup avec la complexité des éléments. (3<sup>ème</sup> et 5<sup>ème</sup> exemple). D'où le grand intérêt de la recherche du cycle le plus long, dans  $G$ , avant le traitement du problème, ce qui peut conduire à une figure de départ,  $G_0$ , totale.



## DEUXIEME PARTIE

### IMPLANTATION DES ELEMENTS ELECTRONIQUES

- Introduction.

Chapitre I - La dispersion barycentrique.

Chapitre II - Non chevauchement.

Chapitre III - Programmes - Résultats.



## INTRODUCTION

Nous abordons ici le problème concret de la réalisation d'un circuit intégré. Sur une plaquette les éléments devront être disposés côte à côte et leurs interconnexions réalisées. La technique planar impose que tout doit être fait sur un seul plan, en respectant diverses contraintes, dont nous avons déjà parlé (cf. 2, ch. 11) et notamment :

- non superposition des éléments, avec respect d'une garde entre eux-ci.
- sorties sur le pourtour de la plaquette, l'ordre de rencontre de ces dernières sur le périmètre de la plaquette étant, soit imposé, soit quelconque.
- rassemblement, des résistances en un nombre réduit de caissons, des transistors à même tension collecteur dans le même caisson.
- recherche de l'occupation minimale de la surface et d'une répartition régulière des éléments.

Nous allons étudier, dans cette 2<sup>ème</sup> partie, le problème de la disposition des éléments sur une surface plane, de façon à obtenir la meilleure occupation possible de cette dernière.

# CHAPITRE I

## DISPERSION BARYCENTRIQUE

### 1 - POSITION DU PROBLEME

Afin d'obtenir une disposition relative des éléments les uns par rapport aux autres, nous avons été amenés à écrire un programme qui simule les liaisons électriques existant entre les différents éléments, par des forces d'attraction.

Nous nous plaçons sur une surface rectangulaire de dimensions fictives, et nous assimilons tout élément du circuit à un petit carré de dimensions réduites.

Nous fixerons a priori la position des sorties sur le pourtour de la plaquette, et, si cela apparaît évident ensuite, nous pourrions fixer, aussi, la position de certains éléments très liés à certaines sorties. Pour résoudre le problème ainsi posé nous avons utilisé, comme algorithme de travail, la dispersion barycentrique, c'est-à-dire : tout élément sera placé au barycentre des éléments auxquels il est électriquement relié.

### 2 - DEROULEMENT DE L'ALGORITHME

Le réseau électrique est assimilé à un graphe, chaque équipotentielle électrique conduisant à un sous-graphe complet, relatif aux éléments de cette équipotentielle.

Il est à noter que le grand intérêt de cette méthode est de pouvoir pondérer différemment, lors de plusieurs passages du programme, les différentes attractions entre éléments.

En effet, il nous est apparu immédiatement que les sorties devaient attirer très fortement les autres éléments, sous peine de voir tous les éléments "libres" groupés au centre de la plaquette.

Cette possibilité est très intéressante car l'on peut alors imposer la proximité d'éléments d'un groupe bien déterminé, en les faisant s'attirer très fortement, s'ils sont reliés électriquement.

### 3 - METHODE DE RESOLUTION PAR RELAXATION

#### 3.1. Généralités.

Cette méthode permet de traiter tous les exemples, mais est à utiliser seulement lorsque le nombre d'éléments du circuit est très grand.

Si le nombre d'éléments est  $\leq 100$ , alors les méthodes directes de résolution sont à conseiller, car elles conduisent à une solution exacte et ceci plus rapidement.

#### 3.2. Principes de la méthode.

La méthode consiste à faire évoluer successivement chacun des éléments au barycentre des éléments qui lui sont liés, et qui sont alors considérés comme fixes.

On itère cette méthode jusqu'à ce que les déplacements relatifs des éléments soient faibles.

### 4 - METHODES DIRECTES DE RESOLUTION D'UN SYSTEME LINEAIRE

#### 4.1. Le système linéaire.

Partant de la matrice de jointivité des éléments, pondérée éventuellement par le concepteur, et d'une liste d'éléments fixes, nous devons résoudre le problème suivant :

Soit  $T$  la matrice de jointivité de tous les éléments du réseau.

$T$  est carrée ( $n \times n$ ) si  $n$  est le nombre d'éléments du réseau.

Placer un élément au barycentre des éléments auxquels il est lié revient à écrire,  $X$  étant la colonne des abscisses (respectivement des ordonnées) de tous les éléments :

$$\forall_i \text{ on a } x_i = \frac{\sum_{j \neq i} t_{ij} \cdot x_j}{\sum_{j \neq i} t_{ij}}$$

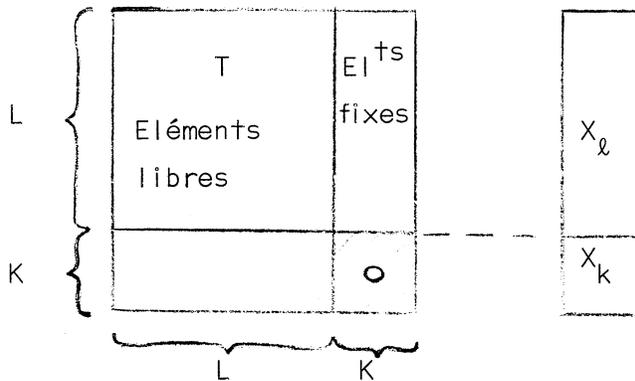
$$\text{soit } \left( \sum_{j \neq i} t_{ij} \right) \cdot x_i = \sum_{j \neq i} t_{ij} \cdot x_j \quad \forall_i$$

Si nous faisons la distinction entre les éléments fixes et les éléments libres, nous obtenons :

$$\text{pour tout } i \quad \left( \sum_{\substack{j \in J \\ j \neq i}} t_{ij} \right) \cdot x_i = \sum_{\substack{k \in K \\ k \neq i}} t_{ik} \cdot x_k + \sum_{\substack{l \in L \\ l \neq i}} t_{il} \cdot x_l$$

où  $l = K + L$      $K$  : ensemble des éléments fixes.  
 et  $J = K + L$      $L$  : ensembles des éléments libres.

Soient les découpages des matrices T et X, suivants :



pour  $i \in L$  nous aurons alors à résoudre :

$$\underbrace{\left( \sum_{\substack{j \in L \\ j \neq i}} t_{ij} + \sum_{\substack{j \in K \\ j \neq i}} t_{ij} \right)}_{\text{connu}} \cdot x_i = \sum_{\substack{l \in L \\ l \neq i}} t_{il} \cdot x_l + \underbrace{\sum_{\substack{k \in K \\ k \neq i}} t_{ik} \cdot x_k}_{\text{connu}}$$

Nous aurons à résoudre le système linéaire suivant :

$$\left( \sum_{\substack{j \in L \\ j \neq i}} t_{ij} + \sum_{\substack{j \in K \\ j \neq i}} t_{ij} \right) \cdot x_i - \sum_{\substack{l \in L \\ l \neq i}} t_{il} \cdot x_l = \sum_{\substack{k \in K \\ k \neq i}} t_{ik} \cdot x_k$$

pour  $\forall i \in L$ .

Posons  $|K| = k$  ;  $|L| = l$  et  $|J| = n$

Soient donc :

$$A = (a_{pq}) ; B = (b_p) ; X = (x_q)$$

avec :  $A = (l \times l)$  ;  $B = (l)$  ;  $X = (l)$

$$a_{pp} = \sum_{\substack{j \in L \\ j \neq p}} t_{pj} / p \in L$$

$$a_{pq} = - t_{pq} / p, q \in L$$

$$x_q = x_q / q \in L$$

$$b_p = \sum_{\substack{j \in K \\ j \neq p}} t_{pj} \cdot x_j / p \in L$$

Nous sommes alors amenés à résoudre :

$A \cdot X = B$  avec  $l$  équations et  $l$  inconnues,  $A$  étant une matrice symétrique.

#### 4.2. Méthodes de résolution utilisées.

Deux méthodes ont été retenues pour résoudre ce système. Ce sont :

a) la méthode de GAUSS-SEIDEL

b) la méthode de CHOLESKI, qui ne peut s'appliquer que si  $A$  est définie positive.

Ces deux méthodes sont exposées en détail dans les cours d'Analyse Numérique.

## C H A P I T R E II

### NON CHEVAUCHEMENT

#### 1 - DEFINITIONS

Nous devons vérifier qu'un dessin à l'échelle respecte les marges d'écartement des éléments et, au cas où il n'en serait pas ainsi, à déplacer les éléments suivant un algorithme très simple.

Les éléments sont représentés, à l'échelle, en dimensions réelles, et nous tiendrons compte de l'appartenance des éléments aux différents caissons. De plus, une orientation des éléments sera introduite, orientation que l'on peut déduire, par exemple, de l'étude de la planéarité du réseau électronique.

Ceci nécessite la création d'une bibliothèque simplifiée des éléments du réseau. Nous assimilerons les éléments à des rectangles et nous ne mentionnerons à l'intérieur de ces rectangles que l'emplacement et la taille des différents contacts.

#### 2 - DEROULEMENT DE L'ALGORITHME

Nous partons d'un dessin à main levée où les éléments sont représentés sans grand souci de leurs proportions. Nous obtiendrons une solution, à partir de laquelle nous chercherons le meilleur résultat possible, en effectuant plusieurs passages en machine. C'est le concepteur, qui, par le choix de certains déplacements, au vu des résultats intermédiaires, guidera le processus vers la solution optimale.

Toute liberté est laissée au concepteur de fixer certains éléments à des positions déterminées.

### 3 - ALGORITHME DE NON CHEVAUCHEMENT

Pour ce problème où il ne s'agit que de détecter des chevauchements de rectangles, nous opérons en coordonnées vraies, sans faire intervenir les notions de zone libre, zone occupée, ou quadrillage de l'espace.

L'algorithme nécessite une répartition préalable des éléments en différentes colonnes.

Tous les éléments sont étudiés l'un après l'autre dans l'ordre où ils sont donnés (par colonne, en partant de la gauche) et sont testés par rapport à tous ceux qui les précèdent.

Prenons le  $n^{\text{ième}}$  élément :

Il est testé par rapport au  $(n-1)^{\text{ième}}$ , puis au  $(n-2)^{\text{ième}}$  etc ... soit à tous les éléments appartenant à la même colonne qui le précèdent. Si les gardes ne sont pas respectées, le  $n^{\text{ième}}$  élément est repoussé vers le bas de la quantité minimale.

A partir de cette position le  $n^{\text{ième}}$  élément est testé par rapport à tous les autres éléments précédents qui n'appartiennent pas à la même colonne. Si il y a un chevauchement illicite, le  $n^{\text{ième}}$  élément est repoussé vers la droite de la quantité minimale.

Les déplacements vers le bas ou vers la droite tiennent compte des exceptions dues aux très gros éléments, ou aux éléments longs et étroits, afin d'occuper au mieux l'espace.

Il est à noter que cet algorithme ne fait que "desserrer" les éléments trop rapprochés. Il ne "tasse" pas la disposition et peut donc respecter les intentions du concepteur, qui aurait espacé deux éléments afin de prévoir le passage des connexions.

## C H A P I T R E III

### PROGRAMMES - RESULTATS

#### 1 - DISPERSION BARYCENTRIQUE

##### 1.1. Parties communes aux trois méthodes.

Les programmes relatifs aux trois méthodes existent en deux versions "mixées" en chaque programme. La version 1 est prise en compte lorsque les sorties seules sont fixes et leur position déterminée à l'avance. Si d'autres éléments sont fixés, alors on passe en version 2 du programme.

Les différentes méthodes ont été programmées en FORTRAN IV. Les passages "machine" ont eu lieu sur IBM 7044, sur IBM 360/40 ou sur IBM 360/65, indifféremment.

Les parties des programmes communes aux trois méthodes sont :

##### 1.1.1. Les entrées.

se composent de :

a) la liste des noeuds équipotentiels du réseau.

Exemple : 1 noeud/carte donnée :

EM10EM09C011AN03

ce qui se traduit par :

émetteur du transistor . 10

émetteur du transistor 09

collecteur du transistor 11

anode de la diode 03

b) la liste des éléments fixes.

Exemple : TR05TR15

c) les éléments à coupler très fortement.

Exemple : TR03Di01

d) les coordonnées des éléments fixes dans l'ordre de la liste.

### 1.1.2. La procédure de sortie.

Afin d'éviter la lecture fastidieuse d'un tableau de coordonnées, nous avons été amenés à écrire un sous-programme, qui dessine les résultats sur une feuille de listing, dans les limites (en largeur) de cette dernière.

```
0 :IBFTC TRACAJ DECK
1   SUBROUTINE TRACAJ(A,C ,AA,BB)
2   DIMENSION A(1),C(1),LINSU1(63),LINSU2(63),IA(99),IB(99),POINT(63)
3   COMMON NS,IT,ID,IC,M,BLANC,RMOINS,BET(15),B(60,90),RESUL(70)
4   1 ,ANUM(99)
5   REAL LINSU1,LINSU2
6   D=AA/60.0
7   IL=INT((BB+10.)*1.2/D+0.5) +1
8   DC 1 I=1,M
9   IA(I)=INT((A(I)+10.)/D+0.5)
10  1 IB(I)=INT((C(I)+10.)*1.2/D+0.5)
11  DC106LM=1,3
12  DC 3 I=1,63
13  LINSU1(I)= BLANC
14  LINSU2(I)= BLANC
15  3 POINT(I)= BLANC
16  WRITE(6,97)
17  97 FORMAT(1H1)
18  WRITE(06,99)
19  DC 2 L=0,IL
20  DO 4 I=1,M
21  IF(IB(I).NE.IL-L+1) GO TO 4
22  IW=IA(I)
23  DO 5 K=1,2
24  IK=IW-1+K
25  POINT(IK)=RMOINS
26  5 LINSU2(IK)= RMOINS
27  LINSU1(IW)=BNATU(I)
28  IK=IW+1
29  JA=INT(DEKOD(I) +0.1)
30  LINSU1(IK)=ANUM(JA)
31  4 CONTINUE
32  WRITE(06,101) ( POINT(I),I=1,63)
33  DO 2 K=1,63
34  POINT(K)=LINSU1(K)
35  LINSU1(K)=LINSU2(K)
36  2 LINSU2(K)= BLANC
37  WRITE(06,99)
38  99 FORMAT(126H -----)
39  1-----)
40  101 FORMAT(2H I,63A2,1H1)
41  106 CONTINUE
42  RETURN
43  END
```



1.1.3. Les sous-programmes de traitement.

```
0 :IBFTC NUM      DECK
1      FUNCTIONNUM(L,K)
2      COMMON NS,IT,ID,IC,M,BLANC,RMCINS,BET(15),B(60,90)
3      BC=B(L,K-1)
4      IF(BC.NE.BLANC) GOTO 1
7      NUM=0
10     GOTO 2
11     1  CCNTINUE
12     IF(BC      .EQ.BET(8) )   NUM=INT(B(L,K)+0.1)
15     IF(BC      .EQ.BET(1) )  NUM=INT(B(L,K)+0.1) +NS
20     IF(BC      .EQ.BET(2) )  NUM=INT(B(L,K)+0.1) +NS
23     IF(BC      .EQ.BET(3) )  NUM=INT(B(L,K)+0.1) +NS
26     IF(BC      .EQ.BET(4) )  NUM=INT(B(L,K)+0.1) +IT
31     IF(BC      .EQ.BET(5) )  NUM=INT(B(L,K)+0.1) +IT
34     IF(BC      .EQ.BET(9) )  NUM=INT(B(L,K)+0.1) +ID
37     IF(BC      .EQ.BET(10) ) NUM=INT(B(L,K)+0.1) +ID
42     IF(BC      .EQ.BET(6) )  NUM=INT(B(L,K)+0.1) +IC
45     IF(BC      .EQ.BET(7) )  NUM=INT(B(L,K)+0.1) +IC
50     2  CCNTINUE
51     RETURN
52     END
```

---

```
0 :IBFTC DEKOD    DECK
1      FUNCTIONDEKOD(I)
2      COMMON NS,IT,ID,IC,M
3      DEKOD= I-IC
4      IF(I.LE.IC)  DEKOD= I-IC
7      IF(I.LE.ID)  DEKOD= I-IT
12     IF(I.LE.IT)  DEKOD= I-NS
15     IF(I.LE.NS)  DEKOD= I
20     RETURN
21     END
```

---



```
0 :IBFTC BNATU DECK
1     FUNCTION BNATU(I)
2     COMMON NS,IT, ID, IC, M, BLANC, RPCINS, BET(15)
3     BNATU= BET(15)
4     IF(I.LE.IC) BNATU = BET(13)
7     IF(I.LE.ID) BNATU = BET(12)
12    IF(I.LE.IT) BNATU = BET(11)
15    IF(I.LE.NS) BNATU = BET(14)
20    RETURN
21    END
```

---

```
0 :IBFTC TYPE DECK
1     FUNCTION TYPE(I,N)
2     COMMON NS,IT, ID, IC, M, BLANC, RPCINS, BET(15), B(60,90)
3     DO80L=1,N
4     DO80K=2,89,3
5     IF(NUM(L,K).NE.I) GO TO 80
10    TYPE = B(L,K+1)
11    GOTO82
12    P0 CONTINUE
15    P2 CONTINUE
16    RETURN
17    END
```



### 1.2. Méthode de relaxation.

Le programme comporte 740 instructions ISN. Sur IBM 7044, le temps de compilation et d'assemblage est de 2'06".

Pour 41 éléments, dont 16 étaient fixés, le temps de travail sur IBM 7044 fut de 10" par passage.

### 1.3. Méthode de GAUSS-SEIDEL.

#### 1.3.1. Sous-programme de traitement.

```
C : IBFTC GAUSS
1     SUBROUTINE GAUSS(A,B,X,EPS,N)
2     DIMENSION A(50,50),B(50),X(50),R(50)
C   RESIDUS INITIALX
3     K=1
4     PHI=C.C
5     DO 200 I=1,N
6     T=C.C
7     DO 201 J=1,N
10    201 T=T+A(I,J)*X(J)
12    R(I)=T-B(I)
13    S=ABS(R(I))
14    IF(S .LE. PHI ) GO TO 200
17    PHI=S
20    200 CONTINUE
C   BOUCLE
22    204 Q=R(K)/A(K,K)
23    X(K)=X(K)-Q
24    R(K)=C.C
25    PHI=C.C
26    DO 202 I=1,N
27    IF(I .EQ. K) GO TO 202
32    R(I)=R(I)-A(I,K)*Q
33    S=ABS(R(I))
34    IF(S .LE. PHI) GO TO 202
37    PHI=S
40    202 CONTINUE
42    IF(PHI.LT. EPS) GO TO 203
45    K=K+1
46    IF(K.LE. N) GO TO 204
51    K=1
52    GO TO 204
C   SORTIE
53    203 CONTINUE
54    RETURN
55    END
```



### 1.3.2. Le programme.

Il fait 782 instructions ISN.

Temps d'assemblage et de compilation :

- sur IBM 360/65 - FORT : 24" - LKED : 19"

- sur IBM 360/40 - FORT : 186" - LKED : 58".

### 1.3.3. Résultats.

Pour 41 éléments, dont 16 étaient fixés, temps de travail :

sur IBM 360/65 : 4,7 secondes par passage

sur IBM 360/40 : 36 secondes par passage.

## 1.4. Méthode de CHOLESKI.

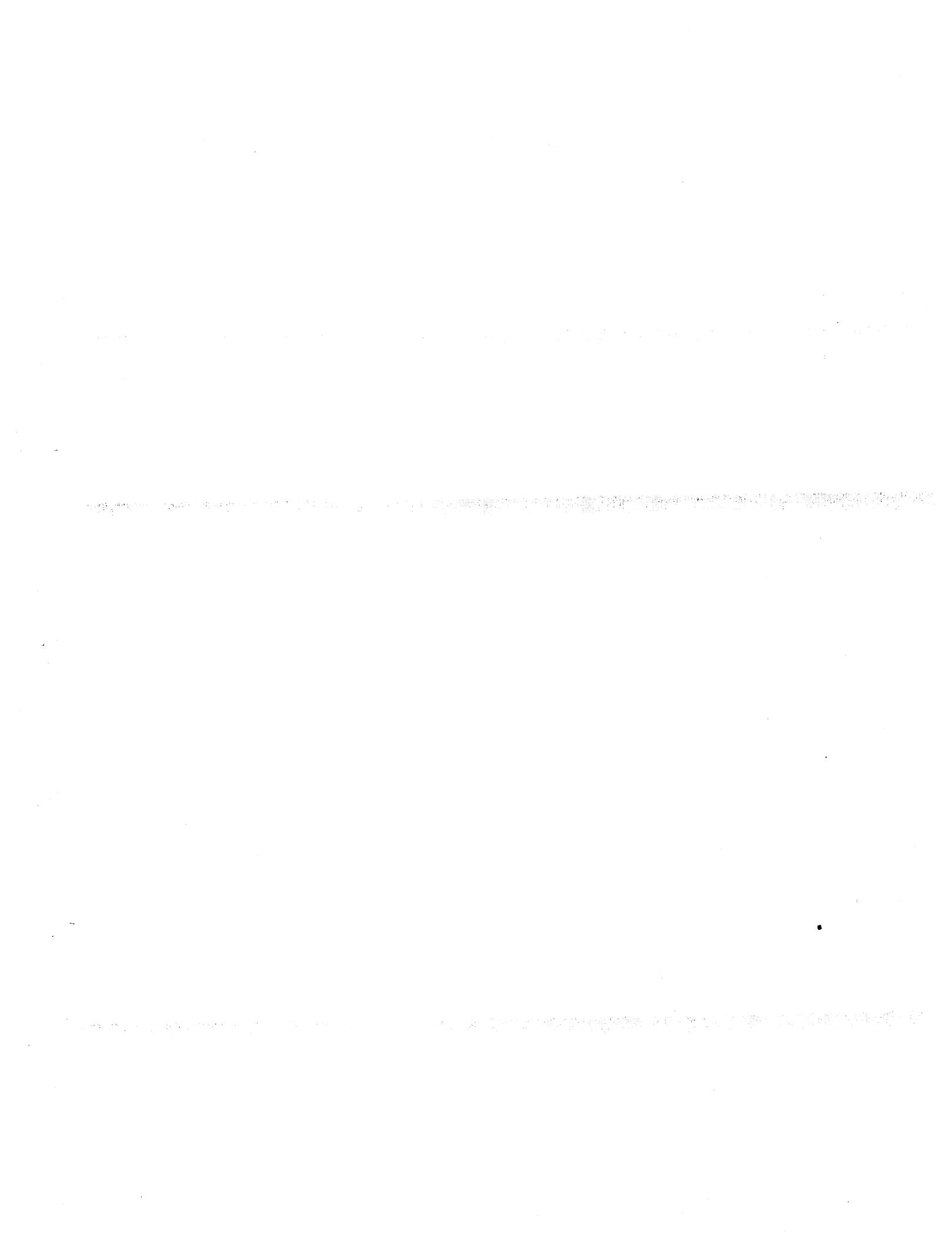
### 1.4.1. Sous-programme de traitement.



```

C :IBFTC RESOL
1   SUBROUTINE RESOL (A,B,N,X,MOVAI)
2   DIMENSION A(50,50), B(1),X(1)
3   LOGICAL MOVAI
4   A1=A(1,1)
5   IF(A1.LE.C.C001) GO TO 100
10  A1= SQRT(A1)
11  A(1,1)=A1
12  DO 101 I=2,N
13  101 A(I,1)=A(I,1)/A1
15  DO 102 J=2,N
16  S=0.0
17  JJ=J-1
20  DO 103 I=1,JJ
21  103 S=S+A(J,I)*A(J,I)
23  S=A(J,J)-S
24  IF(S.LE. C.C001) GO TO 100
27  A1=SQRT(S)
30  A(J,J)=A1
31  JK=J+1
32  DO 104 I=JK,N
33  S=C.0
34  DO 105 K=1,JJ
35  105 S=S+A(I,K)*A(J,K)
37  S=A(I,J)-S
40  A(I,J)=S/A1
41  104 CONTINUE
43  102 CONTINUE
C.....RESOLUTION 1
45  DO 106 I=1,N
46  A1=A(I,1)
47  TX=C.0
50  JJ=I-1
51  DO 107 J=1,JJ
52  107 TX=TX+A(I,J)*X(J)
54  X(I) = (B(I)-TX)/A1
55  B (I)=X(I)
56  106 CONTINUE
C.....RESOLUTION 2
60  DO 108 K=1,N
61  I=N-K+1
62  A1=A(I,1)
63  TX=C.0
64  JJ=I+1
65  IF(JJ.GT. N) GO TO 111
70  DO 109 JK=JJ,N
71  J=N-JK+JJ
72  109 TX=TX+A(J,I)*X(J)
74  111 X(I)=(B(I)-TX)/A1
75  108 CONTINUE
77  MOVAI=.FALSE.
100 GO TO 110
101 100 MOVAI=.TRUE.
102 110 CONTINUE
103 RETURN
104 END

```



#### 1.4.2. Le programme.

Il fait 824 instructions ISN.

Le temps d'assemblage et de compilation sur IBM 360/65 est de :

FORT : 31"    LKED : 24 "

#### 1.4.3. Résultats.

Pour 41 éléments dont 14 étaient fixés : temps de travail = 16"  
par passage.

#### 1.5. Conclusions.

Ayant testé ces différentes méthodes sur plusieurs exemples, chaque exemple étant traité selon les trois méthodes (cf. résultats ci-dessus) nous avons choisi, par la suite, de n'utiliser, pour des circuits de taille moyenne, que la méthode de GAUSS-SEIDEL.

### 2 - NON CHEVAUCHEMENT

Le langage utilisé dans ce programme est le FORTRAN IV. Les passages en machine ont eu lieu sur IBM 7044, sur IBM 360/40 et sur IBM 360/65.

#### 2.1. Les entrées.

Les cartes données sont composées de :

a) une carte donnant le nombre de types différents de transistors, diodes, résistances collecteurs et sorties qui sont dans la bibliothèque.

b) La bibliothèque des différents types d'éléments. Les éléments sont classés dans l'ordre ci-dessus (cf. a), et dans chaque catégorie, par numéro croissant de type.

Chaque type d'élément est défini sur 2 cartes :

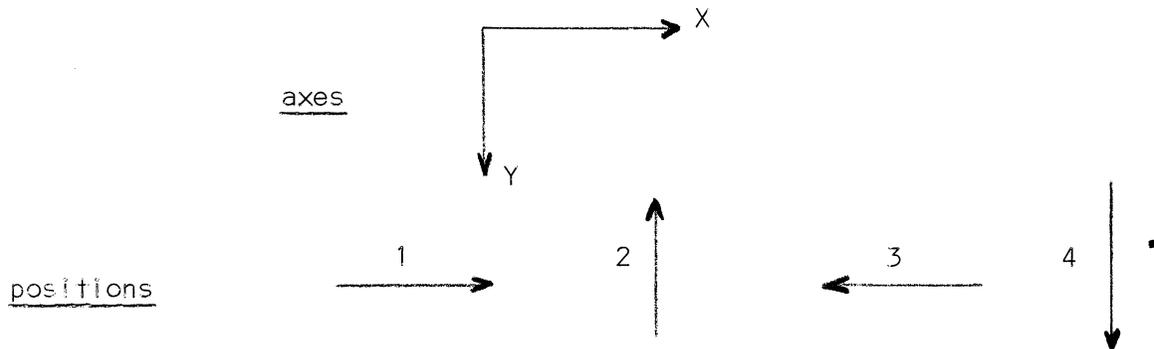
- 1<sup>ère</sup> carte donne, par séries de trois chiffres, la longueur, puis la largeur de :

- l'élément
- 2 contacts base (ou 1 contact résistance)
- 6 contacts émetteur
- 2 contacts collecteur (ou 1 contact résistance).

- 2<sup>ème</sup> carte donne, par séries de trois chiffres, les coordonnées du point supérieur gauche, en position relative/point supérieur gauche de l'élément en position 1, de :

- 2 contacts base (ou 1 contact résistance)
- 6 contacts émetteur
- 2 contacts collecteur (ou 1 contact résistance).

- les coordonnées sont exprimées en microns ; les axes sont pris dans le sens rétrograde et les éléments dans une position conventionnelle notée 1.



c) Une carte donnant le nombre d'éléments du circuit étudié, puis le nombre de colonnes choisi.

d) Une carte indique les gardes, en microns, entre :

- deux caissons différents
- un élément et le bord de son caisson
- deux éléments appartenant au même caisson.

e) Une carte par élément donnant les 7 renseignements suivants :

- nature de l'élément :

Exemple : SO, RE, TR, DI etc ...

- numéro de l'élément dans le circuit
- type de l'élément (numéro dans la bibliothèque)
- abscisse du point supérieur gauche, en microns
- ordonnée du point supérieur gauche, en microns
- orientation de l'élément (1, 2, 3, ou 4)
- numéro du caisson auquel appartient l'élément.

f) Une carte par colonne donnant le nombre d'éléments dans cette colonne.

## 2.2. Procédure de sortie.

Pour des raisons identiques à celles exposées en 1.1.2. de ce chapitre, nous avons été amenés à faire "dessiner" l'imprimante de l'ordinateur, n'ayant pas de table traçante, à cette époque, au Laboratoire.



C : IBFIC MIZPAJ

```

1      SUBROUTINE MIZPAJ (N,M,ISLP)
2      DIMENSION IA(100,52),LIGNE(15,120)
3      COMMON T(100,52),REF(5),DX,DY,BIBLI(2,44)
4      COMMON NB,TYP,NICTIR,NTCTDI,NTCTRE,NTCTCC
5      COMMON CARACT(5),C,CP,DEE,DEC,DCC
6      COMMON IX,IPC,IPC,IBL,IW,II
7      DO 20 I=1,ISLP
10     DO 20 J=1,120
11     20 LIGNE(I,J)=IBL
14     DO 21 I=1,N
15     IA(I,4) =INT( T(I,4) /10.0+0.5)
16     IA(I,5) =INT( T(I,5) /10.0+0.5)
17     DO 21 J=10,M
20     21 IA(I,J) =INT( T(I,J) /10.0+0.5)
23     WRITE(6,22) ((IA(I,J),J=4,5),I=1,N)
34     WRITE(6,23)
35     WRITE(6,24)
36     22 FORMAT(1X,2I8)
37     23 FORMAT(1H1)
40     24 FORMAT(123H 1-----
1-----
2-1)
41     DO 30 I=5,155
42     DO 26 J=1,N
43     IF(IA(J,4) .NE. 1) GO TO 26
44     IB=IA(J,5)
45     IC=IA(J,10)
46     ID=IA(J,11)
47     DO 25 K=1,ID
52     L=121-IB-K
53     LIGNE(I,L)=IW
54     LIGNE(IC,L)=IW
55     25 CONTINUE
57     IC=IC-1
60     DO 26 K=2,IC
61     IR=120-IP
62     LIGNE(K,IR)=IW
63     IR = IR+1-ID
64     LIGNE(K,IR)=IW
65     26 CONTINUE
70     WRITE(6,27) (LIGNE(I,K),K=1,120)
75     27 FORMAT(2H 1,12JA1,1H1)
76     IS=ISLP-1
77     DO 28 L=1,IS
100    IR=L+1
101    DO 28 J=1,120
102    28 LIGNE(L,J)=LIGNE(IR,J)
105    DO 29 J=1,120
106    29 LIGNE(ISLP,J)=IBL
110    30 CONTINUE
112    WRITE(6,24)
113    RETURN
114    END

```



### 2.3. Sous-programmes de traitement.

```

C : IBF1C NLMER
1   FUNCTION NLMER (A)
2   INTEGER A
3   COMMON T(100,52),REF(5),DX,DY,BIBLI(20,44)
4   COMMON NB TYP,NTCTTR,NTCTDI,NTCTRE,NTCTCC
5   COMMON CARACT(5),Q,QP,DEE,DEC,DCC
6   TA=T(A,1)
7   IF(TA .EQ.REF(1))NLMER=INT(T(A,3)+0.1)
12  IF(TA .EQ.REF(2))NLMER=INT(T(A,3)+0.1) + NTCTTR
15  IF(TA .EQ.REF(3))NLMER=INT(T(A,3)+0.1) + NTCTDI
20  IF(TA .EQ.REF(4))NLMER=INT(T(A,3)+0.1) + NTCTRE
23  IF(TA .EQ.REF(5))NLMER=INT(T(A,3)+0.1) + NTCTCC
26  RETURN
27  END

C : IBF1C CHOIPR
1   SUBROUTINE CHOIPR(A,B,PRCC,ECR)
2   EXTERNAL ECR
3   INTEGER A,B
4   COMMON T(100,52),REF(5),DX,DY,BIBLI(20,44)
5   COMMON NB TYP,NTCTTR,NTCTDI,NTCTRE,NTCTCC
6   COMMON CARACT(5),Q,QP,DEE,DEC,DCC
7   TA=T(A,7)
10  TB=T(B,7)
11  IF(TA .GT.0.9.AND.ABS(TA -TB) .LT..001)
1   CALL PROC(A,B,DEE,ECR)
14  IF(TA .LT.0.1.AND.TB .LT.0.1)CALL PREC(A,B,ECC,ECR)
17  IF(TA .LT.0.1.AND.TB .GT.0.9)CALL PREC(A,B,DEC+ECC,ECR)
22  IF(TA .GT.0.9.AND.TB .LT.0.1)CALL PREC(A,B,DEC+DCC,ECR)
25  IF(TA .GT.0.9.AND.TB .GT.0.9.AND.
1   ABS(TA -TB) .GT.0.9)CALL PREC(A,B,ECC+2.*DEC,ECR)
30  END

C : IBF1C DIAGN
1   SUBROUTINE DIAGN(A)
2   INTEGER A,Q,QP
3   COMMON T(100,52),REF(5),DX,DY,BIBLI(20,44)
4   COMMON NB TYP,NTCTTR,NTCTDI,NTCTRE,NTCTCC
5   COMMON CARACT(5),Q,QP,DEE,DEC,DCC
6   DO I=1,5
7   IF(ABS(T(A,I)-REF(I)).LT..001)
1   WRITE(C6,82)(CARACT(I),T(A,I),I=2,5)
22  E1 CONTINUE
24  E2 FORMAT(10X,A6,F6.0,3X,4HTYPE,F6.0,10X,2HEN,F6.0,6X,F6.0)
25  WRITE(C6,E4)
26  E4 FORMAT(1HC)
27  Q=Q+1
30  QP=QP+1
31  RETURN
32  END

```



```

: IBFTC DOWN
1  SUBROUTINE DOWN(A,B,D,ECR)
2  INTEGER A,B
3  COMMON T(100,52),REF(5),DX,DY,BIBLI(20,44)
4  COMMON NB TYP,NTCTTR,NTCTDI,NTCTRE,NTCTCC
5  COMMON CARACT(5),Q,CP,DEE,DEC,DCC
6  IF(.NOT.(T(A,4).LE.T(B,4).AND.DX.LT.T(A,10)+D.OR.
1  T(A,4).GE.T(B,4).AND.DX.LT.T(B,10)+D).AND.
2  (T(A,5).LE.T(B,5).AND.DY.LT.T(A,11)+D.OR.
3  T(A,5).GE.T(B,5).AND.DY.LT.T(B,11)+D)))GOTO 61
1  IF(T(A,1).EQ.REF(3).AND.(T(A,6).EC.2..CR.T(A,6).EC.4.)
1.AND.T(A,10).LE.BIBLI(NBTYP,3)/3..AND.T(B,1).EC.REF(3).AND.
2(T(B,6).EC.2..CR.T(B,6).EC.4.)..AND.T(B,10).LE.BIBLI(NBTYP,3)/3.)
3GOTO 62
4  IF(T(A,52).GT.T(B,52).AND..ACT.
1 (T(B,1).EQ.REF(3).AND.(T(B,6).EC.1..CR.T(B,6).EC.3..0).AND.
2(T(B,10).GT.7.*T(B,11).AND.DX.LT.T(B,10)/2.))GOTO 62
7  IF(T(A,52).GT.T(B,52).AND..ACT.(T(B,10).GT.2.*BIBLI(
NBTYP,3).AND.DX.LT.2.*T(B,10)/3..AND.DY.GT.T(B,11)/2..AND.
2 T(A,5).GT.T(B,5)))GOTO 62
2  T(A,5)=T(B,5)+T(B,11)+D
3  GOTO 63
4  62 T(A,4)=T(B,4)+T(B,10)+D
5  63 CALL ECR(A)
6  61 CONTINUE
7  RETURN
8  END

```

```

: IBFTC RIGHT
1  SUBROUTINE RIGHT(A,B,D,ECR)
2  INTEGER A,B
3  COMMON T(100,52),REF(5),DX,DY,BIBLI(20,44)
4  COMMON NB TYP,NTCTTR,NTCTDI,NTCTRE,NTCTCC
5  COMMON CARACT(5),Q,CP,DEE,DEC,DCC
6  IF(.NOT.(T(A,4).LE.T(B,4).AND.DX.LT.T(A,10)+D.OR.
1  T(A,4).GE.T(B,4).AND.DX.LT.T(B,10)+D).AND.
2  (T(A,5).LE.T(B,5).AND.DY.LT.T(A,11)+D.OR.
3  T(A,5).GE.T(B,5).AND.DY.LT.T(B,11)+D)))GOTO 71
1  IF(T(B,1).EQ.REF(3).AND.(T(B,6).EC.1..CR.T(B,6).EC.3..0).AND.
1(T(B,10).GT.7.*T(B,11).AND.DX.LT.T(B,10)/2..OR.T(B,10).GT.2.*BIBLI(
2NB TYP,3).AND.DX.LT.2.*T(B,10)/3..AND.DY.GT.T(B,11)/2..AND.
3 T(A,5).GT.T(B,5).AND.T(A,52).EC.T(B,52)+1.)GOTO 72
4  T(A,4)=T(B,4)+T(B,10)+D
5  GOTO 73
6  72 T(A,5)=T(B,5)+T(B,11)+D
7  73 CALL ECR(A)
8  71 CONTINUE
9  RETURN
10 END

```



## 2.4. Résultats.

- . Le programme fait 770 instructions ISN.
- . Temps de compilation et d'assemblage :
  - sur 7044 : 2'27"
  - sur 360/40 : FORT : 128"      LKED : 54"
  - sur 360/65 : FORT : 22"      LKED : 19"
- . Pour l'exemple traité au §1 de ce chapitre nous obtenons, comme temps de travail :
  - sur IBM 7044 : 23" pour 60 déplacements.  
(Le dessin des résultats est ici, compris)
  - sur 360/65 : 15" pour 5 déplacements.

## 2.5. Conclusions.

Ce programme supplée en fait le concepteur pour mettre en place des éléments, vérifier leurs gardes respectives et dessiner le résultat obtenu. C'est d'ailleurs cette dernière partie qui prend le plus de temps machine, lors de l'exécution.



# TROISIEME PARTIE

## REALISATION DES INTERCONNEXIONS

- Introduction.

Chapitre I - Tracé automatique d'équipotentiellés.

Chapitre II - Traçage guidé des connexions par évitement d'obstacles.

## INTRODUCTION

Nous avons choisi, dans cette première version, de l'étude globale de la réalisation des circuits intégrés, de séparer l'implantation des éléments de la réalisation des interconnexions.

Nous supposerons donc avoir une disposition des éléments permettant le tracé des connexions. Si les gardes entre éléments et connexions ne sont pas respectées, il sera toujours possible, par passage alterné du programme de non chevauchement et du programme de tracé des connexions, de corriger ce défaut et d'obtenir une solution satisfaisante.

De même, nous avons séparé l'étude des équipotentiels et le tracé des connexions. En fait la première étude se fait sur une grille avec des cases occupées, la deuxième étude, elle, se fait en coordonnées réelles et, est directement utilisable pour la réalisation du circuit. De cette disparité, il reste au concepteur à tenir compte des différents résultats obtenus, afin d'aboutir à la meilleure solution possible.

## C H A P I T R E I

### TRACE AUTOMATIQUE D'EQUIPOTENTIELLES

#### 1 - POSITION DU PROBLEME

Nous cherchons à relier  $n$  points du plan de telle façon, que la somme totale des longueurs de connexion soit minimale.

De plus nous nous interdisons les croisements de connexions.

#### 2 - ALGORITHME DE TRACE D'EQUIPOTENTIELLES [7].

Nous travaillons sur une grille comprenant des cases permises et des cases interdites. Nous nous donnons les contacts  $C_1, \dots, C_p$ , entrant dans l'équipotentielle à tracer, dans un ordre quelconque (les contacts peuvent occuper plusieurs cases).

Le programme fait le produit (concaténation) d'une somme ( $\cup$  booléen) de  $n$  chemins tous permis,  $(C_i, C_{i+1})_k$ ,  $k=1$  à  $\dots, n$ , reliant des contacts successifs et ne passant sur aucune case interdite :

$$\prod_{i=1}^{p-1} \sum_{k=1}^n (C_i, C_{i+1})_k$$

Les chemins joignant deux points se trouvent dans le rectangle construit sur les coordonnées et sont exprimés en fonction de 2 directions perpendiculaires (haut, bas, droite, gauche).

Chaque chemin obtenu est testé pour savoir s'il ne conduit pas à l'utilisation d'une cas interdite.

Les produits comprennent au plus les 4 caractères H, B, D, G ; à chacun on attribue un coût qui est le nombre de caractères restant après avoir éliminé toute chaîne  $\mathcal{E}'$  de caractères telle que :

- il existe une chaîne  $\mathcal{E}\mathcal{E}'$  ou  $\mathcal{E}$  dépend des caractères opposés (H et B ou G et D) à ceux de  $\mathcal{E}'$  et dans l'ordre inverse (réflexion).

En effet, la longueur d'une telle chaîne  $\mathcal{E}\mathcal{E}'$  est égale à celle de  $\mathcal{E}$  puisque  $\mathcal{E}'$  ne fait que reparcourir le chemin, que l'on vient de tracer, en sens inverse.

Exemple.



nous avons réflexion de  $\mathcal{E}'$  sur  $\mathcal{E}$  d'où après élimination :

... D D D H H H ...

Nous prenons alors pour solution le produit qui a le plus faible coût.

2.1. Simplifications éventuelles.

a) Equipotentielle à 3 ou 2 éléments.

Si cette equipotentielle a 2 éléments cela est trivial. Si elle a 3 éléments, nous rajoutons un élément, dit point milieu, dont l'abscisse  $x_m$  (respectivement l'ordonnée  $y_m$ ) est définie à partir des abscisses  $x_1, x_2, x_3$  (respectivement des ordonnées  $y_1, y_2, y_3$ ) telle que :

$$\text{si nous avons } x_{i_1} \leq x_{i_2} \leq x_{i_3}$$

$$\text{alors } x_m = x_{i_2}$$

$$i_1, i_2 \neq i_3$$

$$\text{et } i_1, i_2, i_3 \in \{1, 2, 3\}$$

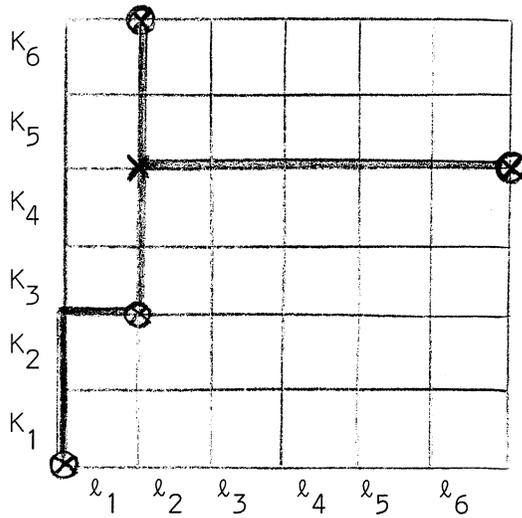
En effet, l'équipotentielle la plus courte passe par ce point milieu.

Remarque.

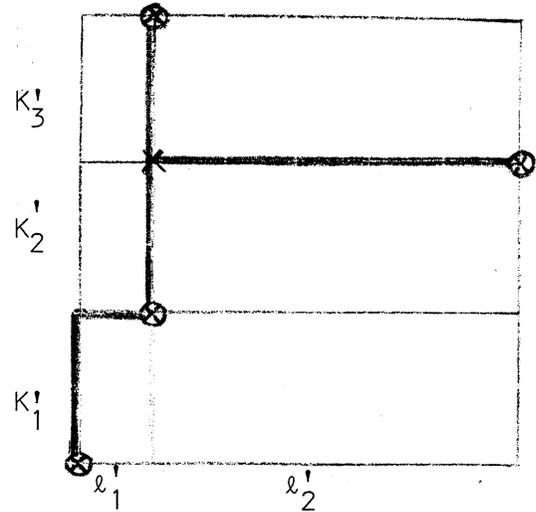
Il est possible que cette équipotentielle ne soit pas permise.

b) Grille réduite aux seules coordonnées effectives des éléments de l'équipotentielle.

Voyons sur un exemple, l'intérêt de cette simplification :



Grille normale



Grille réduite

$$\forall i, l_i = 1$$

$$\forall j, K_j = 1$$

$$l'_1 = 1 \quad l'_2 = 5$$

$$K'_1 = 2 \quad K'_2 = 2 \quad K'_3 = 2$$

2.2. Algorithme général.

Il tient compte du cas général et des deux cas particuliers exposés ci-dessus.

Considérons une équipotentielle à tracer :

A - Si celle-ci a 3 ou 2 éléments on essaie la méthode décrite en (2.1.a).

Si la méthode aboutit, c'est terminé, sinon on passe en B.

B - Soit à tracer un chemin  $(C_i, C_{i+1})$ .

B1 - On trace le chemin sur la grille réduite (cf. 2.1.6.)

Si on obtient un chemin, on passe en B pour le chemin suivant, sinon on passe en B2.

B2 - On trace le chemin sur la grille normale.

En cas de réussite on passe en B pour le chemin suivant, sinon on déclare ne pas pouvoir tracer cette équipotentielle et on passe à l'équipotentielle suivante.

### Remarque.

Il peut s'avérer essentiel de donner des points de "contact" supplémentaires, par exemple, pour éviter des connexions, ou pour sortir du rectangle formé par les contacts, afin de pouvoir tracer toutes les équipotentielles d'un circuit donné. (On peut alors tenir compte du résultat obtenu lors de l'étude de la planéarité).

### 3 - PROGRAMME - RESULTATS

Le programme est écrit en deux versions :

- Algol/7044 pour le compilateur de Grenoble
- Algol/360.

Nous ne présenterons ici que la version IBM/7044.

### 3.1. Sous-programmes standards.

Pour parer aux insuffisances du langage Algol, nous utilisons des procédures écrites en langage machine (7044). Ces procédures sont valables sur IBM 7044 ayant le compilateur de Grenoble.

#### a) Procédures d'écriture.

- procédure RCH ;  
commentaire retour à la ligne ;
- procédure LIGNE(A,B); valeur A,B; entier A,B;  
commentaire défini deux marges entre lesquelles tous les textes suivants seront écrits. Si la procédure n'est pas appelée on écrit sur toute la largeur de la feuille;
- procédure SORCHaine(I,J,X); valeur I,J; entier I,J,X;  
commentaire écrit un texte de I symboles dont le premier est dans la mémoire X en position J. Le texte s'imprime à la suite des textes précédents, s'il y a suffisamment de place. Aucun espace n'est prévu;
- procédure SORTEXTE(X); chaîne X;
- procédure SORENTIER(X); valeur X; entier X;

#### b) Procédures de traitement logique.

- entier procédure DECALAGE(A, ID, IG, UN);  
valeur A, ID, IG; entier A, ID, IG; booléen UN;  
commentaire intercale en position ID, ID-IG zéros en perdant les bits de droite;
- procédure LONGTG(DEA, DANB, POSIT);  
valeur POSIT; entier DEA, DANB, POSIT;  
commentaire longue translation gauche de DEA dans DANB, de POSIT positions, deux fois;

- procédure SELECTION(A,IB,J,MASK,C,D); valeur A,B,I,J,MASK;  
entier A,B,I,J,MASK,C,D;
- procédure SUIVANT(CARAC,IX,IY); valeur CARAC; entier CARAC,IX,IY;  
commentaire si CARAC vaut 0 alors IY=IY+1, si 1 alors IY-1, si 2  
alors IX=IX+1, si 3 alors IX=IX-1;
- booléen procédure INTER2EGO(IA,IB); valeur IA,IB; entier IA,IB;  
commentaire INTER2EGO est vrai si on a les 2 caractères IA et IB  
opposés;

### 3.2. Procédures de sortie.

- procédure SORTKUB(KUBDEF,P); valeur KUBDEF,P; entier KUBDEF,P;
- procédure DESSINER(TI); entier tableau TI;
- procédure SORTIGRAFIK(RESUL,DIM,PL,IX,IY,IX1,IY1,ONECRI); valeur  
DIM,IX,IY,IX1,IY1,PL; entier tableau RESUL; entier DIM,IX,  
IY,IX1,IY1,PL; booléen ONECRI;

### 3.3. Sous-programmes de traitement.

- entier procédure MIDDLE(X,I); valeur I; entier I; entier tableau X;
- procédure EQUIPDETROI(PLA,NB,CAVAPA); valeur PLA,NB; entier PLA,NB;  
booléen CAVAPA;
- procédure CHEMINS(IX,IY,IX1,IY1,MAX,ITABLO,INDICE,NM,COMT,PLA,NB);  
valeur IX,IY,IX1,IY1,MAX,INDICE,PLA,NB; entier IX,IY,IX1,  
IY1,MAX,INDICE,NM,COMT,PLA,NB; entier tableau ITABLO;  
commentaire cette procédure cherche les chemins entre (IX,IY)  
et (IX1,IY1) jusqu'à au plus MAX chemins. Le rangement se  
fait dans la ligne ITABLO [1,...] sous forme de 2, des 4  
caractères H,B,D,G. Le refus est le nombre de chemins re-  
fusés par des cases interdites;
- procédure CHANGRILLE(AUXI,SIMAUxi,ECARTX,PLA,NB); valeur PLA,NB;  
entier PLA,NB; entier tableau AUXI,SIMAUxi,ECARTX;

- procédure SIMULCHEM(IX,IY,IX1,IY1,JX,JY,JX1,JY1,MAX,ITABLO,INDICE,NM,COMT,ECARTX,ECARTY,PLA,NB); valeur IX,IY,IX1,IY1,MAX,INDICE,JX,JX1,JY,JY1,PLA,NB; entier IX,IY,IX1,IY1,MAX,INDICE,JX,JX1,JY,JY1,NM,COMT,PLA,NB; entier tableau ITABLO,ECARTX,ECARTY;  
commentaire cette procédure cherche les chemins entre (IX,IY) et (IX1,IY1) jusqu'à au plus MAX chemins. Le rangement se fait dans la ligne ITABLO [1,...] sous forme de 2 des 4 caractères H,B,D,G. Le refus est le nombre de chemins refusés par des cases interdites;
- procédure COMPARE(MANIP,MEMI,POSA,MEMJ,POSB,COUT,COMT); valeur COMT; entier MEMI,MEMJ,POSA,POSB,COUT,COMT); entier tableau MANIP;  
commentaire elle compare le caractère en position POSA de MEMI avec celui en position POSB de MEMJ. Si ils sont inverses le coût est diminué de 1. Les positions sont numérotées de droite à gauche et de 1 à 18;
- procédure PLUCOU(PLA,MAX,NB,GRIREDUITE); valeur PLA,MAX,NB,GRIREDUITE; entier PLA,MAX,NB; booléen GRIREDUITE ;  
commentaire elle recherche et trace l'équipotentielle numéro PLA avec au plus MAX chemins permis entre deux quelconques de ses éléments dont NB est le nombre ;  
Cette procédure PLUCOU se compose elle-même des 2 procédures suivantes :
- procédure COUT(COL1,COL2,COL3,NB,RESUL,COPTI); entier tableau COL1,COL2,COL3,RESUL; entier NB,COPTI;  
commentaire elle fait la concaténation de NB facteurs et évalue le coût en opérateurs de changement de cases dans la grille, par comptage des opérateurs restant après avoir compté pour 1 deux opérateurs consécutifs opposés, les avoir enlevés et recommencé si besoin était sur les suivants, devenus à leur tour consécutifs. Elle optimise dans RESUL, le coût le meilleur est COPTI;

- procédure BOUCLEPOUR(IND,IPAS,ISUP,NB,NBTOTAL); valeur NB,NBTOTAL;  
entier tableau IND,IPAS,ISUP; entier NB,NBTOTAL;  
commentaire elle génère des boucles pour en nombre variable NBTOTAL,  
sur le modèle pour IND [ ]=1 pas IPAS [ ] à ISUP [ ]  
faire, les unes dans les autres, à la plus petite on  
évalue le coût de la concaténation faite avec ces indi-  
ces variables;

### 3.4. Le programme général.

En Algol 60 (7044) le programme fait 8171 unités syntaxiques Algol.  
Le temps de compilation et d'assemblage est de : 3'37" sur IBM 7044.

### 3.5. Résultats - Conclusions.

Sur l'exemple dont nous avons parlé dans les précédents chapitres et parties, soit 41 éléments électroniques, nous avons tracé toutes les équipotentielles (grâce à l'aide par passage sur des points donnés), au nombre de 19, en 16".

Le nombre d'éléments était, par équipotentielle, de : 2, 2, 2, 2, 3, 4, 3, 3, 2, 2, 6, 5, 3, 5, 3, 5, 2, 11, 9;

Il convient de remarquer que, dans les 16" de travail, est compris le dessin des résultats.

Nous avons donc, avec ce programme, un outil très rapide, qui donne, avec un peu d'habitude des circuits électroniques, une solution très satisfaisante au problème du tracé complet des équipotentielles d'un réseau.

## C H A P I T R E II

### TRACAGE GUIDE DES CONNEXIONS PAR EVITEMENT D'OBSTACLES

#### 1 - POSITION DU PROBLEME

Nous utilisons comme données les coordonnées définitives des éléments c'est-à-dire, directement, les résultats obtenus au chapitre II de la 2<sup>ème</sup> partie.

Une bibliothèque des éléments permet d'avoir en mémoire la définition des éléments  $(\Delta X, \Delta Y)$  et des différents contacts relatifs à ces éléments

$(\underbrace{X_G, Y_G}_{\text{point supérieur gauche du contact}}, \Delta X, \Delta Y)$ .

point supérieur gauche du contact  
en coordonnées relatives.

Nous supposons pour le moment que le passage des connexions, entre deux éléments, ou entre deux contacts (résistance ou émetteur de collecteur), est prévu, c'est-à-dire que la place occupée par ces connexions, et les différentes gardes s'y rapportant, sont réservées.

Nous obtenons les chemins de passage des connexions entre les éléments électroniques, en tenant compte de l'étude de la planéarité (cf. 1<sup>ère</sup> partie).

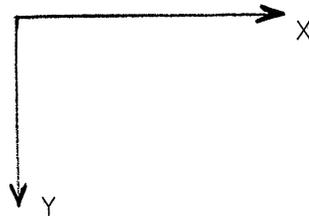
#### 2 - ALGORITHME DE TRACE

Nous définissons une connexion de la façon suivante :

- contact de départ (départ au centre)	{	. élément du contact . type du contact . départ vers, haut, bas, droite, gauche
---	---	---

- liste d'éléments ou de contacts (résistance) à éviter
  - contact arrivée
- . élément à éviter
  - . sens de progression de la connexion
  - . éviter par, nord, sud, est, ouest
  - . position de la connexion/élément
- idem départ avec en plus
  - arrivée quelconque ou au centre du contact.

De ces données nous fournissons une liste de points caractéristiques pour la connexion (points de changement de direction en déplacements perpendiculaires et points de départ et d'arrivée). Les coordonnées sont exprimées selon les axes rétrogrades du plan, soit :



### 3 - PROGRAMME - RESULTATS

Le programme existe en 2 versions :

- Algol 60/IBM 7044.
- Algol 360/IBM 360.

Nous donnons ici la version Algol 60.

#### 3.1. Les données.

- le nombre d'éléments électroniques composant le circuit.
- le nombre de connexions à traiter.
- le tableau de référence (bibliothèque des éléments).
- le tableau : nombre d'éléments à éviter par connexion.
- le tableau décrivant chaque connexion (cf. 2 de ce chapitre).



3.2. Le programme.

```

:ALGOL
'DEPLT' 'ENTIER' N, NN, T, J, R, S, CJ, NC ::
N:=EDCANKFE :: AN:=EDCANKFE ::
'DEPLT' 'ENTIER' 'TABLEAU' T.(1:N,1:52), P.(1:2,1:3),
T1.(1:4,1:6) ::
'POUR' I:=1 'PAS' 1 'A' 2 'FAIRE'
'POUR' J:=1 'PAS' 1 'A' 3 'FAIRE'
P.(I, J).:=EDCANKFE ::
'POUR' I:=1 'PAS' 1 'A' 4 'FAIRE'
'POUR' J:=1 'PAS' 1 'A' 6 'FAIRE'
T1.(I, J).:=EDCANKFE ::
'POUR' I:=1 'PAS' 1 'A' N 'FAIRE'
'DEPLT' ENTREE(05, REFER, 9, T.(I, 1).) ::
ENTREE(05, BIBLI, 42, T.(I, 10).) ::
'FIN' ::
REFER: MODELE(''(1XA2, I2, I3, 2I5, 2I3, 3X, 2I3)'' ) ::
BIBLI: MODELE(''(1X, I3(I3, 1X))'' ) ::
'DEPLT' 'ENTIER' 'TABLEAU' AT.(1:NN) ::
'POUR' I:=1 'PAS' 1 'A' NN 'FAIRE' AT.(I).:=EDCANKFE ::
R:=0 ::
'POUR' I:=1 'PAS' 1 'A' NN 'FAIRE'
'DEPLT' 'SI' AT.(I). 'SUPER' R 'ALORS' R:=AT.(I). ::
'FIN' ::
S:=0 ::
'POUR' I:=1 'PAS' 1 'A' NN 'FAIRE'
S:=S+AT.(I).+2 ::
'DEPLT' 'ENTIER' H, K, A, C, D, B, E, PJ, XE, YF, LXE, Q,
LYE, XC, YC, LXC, LYC ::
'ENTIER' 'TABLEAU' TT.(1:S, 1:6),
U.(1:R+2, 1:2) ::
'POUR' I:=1 'PAS' 1 'A' S 'FAIRE'
'POUR' J:=1 'PAS' 1 'A' 6 'FAIRE'
TT.(I, J).:=EDCANKFE ::
CJ:=1 ::
'POUR' J:=1 'PAS' 1 'A' NN 'FAIRE'
'DEPLT' NC:=AT.(J) ::
H:=TT.(CJ, 2). :: XF:=T.(H, 4). ::
YE:=T.(H, 5). :: LXE:=T.(H, 10). ::
LYF:=T.(H, 11). :: B:=TT.(CJ, 3). ::
'ET' B=C 'ALORS'
'DEPLT' XC:=XE :: YC:=YF ::
LXC:=LXE :: LYC:=LYE ::
'ALLERA' MOMO ::
'FIN' ::
LXC:=T.(H, B). :: LYC:=T.(H, B+1). ::
XC:=T.(H, B+20). ::
YC:=T.(H, B+21). ::
XC:=XE+XC :: YC:=YF+YC ::
D:=TT.(CJ, 4) ::
MOMO:
U.(1, 1).:=XC+T1.(D, 3).*LXC/'2+T1.(D, 2).*LXC ::
U.(1, 2).:=YC+T1.(D, 1).*LYC/'2+T1.(D, 5).*LYC ::
U.(2, 1).:=T1.(D, 3).*U.(1, 1).+T1.(D, 1).*XE+T1.(D, 2).*LYF ::

```



```

U.(2,2).:=T1.(C,1).*U.(1,2).+T1.(C,3).*YE+T1.(D,5).*LYE ::
'PCUR' I:=1 'PAS' I 'A' NO+1 'FAIRE'
'DEBUT' K:=TT.(CJ+I,2). ::
XE:=T.(K,4). :: YE:=T.(K,5). ::
LXE:=T.(K,10). :: LYE:=T.(K,11). ::
E:=TT.(CJ+I,5). ::
D:=TT.(CJ+I,4). ::
PJ:=TT.(CJ+I,6). ::
'SI' I 'NCNEG' I 'ALORS' 'ALLERA'
BEBE ::
'SI' K=H 'ALORS'
'DEBUT'
U.(2,1).:=U.(2,1).+T1.(E,4).*P.(1,PJ). ::
U.(2,2).:=U.(2,2).+T1.(E,6).*P.(1,PJ). ::
L.(2,1).:=T1.(C,3).*U.(2,1).+T1.(C,1).*XE+T1.(D,2).*LXE ::
U.(2,2).:=T1.(C,1).*U.(2,2).+T1.(C,3).*YE+T1.(D,5).*LYE ::
'ALLERA' ARRET ::
'FIN' ::
EERE:
'SI' I=NO+1 'ALORS'
'DEBUT' B:=TT.(CJ+I,3). ::
'SI' B=0 'ALORS'
'DEBUT' XC:=XE :: YC:=YE ::
LXC:=LXE :: LYC:=LYE ::
'ALLERA' STRIP ::
'FIN' ::
LXC:=T.(K,B). ::
LYC:=T.(K,B+1). ::
XC:=T.(K,B+20). ::
YC:=T.(K,B+21). ::
XC:=XE+XC :: YC:=YE+YC ::
A:=TT.(CJ+I,1). ::
'SI' A=0 'ALORS'
'DEBUT'
U.(NO+2,1).:=T1.(C,3).*U.(NO+2,1).+T1.(D,1).*XC+T1.(D,2).*LXC ::
U.(NO+2,2).:=T1.(C,1).*U.(NO+2,2).+T1.(D,3).*YC+T1.(D,5).*LYC ::
'ALLERA' STOP ::
'FIN' ::
U.(NO+2,1).:=XC+T1.(D,3).*LXC/'2+T1.(D,2).*LXC ::
U.(NO+2,2).:=YC+T1.(D,1).*LYC/'2+T1.(D,5).*LYC ::
U.(NO+2,1).:=T1.(C,1).*U.(NO+2,1).+T1.(D,3).*U.(NO+3,1). ::
L.(NO+2,2).:=T1.(C,3).*U.(NO+2,2).+T1.(C,1).*U.(NO+3,2). ::
'ALLERA' STOP ::
'FIN' ::
Q:=P.(2,PJ). ::
'SI' I=NO 'ALORS'
'DEBUT' H:=TT.(CJ+I+1,2). ::
'SI' K=H 'ALORS'
Q:=P.(1,PJ). ::
'FIN' ::
A:=TT.(CJ+I,1). ::
'SI' A=1 'ALORS'
'DEBUT' B:=TT.(CJ+I,3). ::

```



```

LXE:=LXC:=T.(K,B). ::
LYE:=LYC:=T.(K,B+1). ::
XC:=T.(K,B+20). ::
YC:=T.(K,B+21). ::
XE:=XE+XC ::
YE:=YE+YC ::
'FIN' ::
I.(I+1,1). :=T1.(E,3).*U.(I+1,1).+T1.(E,1).*XE+T1.(E,2).*LXE+T1.(E,4).*Q
::
I.(I+1,2). :=T1.(E,1).*U.(I+1,2).+T1.(E,3).*YE+T1.(E,5).*LYE+T1.(E,6).*C
::
I.(I+2,1). :=T1.(D,3).*U.(I+1,1).+T1.(D,1).*XE+T1.(D,2).*LXE ::
I.(I+2,2). :=T1.(D,1).*U.(I+1,2).+T1.(D,3).*YE+T1.(D,5).*LYE ::
ARRET:
'FIN' ::
STOP:
CJ:=CJ+NO+2 ::
SORTIE(C6,RESUL,1,J) ::
RESUL:
MODELE(''(10X,SHCFNFXION,I3)'' ) ::
'POUR' I:=1'PAS'1'A'NO+3'FAIRE'
SORTIE(C6,TABLE,2,U.(I,1).) ::
TABLE:
MODELE(''(20X,2I5)'' ) ::
'FIN' ::
'FIN' ::
'FIN' ::
'FIN' ::
'ALCCL

```

Le temps de compilation et d'assemblage est de :

- 58" sur IBM 7044.
- Algol 31" sur IBM 360/65.

### 3.3. Résultats.

Les sorties ont lieu sous forme de tableaux, un par connexion.

- Pour étudier 11 connexions sur IBM 7044 le temps de travail = 4".

Le nombre d'éléments à éviter est, par connexion de :

1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 3;

- Pour étudier 12 connexions sur IBM 360/65 le temps de travail = 29".

Remarque.

Le programme passant en FAST, le temps ci-dessus comprend en fait, le temps de Link-Edit (LKED) et le temps d'exécution. Le nombre d'éléments à éviter est, par connexion, identique à ci-dessus, pour les 11 premières et de 9 pour la 12<sup>ème</sup> connexion.



## Q U A T R I E M E P A R T I E

### LE PROGRAMME DESMAG

- Introduction.

Chapitre I - Description des divers modules.

Chapitre II - Quelques réalisations.



## INTRODUCTION

Le programme DESMAG

DESSin MASques Grenoble, permet le dessin automatique des masques, en vue de la réalisation d'un circuit intégré, à partir d'un schéma électronique définissant le circuit.

Nous partons donc de la liste des éléments électroniques du circuit et de la liste des divers noeuds équipotentiels.

Le programme est composé de plusieurs modules articulés les uns aux autres (voir Fig. 1), par des interventions humaines. Il est possible, en outre, de modifier les résultats d'un module :

- soit en imposant des conditions supplémentaires et en reprenant ce module

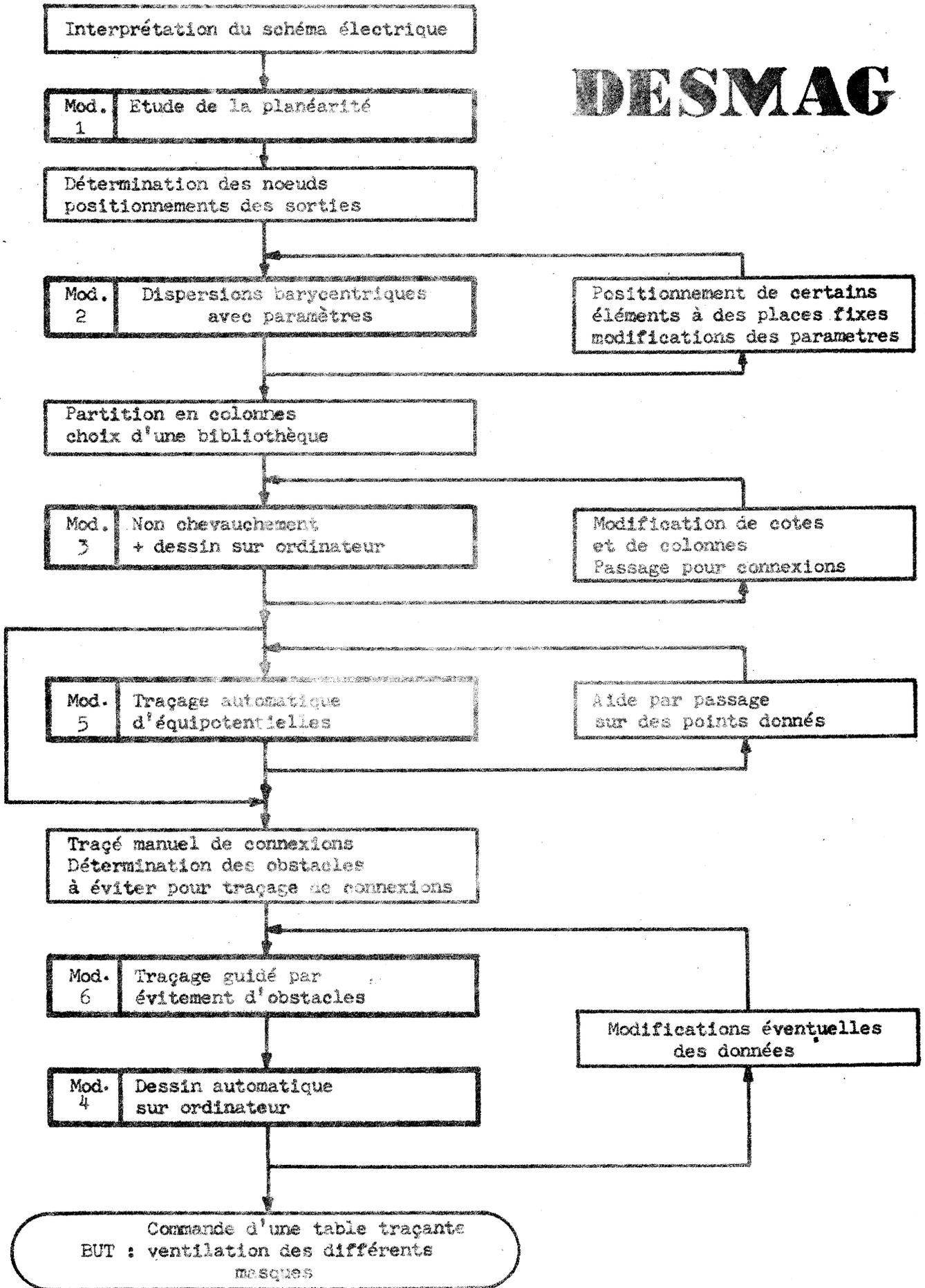
- soit en enchaînant le module suivant, dans des conditions jugées plus raisonnables.

Il est à noter que les résultats obtenus par certains modules sont utilisables, non seulement au module suivant, mais aussi par d'autres modules.

Ce programme a été étudié dans l'optique du C.A.O. conception assistée par ordinateur. (C.A.D, en anglais).



# DESMAG





## C H A P I T R E I

### DESCRIPTION DES DIVERS MODULES

#### 1 - PLANEARITE

Nous disposons des programmes suivants :

##### 1.1. Recherche du nombre minimum de connexions à enlever d'un réseau pour le rendre planaire.

Nous obtiendrons alors le nombre minimum de croisements lorsque nous replacerons les dites connexions.

Cette étude est faite dans la 1<sup>ère</sup> partie de cet ouvrage.

##### 1.2. Reconnaissance et dessin de graphes planaires.

2 programmes sont utilisables :

- programme tiré de [10] et de la thèse de Monsieur PERTUISET.

- programme écrit par FAVRE-MORIN (Grenoble) méthode exposée en [10].

#### 2 - DISPERSION BARYCENTRIQUE

Méthode exposée au chapitre 1, 2<sup>ième</sup> partie de cet ouvrage.

Nous avons pu constater, sur les exemples traités (cf. ch II, 4<sup>ième</sup> partie) que les résultats obtenus par ce module sont, en fait, concordants avec ceux obtenus par le module 1, programme 1.2.

### 3 - NON CHEVAUCHEMENT

Méthode exposée au chapitre II, 2<sup>ième</sup> partie de cet ouvrage.

Il est à remarquer que l'on peut initialiser la disposition des éléments, soit en se servant des résultats du module précédent, dispersion barycentrique, soit en utilisant le dessin obtenu au module 1 (programme 1.2.).

### 4 - DESSIN SUR ORDINATEUR

Nous obtenons par ce module un dessin des résultats précédents éventuellement modifiés, facilitant la lecture fastidieuse de coordonnées. Ce dessin est très rapidement obtenu, mais ne peut être consulté qu'à titre indicatif. Il ne peut en aucun cas remplacer un dessin sur table traçante.

Après le passage de ce programme, les contours des divers éléments apparaissent sur la feuille de papier ; il est également possible de faire apparaître les contacts et les connexions résultant du module 6, sur le dessin. Ce dernier est moins précis que sur table traçante puisque tous les points se trouvent placés sur une grille dont le pas est  $10\mu$  (il est nécessaire pour cela d'arrondir les diverses grandeurs). Ce pas est tout à fait raisonnable car la moindre connexion a  $10\mu$  d'épaisseur au minimum.

Le module est écrit en 3 versions.

- Algol 60/IBM 7044 - 1644 Unités Syntaxiques
- Algol/IBM 360
- Fortran IV.

Temps de compilation et d'assemblage :

- Algol/7044 : 1'06"
- Fortran IV/360.40 : FORT : 76"      LKED : 46"

## 5 - TRACE AUTOMATIQUE D'EQUIPOTENTIELLES

Méthode exposée au chapitre I de la 3<sup>ième</sup> partie de cet ouvrage.

La disposition des éléments nous est donnée par le résultat du module 3, non chevauchement.

## 6 - TRACAGE GUIDE DES CONNEXIONS PAR EVITEMENT D'OBSTACLES

Méthode exposée au chapitre II de la 3<sup>ième</sup> partie de cet ouvrage.

La disposition des éléments est connue à la fin du module 3.

Le guidage des connexions est obtenu en faisant apparaître sur le dessin terminal du module 3, le résultat donné par le module I (progr. 1.2.).

## 7 - GESTION DE LA LISTE DES NOEUDS

Ce module permet d'assurer une gestion de la liste des noeuds. Lors d'un oubli, d'une erreur ou d'une modification de schéma électrique il est long et fastidieux de supprimer des éléments ou d'en rajouter. En effet, la suppression du transistor 1 nous oblige à renuméroter tous les transistors en abaissant leur numéro de 1. Tous les éléments sont rangés dans un seul tableau, leur rang étant calculé en fonction de leur nom et de leur numéro.

Il est intéressant, lorsque le circuit présente une symétrie de certains de ses éléments, de les numéroter ainsi qu'exposé ci-dessous :

La numérotation est faite par type d'éléments, c'est-à-dire transistors, diodes, résistances, capacités, sorties.

### Exemples.

$TR_1, TR_2 \dots TR_n, DI_1, \dots, DI_p, RE_1, \dots, RE_q, \text{ etc } \dots$

Pour chaque type d'élément, lorsque certains éléments ont un symétrique, nous numéroterons de 1 à  $l$  les éléments ayant un symétrique, de  $l+1$  à  $m$  les éléments n'ayant pas de symétrique et de  $m+1$  à  $n$  les éléments symétriques des premiers de telle façon que le numéro d'un élément, symétrique d'un élément déjà numéroté, soit égal à la somme :

N° de l'élément déjà numéroté + nombre d'éléments n'ayant pas de symétrique  
+ nombre d'éléments ayant un symétrique et déjà numérotés dans la 1<sup>ère</sup> partie (1 à  $l$ ).

Le programme permet de rajouter aussi, autant de noeuds que l'on veut.

Ce module est écrit en FORTRAN IV.

Temps de compilation et d'assemblage sur IBM 360/65 :

FORT : 7"      LKED : 3".

### 8 - CONFORMITE AU SCHEMA DE PRINCIPE

Ayant en mémoire d'une part, les coordonnées de tous les "centres" des contacts et les coordonnées des points caractéristiques de toutes les connexions, d'autre part, la liste des noeuds équipotentiels du schéma électronique de départ, le programme :

- reconstitue la composition des équipotentiels, telles qu'elles apparaissent dans l'implantation physique. (résultat du module 6 dessiné à l'aide du module 4).

- compare ces équipotentiels aux noeuds du schéma électronique de départ et signale éventuellement les erreurs à rectifier.

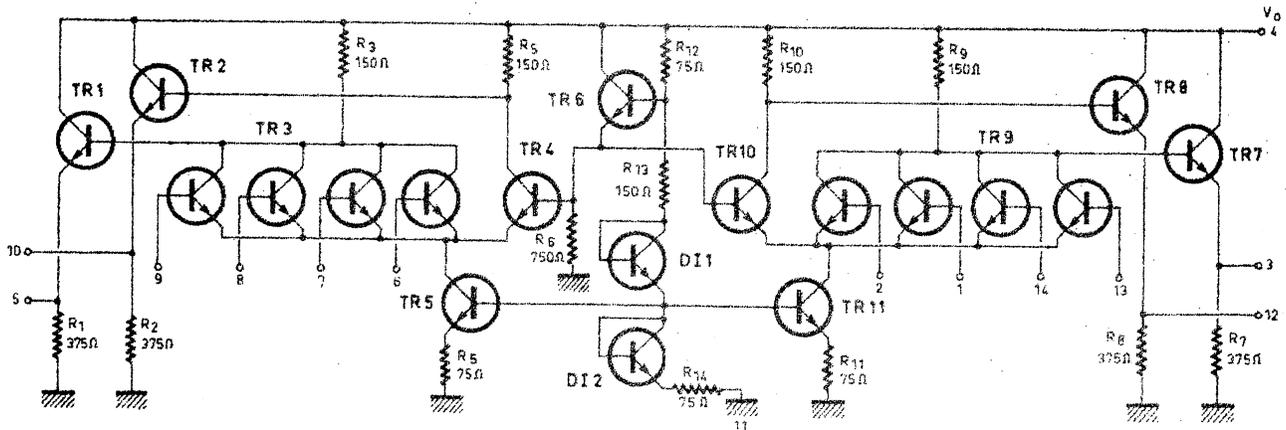
Ce module est écrit en Algol 60/IBM 7044. Temps de compilation : 1'16".

# CHAPITRE II

## QUELQUES REALISATIONS

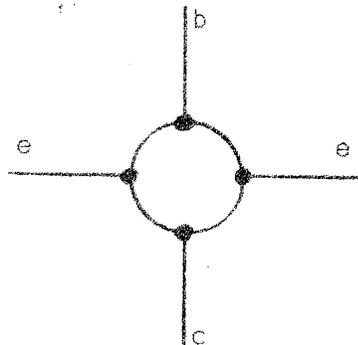
### 1 - DOUBLE PORTE, TYPE CENG

Le point de départ est le schéma électronique donné par le circuiteur :



#### 1.1. Etude de la planéarité.

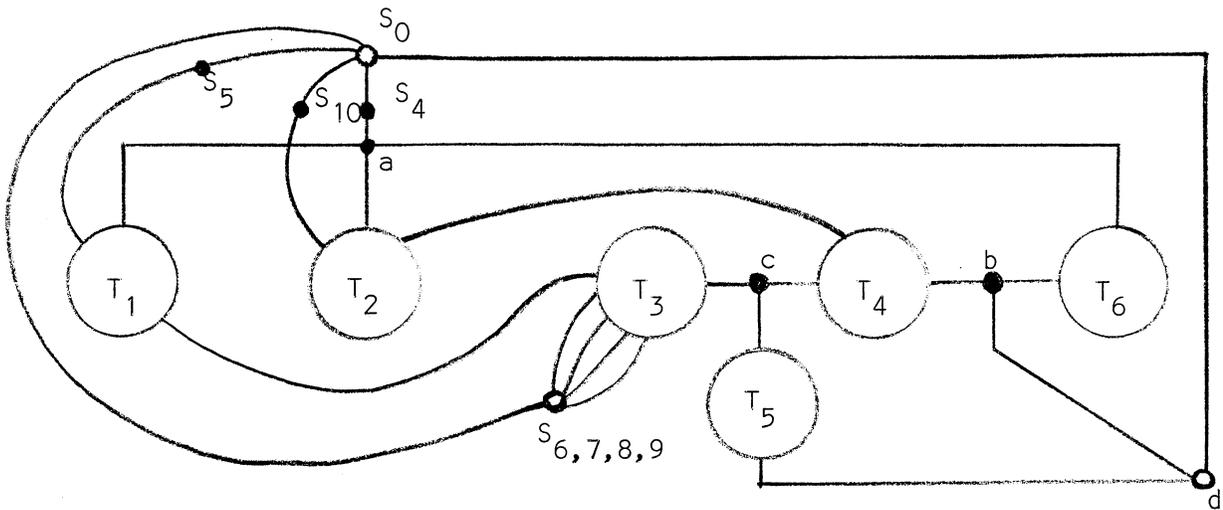
Ce circuit est réalisé avec le transistor du premier type, c'est-à-dire :



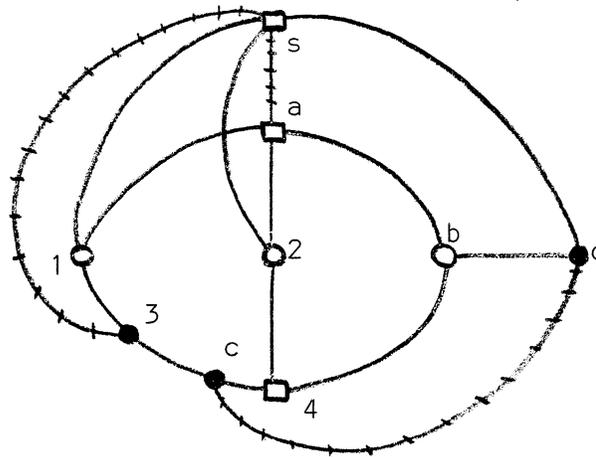




Le circuit présentant une symétrie par rapport à un axe vertical, nous avons opéré une 2-fracture sur deux connexions de ce circuit (d), et nous obtenons le réseau de noeuds et d'étoiles suivant :



Nous en déduisons le graphe suivant :

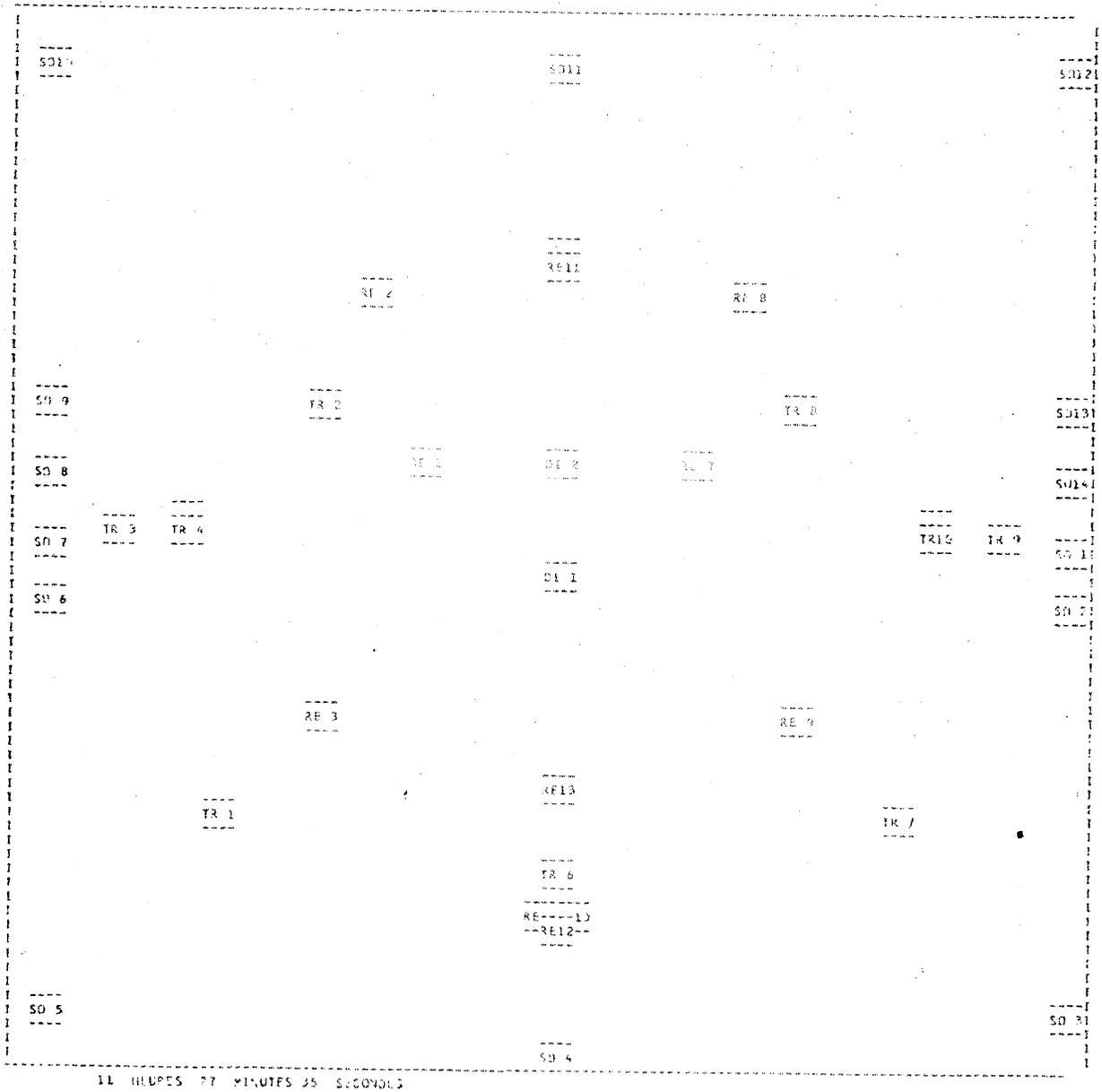


Les solutions planaires en enlevant une arête sont :

arête :  $S, 2 - a, 1 - a, b(T_6) - b, d - S, d - 4, 2 - 1, 3 - 3, C - 4, C -$

1.2. Dispersion barycentrique des éléments.

En plus des 14 sorties, nous avons fixé la position des transistors TR3 et TR9; l'ordre de rencontre des sorties sur le périmètre de la plaquette nous était donné par le résultat obtenu précédemment en 1.1.





Il est à remarquer que la symétrie observée sur le schéma électro -  
nique de départ, se conserve dans la dispersion barycentrique. Les sorties ont  
été, bien entendu, disposées au préalable, de façon symétrique.

### 1.3. Non chevauchement.

Sur le dessin des résultats, on peut remarquer que les échelles sont  
différentes sur les horizontales et sur les verticales. Ceci est dû au fait que  
le dessin est exécuté par une machine à écrire. Pour pallier à cet inconvénient,  
nous avons réécrit le module 4, et obtenons maintenant un dessin respectant, à  
une certaine échelle, les dimensions des éléments (cf. module 6 en 1.6.).

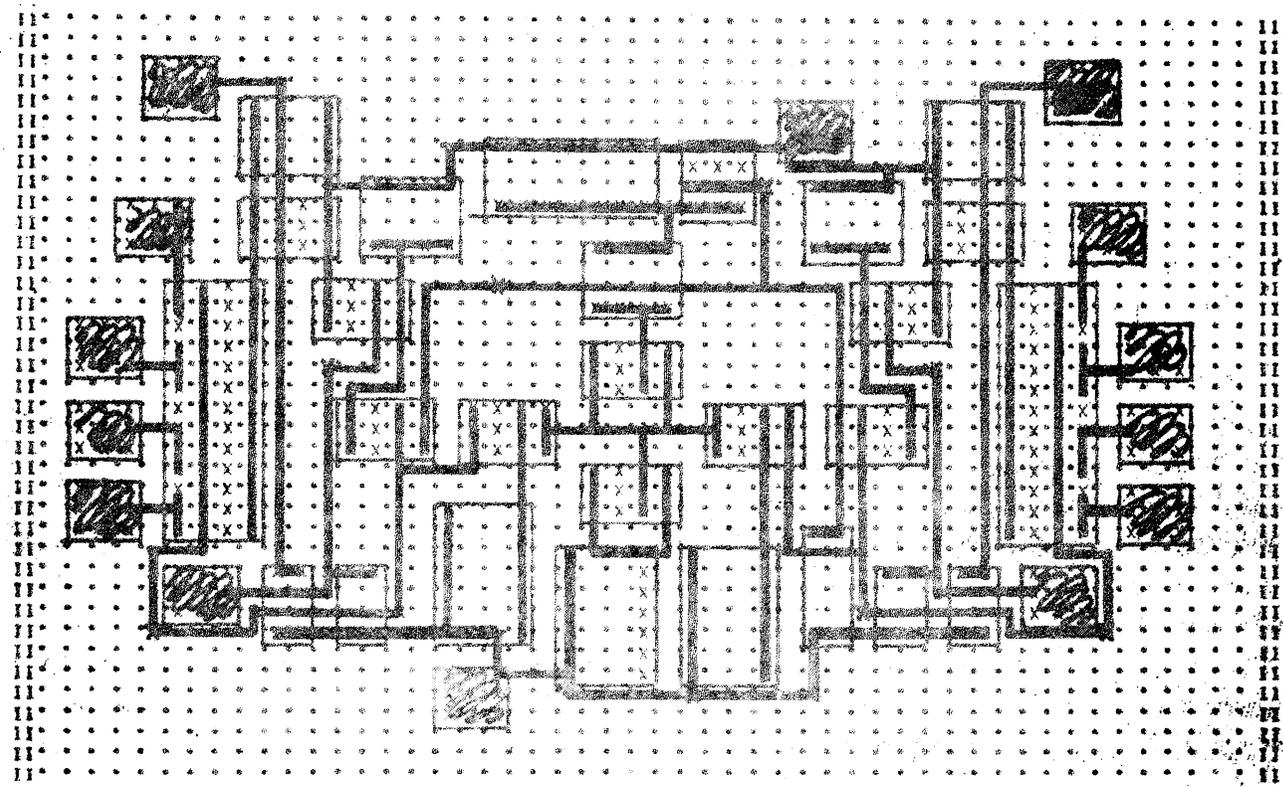






### 1.4. Tracé automatique d'équipotentielles.

Nous avons tracé toutes les équipotentielles en utilisant, pour certains, le passage sur des points donnés.



### 1.5. Tracage guidé des connexions.

Seulement 12 connexions ont été tracées par cette méthode (afin de tester le programme, mais on peut toutes les tracer).

Seules n'apparaissent pas sur ce dessin, qui est la fin de l'étude, les différentes zones de diffusion, se rapportant à chaque élément, et les zones délimitant les caissons d'isolement.

Afin d'éviter le croisement obligatoire (cf. résultat du module 1) nous avons choisi d'entourer la sortie 10 (et sa symétrique, la sortie 12).





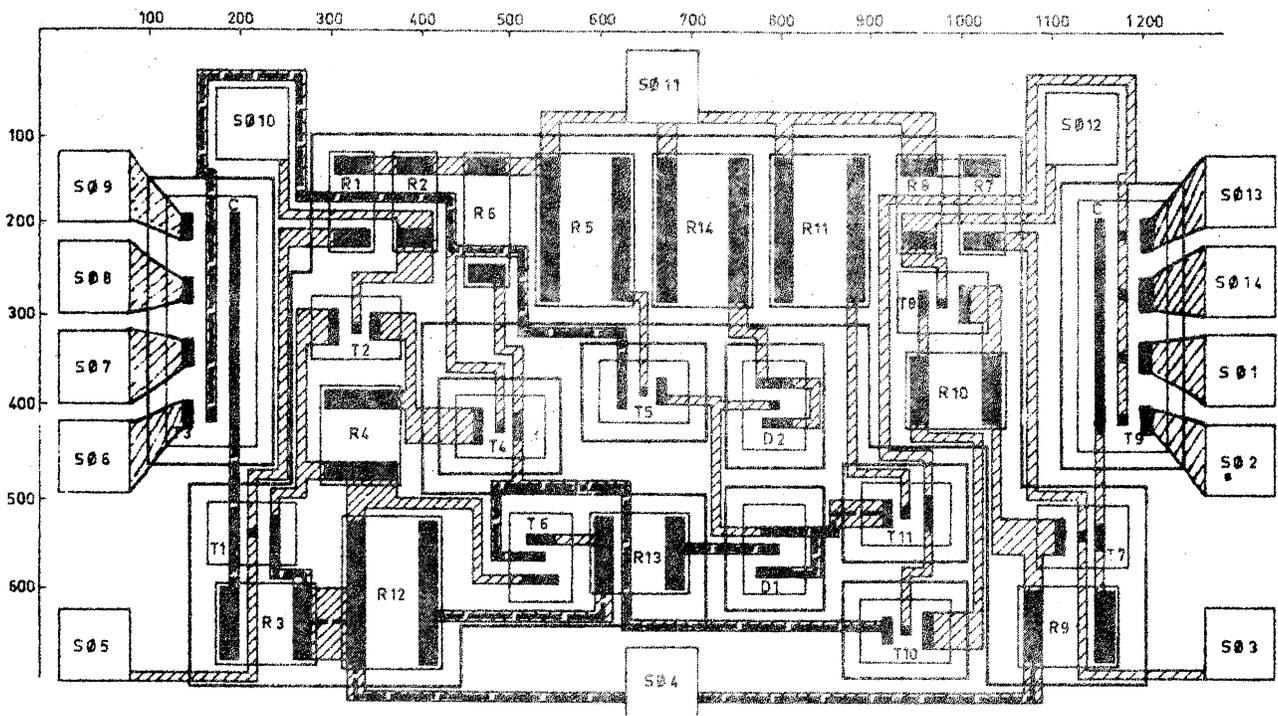


### 1.6. Circuit intégré avant ventilation des masques.

Ceci est le point d'arrivée de DESMAG.

Tous les résultats sont conservés sur cartes perforées, et, par un simple travail de repérage, il sera facile au concepteur, d'en déduire la bande magnétique qui servira à commander la table traçante.

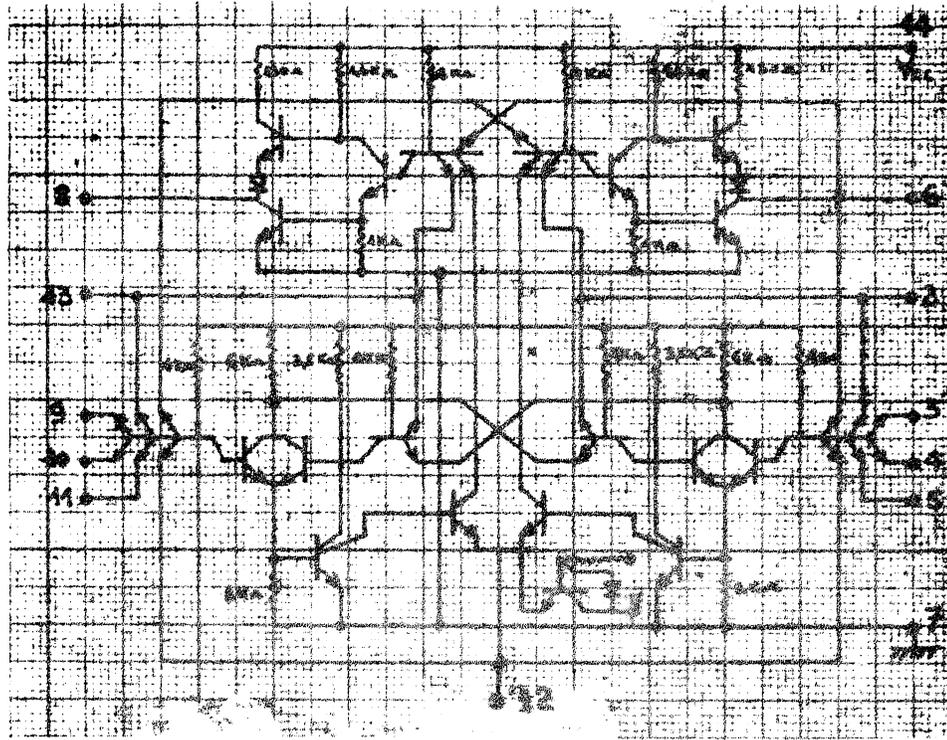
Les différents masques de diffusion seront déduits facilement du tableau résultat final, qui sera conservé sur cartes perforées ou sur un autre support.





## 2 - BASCULE J-K. MAITRE-ESCLAVE, TTL NORMAL, TYPE COSEM

Schéma électronique de départ :

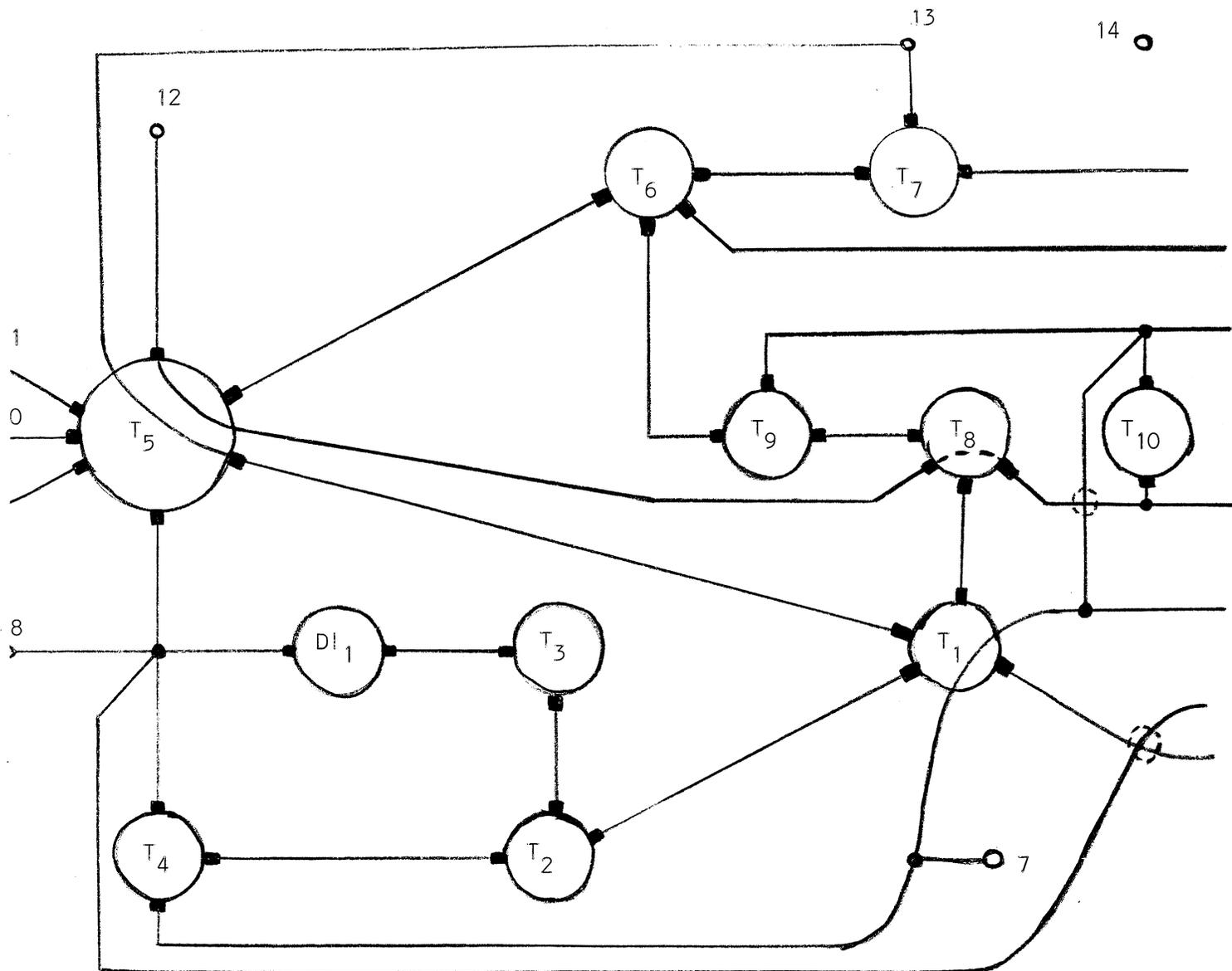


### 2.1. Etude de la planéarité.

Le circuit présentant une symétrie par rapport à un axe vertical, nous n'avons traité que la partie gauche du réseau.

Nous obtenons la solution suivante (les résistances ne sont pas mentionnées) :





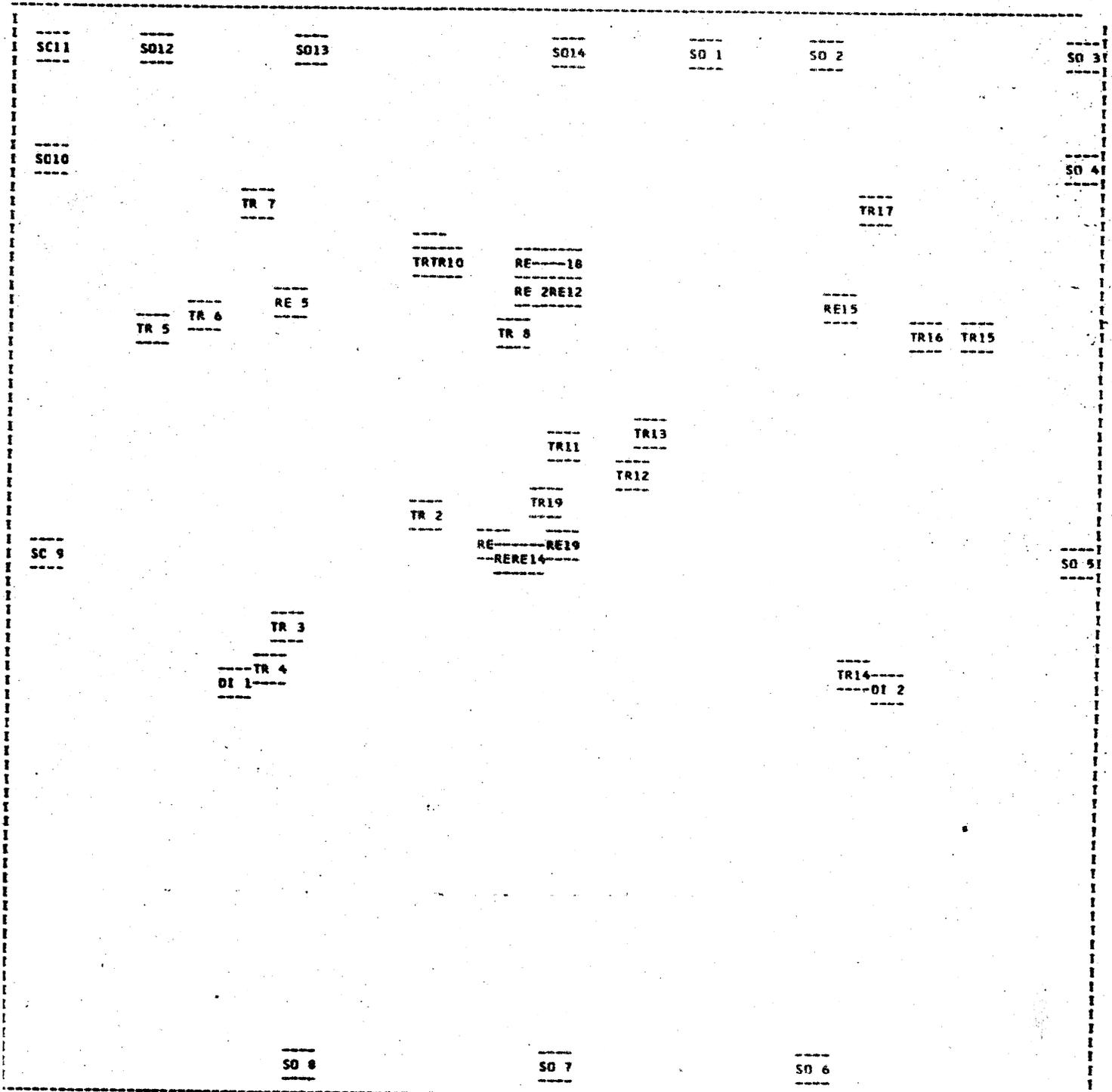
Remarques.

Les passages entre émetteurs, de transistors multi-émetteurs, sont ici autorisés. Nous obtenons 2 croisements obligatoires que l'on reporte sur la partie non symétrique du réseau.



## 2.2. Dispersion barycentrique des éléments.

L'ordre de rencontre des sorties, sur le périmètre de la plaquette nous était imposé.





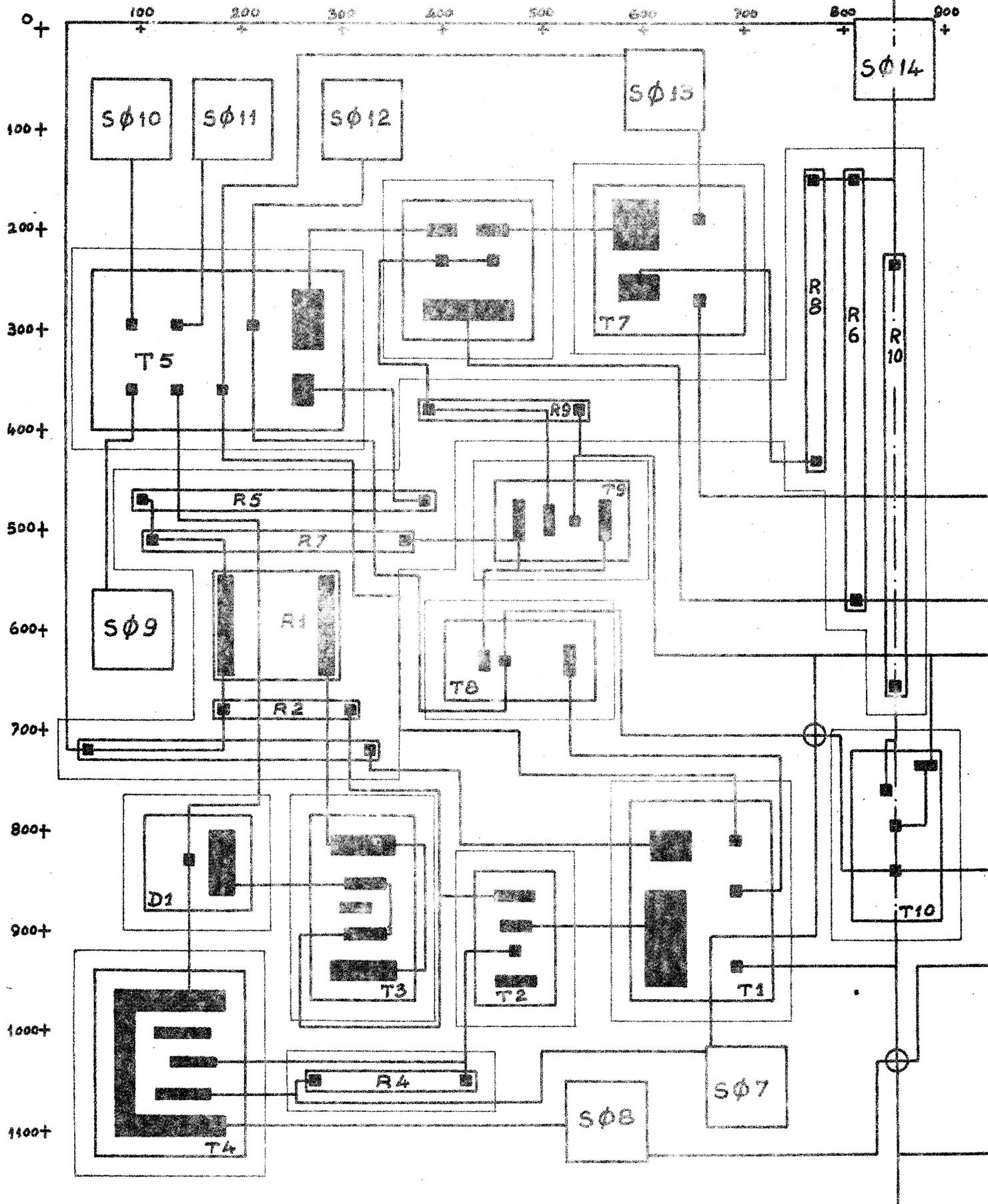




Nous n'avons traité que la partie gauche du circuit et les éléments apparaissant sur l'axe de symétrie (S0 14, S0 7, R10, T10).

#### 2.4. Traçage guidé des connexions - Résultat final.







C O N C L U S I O N



Nous avons essayé, dans ce qui précède, de montrer que la conception des circuits intégrés peut être, de nos jours, facilitée et surtout accélérée grâce à l'aide fournie par les ordinateurs.

DESMAG ne doit être qu'une première version d'un programme résolvant ce problème, bien que des améliorations soient actuellement en cours d'étude : non chevauchement des éléments avec les connexions, détermination des équipotentielles et tracé guidé des connexions faits en simultanéité, etc ... .

Dans une version ultérieure l'on pourrait, traiter ensemble les connexions et la disposition des éléments, tasser un circuit très "aéré" au départ au lieu de desserrer seulement des éléments. Mais dans cette nouvelle étude, un apport considérable serait fait au concepteur, par la possibilité d'utiliser une console de visualisation, car c'est le grand enseignement que l'on peut tirer du travail effectué pour la mise au point de DESMAG :

Le problème de la conception des circuits électroniques doit être résolu, avec l'aide de plusieurs programmes intervenant en des points précis de l'étude, en mode conversationnel et avec visualisation à tout instant du résultat partiel obtenu.



B I B L I O G R A P H I E

- [ 1 ] AUSLENDER, L., PARTER, S.V., On embedding graphs in the sphere,  
J. Math. Mec., vol. 10, 517-523, Mai 1961.
- [ 2 ] BENZAKEN, C., Synthèse des réseaux de contact,  
Séminaire de Logique, Grenoble 1965 (non publié).
- [ 3 ] BERGE, C., Théorie des graphes et ses applications,  
DUNOD - 1958.
- [ 4 ] BUI-NGOG, M., SARRET, M.J., Problèmes d'implantation - Etude de la planéa-  
rité pour les réseaux électroniques,  
Contrat DRME - 66.34.215/6544.
- [ 5 ] CARREFOUR, M.V., Etude de la définition des schémas de câblages d'ensem-  
bles électroniques complexes à l'aide d'un ordinateur élec-  
tronique,  
C.N.E.T. de LANNION, 9 Novembre 1964.
- [ 6 ] CASE, P.W., GRAFF, H.H., GRIFFITHS, L.E., LECLERCQ, A.R., MURLEY, W.B.,  
SPENCE, T.M., Solid logic design automation,  
IBM-Journal, pp. 127-140, April 1964.
- [ 7 ] CHEIN, M., Un algorithme pour relier n points,  
Calcolo, 4, vol. 5, pp. 537-547, Décembre 1968.
- [ 8 ] CONGRES, Microelectronics and large systems.  
Edited by MATHIS, J.J., WILLEY, T.E., SPANDORFER, L.M.,  
Spartan books - Mac Millan and C°, 1965.
- [ 9 ] DAVIS, E.M., HARDING, W.E., SCHWARTZ, R.S., CORNING, J.J.,  
Solid logic technology : versatile, high - performance  
microelectronics,  
IBM-Journal, pp. 102-114, April 1964.

- [10] DEMOUCRON, G., MALGRANGE, Y., PERTUISET, R., Graphes planaires : reconnaissance et construction de représentations planaires topologiques,  
S.F.R.O., 8<sup>ième</sup> année N° 30, pp. 33-47, 1<sup>er</sup> tr. 1964.
- [11] FISHER, G.J., WING, O., A correspondence between a class of planar graphs and bipartite graphs,  
I.E.E.E. Trans on C.T., vol. 12, N° 2, pp. 266-269, June 1965
- [12] FISHER, G.J., WING, O., Computer recognition and extraction of planar graphs from the incidence matrix,  
I.E.E.E. Transactions on C.T., vol. 13, N° 2 pp. 154-163  
June 1966.
- [13] HARARY, F., A complementary problem on non planar graphs,  
Math. Magazine, pp. 301-303, Nov. Déc. 1962.
- [14] KUNTZMANN, J., Théorie des relations et des réseaux,  
Cours Grenoble 1967.
- [15] LEDLEY, R.S., Digital Computer and Control engineering,  
Mc Graw Hill, 1960.
- [16] LEE, D.Y., On algorithm for path connection and its applications,  
I.R.E. Trans. on E.C., pp. 346-353, Sept. 1961.
- [17] LEMPEL, A., EVEN, S., CEDERBAUM, I., An algorithm for planarity testing of graphs,  
International symposium, Théory of graphs,  
Rome, pp. 215-232, July 1966.
- [18] LERDA, F., MAJORANI, E., An algorithm for connecting N points with a minimum number of crossings,  
Calcolo i, pp. 257-265, 1964.

- [19] LOBERMAN, H., WEINBERGER, A., Formal procedures for connecting terminals with a minimum total wire length, Journal ACM, 4, pp. 428-433, 1957.
  
- [20] MAC LANE, S., A combinatorial condition for planar graphs, Fund. Math. t. 28, pp. 22-32, 1937.
  
- [21] MAJORANI, E., Simplification of Lee's algorithm for special problems, Calcolo 1, pp. 247-256, 1964.
  
- [22] PONENTALE, T., An algorithm for minimizing backboard wiring functions, Communications of the ACM, vol. 8, number 11, pp. 699-703, November 1965.
  
- [23] RICE, R., The packaging revolution, Electronics, vol. 39, N° 3, pp. 109-116, Fév. 1966.
  
- [24] SAILLARD, J.C., SARRET, M.J., Dessin automatique des masques : le programme DESMAG, Onde électrique, vol. 49, fasc. 1, pp. 92-97, Janvier 1969.
  
- [25] WING, O., On drawing a planar graph, I.E.E.E. Trans. on C.T., vol. 13, N° 1, pp. 112-114, March 1966.



VU

Grenoble, le

*Le Président de la Thèse*

VU

Grenoble, le

*Le Doyen de la Faculté des Sciences*

Vu, et permis d'imprimer,

*Le Recteur de l'Académie de GRENOBLE*

