

50376

1973

N° d'ordre : 143

143

50376

1973

143

THÈSE

présentée à l'

UNIVERSITÉ DES SCIENCES ET TECHNIQUES DE LILLE 1

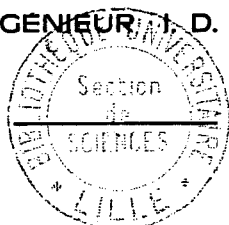
pour l'obtention du titre de

DOCTEUR INGÉNIEUR

par

Jean-Claude TARTE

INGÉNIEUR I. D. N.



**ÉTUDE ET RÉALISATION D'UN COUPLAGE
ENTRE UNE CALCULATRICE HYBRIDE
ET UN CALCULATEUR DE PROCESSUS**

soutenue le 6 Juillet 1973 devant la Commission d'examen

Messieurs : P. VIDAL, Président

J. M. TOULOTTE, Rapporteur

L. POVY, Examineur

CORDONNIER, Examineur

M. MORIAMEZ, Invité

C. SOURISSE, Invité

A ma Femme,

A mes Parents,

A mes Beaux - Parents,

AVANT PROPOS

Le travail présenté dans ce mémoire a été effectué au Laboratoire d'Automatique de l'Université des Sciences et des Techniques de Lille I.

Nous remercions Monsieur le Professeur VIDAL de son accueil au sein du Laboratoire. Il nous a permis d'entreprendre des recherches très intéressantes. Qu'il trouve ici l'expression de notre gratitude pour l'honneur qu'il nous fait d'accepter la présidence de notre Jury.

Nous sommes également très honorés par la présence de Monsieur le Professeur MORIAMEZ, et nous lui exprimons notre plus profonde gratitude pour le soutien qu'il nous a toujours apporté.

Que Monsieur TOULOTTE, Maître de conférences à l'Université des Sciences et Techniques de Lille I, trouve ici l'expression de notre reconnaissance, pour les encouragements, les conseils qu'il nous a prodigués tout au long de notre recherche.

Nous tenons à remercier Monsieur POVY, Maître de conférences à l'Université des Sciences et Techniques de Lille I, pour l'aide matérielle qu'il nous a apportée, pour l'intérêt qu'il nous a témoigné dans notre travail et pour les facilités qu'il nous a accordées lors de la réalisation de notre étude.

Nous adressons à Monsieur CORDONNIER, Maître de conférences à l'Université des Sciences et Techniques de Lille I, nos remerciements pour sa présence dans notre jury.

Monsieur SOURISSE, Responsable du Département de calcul hybride à la Télémécanique Electrique, nous a honorés, en acceptant de participer à notre jury. Qu'il trouve ici l'expression de notre reconnaissance.

Nous remercions sincèrement les professeurs et tous les chercheurs du Laboratoire pour l'aide et l'amitié qu'ils nous ont apportées.

Enfin, nous ne saurions terminer cet avant-propos sans exprimer notre reconnaissance à tout le personnel du Laboratoire pour la diligence avec laquelle il a participé à la réalisation de notre étude et à la confection du mémoire.

AVERTISSEMENT

Le travail présenté dans ce mémoire est le fruit des efforts d'une équipe de recherche du Centre d'Automatique de l'Université des Sciences et Techniques de Lille I.

Dans une première partie, nous avons cherché à donner une définition du calcul hybride de type II. Nous citons les avantages et les inconvénients du calcul analogique et du calcul numérique, puis définissons les contraintes d'un couplage entre un calculateur de processus et une calculatrice hybride à logique parallèle.

La deuxième partie est consacrée à la présentation des deux machines à coupler, de leurs caractéristiques et de leurs performances. Nous étudions ensuite les divers signaux de liaisons à développer pour assurer la commande des calculatrices hybrides par le calculateur numérique. Parallèlement à cette réalisation, s'est développé un organe de test, destiné à détecter les composants défectueux de l'interface.

La troisième partie de ce travail, consiste en un résumé du langage de programmation propre au couplage.

Enfin, dans une dernière partie, des exemples illustrent diverses applications du système élaboré. Ces exemples cherchent surtout à montrer les possibilités et la précision que peut atteindre le couplage du calculateur numérique et de la calculatrice hybride.

La partie langage n'est pas développée dans ce mémoire ; elle concerne le travail effectué plus particulièrement par Monsieur DU BOIS de MEYRIGNAC.

La réalisation du système a constamment nécessité des mises au point entre la partie technologique et la partie programmation, aussi est-il difficile de dissocier les travaux de Messieurs DU BOIS de MEYRIGNAC et TARTE.

GENERALITES

Devant les problèmes de plus en plus complexes qui se posent, nous nous apercevons de notre inaptitude au maniement rapide des nombres, aussi a-t-on été amené à construire des outils de calcul de plus en plus perfectionnés, généralement appelés: "machines mathématiques".

D'un point de vue général, on peut classer celles-ci en deux catégories selon leur mode de fonctionnement. La première englobe les machines à traitement continu et parallèle des informations : ce sont les machines analogiques. La seconde concerne les machines à traitement séquentiel d'informations discrètes : ce sont les machines numériques.

Les nombreux opérateurs disponibles sur les calculatrices analogiques permettent une très grande souplesse lors de l'étude d'un problème physique. Ainsi ces calculatrices sont-elles particulièrement adaptées à la résolution de problèmes différentiels et integro différentiels. Une analyse de ces machines indique plusieurs caractéristiques essentielles : d'une part la technique du calcul parallèle impose un mode de fonctionnement uniforme à toute la machine ; d'autre part certains éléments non linéaires sont difficiles à réaliser tandis que la mise en mémoire de l'information pose des problèmes délicats.

Quant aux ordinateurs, ils réalisent à grande vitesse des calculs numériques complexes, les opérations se font en série et les grandeurs mises en oeuvre apparaissent sous forme quantifiée. Ce type de machine impose donc une description mathématique complète du phénomène à étudier et sa réduction aux opérations élémentaires. Parmi les avantages, nous pouvons citer : une grande précision, liée au temps de calcul, une mise en mémoire très facile de l'information, un rôle de gestion puissant commandé par un programme logique. Parmi les inconvénients, figurent la nécessité de discrétiser l'opération d'intégration et une méthode en série.

Finalement l'ordinateur et le calculateur analogique apparaissent essentiellement complémentaires et l'idée de les associer afin de bénéficier des avantages de l'un et de l'autre est toute naturelle.

I PARTIE

DEFINITION DU CALCUL HYBRIDE

On peut définir le calcul hybride comme "l'art et la manière" d'utiliser simultanément un calculateur analogique et un ordinateur liés entre eux par un sous ensemble nommé interface.

Selon les fonctions dévolues au calculateur numérique on distingue le calcul hybride série et parallèle. Par le premier cas les deux calculateurs travaillent l'un après l'autre. Généralement le calculateur analogique résoud le système d'équations, au cours d'un cycle de calcul, alors que le calculateur numérique traite des données ou des résultats afin de préparer un nouveau cycle de calcul analogique.

Dans le calcul hybride parallèle les deux calculateurs travaillent simultanément. Au cours d'un même cycle de calcul, il y a partage des tâches entre les deux calculateurs et donc échanges d'informations (ce mode de fonctionnement parallèle n'excluant évidemment pas le précédent).

Initialement on vit apparaître des ensembles hybrides formés de machines analogiques couplées à des calculateurs numériques de caractéristiques variées, puis des ensembles standards, étudiés et mis au point par une même firme (EAI 690), si bien qu'il est possible de distinguer trois étapes :

- couplage d'un calculateur analogique et d'un calculateur numérique,
- réalisation d'ensembles hybrides avec software limité,
- ensemble hybride avec software évolué.

Le but du travail que nous nous sommes fixé est d'arriver à ce troisième stade à partir de deux machines apparemment disparates ; les deux calculateurs n'étant pas fabriqués par la même société.

I - STRUCTURE D'UN CALCULATEUR HYBRIDE

Nous décrivons ici les deux machines d'une manière générale, la description des éléments spécifiques au couplage sera détaillée par la suite.

I.1 - Le Calculateur analogique

I.1.1 - Description

Un calculateur analogique peut être divisé en cinq parties :

- les éléments analogiques de calcul parallèle (bloc opérationnel)
- les éléments de logiques (simulateur logique)
- les éléments de liaison analogique - logique (interface logique)
- le système de contrôle et de commande
- les appareils de sortie, lents et rapides

Toutes les entrées et sorties des éléments analogiques et logiques ainsi que les commandes sont rassemblées sur deux panneaux de câblage. D'une part le panneau de câblage analogique comprend les entrées et sorties des éléments analogiques, les entrées analogiques de l'interface logique et les entrées des appareils d'enregistrements.

D'autre part, le panneau de câblage logique rassemble les entrées logiques de l'interface logique et la commande des appareils d'enregistrements.

Les éléments logiques permettent de réaliser soit des circuits combinatoires, soit des circuits séquentiels synchrones. Le système de contrôle et de commande comprend le réglage des potentiomètres, la sélection et la lecture sur voltmètre numérique des valeurs de sortie des blocs opérationnels, la commande des modes analogiques et logiques, la visualisation de l'état de chaque élément logique, la commande de l'horloge de synchronisation, etc... (fig. I.1)

I.1.2 - Caractéristiques fondamentales

Le calculateur analogique est composé de blocs opérateurs élémentaires, assemblés par l'intermédiaire d'un panneau de câblage, afin de simuler un modèle mathématique d'un ensemble physique. Les grandeurs mises en jeu par le calculateur sont des variables dépendant du temps et les opérations possibles sur le calculateur sont l'addition, l'intégration, la multiplication, etc...

Il en résulte que le modèle mathématique doit lui aussi ne dépendre que d'une variable indépendant à support positif. Si cette condition n'est pas remplie, il sera nécessaire de faire des transformations ou des approximations préalables pour s'y ramener quand de telles transformations existent.

Les différentes opérations s'effectuent simultanément et instantanément du fait de son fonctionnement en parallèle. Il en résulte que le temps nécessaire à la résolution d'un problème est indépendant du nombre d'opérations à simuler. Cette qualité n'intervient que sur le nombre de blocs opérateurs à mettre en oeuvre, et caractérise par ailleurs la puissance du calculateur.

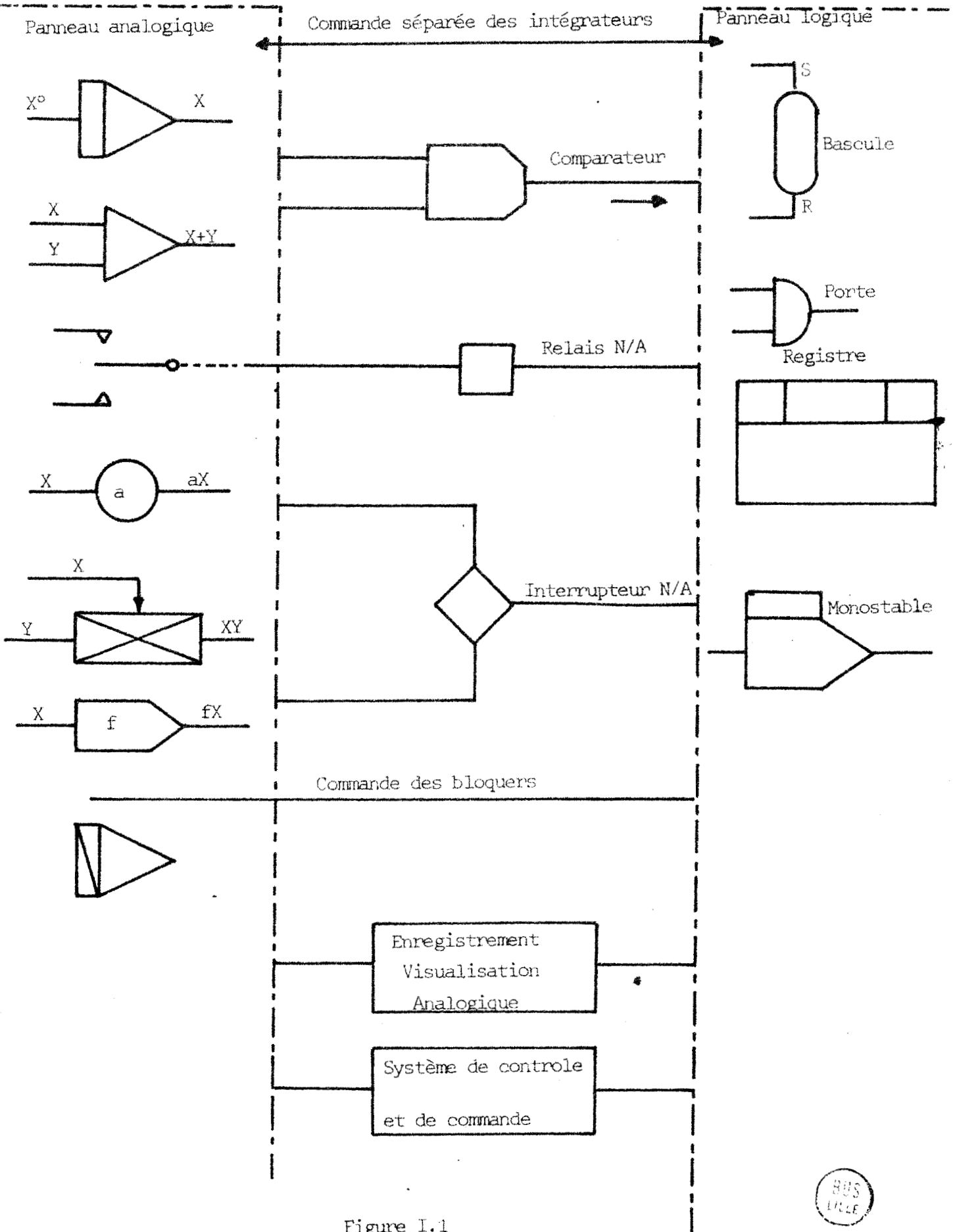


Figure I.1
Organisation générale d'une machine analogique



Nous entendons par temps nécessaire à la résolution du problème, le temps fixé à l'avance, pendant lequel la machine partant des conditions initiales déterminées, fournira l'évolution dynamique des grandeurs physiques ; plus brièvement c'est la durée d'un essai. Nous reviendrons ultérieurement sur cette machine, par comparaison avec le calculateur numérique.

La machine analogique fournit une représentation continue du phénomène étudié, c'est-à-dire qu'au cours d'un essai, chaque tension électrique représentative d'un variable du système peut être notée $X(t)$, le temps t variant de façon continue de 0 à T (fin de l'essai). Cette caractéristique permet de dire que le calcul analogique pur fournit une représentation plus "hette" des phénomènes. Il est en effet possible de définir l'état du système simulé à tout moment, ce que ne permet pas le calcul numérique.

Les blocs opérateurs ne réalisent pas des opérations parfaites. Celles-ci sont limitées par les performances électriques, mécaniques, etc... des éléments.

D'une façon générale, le calcul analogique permet d'aborder l'étude des phénomènes physiques dont le formalisme physique se traduit par un système d'équations. Le calculateur dispose d'organes de commande, de lecture, d'enregistrements qui en font un outil de travail puissant. La conversation avec la machine est relativement aisée grâce à la grande souplesse des organes d'entrée et de sortie d'informations.

I.2 - Le calculateur numérique

I.2.1 - Description sommaire

En vue d'être intégré dans un senssemble hybride, le calculateur numérique doit se caractériser par une grande

vitesse de calcul, un répertoire très complet d'instructions, une largeur de mots permettant une précision suffisante afin de dialoguer avec la machine analogique, une grande souplesse dans les échanges d'entrée-sortie, et être doté d'un système d'interruption efficace.

Nous utilisons pour cela un calculateur de processus, qui permet de par sa structure la gestion d'un grand nombre de périphériques différents (Fig. I.2)

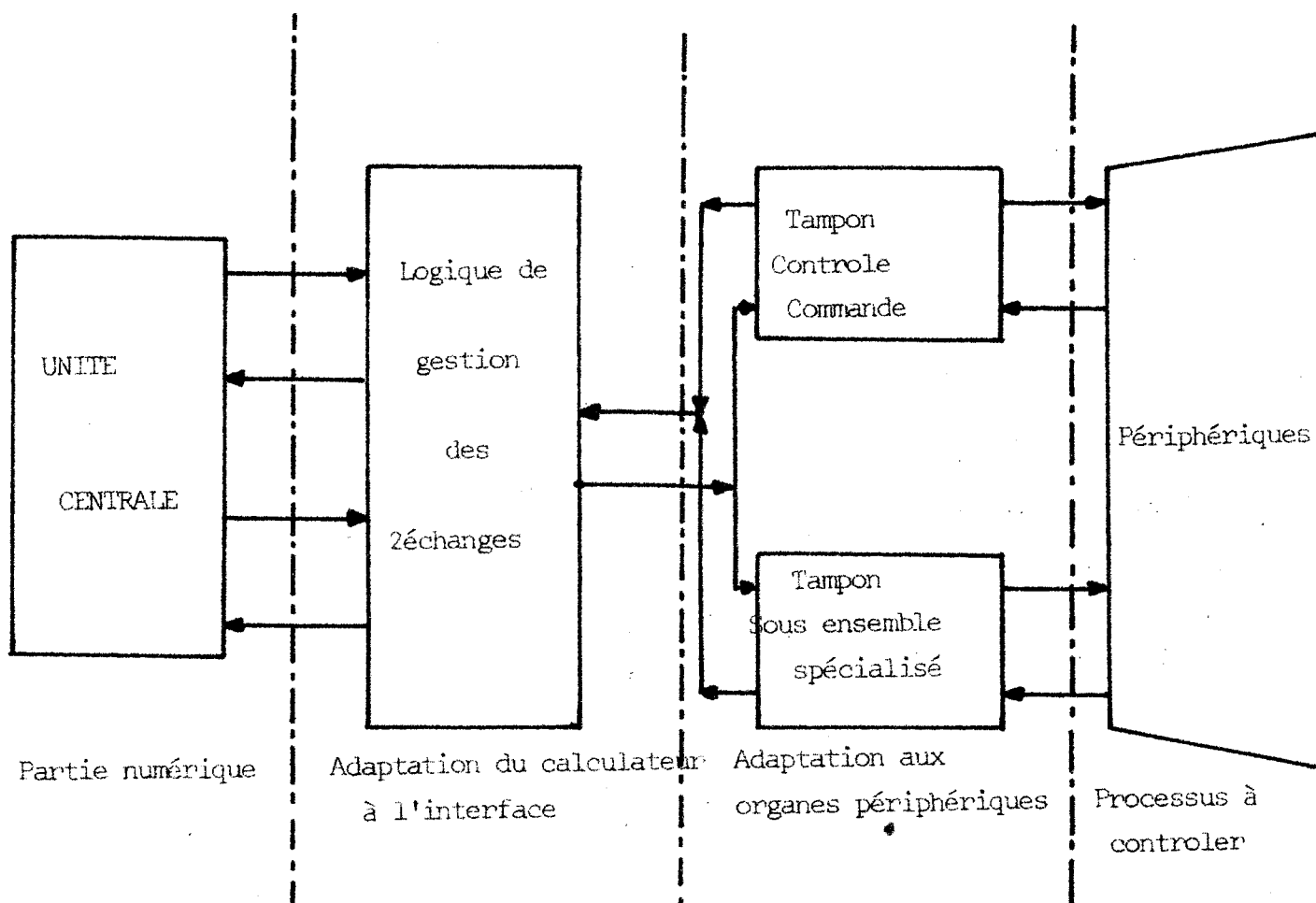


Figure I.2

Les différentes parties d'un ensemble hybride

Quant aux différentes instructions de l'ordinateur, elles permettent d'effectuer les opérations usuelles, aussi bien arithmétiques que logiques. D'autre part, le calculateur dispose d'un canal de transfert automatique qui permet de minimiser le temps d'occupation de l'unité centrale en accroissant la rapidité des échanges. Au niveau de l'adaptation du calculateur avec les différents organes périphériques, il existe une mémorisation des informations données ou reçues par le périphérique, l'adaptation des signaux (format, nature et niveau) est réalisée par l'intermédiaire d'un coupleur. La principale caractéristique de ce dernier est d'être dotée d'une certaine autonomie de fonctionnement, vis à vis de la fonction "adaptation à l'interface unité centrale".

I.2.2 - Caractéristiques

La machine utilisée est un ordinateur de processus, car sa structure permet aussi bien de traiter des problèmes scientifiques que de contrôler un procédé, en relevant périodiquement l'état du système à l'aide de capteurs. Il permet en outre d'allier une grande précision dans les contrôles, à sa rapidité des décisions dans les tests, afin de déterminer les marges de fonctionnement d'un système physique. Afin de diversifier le mode de commande du processus, il est possible de hiérarchiser les interruptions, permettant de connaître l'état actuel du processus (délais, anomalies, changement de consigne...).

Le calculateur est caractérisé par une possibilité de mémorisation importante, que ce soit en utilisant la mémoire rapide, ou en stockant les informations sur les éléments périphériques tels que disques, rubans ou bandes. Cependant, afin d'éviter des pertes de temps, et de compromettre un fonctionnement en temps réel, il est nécessaire de prévoir une mémoire rapide assez importante, permettant

d'éviter un trop grand nombre de transferts de données du calculateur, vers la mémoire lente, ou vice versa.

La précision des calculs dépend en partie de la structure interne de la machine. De nombreux ordinateurs traitent des mots de 16 bits, d'autres, comme l'appareil que nous utilisons utilisent un format de 19 bits, ce qui permet de meilleurs résultats dans les calculs en simple précision.

La machine disposant d'un compilateur Fortran IV, la mise au point des programmes se trouvent simplifiée pour l'opérateur.

En conclusion, nous pouvons dire que le rôle de l'ordinateur dans un système hybride est difficile à définir, car, la plupart du temps les techniques de calcul employées sont fonction du type de problème à résoudre. Toutefois, deux grandes options se dégagent. La première utilise l'énorme pouvoir de gestion des calculateurs numériques, la seconde exploite la facilité de mise en mémoire des informations (table de valeurs numériques par exemple). De plus, la partie numérique pourra effectuer certaines opérations arithmétiques ou contribuera à la réalisation de routines spécifiques à certaines tâches.

I.3 - L'interface

I.3.1 - Constitution

Trois fonctions principales sont confiées à cet organe de liaison :

- d'une part la conversion des tensions analogiques issues du milieu extérieur en données compatibles avec la structure du calculateur numérique, est réalisé à grande vitesse au moyen d'une chaîne de mesure comprenant

un multiplexeur analogique reliant séquentiellement un convertisseur analogique numérique aux canaux transmettant les tensions analogiques, sous contrôle du calculateur numérique. Des bloqueurs sont incorporés sur le trajet de l'information analogique, avant le multiplexeur, pour échantillonner au même instant les valeurs analogiques contenues sur chacun des canaux.

- d'autre part, la conversion des informations numériques ou mots délivrées par le calculateur numérique en valeurs analogiques ; cette opération est effectuée au moyen de convertisseurs numériques analogiques.

- enfin la transmission et l'aiguillage de tous les ordres échangés par les machines lors du déroulement des différentes étapes de résolution d'un problème.

C'est au niveau de l'interface que se font tous les tests d'occupation des périphériques nécessaires au couplage, ainsi que la synchronisation entre les deux machines (fig. I.3)

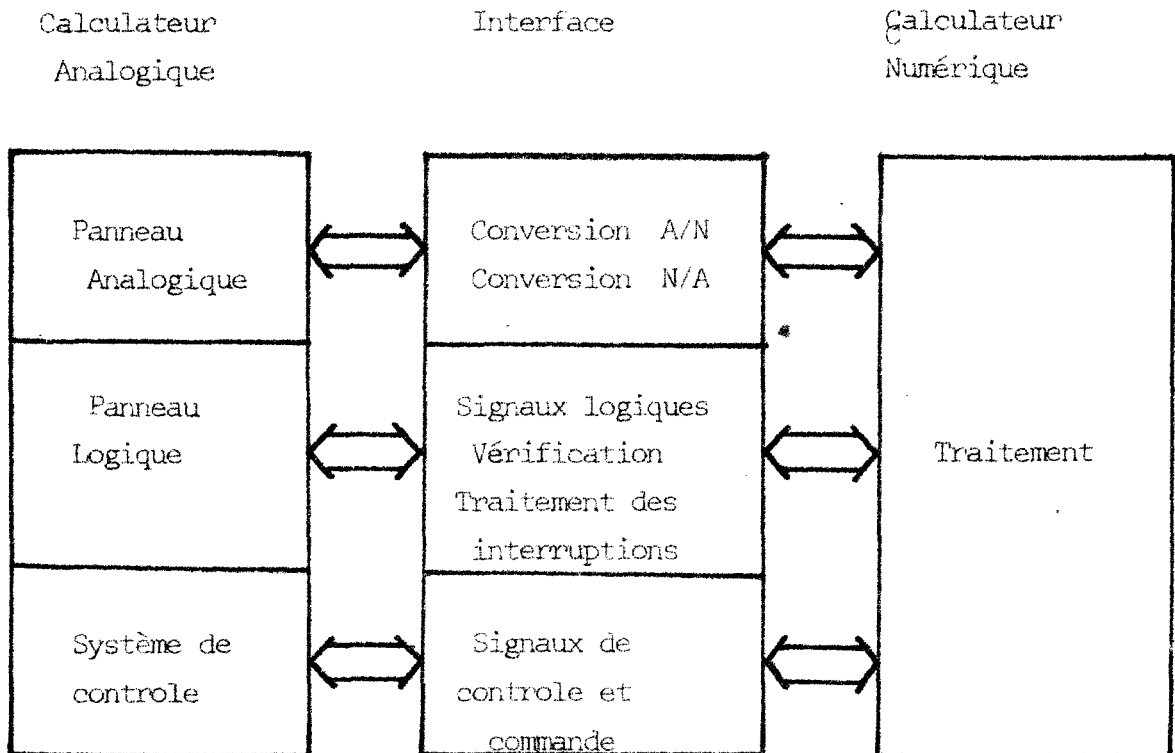


Figure I.3 r -Role de l'interface

Un autre rôle de l'interface est de permettre à l'opérateur de contrôler l'ensemble du problème et le modèle réalisé . L'interface doit aussi pouvoir réaliser directement toutes les fonctions requises par vue d'une autoMatisation de la sélection des organes analogiques, de la lecture des grandeurs de sortie, de l'adressage et de l'affichage des potentiomètres et de la commande du mode opératoire analogique.

L'interface permet aussi de fournir divers états du calcul à la mémoire de l'ordinateur par l'intermédiaire de lignes d'interruptions ou de tests. Dans ce but, la lecture de registres spécialisés permet de connaître l'état du système.

L'interface de commande s'adressant à des unités qui possèdent des temps de réponse différents doit pouvoir être programmée indépendamment de l'ensemble ce qui soulève le problème du "software". En effet un fonctionnement en temps réel nécessite une synghronisation des tâches au niveau du calculateur, et une structure précise dans la modalité des échanges.

I.3.2 - Conclusion

Toutes les possibilités d'un ensemble hybride dependent étroitement de la conception de l'interface qui doit alléger la tâche de l'opérateur mais également lui permettre des interventions rapides et faciles au cours du calcul. A ce niveau se place le problème du langage hybride : dans sa version la plus élaborée, l'interface tend à supprimer le travail de vérification de l'opérateur et du moniteur. La structure des machines utilisées étant fixée une fois pour toute, les vérifications à faire lors des échanges se présentent toujours sous le même format pour les mêmes fonctions; le système étant alors figé, les qualités d'adaptation par software des programmes d'échanges ne sont

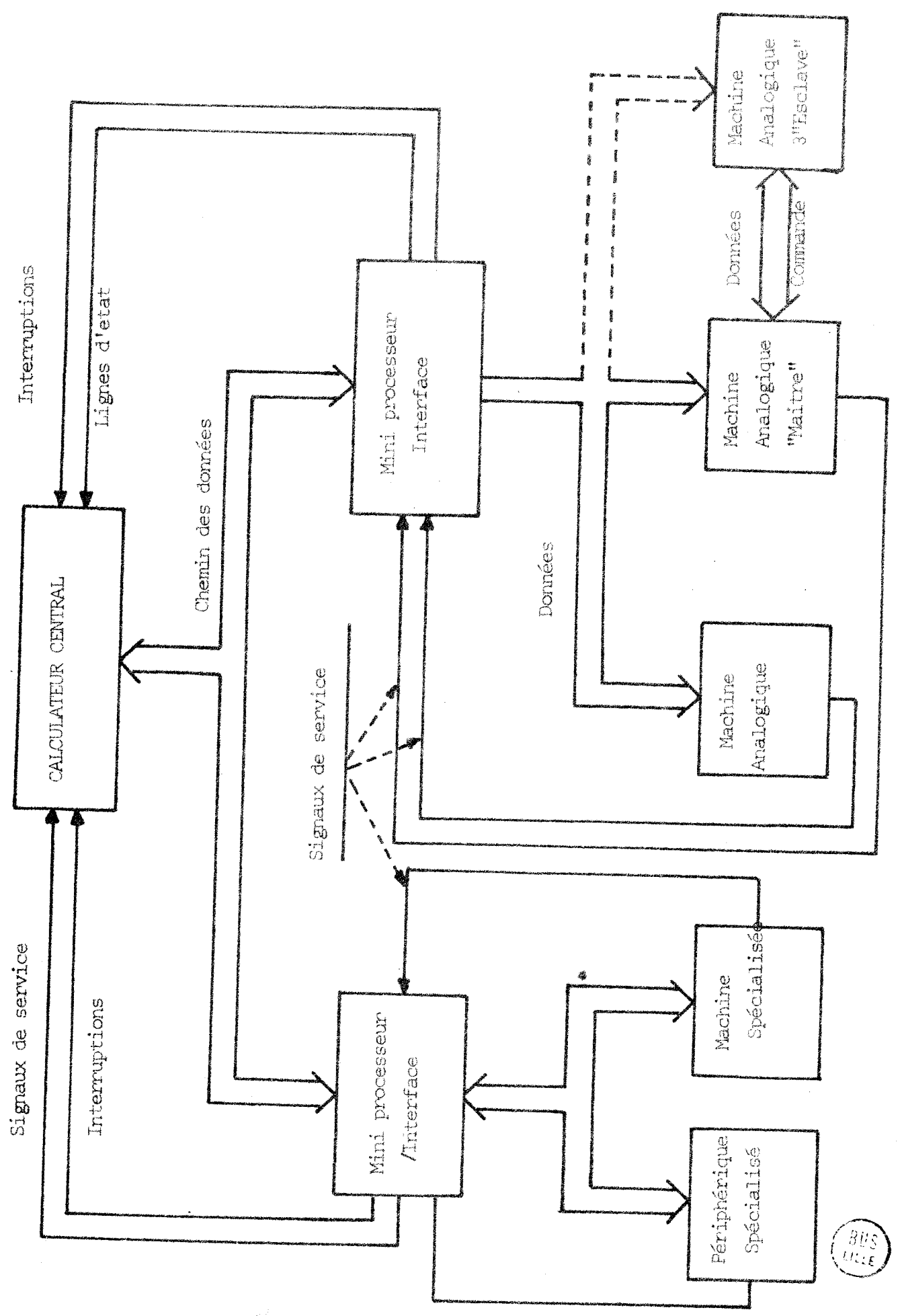
plus utiles, il est alors possible de repousser les vérifications au niveau de l'interface par l'utilisation de la microprogrammation. En allant plus loin, il sera possible de considérer l'interface comme un "minicalcateur" hautement spécialisé, traitant les échanges dont l'ordre et les données initiales seront fournies par le calculateur numérique central. Il sera alors possible à ce dernier, puisque son taux d'occupation diminue, de se consacrer à d'autres travaux qui lui sont plus spécifiques. Du fait que l'ordinateur central ne lance que les échanges, il lui est possible de commander un grand nombre de machines qui lui sont couplées, qu'elles soient analogiques ou de toute autre nature, la fin des échanges étant signalée par une interruption sur un niveau préétabli (fig. I.4). Le calculateur central reconnaît alors l'origine de l'interruption, et teste si l'échange s'est bien déroulé pour valider des informations émises ou reçues, vis à vis de l'utilisateur.

II - FONCTION DES ORGANES CONSTITUTIFS D'UN ENSEMBLE HYBRIDE :

PARTAGE DES TACHES

II.1 - Le calculateur analogique

La qualité essentielle du calculateur analogique est sa très grande vitesse de résolution des équations différentielles : cette tâche lui sera donc confiée. D'autre part la large bande passante des blocs opérationnels et le faible temps de commutation des relais électroniques offrent la possibilité d'effectuer un calcul répétitif commandé par des éléments logiques. Ainsi peut-on confier au calculateur analogique la résolution, à grande vitesse d'un système d'équations différentielles, alors que la logique parallèle



BUS
LISE

commande le déroulement des cycles de calcul, la modification des conditions initiales ainsi que les organes d'enregistrement ou de visualisation. Pendant que les opérateurs analogiques effectuent les opérations continues, la logique parallèle peut également simuler les opérations logiques d'un problème, voire même certains traitements d'informations à grande vitesse. En effet pour les opérations simples la logique parallèle est beaucoup plus rapide que le calculateur numérique. D'autre part, pouvant communiquer facilement avec le calculateur analogique et avec l'ordinateur, elle peut être utilisée avec intérêt pour le contrôle de l'exécution d'un problème hybride.

Nous avons remarqué précédemment qu'en principe l'opération de multiplication, et même de division était possible. Cette dernière étant admise sous toute réserve, il reste que les multiplieurs existent sur le panneau analogique, mais présentent rarement des garanties de précision ; il est de loin préférable de passer par le calculateur numérique, quitte à perdre légèrement en précision par les deux conversions successives analogique numérique puis numérique analogique. A ce niveau, le problème important est le temps. En effet, même si nous disposons de convertisseurs extrêmement rapides, les différents échanges peuvent introduire des retards dans le déroulement d'une simulation dans le cas d'une résolution rapide. Il faut alors prévoir une mise temporaire en condition de "gel" de la machine analogique lors de la multiplication numérique. Une autre solution consiste à utiliser les potentiomètres numériques, qui formés à partir de multiplieurs analogiques numériques très précis, offrent l'avantage de la rapidité, sans nécessiter de passer par le calculateur.

Un autre problème à considérer est la mémorisation analogique. En effet, le plus souvent si les éléments constitutifs de la machine sont de bonne qualité, il existe néanmoins une dérive, qui peut présenter des inconvénients pour certains problèmes. Toutefois, si l'on se satisfait de la précision ainsi obtenue, la mémorisation de plusieurs données

risque de monopoliser un grand nombre d'éléments, ce qui réduit le champ des possibilités de câblage du modèle.

Sur le calculateur analogique, il est difficile de réaliser certaines non linéarités. Bien qu'existent des générateurs de fonctions (à diodes généralement) leur utilisation demeure délicate et nécessite parfois de longues mises au point afin de générer la fonction désirée.

Enfin il est nécessaire lors de l'élaboration d'un problème de prévoir une mise à l'échelle des variables afin de rester dans les normes de fonctionnement de la machine et éviter la saturation des amplificateurs. Cette mise à l'échelle est faite préalablement sur les équations mathématiques décrivant le phénomène physique, et pourra se faire aisément sur le calculateur numérique en utilisant les programmes en bibliothèques.

Une propriété très importante du calculateur analogique est, une fois la durée de résolution du problème choisie, la possibilité de répéter indéfiniment le résultat du calcul suivant le cycle invariant : repos, conditions initiales, calcul. Il est alors très facile d'étudier l'influence du paramètre et d'obtenir ainsi une solution optimale.

II.2 - Le calculateur numérique

L'ordinateur a pour tâches principales le réglage et la lecture des composants analogiques, l'automatisation des opérations sur le calculateur analogique et le calcul hybride en boucle fermée.

Nous pouvons remarquer que l'automatisation des opérations est une généralisation des possibilités déjà offertes par la logique parallèle du calculateur analogique. Le calculateur numérique peut avoir en mémoire un programme

élaboré comprenant, par exemple, la comparaison d'un résultat de calcul à des données physiques, également en mémoire, soit en vue de déterminer de nouvelles valeurs des coefficients, soit pour choisir un modèle mathématique différent. Ces changements seront alors effectués automatiquement sur ordre du calculateur numérique. Il est possible d'automatiser un calcul par son emploi conjugué avec la logique parallèle.

Le calculateur numérique opère rapidement, mais en séquence, si bien que certaines fonctions présentent des difficultés de réalisation. C'est le cas de l'intégration par exemple, où il est nécessaire de discrétiser les variables et d'intégrer sous forme de développement en série. La précision de la solution dépend alors du temps de calcul. La résolution d'équations aux différences, ou d'équations de récurrence nécessite lors du traitement numérique un temps d'occupation important de l'unité centrale, alors qu'il peut être plus facile parfois de considérer l'équation continue décrivant le phénomène, à l'aide du calculateur analogique, et si un traitement numérique particulier est nécessaire, d'échantillonner à partir de la partie numérique. Suivant la méthode employée, les résultats ne sont pas nécessairement identiques.

Une autre possibilité d'utiliser le calculateur numérique en intégration, est de l'utiliser conjointement avec la machine analogique pour les problèmes résolus en mode de calcul multivitesse : l'intégration lente est effectuée sur l'ordinateur, les intégrations rapides seront calculées analogiquement.

D'autre part, il est possible de commander les intégrateurs séparément à partir du calculateur, ce qui peut simplifier l'élaboration de la logique parallèle.

Une autre possibilité du calculateur numérique est son utilisation en générateur de fonctions, opérateurs souvent absents sur la machine analogique. Il existe d'ailleurs d'autres "creneaux", tel que la mémorisation et la

restitution de fonctions, de variables, la génération de retard dont la durée peut être programmée, le traitement d'informations statistiques, etc...

D'une façon générale on considère le calculateur numérique d'un côté, et le calculateur analogique comme un de ses périphériques plus ou moins particulier. Lors de l'élaboration d'un problème hybride, on s'attache à diminuer le temps d'occupation de l'unité centrale de l'ordinateur et l'on se soucie peu de la partie analogique. Cela est dû au fait que la partie analogique calcule en parallèle, donc nettement plus rapidement que la partie numérique ; c'est cette dernière qui a la possibilité de décision et qui commande les différentes étapes du travail. Mais si nous revenons sur la possibilité d'utiliser le calculateur en tant que multiplieur ou comme générateur de fonctions, d'une ou plusieurs variables, l'ensemble numérique est alors utilisé comme un périphérique, la partie principale du travail se développant analogiquement ; il serait tout aussi légitime lorsqu'on détermine le "côût" de temps d'un travail, de compter le temps imparti au calculateur analogique et au calculateur numérique.

Si nous entendons par périphérique toute machine ne pouvant fonctionner véritablement qu'avec un calculateur, (cas du lecteur de cartes, de l'imprimante, etc...) la calculatrice analogique n'est pas un périphérique au sens plein du terme. Le fait d'être utilisée avec un calculateur élargit le champ de ses possibilités, mais n'empêche en rien son fonctionnement en local, c'est-à-dire l'utilisation de ses propriétés . En fait nous entendons par périphérique toute machine utilisée par le calculateur sans en faire partie intégrante.

Nous avons défini le calculateur analogique comme étant capable d'intégrer aisément les équations différentielles, à conditions toutefois que la précision globale demandée par le calcul ne dépasse pas 10^{-4} . Au delà le calculateur numérique est

nécessaire ; grâce au traitement en virgule flottante, il supplée efficacement l'analogique pour le traitement des grandeurs à fortes variations dynamiques.

Dans la simulation de la dynamique des systèmes, la dispersion des fréquences des solutions est souvent importante. En effet, pour observer les transitoires rapides, on travaillera analogiquement avec une échelle des temps dilatée : les dérives des amplificateurs perturbent alors les phénomènes lents. Inversement si l'on choisit une échelle contractée pour étudier les phénomènes transitoires lents, les phases rapides sont cachées.

La même difficulté se retrouve lors de la résolution numérique lorsqu'il s'agit de définir le pas d'intégration. Cependant, les parties hautes et basses fréquences sont souvent assez découplées pour qu'on puisse les traiter séparément. Les premières sont câblées en analogique, les secondes programmées en numérique.

Un programme de simulation numérique doit être exécuté d'une manière quasi parallèle s'il veut conserver la notion du temps réel. Par cela le pas d'intégration minimal est égal au temps de calcul élémentaire, c'est-à-dire à la durée totale d'exécution de toutes les opérations qui sont incluses dans ce pas. Si nous voulons simuler un problème nécessitant dans le cas d'une résolution analogique l'emploi d'une centaine d'amplificateurs nous sommes obligés de le traiter numériquement.

Si nous supposons un temps de résolution de l'ordre de grandeur de 4 000 à 5 000 cycles de bases du calculateur, nous arrivons, suivant la méthode que nous employons à une résolution d'une durée de 5 ms, pour une précision de l'ordre de 10^{-4}

Dans un cas tel que celui-ci, la simulation en temps réel purement numérique n'est alors pas possible, et la méthode purement analogique non plus. Le calcul hybride se présente alors comme la seule solution à condition de

répartir convenablement les tâches. Cette situation se reproduira lorsqu'un système présente un grand nombre de cellules semblables ; bien que le modèle mathématique relatif à chaque cellule soit le plus souvent bien adapté à la résolution analogique, cet avantage peut se transformer en un inconvénient majeur. Il est en effet nécessaire de disposer d'autant d'éléments de calcul que de cellules, ce qui peut devenir rapidement prohibitif. On peut alors simuler le fonctionnement d'un nombre restreint de cellules et appliquer une méthode voisine du calcul répétitif ou itératif en calculant successivement la propagation des variations des éléments le long des cellules.

III - CONCLUSION

Il existe de nombreux domaines d'application du calcul hybride, n'offrant pas seulement une amélioration des possibilités déjà offertes par des méthodes purement analogiques ou numériques, mais aussi, comme nous l'avons vu précédemment, présentant une méthode originale de traitement de problèmes impossibles à résoudre par ailleurs.

Pour tout système géré par ordinateur tel qu'une usine par exemple, il est nécessaire de tester par avance le comportement de l'ensemble géré par programmes. Une simulation hybride permet d'atteindre cet objectif dans lequel le système lui-même est représenté par une structure analogique ou hybride et l'ordinateur de commande par une partie de l'unité numérique du calculateur hybride.

D'autres possibilités se sont généralisées, comme l'identification des problèmes, l'optimisation, la physique nucléaire, etc...

On ne peut évidemment pas conclure que le calcul hybride peut tout faire parce qu'il allie les avantages des méthodes numériques et analogiques. Il en reçoit aussi

certains inconvénients et présente par là même des limitations; de nombreux efforts doivent encore être réalisés tant dans le domaine des méthodes que dans celui des langages. A notre niveau nous décrivons le système de base de l'interface réalisée, et mettons en évidence sur des exemples, les possibilités de résolution de quelques problèmes, les comparant avec ce qui avait pu être obtenu par une solution uniquement analogique ou numérique.

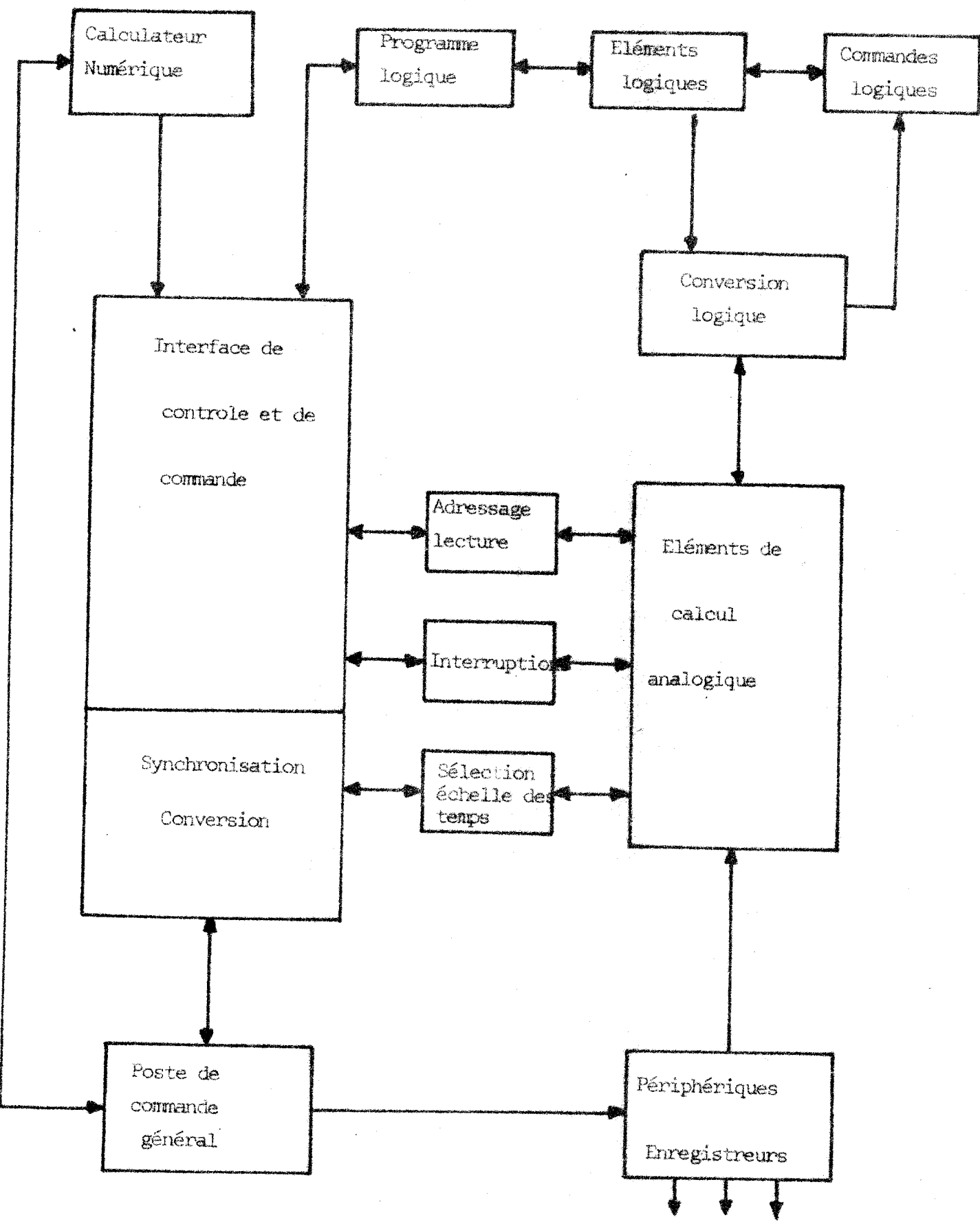


Figure I.5
Ensemble Hybride



II PARTIE

LE COUPLAGE

II - LES CALCULATEURS

Le couplage envisagé dans ce travail, a été effectué en utilisant un calculateur de processus "Teleme- canique T 2000 " et une calculatrice hybride à courant continu "EAI 580" ; notons toutefois que cette étude reste valable pour n'importe quel type d'appareil, et que les performances du système global, dépendront surtout du matériel utilisé.

II.1 - L'ordinateur de processus

Le calculateur T 2000, est conçu pour prendre en compte et traiter des informations de toute forme, grâce à une gamme complète d'organes élémentaires réalisant des fonctions caractéristiques avec des performances variables.

II.1.1 - Organisation générale du calculateur

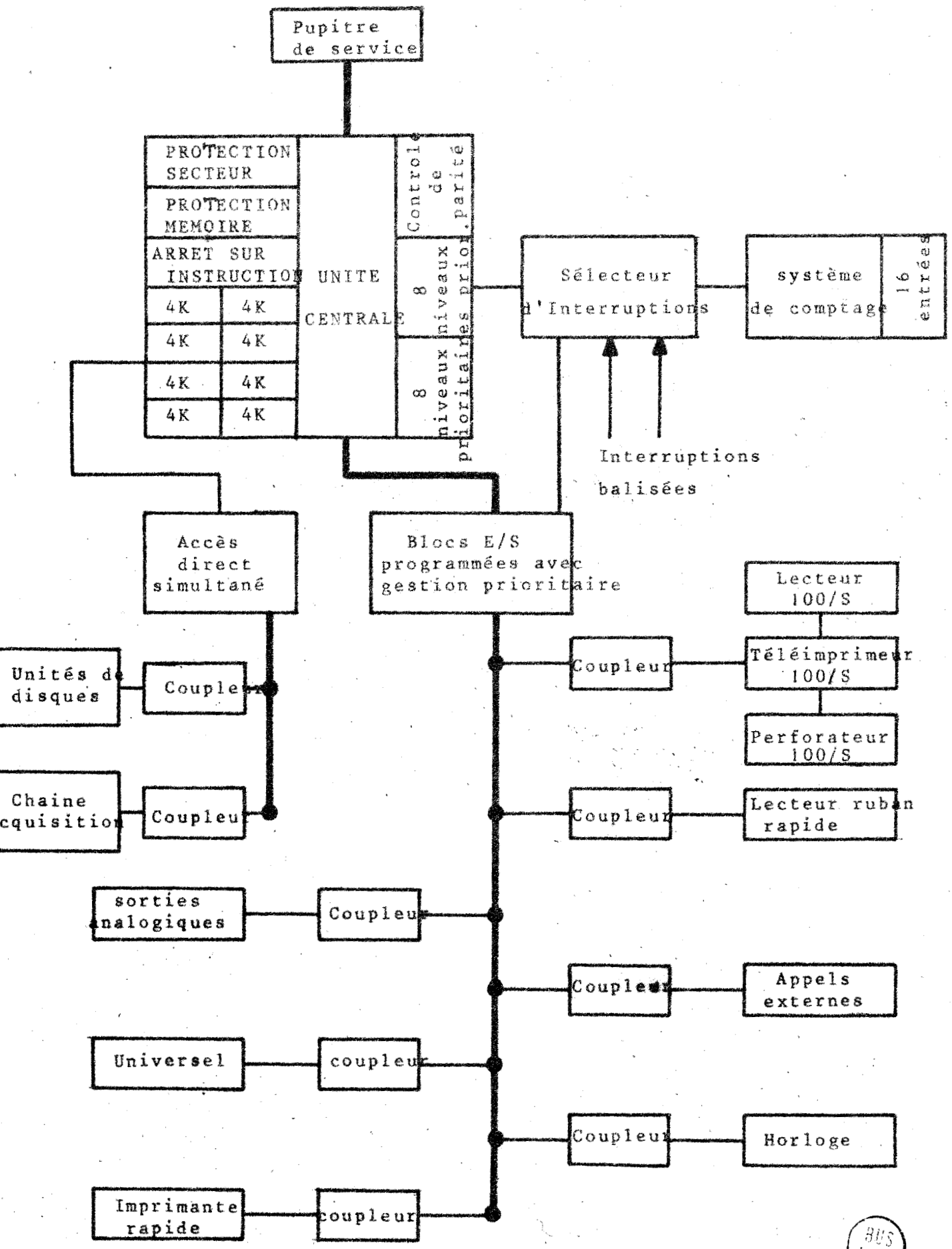
La chaîne numérique d'Automatisme comprend :

- un organe central : le calculateur
- un système d'interruptions prioritaires et de voies de comptages
- des voies d'entrée et de sortie

a) Organe central

Le calculateur T 2000, a un cycle de base de $1,5 \mu s$. Les circuits de calculs parallèles confèrent au système une rapidité dans l'exécution. La mémoire, à tores, est extensible par bloc de 4 K, à concurrence de 32. Les mots traités sont de 19 chiffres binaires, et un vingtième appelé bit de parité assure le rôle de protection de la mémoire.

+



BUS
LILLE

Figure II.1
Organisation générale du calculateur numérique

b) Le système d'interruptions

Le système d'interruptions comporte jusqu'à 16 niveaux hiérarchisés, masquables par programmation

Il assure les fonctions suivantes :

- acquisitions et mémorisation des appels
- tri des appels suivant leur hiérarchie
- exécution de l'interruption
- libération du niveau.

Un système de comptage permet de traiter sur un niveau de priorité 16 à 240 voies de comptage.

Le premier niveau dans la hiérarchie est toujours réservé aux alarmes internes, les 15 autres niveaux correspondant aux modules implantés dans un bac d'entrées-sorties.

c) Le système d'entrées-sorties

On distingue deux types de dispositifs d'entrées-sorties :

- les entrées-sorties conventionnelles : téléimprimeur, imprimante rapide, lecteur de cartes, de rubans...
- les entrées-sorties industrielles, numériques ou analogiques.

Ces derniers sont très importants dans la réalisation du couplage, nous allons en détailler la structure.

Les entrées-sorties numériques permettent de correspondre avec le milieu extérieur en utilisant comme support d'informations des mots de 19 bits, auxquels il faut ajouter les signaux de service ; ces derniers indiquent l'occupation et l'état du périphérique considéré, ainsi que la validation de l'information.

L'initiative de cet échange est réservée à l'unité centrale, qui peut être motivée par des interruptions prioritaires.

En particulier, le coupleur universel permet d'échanger quatre mots d'informations (2 mots d'entrée et 2 mots de sortie) en logique TTL. Comme le montrent les figures II.2 et II.3, des signaux de service valident les informations d'entrée et rendent le périphérique libre ou occupé.

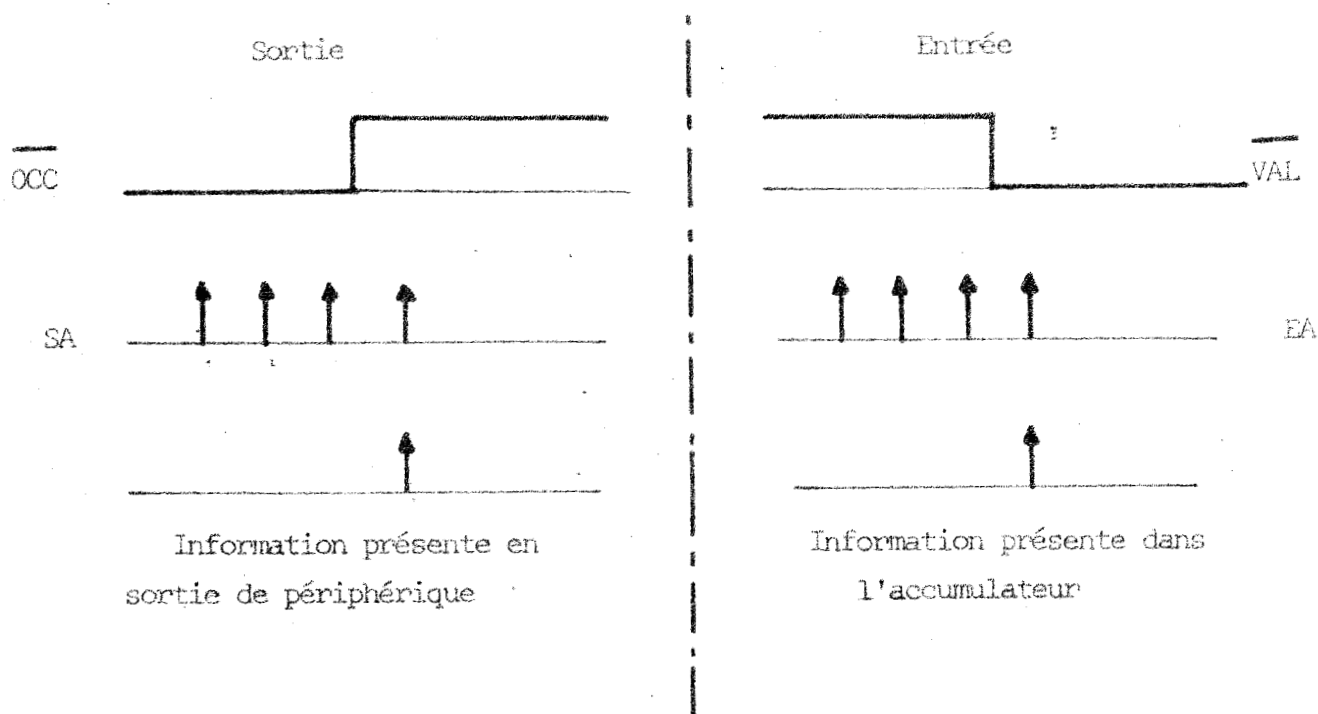


Figure II.3
Signaux de service.

$\overline{\text{OCC}} = 0$, le périphérique est occupé, et toute sortie est impossible.

$\overline{\text{VAL}} = 0$, l'information est considérée comme non valide et son acquisition ne peut se faire.

La borne $\overline{\text{VAL}}$ a une autre utilisation; si le bit 0 du mots de sortie est à 1, lorsque le signal sur $\overline{\text{VAL}}$ passe de 1 à 0, le coupleur crée au niveau du bac d'entrées-sorties, un appel prioritaire.

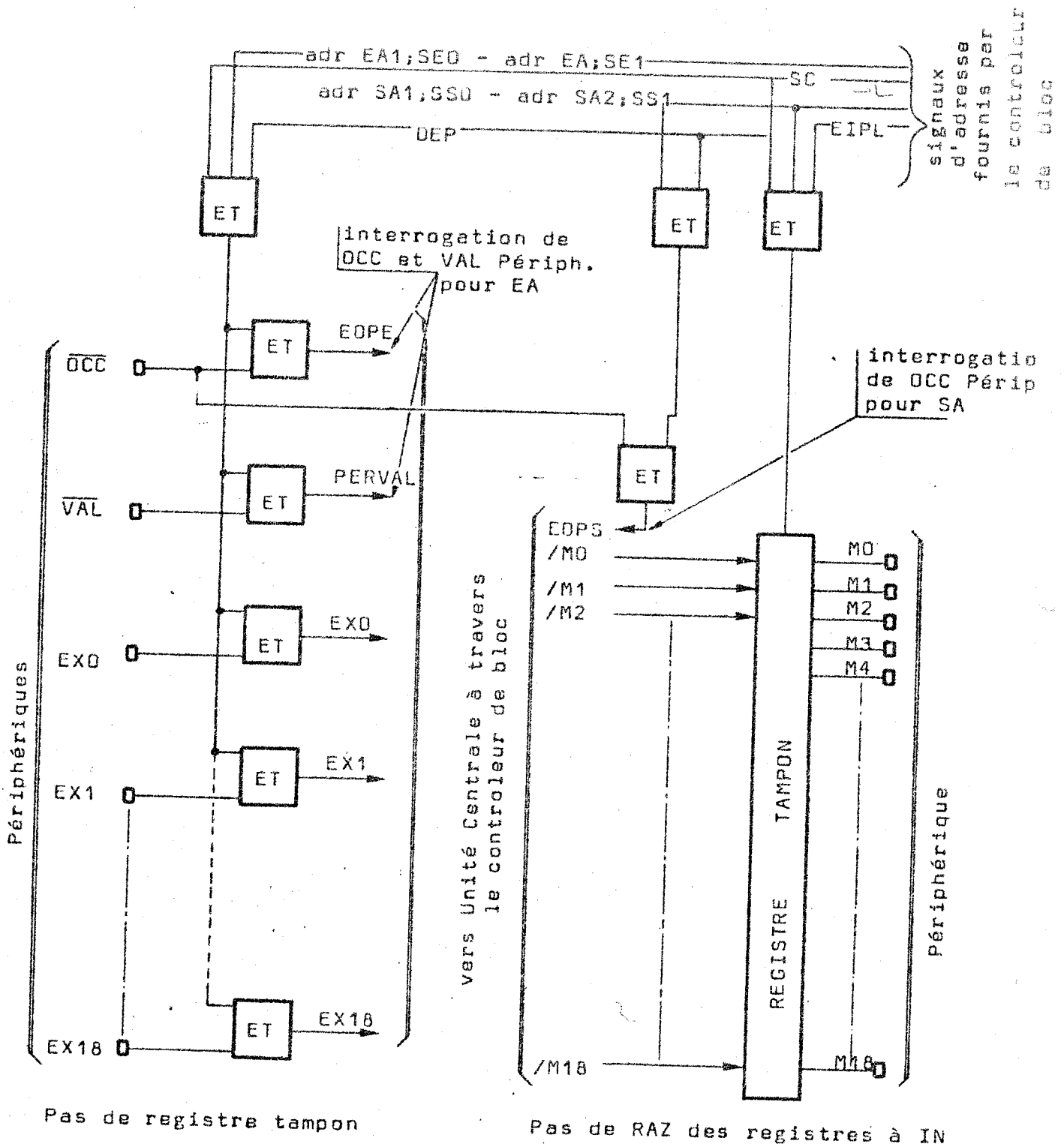


Figure II.2
Le coupleur universel.



D'autres signaux sont disponibles, mais ceux que nous avons cités, sont les plus intéressants pour notre utilisation ; ils définissent les modalités d'échanges avec l'interface.

Les entrées-sorties analogiques sont plus particulières ; leur choix dépend surtout de la nature des problèmes à étudier. Le principe reste le même pour tous les modèles, et le périphérique est choisi en fonction d'un certain nombre de caractéristiques : précision, vitesse, tension...

Le calculateur du laboratoire est équipé d'une chaîne d'acquisition "PRESTON" ; la mesure est convertie sur 14 bits plus signe en un temps de 10 μ s. La sélection de la voie étant de 1 μ s, la chaîne peut donc lire 9000 voies par seconde. Cette vitesse est limitée par le canal d'acquisition et elle n'atteint que 8 000 voies-seconde. Le nombre de voies actuel est de 8, extensible jusqu'à 32. La précision étant de 1 à $2 \cdot 10^{-4}$, il est possible d'apprécier le millivolt pour 10 volts de référence.

Les sorties analogiques, en liaison avec le bloc d'entrées-sorties délivrent et mémorisent entre deux remises à jour, un signal analogique dont l'amplitude est fixée par une valeur numérique. La résolution se fait en ± 512 points soit 9 bits plus signe avec une précision de $2 \cdot 10^{-3}$; la gamme de sortie est de - 10 volts à + 10 volts, avec un temps de stabilisation de 30 μ s. Les sorties analogiques forment une source d'alimentation programmable utilisable soit pour un affichage de tension, soit pour une commande de système.

II.1.2 - L'unité de traitement

Le calculateur possède un code d'ordre de 56 instructions, deux accumulateurs et un index. Tous les modes d'adressage sont possibles et une barre de verrouillage pro-

grammable assure la protection de la mémoire par zones. Les circuits de calculs sont parallèles, la multiplication et la division étant des opérations cablées.

Enfin un dispositif de sécurité permet la sauvegarde de l'état des registres en cas de défauts d'alimentation. Il contrôle la durée des instructions, teste certaines erreurs de programmation par verrouillage d'une zone mémoire et détecte les instructions illégales ou impossibles. Une défaillance de la mémoire est contrôlée par le bit de parité.

II.1.3 - Modes d'entrées-sorties

On peut distinguer trois modes de fonctionnement pour les entrées-sorties.

En mode programmé simple, chaque opération d'entrée ou de sortie se fait par l'intermédiaire de l'accumulateur. Il convient alors de vérifier que le périphérique sur lequel on effectue un échange est libre.

Dans le mode programmé prioritaire, le périphérique indique à l'unité de traitement par le système d'interruptions qu'il est prêt à prendre en compte ou qu'il dispose de l'information.

Quant au mode canal, il permet de réduire la programmation et de libérer l'unité centrale des opérations de dialogue avec le périphérique. La gestion des échanges se fait au rythme du périphérique, et en réalisant un blocage de l'unité centrale, le débit d'informations peut atteindre 200 000 mots par seconde.

II.2 - Le calculateur hybride

II.2.1 Organisation générale

La calculatrice hybride, est une machine à courant continu. Elle permet la résolution simple d'équations différentielles linéaires ou non linéaires ainsi que le fonctionnement simultané de plusieurs boucles de calcul à des vitesses d'intégration différentes. Cette propriété est le trait le plus fondamental de ce calculateur, et est à l'origine des méthodes de sous-routine et de calcul multivitesse.

Tout ceci n'est rendu possible que par l'adjonction à la partie analogique désignée, d'une logique synchrone, capable de commander séparément chaque intégrateur.

II.2.2 - Les fonctions analogiques

Les tensions de référence de la calculatrice à courant continu, sont de ± 10 V.

L'amplificateur opérationnel est l'élément majeur de cet ensemble ; il existe plusieurs types de modules dont :

- l'intégrateur commandable par une logique parallèle, et disposant de deux constantes de temps d'intégration, par choix des capacités de bouclage,
- le sommateur et l'inverseur,
- le comparateur délivrant un signal logique et son complément,
- les relais, commandés par des variables logiques,
- le potentiomètre,
- la mémoire paire qui permet l'échantillonnage et la mémorisation d'une information.

II.2.3 - Les fonctions logiques

La logique de l'EAI 580, est une logique positive 0volt-5volts, possédant un certain nombre de fonctions de

base :

- les portes "ET",
- le différenciateur,
- les compteurs à décades,
- les bascules synchrones,
- les compteurs-décompteurs,
- des boutons poussoirs.

II.2.4 - Le pupitre de commande

Un pupitre de commande contrôle l'ensemble et permet le choix d'un certain nombre de fonctions analogiques et logiques.

L'utilisateur choisit la constante d'intégration des amplificateurs, et la fréquence de l'horloge.

Le temps de répétitif est sélectionné en fonction du problème ; le pupitre exécute la sélection des potentiomètres et l'affichage de la valeur analogique.

Notre problème consiste donc, à effectuer une simulation numérique du pupitre, afin de réaliser les mêmes opérations par le calculateur de processus.

II.3 - Conclusion

La description des deux calculateurs, nous amène maintenant à aborder l'étude du couplage entre ces deux machines.

Les fonctions du pupitre de commande, sont facilement simulées par le T 2000, les références logiques des deux machines étant les mêmes.

L'affichage des potentiomètres, par contre risque d'être plus délicat à réaliser. En effet, sur l'EAI, la sélection d'un potentiomètre est faite à partir d'une matrice de relais, dont la tension d'alimentation est de -20 volts- Cela nécessite un changement de logique (0v, 5volts) en (0v, -20 volts). La valeur du potentiomètre à afficher est obtenue grâce au convertisseur digital-analogique du calculateur.

Le problème est surtout d'assurer un grand nombre de commandes avec le minimum d'informations venant du calculateur, donc de minimiser le nombre de coupleurs universels du T 2000.

L'étude a été envisagée pour le couplage de deux EAI, et de ce fait nous avons adopté une configuration qui n'est pas nécessairement celle convenant au couplage avec une seule calculatrice.

III - INTERFACE

La première étape dans la réalisation d'un ensemble hybride par couplage d'un ordinateur de processus et d'une calculatrice hybride, est d'effectuer par programmation, toutes les opérations manuelles réalisables au pupitre de la calculatrice.

Nous devons noter dès maintenant que les regroupements d'informations effectués, ont été choisis suivant plusieurs critères :

- minimiser le nombre de signaux de commande
- envisager un décodage simple et des vérifications au niveau de l'interface
- assurer des facilités pour le codage des informations au niveau de la programmation

- et enfin tenir compte de la technologie utilisée.

Dans toute notre réalisation, nous nous sommes intéressés des liaisons hardware-Software, afin d'assurer pour chaque partie un travail minimum, avec un maximum d'efficacité et de rapidité.

III.1 - Commande du pupitre

Dans cette partie, nous distinguons plusieurs groupes de commandes. L'utilisateur a le choix des constantes d'intégration, et de la fréquence d'horloge, mais également, des commandes analogiques et logiques.

III.1.1 - Etude des signaux EAI

L'analyse du système de commande des différentes fonctions est indispensable avant l'exposé de notre réalisation.

Cette étude peut se décomposer en quatre parties, suivant la nature de chacune des commandes.

a) Les Commandes analogiques.

Les commandes analogiques sont au nombre de cinq :

SP : "Set Pot", arrêt de calcul et remise à zéro des
intégrateurs

IC : "Initial condition", mise en conditions initiales

OP : "Operate", résolution, calcul

HD : "Hold", gel des calculs

PP : "Program Panel", calcul en mode répétitif.

L'existence de l'un de ces signaux élimine les autres, sauf pour le répétitif, qui assure les opérations de mise en conditions initiales et en calcul.

La figure III.1, permet d'étudier la réalisation de ces commandes.

La présence des bascules assure une mémorisation de l'information mais en même temps, élimine les phénomènes de rebondissement dus à l'action sur l'un des poussoirs.

Sur les lignes KBP14-2, KBP14-1, KBP14-3, KBP14-4, KBP14-5 et KBP14-6, nous disposons après décodage, du signal correspondant à la fonction affichée. Cette information reste présente jusqu'au prochain état du à un changement de fonctions.

b) Les commandes logiques

Nous distinguons quatre types de commande logique :

C : "Clear" , remise à zéro des bascules et compteurs

S : "Stop" , arrêt des horloges

R : "Run" , exécution du programme logique

PP : "Patch Panel" , répétitif de la logique.

La figure III.2 nous montre l'organisation de ce système. Nous recueillons, sur les lignes KBP21-1, KBP21-2, KBP21-3 et KBP21-4, après décodage, l'information que nous avons affichée.

c) Le temps d'intégration

L'utilisateur a le choix entre deux constantes d'intégration : 1 s ou 2ms ; ces constantes sont modifiables par action directe au niveau du câblage, en choisissant la capacité de bouclage de l'ampli-intégrateur.

Comme précédemment, une bascule mémorise l'information qui après décodage est disponible sur la ligne KBP8-7 (figure III.3).

d) L'horloge de la logique

Les informations décodées et mémorisées sont disponibles sur les lignes KBP21-5, KBP21-6 et KBP21-8.

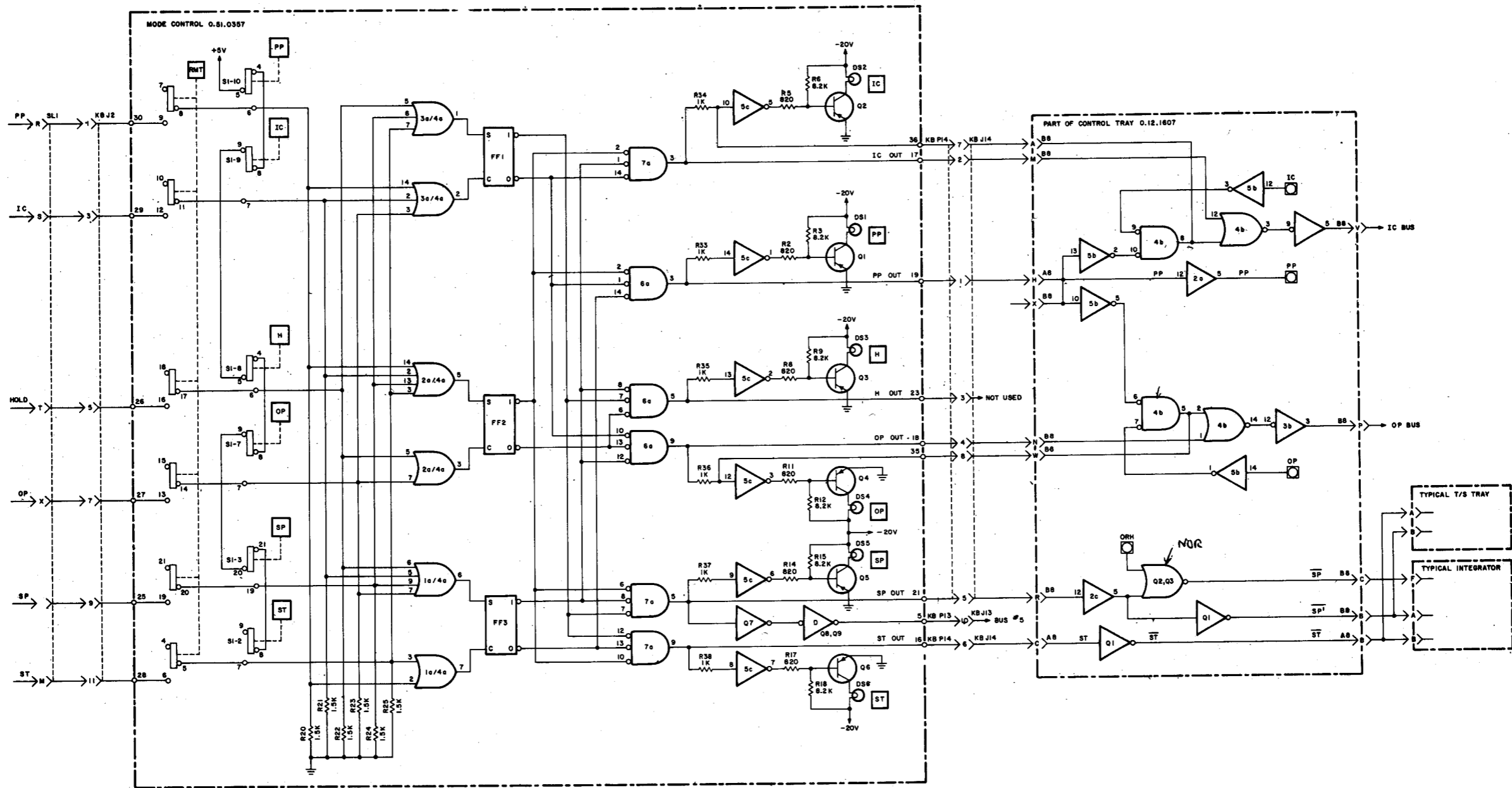
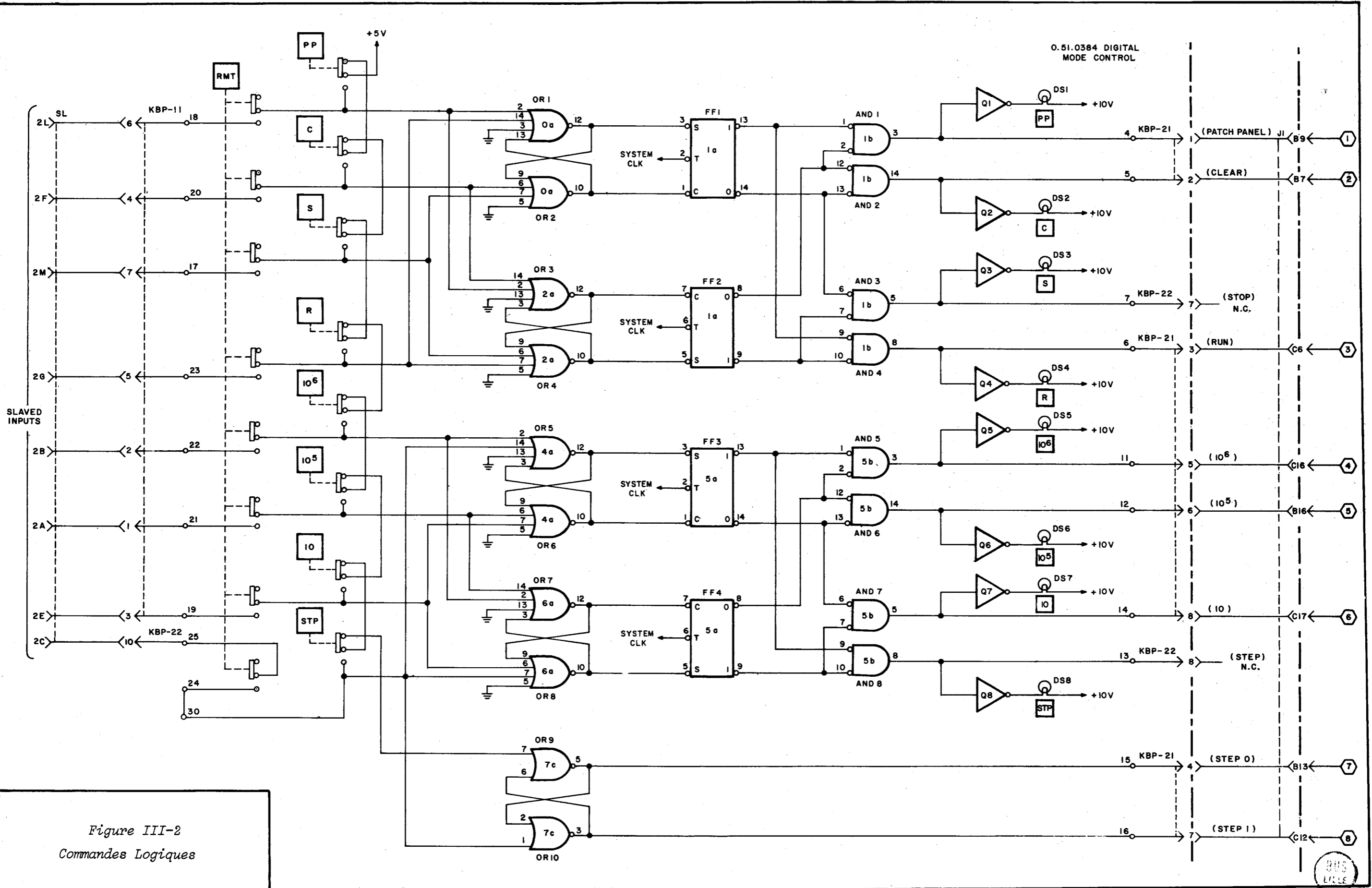


Figure III-1
Commandes Analogiques



0.51.0384 DIGITAL
MODE CONTROL

Figure III-2
Commandes Logiques



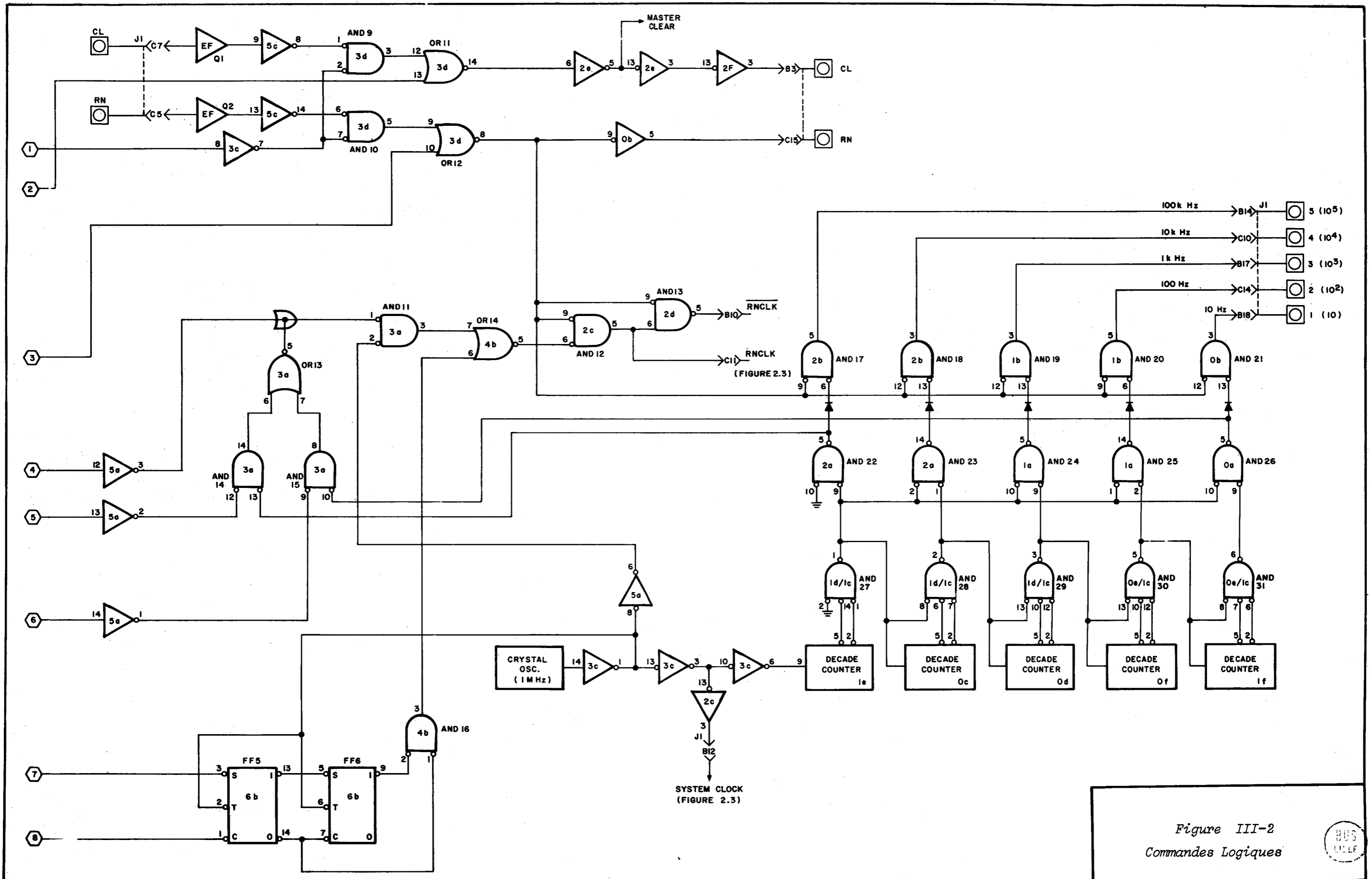


Figure III-2
Commandes Logiques



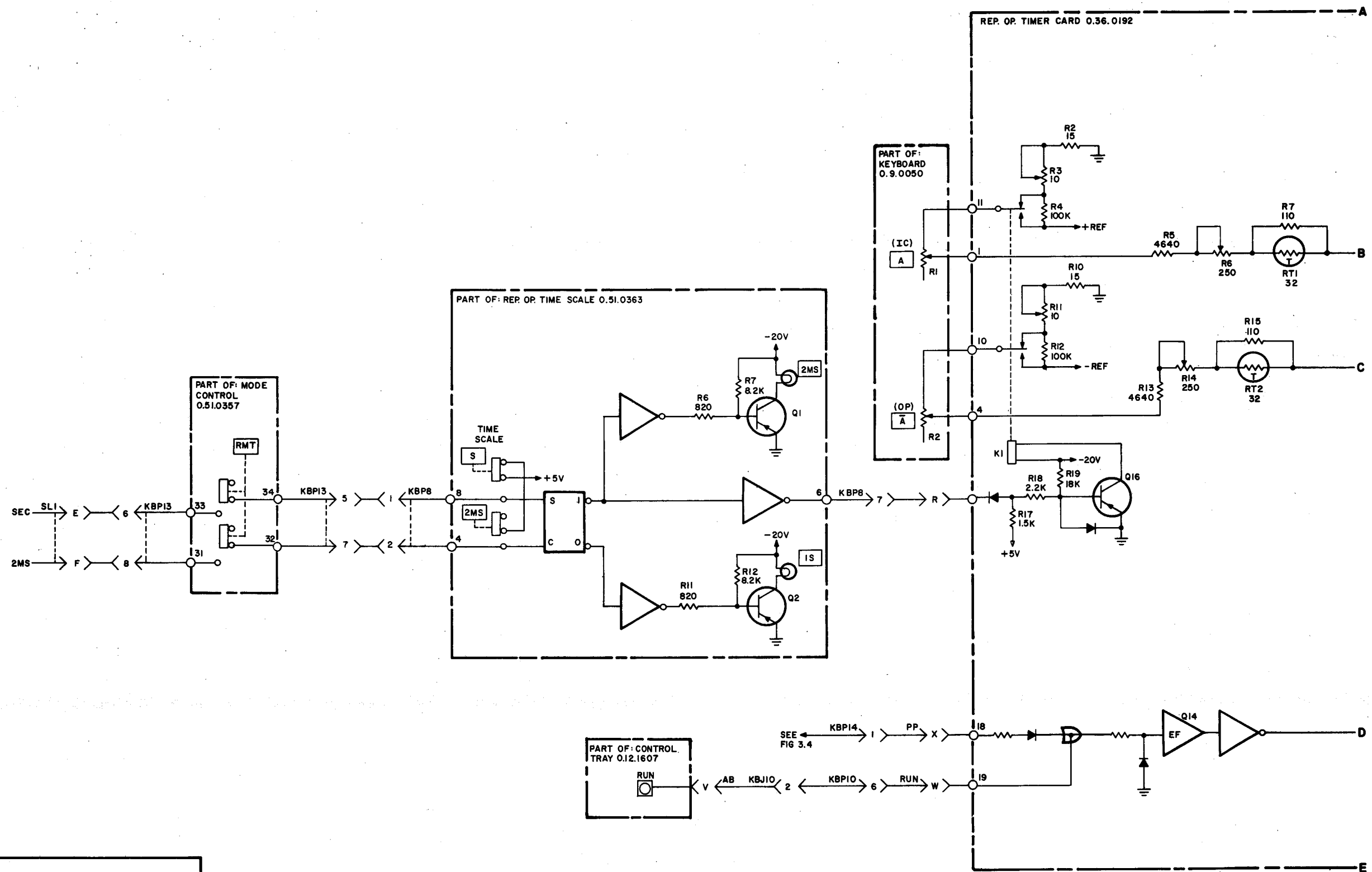


Figure III-3
 Constante d'integration



L'observation de la figure III.2, montre, que l'horloge de la logique pilote toutes les bascules du pupitre de commande.

e) Durée du répétitif

L'utilisateur a accès à trois commandes de répétitif (figure III.4).

La sélection du poussoir X1, relie les lignes KBJ19-1 et KBJ19-2

La sélection du poussoir X10, relie la ligne KBJ19-1 aux lignes KBJ19-2 et KBJ19-3.

La ligne KBJ19-1 est à un potentiel de + 10 volts.

III.1.2 - But à atteindre

D'après l'étude précédente, plusieurs réalisations sont possibles, Mais il faut remarquer qu'une EAI peut en piloter une autre en transmettant tous les signaux de commande.

Un bouton "RMT", ("remote"), permet de mettre en relation directe, les bascules du système avec les informations extérieures venant de l'organe maître.

Il semble intéressant d'utiliser cette propriété afin d'assurer la commande du système.

Nous envisageons d'effectuer un couplage avec deux calculatrices EAI. Les signaux de commande doivent donc être distribués simultanément aux deux machines ; or comme les informations sont délivrées par plusieurs coupleurs universels, nous sommes obligés de considérer un système de bascules au niveau de l'interface ; ce dernier ne prend en compte l'information d'entrée qu'à l'instant où toutes les sorties sont effectuées, sauf pour les changements des constantes d'intégration et d'horloge.

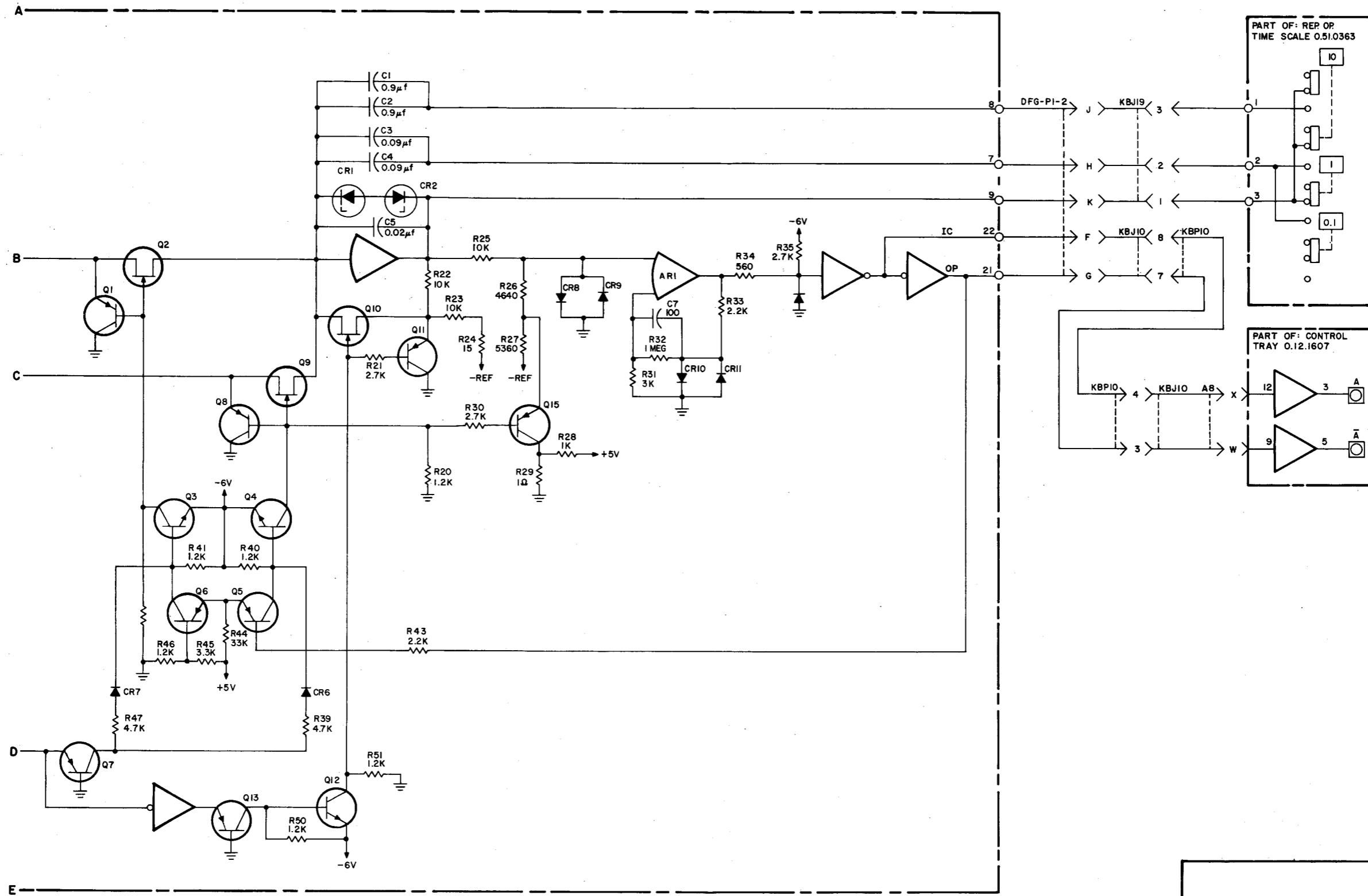


Figure III-4
Duree de répitif



Le problème consiste maintenant à regrouper les informations entre-elles, afin de minimiser le nombre de bits de commande, tout en essayant de garder une programmation simple.

Le regroupement qui semble le plus avantageux est le suivant :

- commandes analogiques et durée du répétitif
- commandes logiques
- temps d'intégration
- horloge de la logique.

III.1.3 - Réalisation pratique

a) Les commandes analogiques

Aux informations de commandes analogiques, nous associons trois états notés PP1 (X0.1), PP2 (X1.), PP3 (X10.), représentant les différentes durées de répétitif.

Si CANA1, CANA2, CANA3 sont les bits de commande venant du T 2000, la table de vérité (figure III.5) du système s'écrit :

CANA1	CANA2	CANA3	SP	IC	OP	HD	PP	X1	X10	Commandes
0	0	0	1	0	0	0	0	0	0	Arrêt
0	0	1	0	1	0	0	0	0	0	Cond. Initiales
0	1	0	0	0	1	0	0	0	0	Calcul
0	1	1	0	0	0	0	1	0	0	Répétitif 1
1	0	0	0	0	0	1	0	0	0	Gel
1	0	1	0	0	0	0	1	1	0	Répétitif2
1	1	0	1	0	0	0	0	0	0	Arrêt
1	1	1	0	0	0	0	1	1	1	Répétitif3

Figure III.5

Table de vérité des commandes analogiques

Chaque information (SP), (JC), (OP), (HD), (PP) est mémorisée par une bascule D, pilotée par l'information "RAFFICH" de fin de commande venant du T 2000.

Les différentes lignes de sortie sont raccordées aux lignes KBJ2-9, KBJ2-3, KBJ2-7, KBJ2-5, KBJ2-1 de la figure III.1.

Les informations (X1.) et (X10.) doivent subir un changement de logique (0V - 5 volts) en (0V - 10 volts) et vont respectivement sur les lignes KBJ19-2 et KBJ19-3 de la figure III.4.

La figure III.6 présente la réalisation de la commande des fonctions analogiques.

b) Commandes logiques

Deux informations notées CLOG1 et CLOG2 définissent les quatre commandes logiques possibles.

Nous avons regroupé (figure III.7) dans une table de vérité les différentes informations.

CLOG1	CLOG2	STOP	CLEAR	PPL	RUN	Commande
0	0	1	0	0	0	Arrêt
0	1	0	1	0	0	Remise à zéro
1	1	0	0	1	0	Répétitif
1	0	0	0	0	1	Calcul

Figure III.7 - Table de vérité des commandes logiques

Ces quatre signaux sont mémorisés comme précédemment par des bascules "D", commandées par l'information RAFFICH (figure III.8)

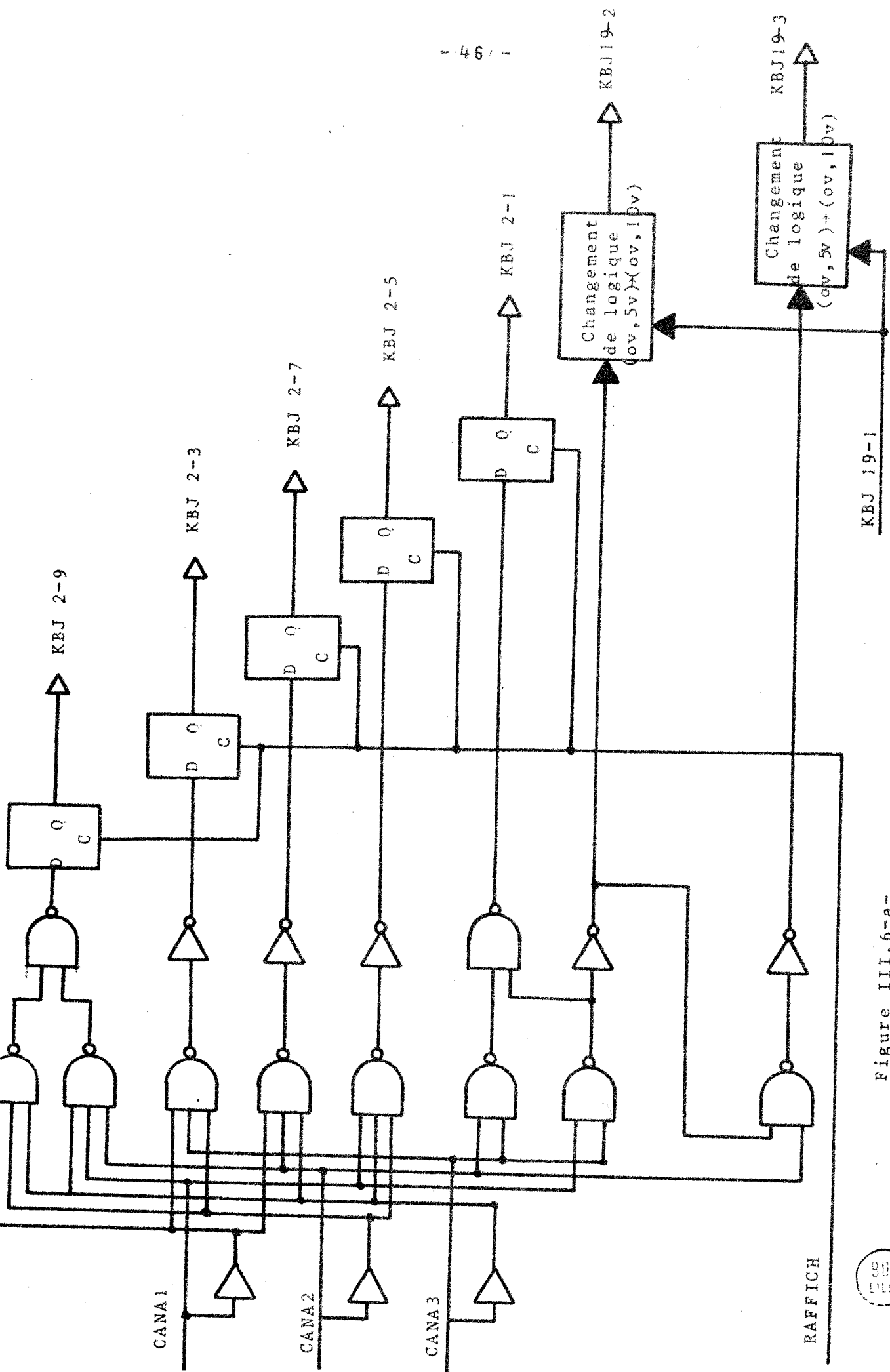


Figure III.6-a-

Schéma logique de la commande des fonctions analogiques.



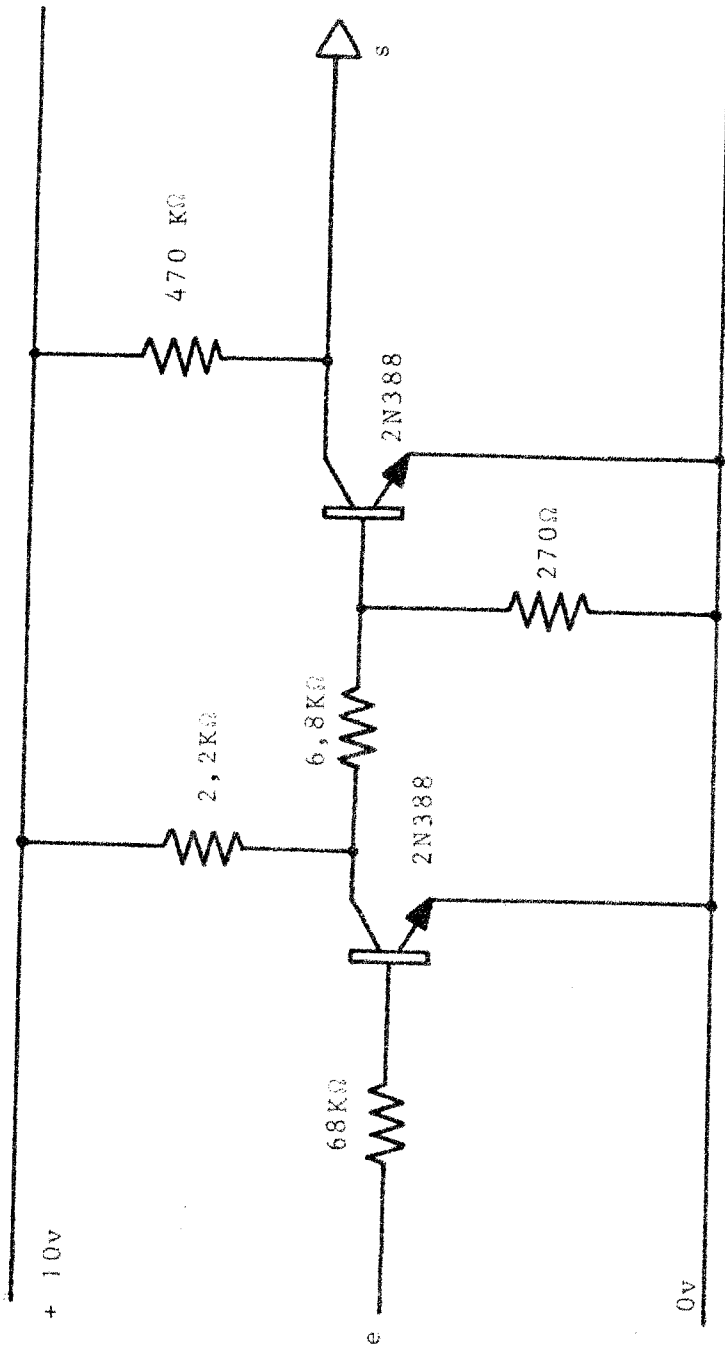


Figure III.6.b
Changement de logique (0v, 5v) en (0v, 10v)



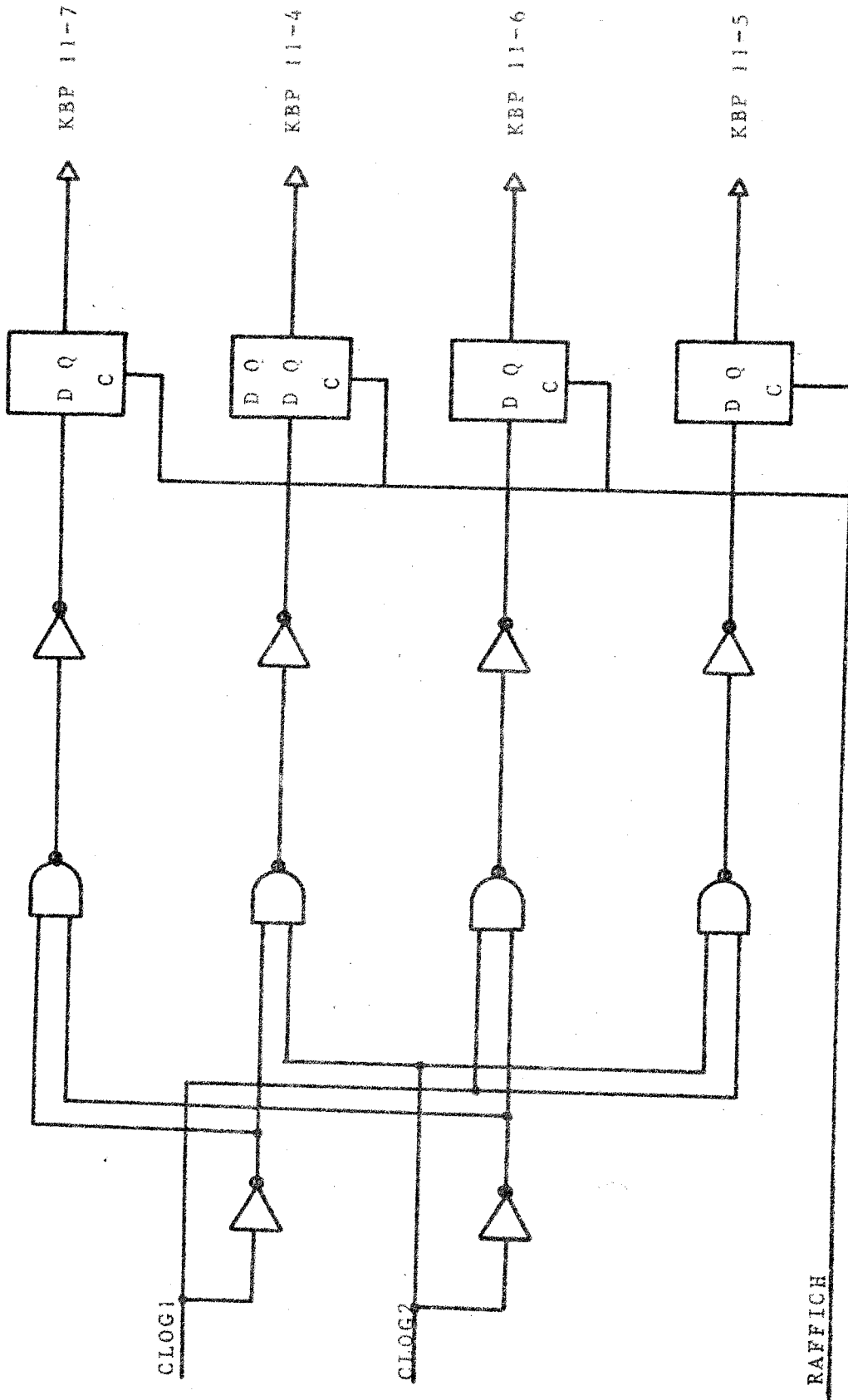


Figure III.8

Schéma logique des commandes logiques



c) Temps d'intégration

La table de vérité est la suivante :

TINTEGR	1 s	32 ms	Commande
0	1	0	1 s
1	0	1	2 ms

Figure III.9 - Table de vérité des constantes d'intégration

La réalisation en est donnée figure III.10

d) Horloge de la logique

Les informations venant du T 2000 sont :

TORLOG 1 et TORLOG 2 ;

TORLOG 1	TORLOG 2	106	105	101	Commande
0	0	1	0	0	10^6
0	1	0	1	0	10^5
1	1	0	0	1	10^1
1	0	1	0	0	10^6



Figure III.11 - Table de vérité des horloges de la logique

La figure III.12 donne le schéma logique correspondant à cette table.

III.1.4 - Le Mot d'Etat

Les informations de commande de l'EAI sont mémorisées dans des bascules, formant un registre fictif que nous appelons "Registre d'Etat du Système". Il nous permet de connaître les différentes fonctions en cours.

Par l'analyse de ce mot d'état, nous détectons les erreurs, dues à des modules défectueux. Une telle



Figure III.10

Schéma logique des constantes d'intégration

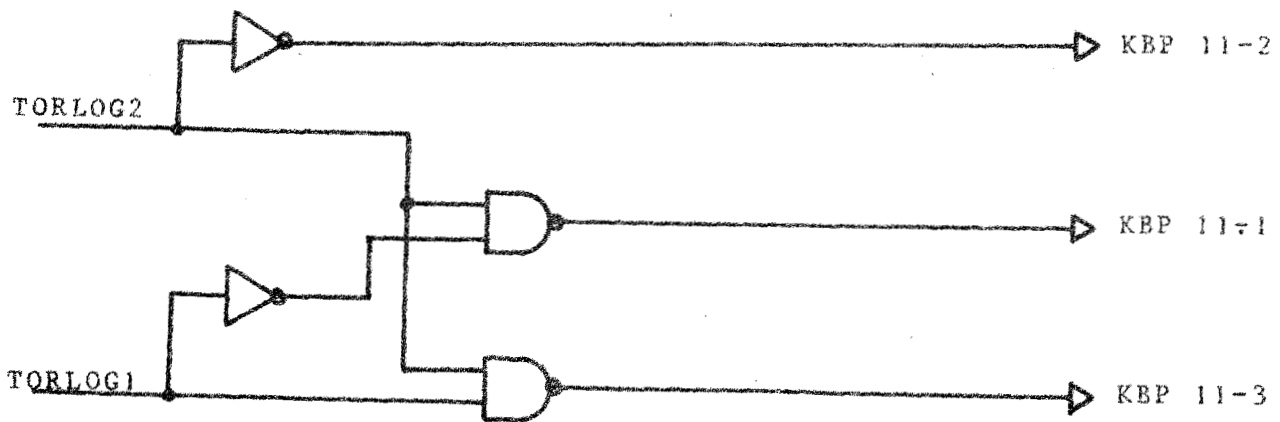


Figure III.12

Schéma logique des horloges



détection entraîne l'émission d'un message invitant l'utilisateur à effectuer le test de la plaque de commande du pupitre.

Le mot d'état comporte 13 informations ; la présence d'un 1 logique, indique la réalisation de la fonction correspondante, sauf le bit n° 9, qui possède deux états significatifs.

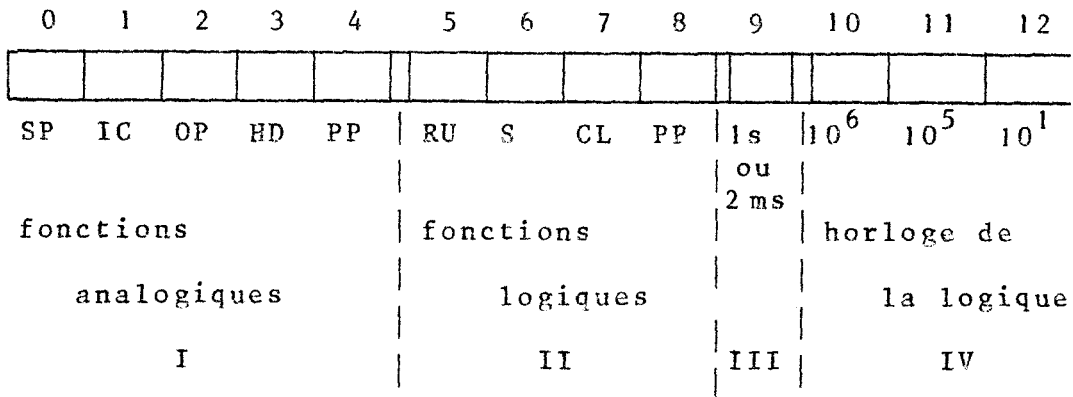


Figure III.13 - mot d'état

L'acquisition du mot d'état est toujours possible. Nous pouvons réaliser un codage permettant de déterminer pour les groupes I, II, IV, un défaut du système de commande, cette information déclenchant une demande de test de la carte de commande du pupitre.

III.1.5 - Conclusion

La réalisation de ce système entraîne un certain nombre de contraintes. En effet, lors d'une utilisation des EAI en local, c'est-à-dire indépendamment du calculateur, nous ne devons pas être influencé par l'interface. Nous avons constaté qu'il était indispensable de pouvoir débrancher certaines lignes afin d'éviter d'éventuels ennuis.

Cette modification a été réalisée à l'aide de relais qui sont commutés une fois pour toute en début d'utilisation grâce à une information venant du calculateur

Cette information notée SELECT a aussi un autre but, celui de la sélection de la calculatrice EAI. Le principe de fonctionnement de ces lignes de coupure est donné figure III.14.

Les différentes lignes coupées sont :

- les lignes des différentes fréquences d'horloge,
- le temps d'intégration
- la durée du répétitif.

Les autres lignes sont directement deconnectées par le bouton poussoir "REMOTE". La réalisation de la commande du pupitre nécessite donc neuf informations venant du T 2000 ; soit un demi-coupleur.

La disposition de ces informations est donnée figure III.15.

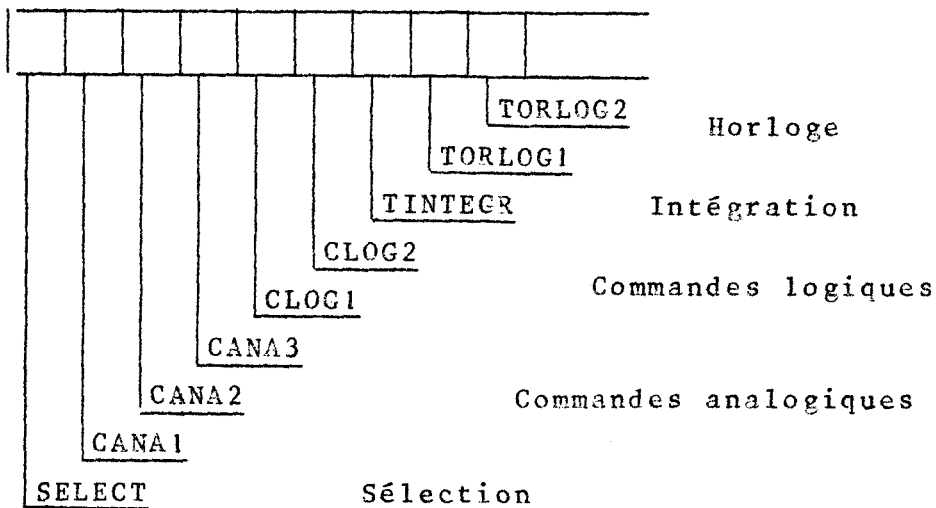


Figure III.15 - Organisation du mot de commande

La conception de cette commande, ainsi que toutes les réalisations ultérieures sont toujours prévues de manière à assurer, une programmation simple et une possibilité de test des cartes logiques.

Le but de notre étude étant la commande intégrale des machines hybrides, il convient de pouvoir piloter individuellement chaque intégrateur.

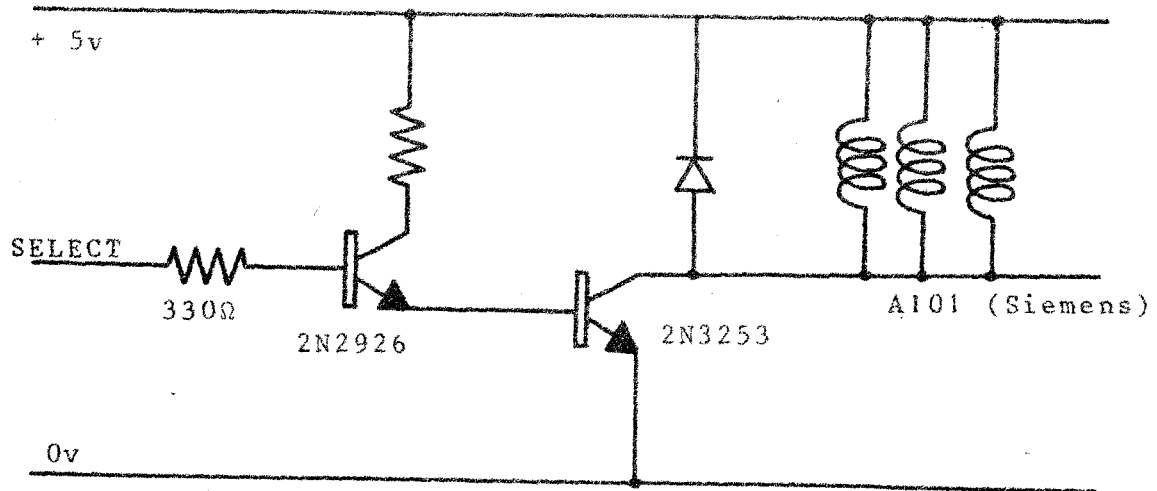


Figure III.14

Information de sélection des calculatrices



III.2 - Commande individuelle des intégrateurs

Il est possible de commander les intégrateurs par la logique parallèle de l'EAI.

En mode normal, un cavalier de bouclage sur le panneau analogique fournit à l'intégrateur, les signaux (OP) bus et (IC) bus qui déterminent le mode de fonctionnement de l'intégrateur. Ces signaux sont donnés par la table de vérité (figure III.16).

Si l'on désire commander l'intégrateur par la logique parallèle, les signaux logiques sont alors envoyés directement sur le panneau avant, et remplacent ceux fournis par le cavalier de bouclage.

IC bus	OP bus	Commande
0	0	Gel des calculs
0	1	Calcul
1	1	Cond. Init.
1	0	"

Figure III.16

Table de vérité de la commande des intégrateurs

III.2.1 - Etude des signaux

Nous ne pouvons pas traiter directement les signaux fournis par les bus (IC) et (OP), au niveau du pupitre de l'EAI, car cela entraîne une commande identique pour les intégrateurs.

Par contre, ces signaux sont délivrés à chaque intégrateur au niveau du module. Il est alors possible de remplacer ces deux signaux d'entrée que nous appelons "ENTIC" et "ENTOP" par des signaux élaborés au niveau de

l'interface ceux-ci, devant jouer le même rôle.

III.2.2 - But à atteindre

Nous devons pouvoir, à partir du calculateur, commander chaque intégrateur individuellement, mais aussi laisser cette commande à l'initiative du pupitre.

Il faut pour cela disposer d'une information qui nous définisse quel est l'organe assurant la commande.

La table de vérité précédente nous indique un état redondant que nous utilisons afin de minimiser le nombre d'informations. Toutes ces commandes sont délivrées simultanément : l'utilisation de bascules "D" commandées comme précédemment par l'information RAFFICH, assure ce rôle.

III.2.3 - Réalisation pratique

Considérons une information "INFIC" jouant le même rôle que "ENTIC" et une information "INFOP" jouant le même rôle que "ENTOP" mais venant du calculateur.

Soit T une information, indiquant la nature de la commande. Pour T = 1, la commande se fera par le pupitre, pour T = 0 par le calculateur.

Nous pouvons donc écrire

$$\text{SORIC} = T.\text{ENTIC} + \bar{T}.\text{INFIC}$$

$$\text{SOROP} = T.\text{ENTOP} + \bar{T}.\text{INFOP}$$

T	INFIC	INFOP	
1	-	-	Commande par le pupitre
0	0	1	Calcul
0	0	0	HD
0	1	0	IC

Commande par le calculateur

Figure III.17a - Table de vérité initiale

On constate par cette table de vérité qu'il n'existe que 4 états pour les variables "T", "INFIC", "INFOP". Elles peuvent donc être formées à partir de deux informations venant du T 2000.

Appelons ces informations "BITIC", et "BITOP", d'où la table de vérité (figure III.17b)

BITIC	BITOP	T	INFIC	INFOP
0	0	1	-	-
0	1	0	0	1
1	1	0	0	0
1	0	0	1	0

Figure III.17b

Table de vérité finale

On peut donc choisir

$$T = \overline{\text{BITIC}} \cdot \overline{\text{BITOP}}$$

$$\text{INFIC} = \overline{\text{BITOP}}$$

$$\text{INFOP} = \overline{\text{BITIC}}$$

d'où

$$\text{SORIC} = \overline{\text{BITIC}} \cdot \overline{\text{BITOP}} \cdot \text{ENTIC} + (\text{BITIC} + \text{BITOP}) \cdot \overline{\text{BITOP}}$$

$$\text{SOROP} = \overline{\text{BITIC}} \cdot \overline{\text{BITOP}} \cdot \text{ENTOP} + (\text{BITIC} + \text{BITOP}) \cdot \overline{\text{BITIC}}$$

En sortie, de ces bascules, nous disposons donc des signaux "MEMOIC" et "MEMOOP".

L'utilisation des calculatrices en local entraîne une restitution des signaux (IC Bus) et (OP Bus) du pupitre.

Un bouton poussoir au niveau de l'interface va permettre de restituer des informations initiales aux calculatrices.

Soit "CHOISI", le nom de ce signal.

Nous pouvons écrire alors :

$$ICDEOR = CHOISI \cdot MEMOIC + \overline{CHOISI} \cdot ENTIC$$

$$DPDEOR = CHOISI \cdot MEMOOP + \overline{CHOISI} \cdot ENTOP$$

Ces signaux élaborés commandent un module d'intégrateurs, c'est-à-dire de 2 intégrateurs.

Le nombre de modules intégrateurs étant de 8, 16 informations venant du T 2000, sont nécessaires à la commande et sont utilisables, soit individuellement (pour le calcul multivitesse), soit simultanément suivant les besoins de l'utilisateur.

L'information "CHOISI" est une information commune à tous les circuits.

La figure (III.18) nous montre la réalisation de ces circuits de commandes.

III.2.4 - Conclusions

La possibilité de commander chaque intégrateur, indépendamment du pupitre, donne au système une puissance de calcul assez importante. Cette possibilité réduit le volume de commande par la logique parallèle ; les décisions de mise en conditions initiales ou de calcul sont fournies par le calculateur à la suite d'un calcul, de tests ...

Cette réalisation permet l'élaboration de boucles de calculs indépendantes ayant la possibilité d'avoir des vitesses variables. Cette organisation offre des avantages précieux et importants dans la réalisation des différents problèmes.

III.3 - Affichage des potentiomètres asservis

Les problèmes de recherche paramétrique, ou de modèles, nécessitent la possibilité de changer les conditions

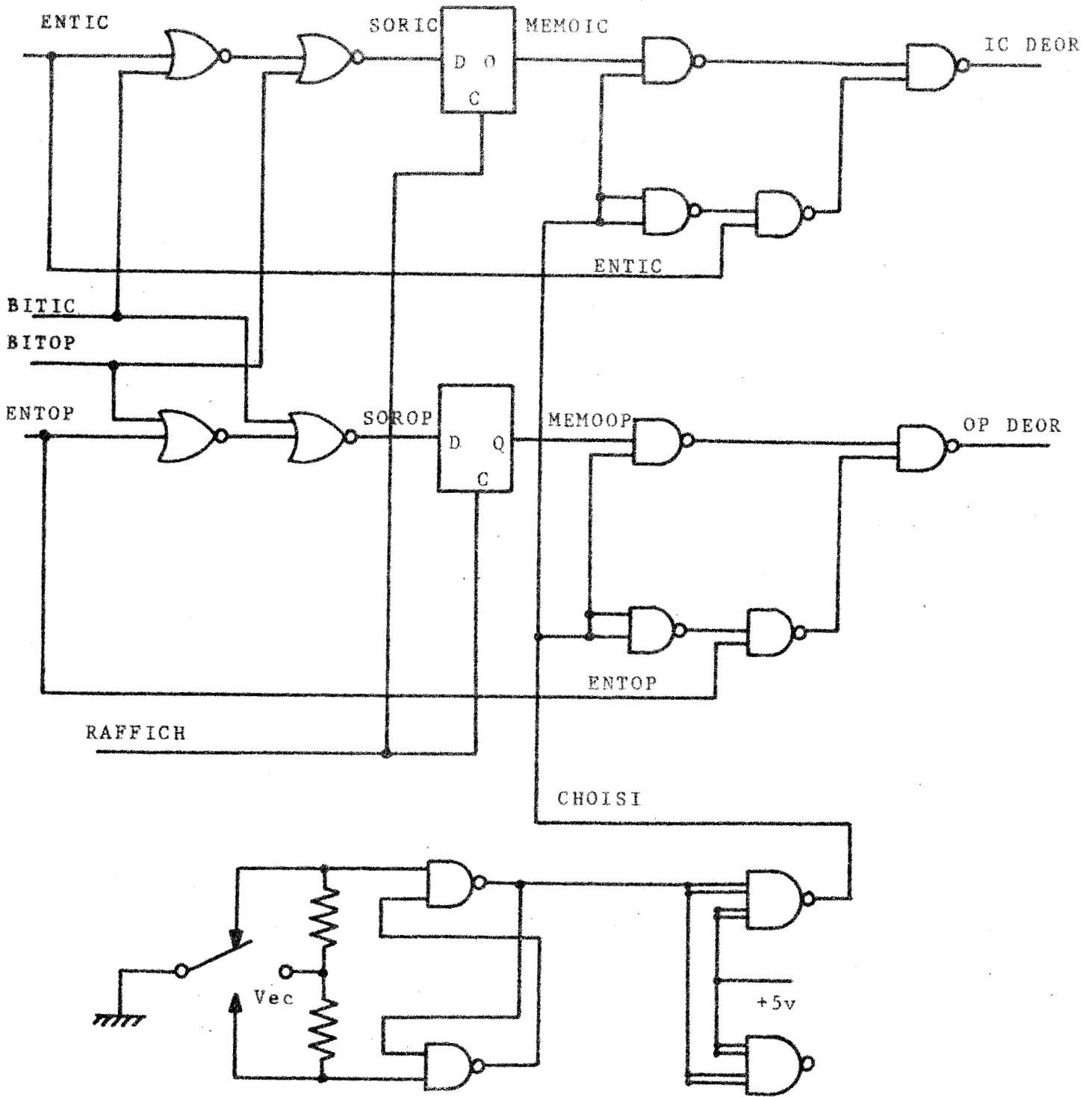


Figure III.18

Schéma logique de la commande individuelle des intégrateurs



initiales et les paramètres de bouclage.

Pour cela, il faut assurer la sélection du potentiomètre, le décodage de la valeur à afficher, puis transmettre cette valeur.

III.3.1 - Réalisation au niveau des EAI

Un potentiomètre ne peut être affiché qu'en mode arrêt des calculs. Cette restriction entraîne un certain nombre de contraintes au niveau de l'utilisation de la calculatrice dans le couplage. Le temps de sélection et d'affichage du potentiomètre n'est pas négligeable. Des essais indiquent qu'en moyenne, il faut 200 ms pour assurer l'opération, soit une fréquence de 5 Hz. Examinons plus en détail la réalisation au niveau EAI, en notant que sur 80 potentiomètres, 70 sont asservis, les 10 autres étant manuels.

a) Signaux EAI

L'affichage d'un potentiomètre, s'effectue en plusieurs étapes. Nous devons d'abord sélectionner l'adresse du potentiomètre (de 10 à 99), décoder la valeur sur 4 digits décimaux, puis afficher cette valeur grâce au bouton poussoir "SET".

Nous avons déjà à ce niveau, la décomposition des différents ordres à effectuer pour réaliser notre commande.

b) Sélection de l'adresse

Une matrice de relais sélectionne le potentiomètre ; son adresse est un nombre de deux digits décimaux. La figure III.19 nous montre l'organisation de ce système.

Chaque ligne des unités est connectée au point haut de 8 relais. Sur chaque ligne de dizaines, la tension est inversée et va au point bas de 10 relais.

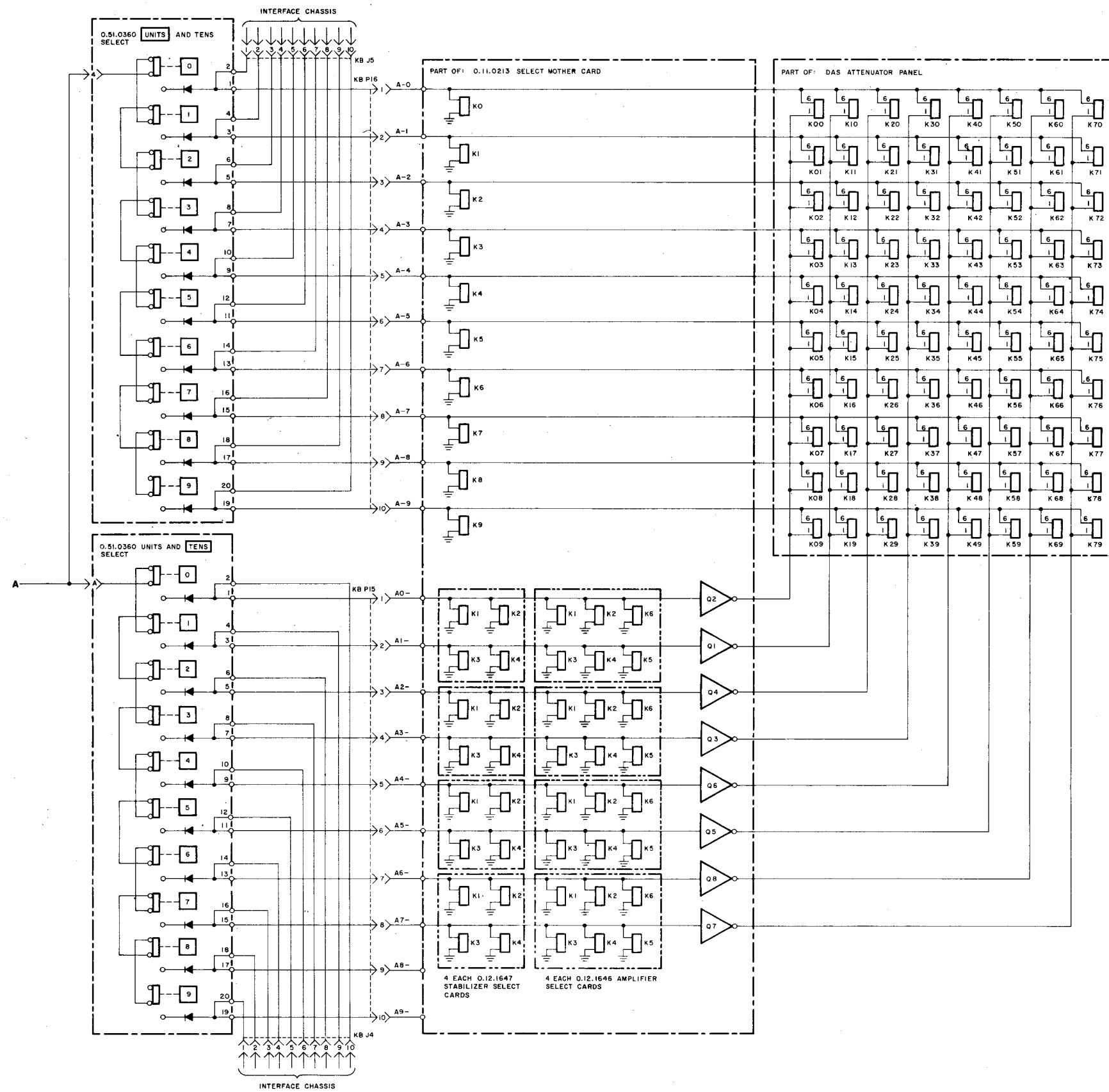


Figure III-19
 Sélection des potentiomètres

BUS
 LINE

Lors de la sélection d'un potentiomètre, sur les lignes d'adresse, apparait une tension de - 20 v. En conséquence, un seul relais à son point haut à - 20 v et son point bas à 0 v ; il est donc excité, et sélectionne un potentiomètre ou un ampli, puisque le même système de décodage est utilisé dans les deux cas.

c) Affichage de la valeur

L'affichage de la valeur se fait par 4 digits décimaux. Un réseau de résistances pondérées délivre un courant proportionnel aux valeurs affichées. Un amplificateur différentiel, transforme ce courant en une tension directement utilisable.

La précision que l'on peut attendre de cet affichage est de 3 à $4 \cdot 10^{-4}$ soit un pas de 3 à 4 mv.

Cette tension connectée à la ligne "SERNO IN" (KBP 18-6) servira de comparaison pour le positionnement du potentiomètre (figure III.20).

d) Positionnement

Le positionnement du potentiomètre est assuré par le bouton SET (figure III.21).

Cet affichage ne peut se réaliser que si SP = 1, c'est-à-dire en arrêt de calcul.

La différence entre la tension du potentiomètre et celle de référence, sert à asservir le moteur. Lorsque les deux signaux sont égaux, sur la ligne KBP 7-9 le signal de 15 volts apparait en début d'affichage, retombe à zéro. Ce signal permet de détecter la fin de résolution (figure III.22).

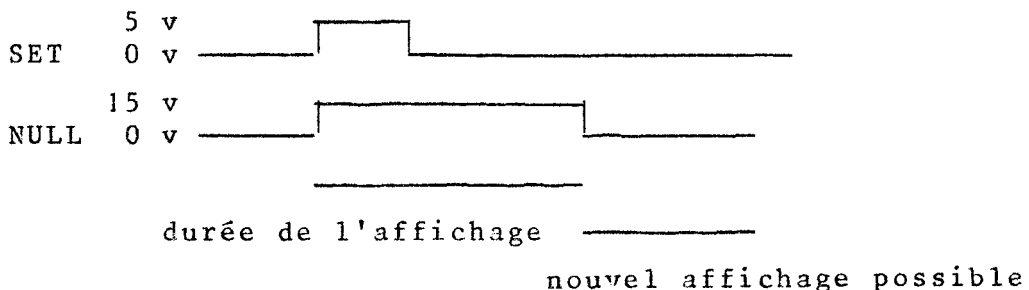


Figure III.22 - Signaux d'affichage

- NOTES:
1. SWITCH SECTIONS ARE MOMENTARY.
 2. a. PIN 4 ON ICP GOES TO DIG GRD
b. PIN 11 ON ICP GOES TO +5V.

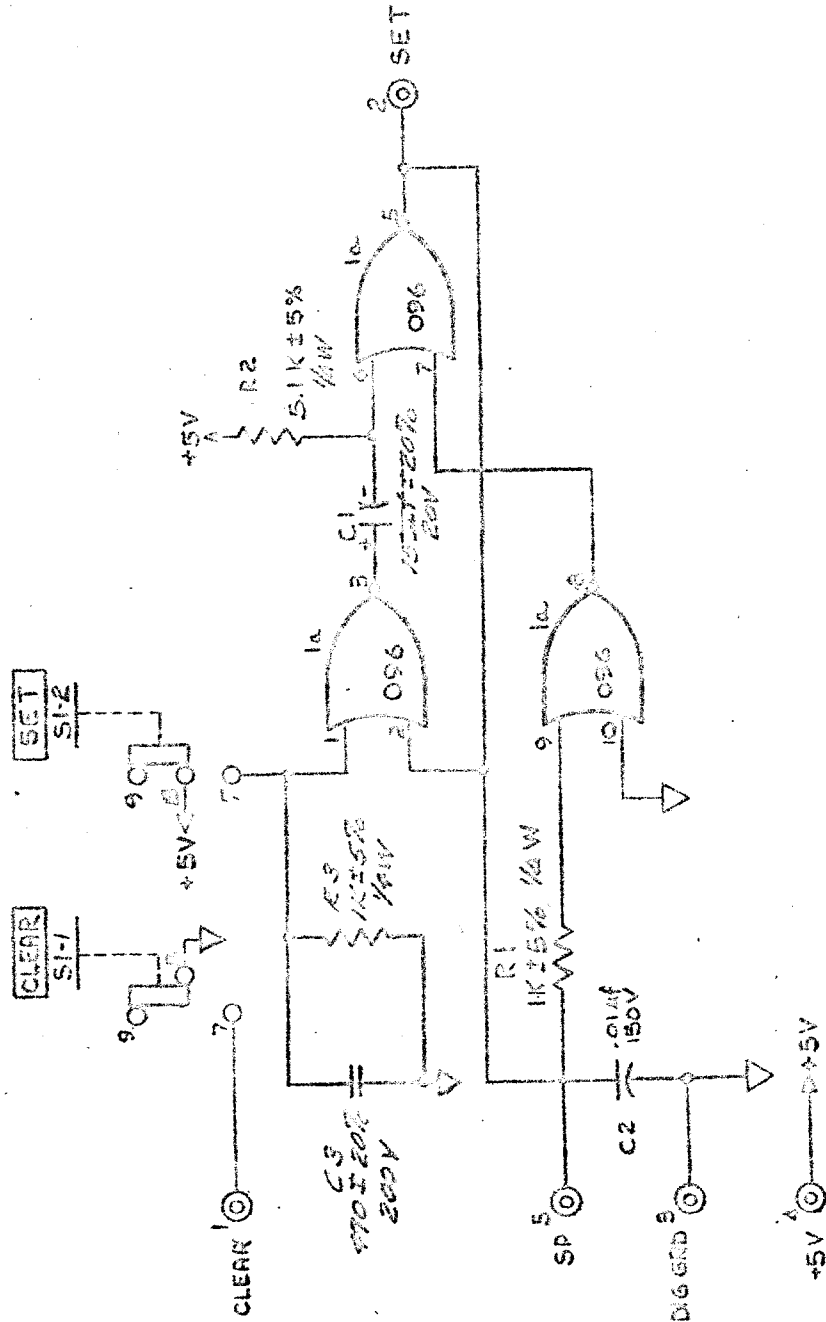


Figure 11.21

Commande d'affichage du potentiomètre.



III.3.2 - But à atteindre

Nous devons effectuer toutes ces opérations de la manière la plus rapide, en assurant la précision la plus grande. Il est nécessaire au niveau de coupleur universel, de transformer la tension de sortie 5 volts en tension de - 20 volts. L'adresse du potentiomètre est sélectionnée par deux informations codées en BCD.

La valeur analogique du potentiomètre est fournie par un convertisseur digital analogique (sorties analogiques du calculateur numérique) à 9 bits + signe, soit un pas de discrétisation de 19,53 mv.

Le signal logique, que nous appelons POTSET, permet l'affichage du potentiomètre.

La retombée du signal sur la ligne "NULL" nous indiquera la fin de l'opération.

III.3.3 - Réalisation

Les opérations que nous devons effectuer pour la sélection de l'adresse sont les mêmes pour les dizaines que pour les unités (figure III.23)

Le changement de logique doit être précis et tenir compte des charges apportées par la matrice de relais (figure III.24).

Nous avons signalé précédemment qu'un convertisseur, assurait le décodage de la valeur à afficher. Cette valeur est transmise à la ligne "SERVO IN" (KBJ 6-8). Le signal "SETPOT", a la durée nécessaire à l'établissement du signal "NULL". Ce dernier pilote l'affichage, et sa retombée déclenche un monostable qui assure la remise à zéro en fin d'affichage (figure III.25).

III.3.4 - Performances

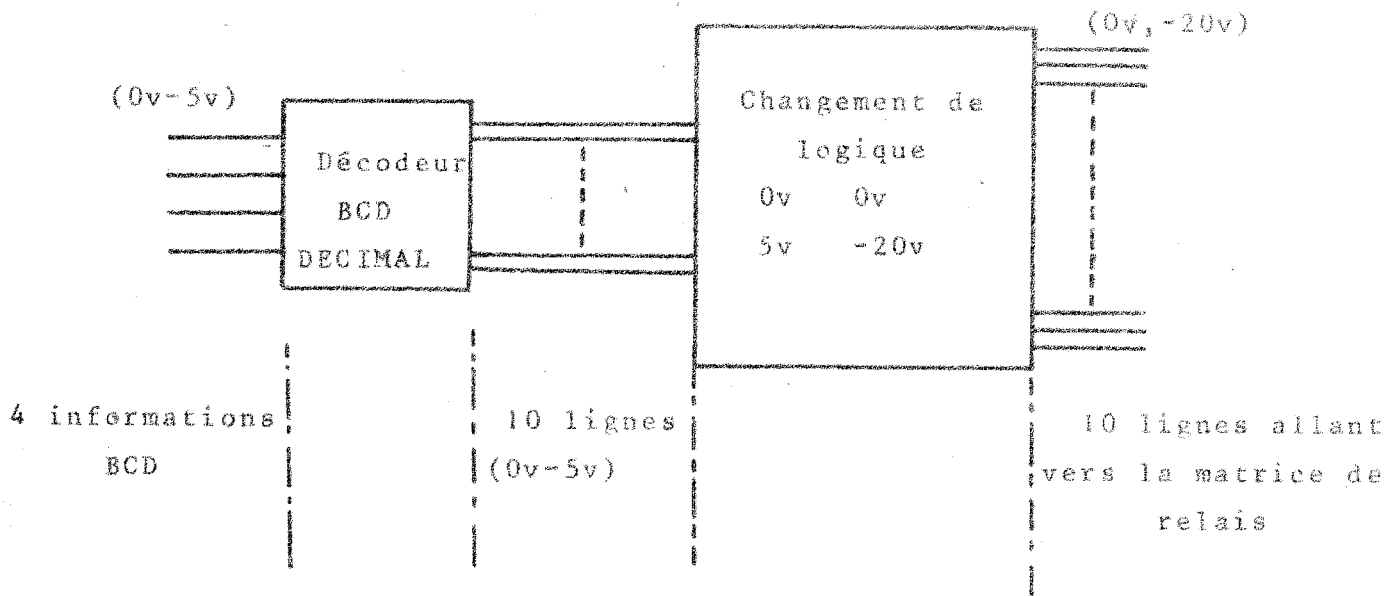
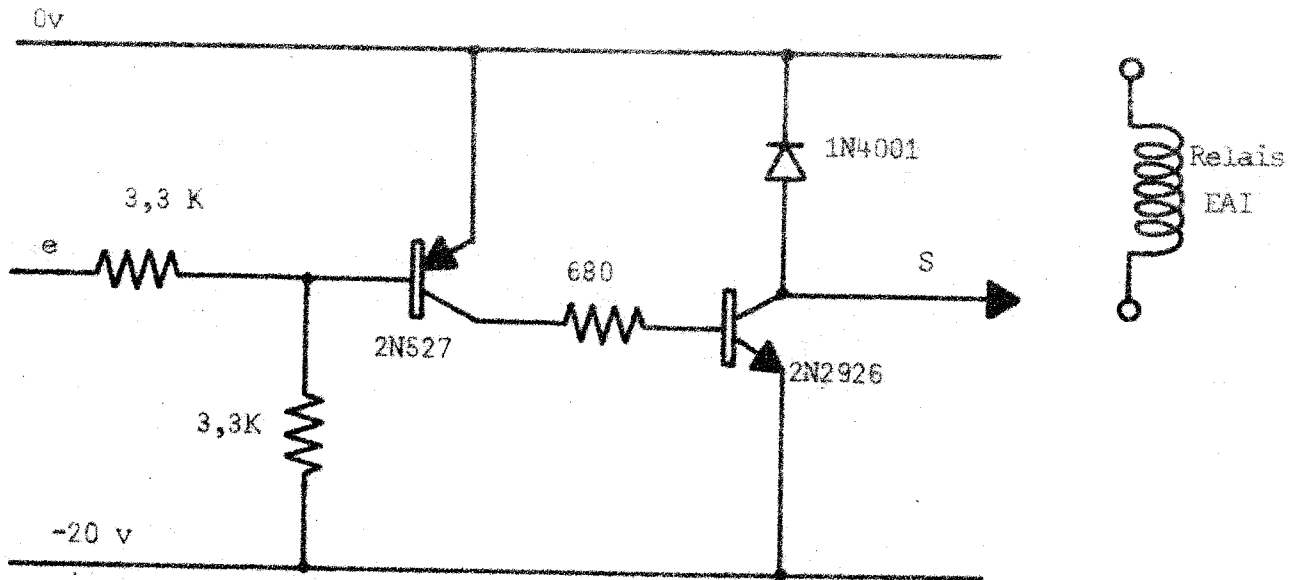


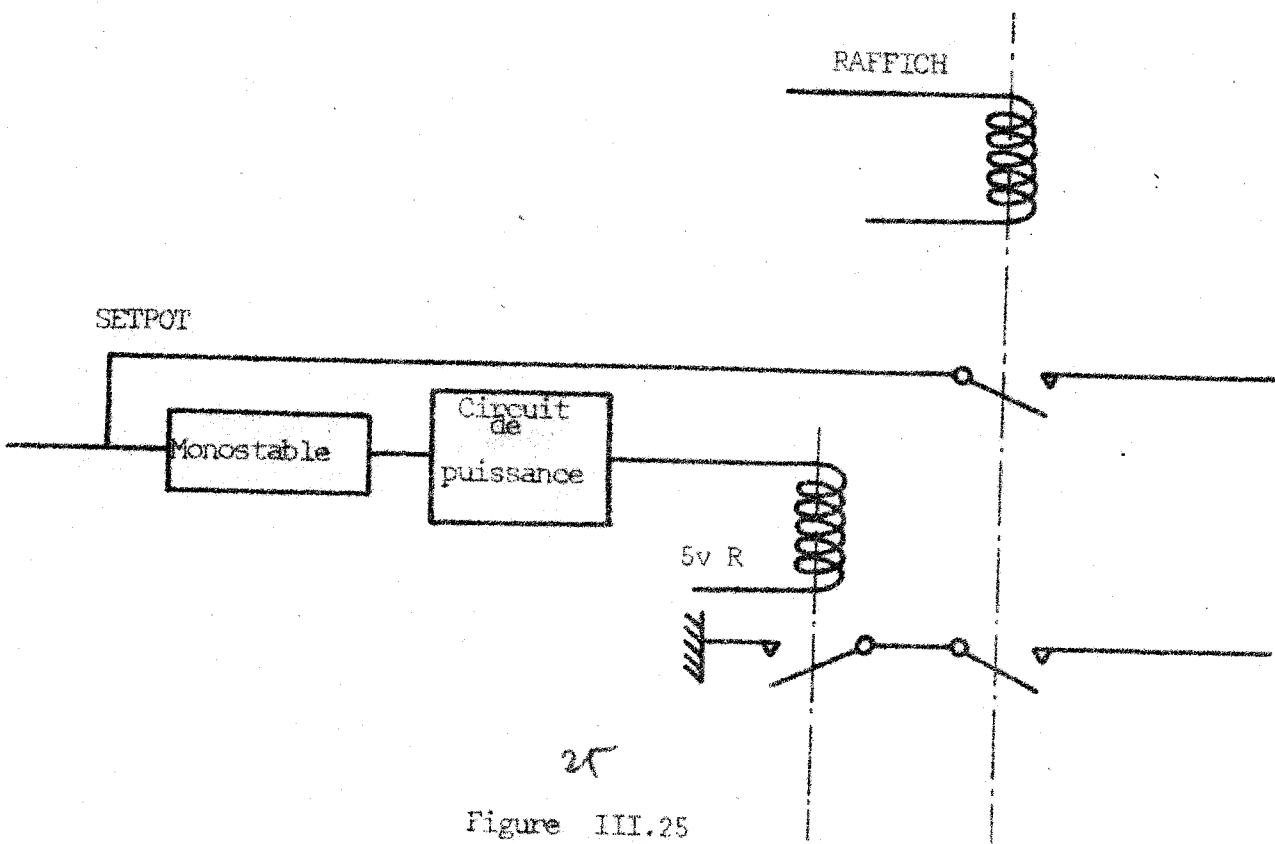
Figure III.23

Sélection du changement de logique





94
Figure III.24
Changement de logique pour les relais



25
Figure III.25
Signaux d'affichage du potentiomètre

L'enchaînement de programmation s'effectue de manière à assurer la plus grande rapidité, mais cette durée est uniquement fonction du moteur d'entraînement du potentiomètre.

La précision du système est celle du convertisseur. Les erreurs que nous introduisons suite à une série de calcul, sont dues au pas de discrétisation du convertisseur. On peut réduire ce pas de moitié en testant au niveau de la programmation, la position du résultat de calcul suivant les valeurs de discrétisation.

Afin d'éviter d'éventuels ennuis, chaque ligne de sélection du potentiomètre a été coupée par un relais commandé par le bit SELECT. Cette précaution, permet d'utiliser les deux EAI, en système maître esclave, indépendamment du calculateur et du couplage.

A chaque affichage, la valeur est relue par la chaîne d'acquisition afin de contrôler si l'opération s'est déroulée correctement.

L'affichage de ces potentiomètres présente une contrainte d'utilisation, car la commande ne peut s'effectuer qu'en arrêt des calculs. Notre intention est donc de réaliser des potentiomètres numériques qui peuvent changer de valeur en cours de calcul, grâce à leur temps de conversion faible.

III.4 - Les potentiomètres numériques

Les potentiomètres numériques que nous envisageons, doivent avoir un temps de conversion faible et une précision de l'ordre de 10^{-3} , pour répondre aux deux utilisations suivantes :

- en potentiomètres de simulation, qui effectuent le produit d'une tension analogique par une valeur digitale,
- en potentiomètres pour l'affichage des conditions

initiales. Des simples convertisseurs digitaux-analogiques suffisent pour résoudre ce dernier point.

III.4.1 - Position du problème

Un certain nombre de contraintes sont à respecter :

- permettre l'affichage d'un potentiomètre entre 0 et 1 avec une précision de 10^{-3}
- effectuer l'opération en 1 μ s
- avoir une compatibilité TTL
- accepter à l'entrée une variation de tension entre -10 volts et + 10 volts.

Il faut considérer la sélection du potentiomètre, donc prévoir une matrice d'adressage des potentiomètres et une sauvegarde du paramètre α , valeur de l'affichage, entre deux commandes. Ces potentiomètres numériques ne doivent pas uniquement servir dans le couplage, mais être commandables en local, comme les autres potentiomètres des EAI.

La valeur du paramètre α , doit pouvoir évoluer dans le temps permettant l'utilisation du potentiomètre en multiplieur. Il suffit de digitaliser l'une des variables, celle dont les variations sont les moins importantes.

Les informations de sortie sont alors échantillonnées, mais à fréquence très rapide.

III.4.2 - Solution envisagée pour le potentiomètre

La solution qui a été retenue pour notre problème consiste en un convertisseur digital-analogique multiplieur - 10 bits.

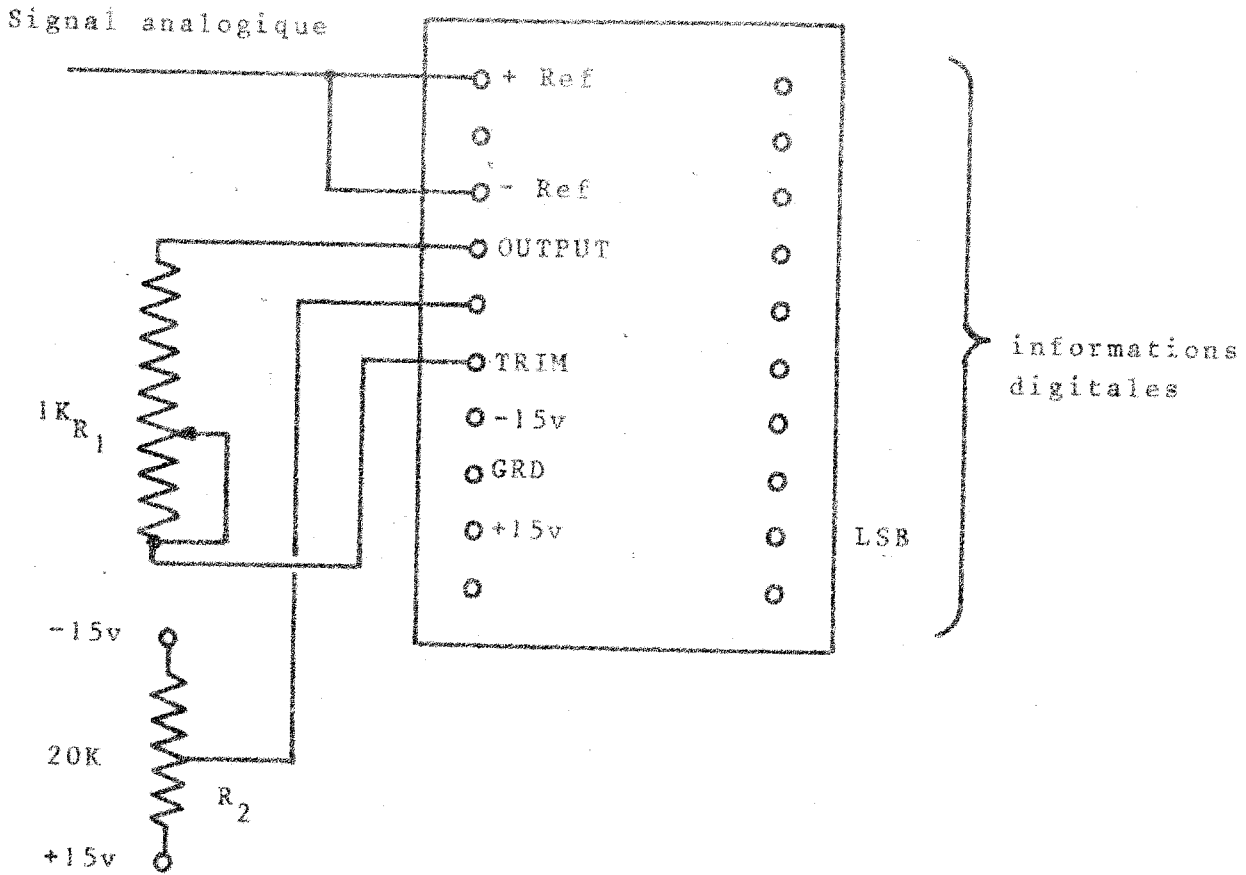


Figure III.26

Convertisseur digital analogique multiplieur

(Hybrid System - DAC 315-10)

La résistance R_2 règle le zéro du système pour une entrée nulle.

La résistance R_1 ajuste la sortie à + 10 v, lorsque le signal d'entrée est de 10 v, et tous les bits d'information à 1.

La vitesse d'évolution de ce système est de 5 v par μ s, la bande passante de 100 KHz à 3db et l'impédance d'entrée de 10 K Ω . Afin de rétablir le signe en sortie, il faut ajouter au système un amplificateur opérationnel.

III.4.3 - Mode de sélection du potentiomètre

Dès maintenant, nous envisageons l'utilisation du potentiomètre à la fois dans le couplage et en local.

Chaque potentiomètre est défini par une adresse qui est celle de son emplacement dans le bac d'entrée-sortie du système Interface.

Le nombre de potentiomètres ne peut dépasser 30.

Le mode de sélection est effectué par un bouton poussoir du panneau de l'Interface, appelé "MODE".

MODE = 1 (enfoncé) sélection par le T 2000

MODE = 0 sélection par le clavier de l'interface

L'adresse fournie par le T 2000, codée en BCD, afin de minimiser le nombre d'informations, est transformée en décimal au niveau de l'interface ; la sélection au clavier se fait directement en code décimal.

Sur un emplacement de carte, arrivent 4 lignes, qui sont les dizaines et les unités de l'adresse, et l'information de mode : l'adresse du potentiomètre est fixée par le câblage de la plaque.

Nous avons donc

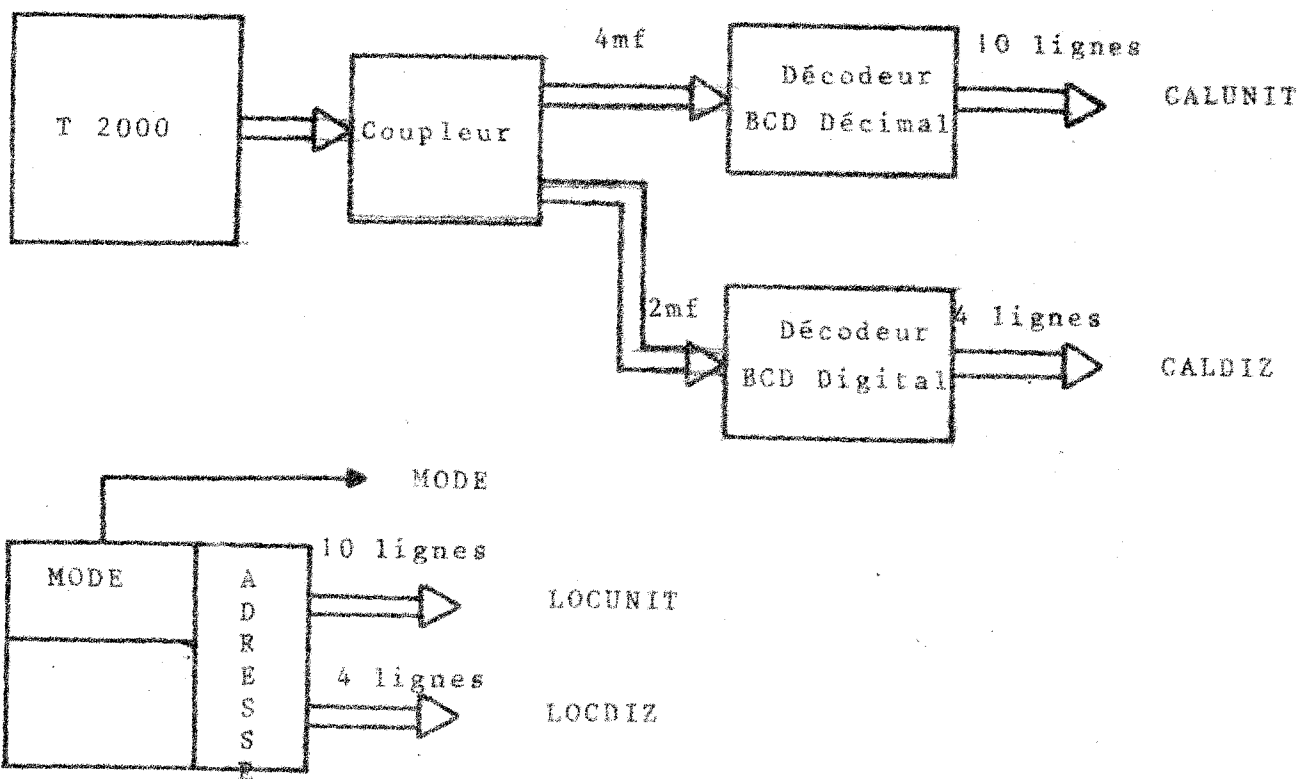


Figure II.27

Signaux de sélection du potentiomètre

LOCUNIT : ligne d'unité de la sélection locale de l'adresse
LOGDIZ : " de dizaine " " " "
CALUNIT : " d'unité de la sélection par calculateur "
CALDIZ : " de dizaine " " " "

L'adresse du potentiomètre s'écrit :

$ADRPOT = (LOCUNIT . LOGDIZ) \overline{MODE} + (CALUNIT . CALDIZ) MODE$

Lors de la commande une seule ligne "ADRPOT"
est à 1.

III.4.4 - Sélection de la valeur

Le bouton "MODE" de l'interface fixe l'origine de la commande. En mode local, l'utilisation du potentiomètre en mode Pas à pas avec une valeur d'incrément fixe, doit pouvoir être faite.

Nous appelons :

"CALVAL" : la valeur venant du calculateur sur 10 bits
"LOCVAL" : la valeur venant du pupitre local sur 10 bits
(au niveau du pupitre, nous supposons déjà les informations de valeur sous forme binaire)
"PASAPA" : un bouton de sélection du mode pas à pas
"INCRE" : bouton d'affichage dans un registre de la valeur d'incrementation
"CALSET" : information d'affichage venant du T 2000
"LOCSET" : " " venant du pupitre local
"PAS VAL" : la valeur d'incrementation du pas à pas
"VALSET" : signal d'horloge du registre tampon du potentiomètre
"POTVAL" : la valeur fournie au registre tampon du potentiomètre

La figure III.2.8 montre le principe du système de sélection de la valeur.

$COMVAL = LOCVAL . \overline{MODE} + CALVAL . MODE$

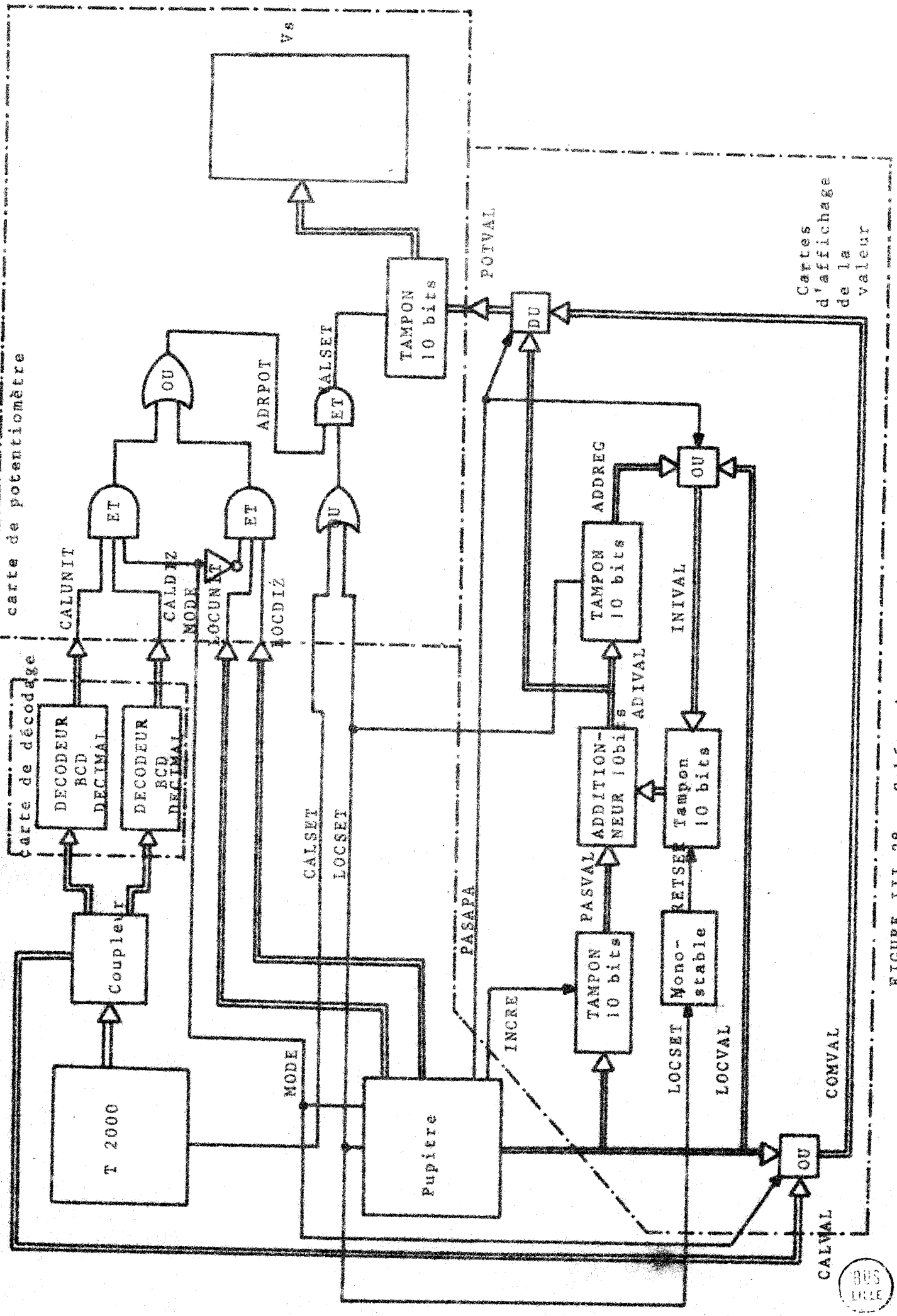


FIGURE III.28 - Schéma des potentiomètres numériques

d'où

$$\text{POTVAL} = \text{COMVAL} . \overline{\text{PASAPA}} + \text{ADIVAL} . \text{PASAPA}$$
$$\text{ADIVAL} = (\text{PREVAL}) + (\text{PASVAL}) \text{ (addition des 2 valeurs)}$$
$$\text{PASVAL} = \text{LOCVAL} \text{ initialisé par INCRE}$$
$$\text{INIVAL} = \text{LOCVAL} . \overline{\text{PASAPA}} + \text{ADBREG} . \text{PASAPA}$$

"ADBREG" prend la valeur "ADIVAL", lors de l'affichage du potentiomètre et, "PREVAL" prend la valeur "INIVAL" grâce au signal "LOCSET", retarde du temps nécessaire à l'affichage du potentiomètre.

On peut remarquer que pour simplifier la logique nous utiliserons des multiplexeurs 2 bits réalisant les signaux "INIVAL", "COMVAL", "POTVAL"

III.4.5 - Affichage de la valeur

Nous disposons maintenant de la valeur du potentiomètre "POTVAL". Cette valeur doit être mémorisée lorsque le potentiomètre est sélectionné. L'affichage est effectué, soit par "CALSET", soit par "LOCSET".

Nous avons donc :

$$\text{VALSET} = (\text{LOCSET} + \text{CALSET}) . \text{ADRPOT}$$

La figure III.28 nous montre la réalisation complète de ce système.

III.4.6 - Conclusions

Le système élaboré fournit un outil puissant, vu la rapidité de travail et de conversion. L'erreur apportée intervient uniquement dans le choix du convertisseur numérique analogique. Le temps de réponse du potentiomètre pour une variation de 0 v est de 7 μ s, ce qui pratiquement, en tenant compte des différentes opérations de sélection, donne une fréquence d'affichage de 100 KHz, donc une fréquence semblable à celle de la chaîne d'acquisition.

III.5 - Acquisitions de données

Nous avons considéré jusqu'ici, la commande logique de tous les éléments de l'EAI. Pour avoir un système complet, il convient maintenant de se pencher sur le problème de l'acquisition de valeurs analogiques par le calculateur numérique.

III.5.1 - Etude au niveau EAI

Sur l'EAI, il existe un dispositif de lecture d'un certain nombre de paramètres, grâce à un voltmètre à cadran et à un voltmètre digital.

Deux lignes spécialisées au niveau du calculateur, "PSEL" et "ASEL", transmettent les valeurs du potentiomètre ou de l'amplificateur sélectionné.

III.5.2 - But à atteindre et réalisation

La lecture des tensions de sortie, doit s'effectuer le plus rapidement possible pour éviter les erreurs dues à une évolution trop sensible du système.

La sélection d'une information par la matrice de relais est trop lente pour assurer une lecture de tensions du système.

Nous disposons d'une chaîne analogique numérique possédant huit voies ; la vitesse de la chaîne est limitée par le système d'acquisition et n'atteint que 8000 voies à la seconde.

Le schéma de principe de la chaîne est le suivant (figure III.30)

Le T 2000, par l'intermédiaire du coupleur envoie et reçoit des informations du contrôleur Haut niveau.

Le contrôleur assure la sélection de la voie à lire. L'échantillonneur à mémoire bloque cette valeur, qui est alors traduite sur 14 bits plus signe par le convertis-

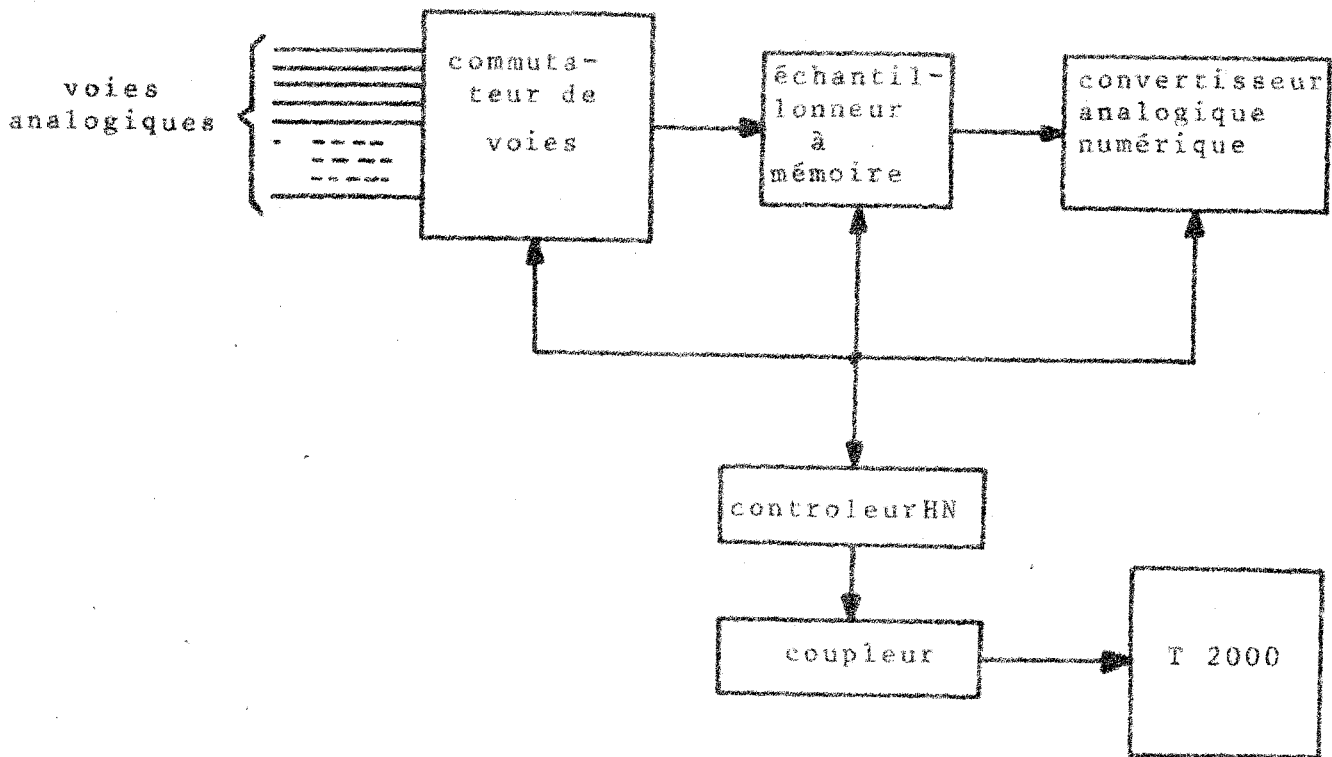


Figure III.30

Principe de la chaîne d'acquisition

seur analogique digital.

La rapidité du système étant grande, le calculateur travaille en mode canal. Il suffit d'initialiser l'échange, et le canal avertit l'unité centrale de la fin de travail.

III.5.3 - Conclusions

Le système envisagé garde toute sa rapidité et la précision attendue est de 1 à 2.10^{-4} . Avec les huit voies dont nous disposons actuellement, le temps d'observation et de conversion total est de 88 μ s. Ceci donne un ordre de grandeurs des variations possibles des signaux à prendre en compte.

III.6 - Etude de la saturation des amplificateurs

Lors de l'étude d'un problème, la saturation d'un amplificateur occasionne des erreurs de calcul, même si cette saturation n'est que momentanée. Ces saturations viennent de boucles initiales ou de divergence de solutions.

Au niveau du pupitre, une série de voyants indiquent les amplificateurs saturés.

III.6.1 - Etude des signaux EAI

Un amplificateur est considéré comme saturé, lorsque sa tension de sortie devient supérieure à 11,5 volts. Sur une ligne que nous notons "SATUR", apparaît alors un signal de la forme suivante (figure III.31).

Ce signal peut devenir négatif, lors de l'arrêt des calculs. Cette ligne "SATUR" est reliée directement à un voyant indiquant si l'amplificateur est saturé (figure III.32)

Dans le même temps, le système élabore sur une ligne appelée "OVLD", un signal dont le niveau bas correspond à une saturation d'amplificateur (figure III.33)

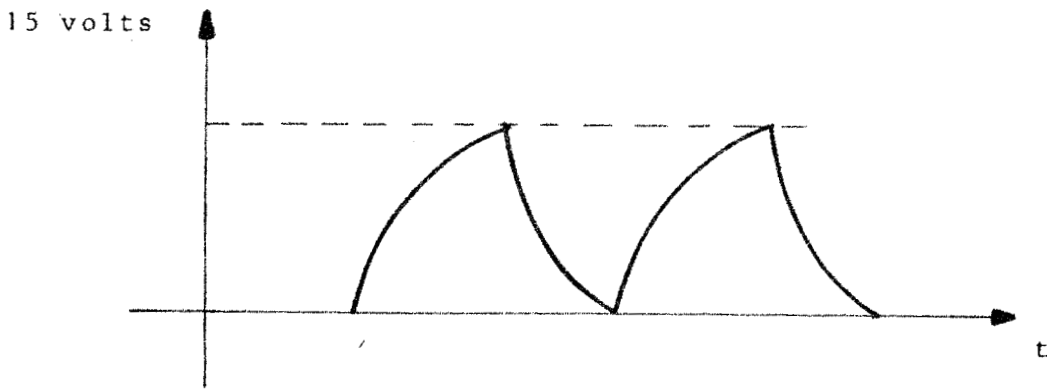


Figure III.31

Signal de saturation

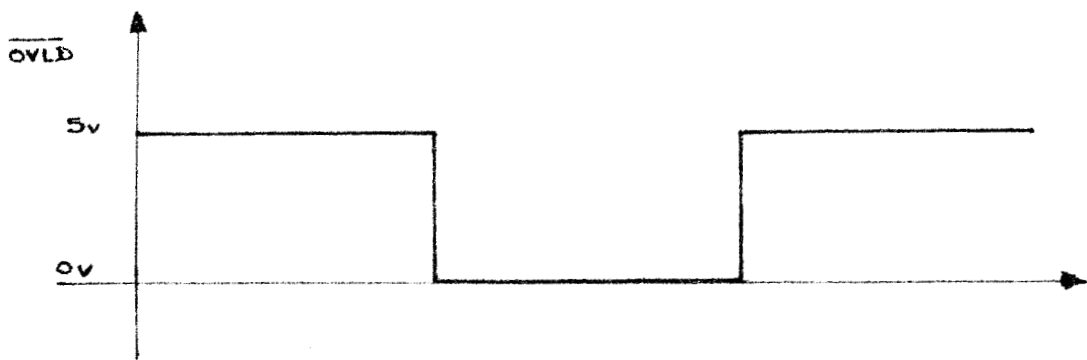


Figure III.33

Signal général de saturation



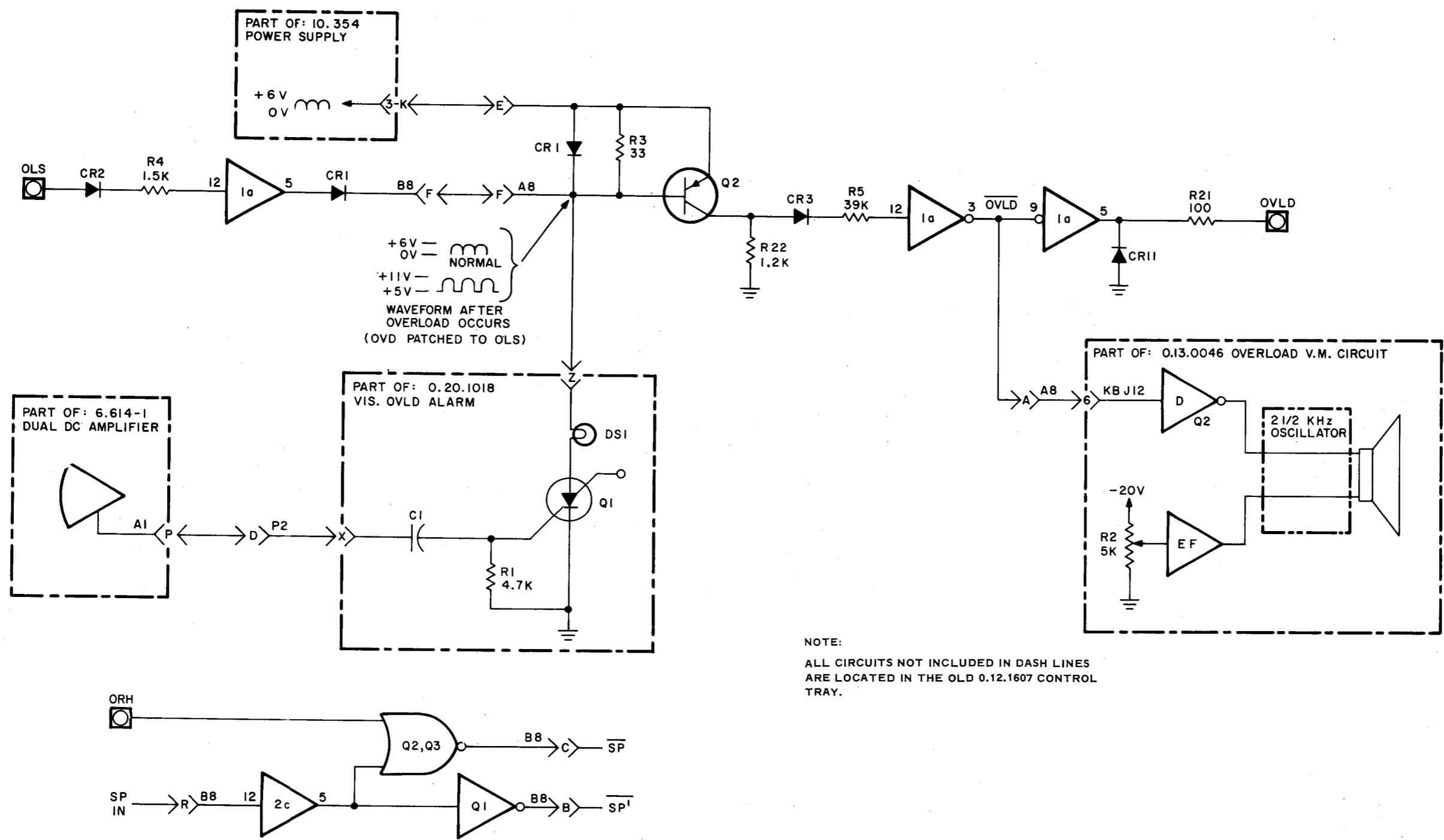


Figure III-32
Saturation



Le signal "OVLD" est commun à tous les amplificateurs.

III.6.2 - But à atteindre

Nous devons être averti le plus rapidement, de la saturation d'un amplificateur, afin de bloquer les calculs en vue d'une vérification du montage. Cet avantage est intéressant dans la mesure où l'utilisateur connaît les numéros des amplificateurs saturés.

Le signal "OVLD" fournit au calculateur, l'indication d'une saturation, en créant un appel prioritaire sur un coupleur universel. Il devient alors possible, après mise en forme des signaux "SATUR", de repérer les amplificateurs qui ont déclenchés l'appel.

III.6.3 - Réalisation

Le signal "OVLD" sera branché sur la borne "Appel" d'un coupleur universel. Le front descendant du signal déclenche au niveau du bac d'entrées-sorties, un appel prioritaire traité par l'unité centrale.

Les informations venant de chaque ligne "SATUR" sont mises en forme, pour avoir un niveau logique 1 lorsque l'amplificateur est saturé (figure III.33)

Comme il existe 80 amplificateurs, le nombre d'informations à lire est considérable. L'emploi de multiplexeurs entraîne plusieurs lectures, avec un nombre d'informations moins important.

Pour des raisons technologiques liées à la réalisation des plaques de circuits, nous adoptons des multiplexeurs à 8 voies.

Nous avons donc quatre plaques identiques pour chacune desquelles 20 signaux sont à traiter suivant une

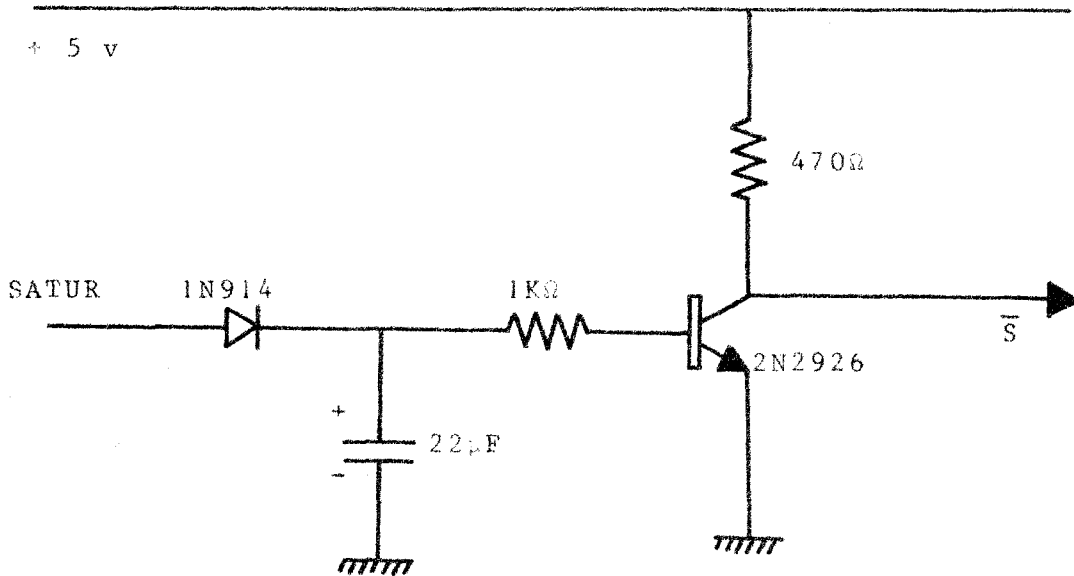


Figure III.33

Mise en forme du signal "SATUR"

répartition de 7, 7, 6 sur chacun des multiplexeurs. Le nombre d'informations de sortie étant de 3, il est nécessaire d'effectuer 7 lectures.

Le montage est donc celui de la figure III.34.

Les trois signaux "ADRES1", "ADRES2", "ADRES3" assurent la sélection d'un certain nombre de voies.

III.6.4 - Conclusions

La répartition des lignes de saturation au niveau des différentes plaques est faite pour assurer un décodage très rapide des numéros d'amplificateurs saturés. L'EAI, n'est

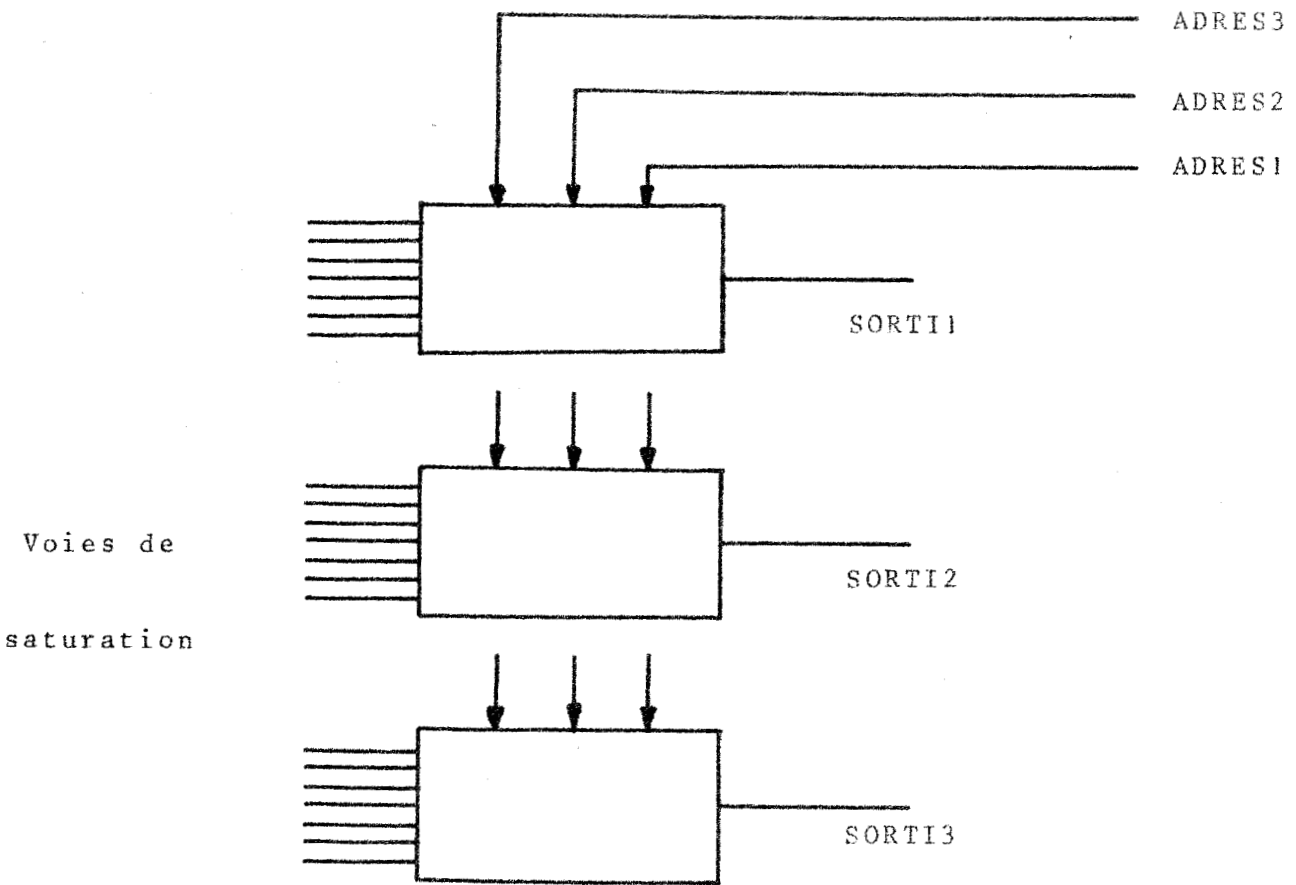


Figure III.34

Multiplixage des voies de saturation



remise en arrêt de calcul qu'après la lecture des informations de saturation. Cette restriction n'est pas gênante, car chaque lecture prenant un temps maximum de 20 μ s, soit un temps global de 150 à 200 μ s, les amplificateurs ne risquent pas d'être endommagés par cette attente.

De plus, lorsque la tension de sortie d'un amplificateur est supérieure à 11,5 volts, celui-ci est mis en position gel

III.7 - Conclusions

Les contraintes de rapidité et de précision que nous avons envisagées pour notre dispositif, ne peuvent être valablement appréciées que sur des exemples. Cela fera l'objet de la dernière partie de ce mémoire. Mais la qualité de l'ensemble hybride ne peut être assurée que si chacune des parties est en parfait état de marche ; nous allons donc considérer dans le chapitre suivant le problème des tests de notre système.

CHAPITRE IV

LES TESTS DU SYSTEME

Pour permettre au système d'assurer complètement son rôle, il est intéressant d'effectuer un certain nombre de tests, tendant à détecter et à localiser les défauts des divers modules.

Cet ensemble de test se décompose en trois parties portant sur :

- Le calculateur numérique
- L'interface
- Les calculatrices hybrides

Cette série de tests n'est pas obligatoire à chaque utilisation, car des vérifications sont faites au niveau du système câblé et dans la programmation des ordres de commande.

IV.1 - Test du calculateur et des périphériques

La Société Télémécanique fournit différentes bandes de test du calculateur et de ses organes d'entrées-sorties. Ces tests doivent être effectués régulièrement afin d'assurer la maintenance du matériel.

Les programmes de tests sont parmi les plus importants :

- Marathon : test des différentes instructions,
- Mémoire : test de la mémoire,
- Canal, test du canal,
- coupleur universel,
- cadenceur,
- voies de comptages.

IV.2 - Test de l'interface

La deuxième étape à résliser, est le test de l'interface, c'est-à-dire la vérification de tous les circuits de commande. Pour réduire le nombre d'informations à lire ou à transmettre, les signaux sont regroupés sur des multiplexeurs.

IV.2.1 - Présentation du problème

Quatre modèles différents de plaques sont à tester :

- cartes de commande pupitre,
- cartes de changement de logique,
- cartes de commande individuelle des intégrateurs,
- cartes de saturation.

Nous avons cherché à minimiser le nombre d'informations nécessaires au niveau des séquences d'entrée et de sortie. Nous adoptons pour cela la structure de la figure IV.1.

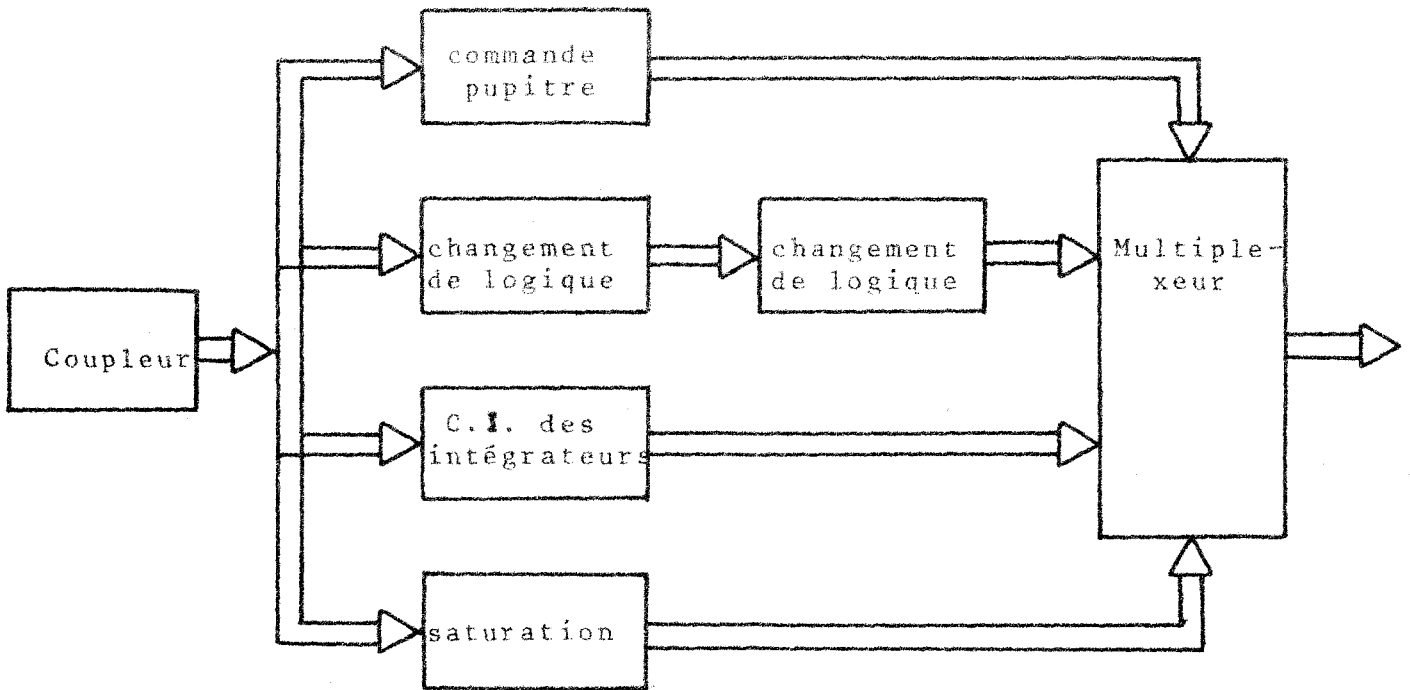


Figure IV.1

Regroupement des informations d'entrées-sorties

Le nombre d'entrées-sorties sur chacune des plaques est le suivant :

	entrées	sorties
Commande pupitre	8	16
Changement de logique	4	10
Commande Individuelle des intégrateurs	16	8
Saturation	20	3

Le multiplexage des informations, n'est nécessaire que pour les signaux de sorties.

IV.2.2 - Réalisation pratique

Des multiplexeurs 4 bits, commandés par les informations "VOIEX" et "VOIEY", sélectionnent les informations. La structure adoptée est donnée figure IV.2.

Elle permet de tester simultanément quatre plaques du système, ce qui entraîne un gain de temps.

Nous avons cherché à déterminer pour chacune des plaques de circuit, la séquence optimale de test à effectuer pour détecter un défaut.

L'analyse du mot de sortie, nous permet de générer, lorsqu'il existe un défaut, diverses séquences, afin de localiser les éléments défectueux. Il n'est pas toujours possible de mettre en cause un élément ; le diagnostic se limite alors à une classe de modules.

Dans l'annexe A, nous indiquons comment nous avons déterminé la séquence optimale de signaux permettant de détecter un défaut et les séquences de localisation du module défectueux.

Un rapport est édité sur l'imprimante et indique les numéros des éléments défectueux, on la classe de module lorsque la panne ne peut être localisée.

Les plaques de sélection de potentiomètre nécessitent un changement de logique. Nous employons une série de relais dont la tension d'excitation est de 20 volts. En sortie l'information est 0 volt ou 5 volts suivant l'état du relais.

Pour effectuer cette série de tests, il faut être certain que les plaques de multiplexage n'apportent aucune erreur. Un emplacement dans le bac de test, leur est réservé, afin de déterminer les opérateurs hors de fonctionnement.

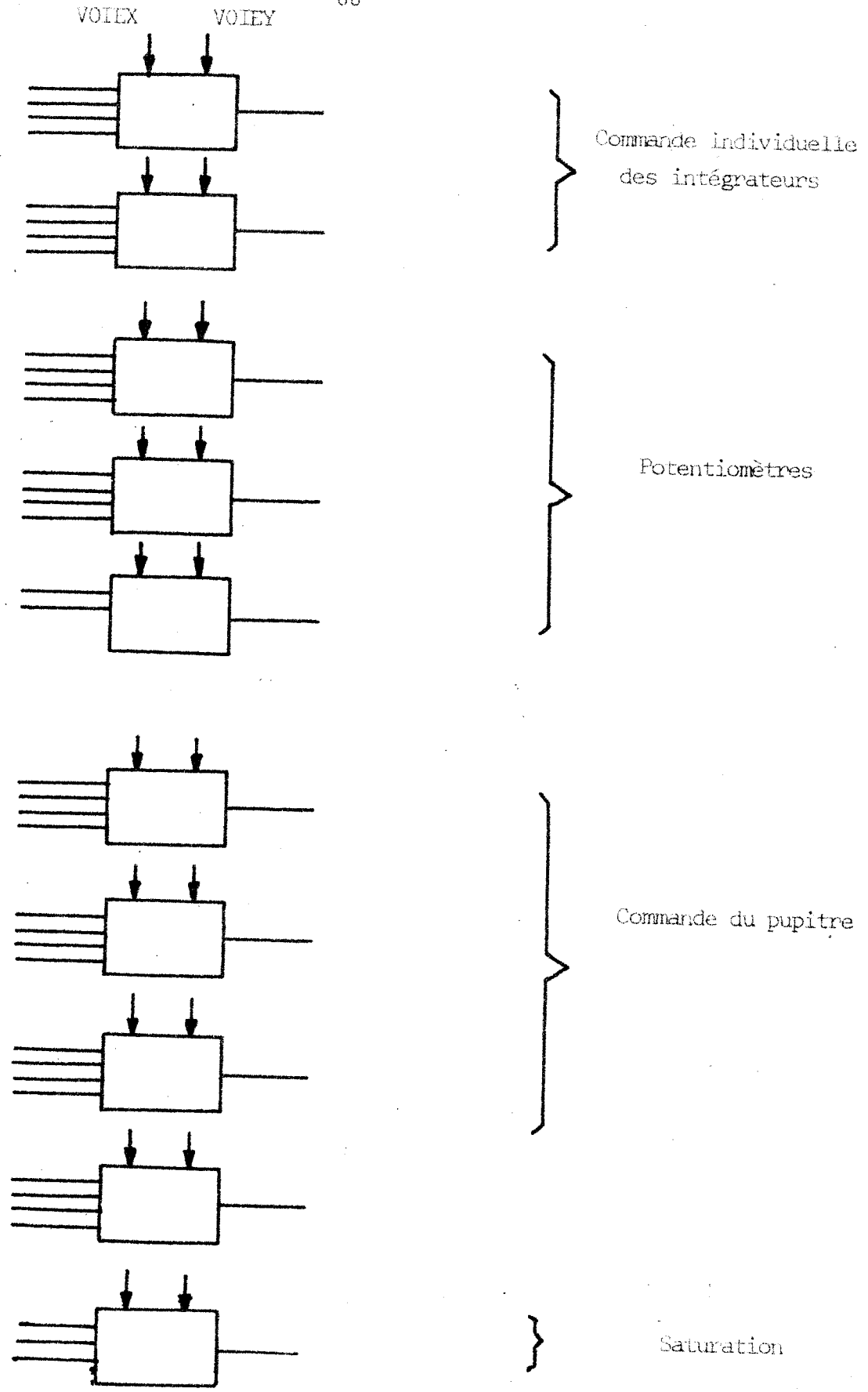


Figure IV.2

Disposition des signaux testés



IV.3 - Test des calculatrices hybrides

Le principe de ce test est simple. Nous analysons tous les amplificateurs afin de s'assurer de leur bon fonctionnement, tant du point de vue analogique que logique.

Les tests effectués sont fonction de la nature du module.

IV.3.1 - Test des inverseurs

Nous vérifions que la tension de sortie est l'opposé de la tension d'entrée, à la précision des appareils de mesure.

Nous admettons un pourcentage d'erreur dû uniquement aux organes d'affichage et de lecture.

IV.3.2 - Test des sommateurs

Sur les sommateurs, nous vérifions la somme de deux signaux d'entrée et le gain de 10.

Nous utilisons le montage de la figure IV.3, qui doit vérifier :

$V_s = \alpha \text{ Ref1} + 10 \beta \text{ Ref2}$, avec Ref1 et Ref2 positive ou négative.

Si l'erreur dépasse la précision des appareils, nous faisons varier les valeurs de α ou de β , afin de déterminer la source de l'erreur.

IV.3.4 - Test des intégrateurs

Les parties analogiques et logique sont à tester pour les intégrateurs.

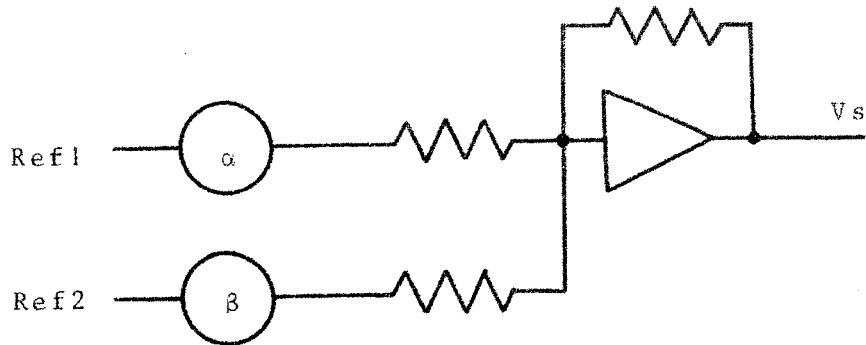


Figure IV.3

Montage du sommateur

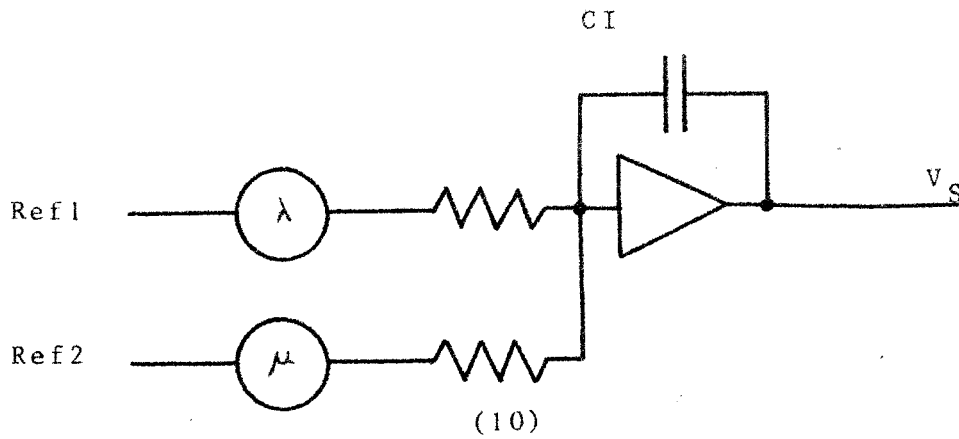


Figure IV.4

Montage de l'intégrateur

Les divers modes de fonctionnement permettent de vérifier la commande du module.

Le test des intégrateurs se déroule en plusieurs étapes :

- contrôle de la sortie nulle avec entre nulle et conditions initiales nulles
- contrôle des conditions initiales avec entrée nulle
- contrôle de l'intégration sur un temps t_0 avec conditions initiales nulles

Nous utilisons le montage de la figure IV.4

La sortie de l'intégrateur prend la forme de la figure IV.5, en fonction de λ et μ .

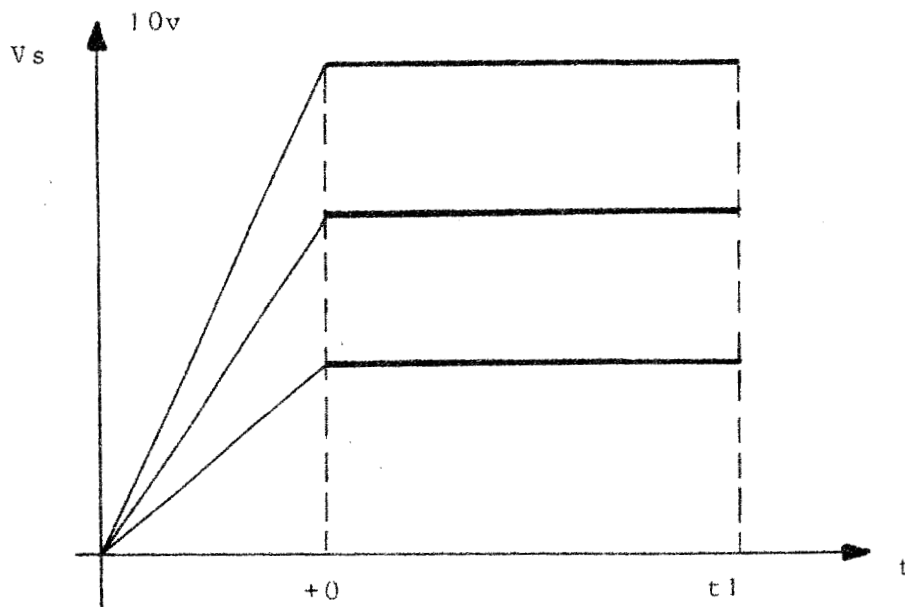


Figure IV.5

$$V_s = -(\lambda \text{Ref1} + 10v \text{Ref2}) t$$

Au bout du temps t_0 , l'intégrateur est placé en position gel. On relève la tension de sortie au temps t_0 et t_1 .

On peut déterminer le pourcentage de perte de l'information, donc la dérive des amplificateurs.

Comme précédemment, lorsqu'un module est défectueux, par variation des paramètres λ et μ , nous déterminons les causes de l'erreur.

IV.3.4 - Test des autres éléments

Ces tests sont réalisés en effectuant les montages suivants :

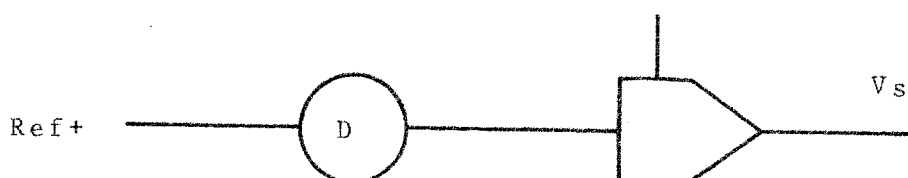


figure IV.6

Les mémoires paires

Nous comparons pour les mémoires paires la tension V_s avec $y \text{ Ref}^+$.

L'information de gel ou de résolution est fournie par le T 2000.

Pour les multiplieurs nous vérifions que

$$V_s = \frac{\text{Ref}_1 \text{ Ref}_2}{10}$$

Les variations de λ et μ , nous permettent d'exploiter le domaine de cette non-linéarité.

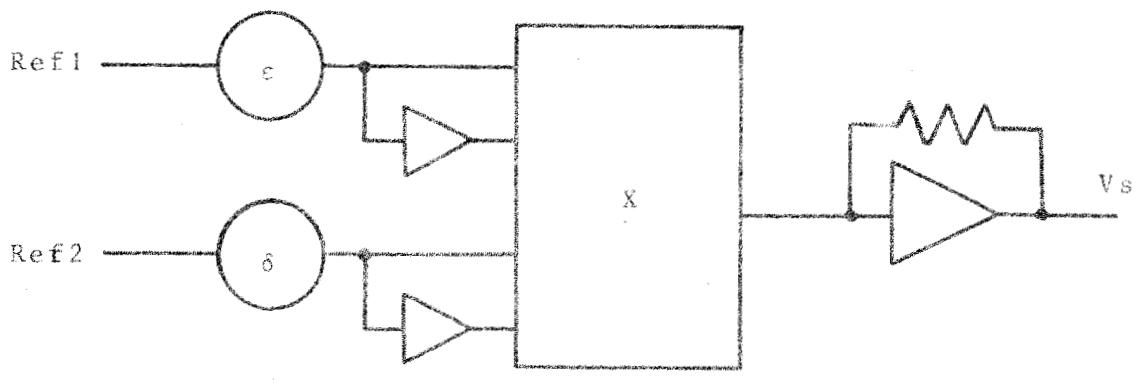


Figure IV.7
Les multiplieurs

Vs est un signal logique tel que :

Ve + a	0	Vs = 1
Ve + a	0	Vs = 0

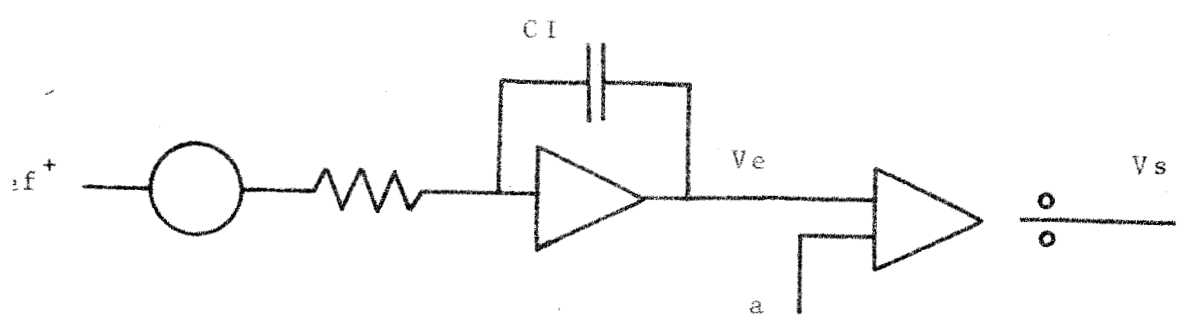


Figure IV.8
Les comparateurs

L'exploitation du domaine est assurée par la variation de la constante "a".

IV.3.4 - Conclusions

Cette série de tests s'effectue de manière continue, grâce à un panneau pré-cablé. Pour éliminer les risques d'erreurs dus aux affichages de potentiomètres, nous ne tenons compte dans cette série de tests que des

informations lues par la chaîne d'acquisition.

Les éléments défectueux sont signalés par message sur l'imprimante.

Après la description de l'interface, notre but est de définir le langage hybride adopté pour notre système. Ce langage a été élaboré afin de permettre à l'utilisateur, une utilisation simple, sans l'obligation de connaître la structure du couplage.

3ème PARTIE

LANGAGE et PROGRAMMATION

LANGAGE et PROGRAMMATION

Nous traitons dans cette partie les méthodes de programmation propres à un ensemble hybride. A partir des données relatives à une liaison calculateur - périphérique, nous mettons en évidence les modules programmés nécessaires à la gestion des échanges, nous permettant d'élaborer le système de base.

Nous ne pourrons éviter par la suite l'emploi des termes de "hardware" et "software". Le premier comprend toutes les fonctions câblées et fixées une fois pour toute, par le second, nous entendons tout ce qui est programmé, c'est à dire la liaison entre l'interface et l'opérateur.

Nous tenons à ajouter que ce chapitre est un résumé d'un exposé plus long présenté dans le mémoire de thèse de Monsieur MEYRIGNAC. Notre but est surtout de donner une vue d'ensemble du travail commun effectué sur le couplage, en insistant surtout sur la partie Hardware qui nous concerne un peu plus.

1 - INTRODUCTION

L'absence de langage permettant de faire fonctionner simultanément la machine numérique et la machine analogique a été initialement un frein au développement du calcul hybride.

Actuellement, un software hybride existe permettant d'utiliser avec efficacité et rapidité l'ensemble des deux calculateurs. Cela présente toutefois l'inconvénient d'éloigner l'analyste et le programmeur de la machine et par là même de ne profiter d'avantages liés à certaines caractéristiques propres aux calculateurs. En résumé, le software hybride peut être comparé à un organe de liaison programmé facilitant la communication entre l'utilisateur et l'ensemble hybride.

Les traits caractéristiques du software hybride répondent aux exigences propres au calcul hybride. Les principales sont les suivantes :

1.1 - Exigences liées à la nature du problème

Dans de nombreux problèmes, s'exerce une contrainte de temps minimal provenant du besoin de communiquer avec un système externe au calculateur numérique qui, lui, travaille sur une base de temps réelle. Le programme hybride doit donc être capable d'utiliser des informations continues ou discrètes provenant de supports externes, de les traiter et de répondre à la fonction demandée dans un intervalle de temps prédéterminé. Il travaille en temps réel.

1.2 - Exigences liées à la préparation d'un programme

Il est d'une grande importance que, quel que soit le langage de programmation utilisé, le programme hybride fonctionne comme une entité. Sa fonction est d'adapter le mode d'opérations séquentiel du calculateur numérique aux propriétés parallèles du calculateur analogique. Il doit donc réaliser

la synchronisation des opérations analogiques et numériques, l'optimisation du temps de calcul nécessaire, la communication de données et des informations de commande entre les programmes analogique et numérique.

1.3 - Exigences liées aux procédés opératoires

Le calculateur hybride, d'après sa conception offre des possibilités de dialogue avec l'opérateur. Le software hybride doit donc tenir compte de la possibilité d'intervention de celui-ci.

Enfin une caractéristique importante est de posséder des programmes de diagnostic permettant de vérifier les opérations de calcul de l'ensemble et plus spécialement les opérations analogiques.

On peut donc décomposer le système hybride en plusieurs classes : les langages de programmation, les systèmes de commande, les programmes de bibliothèque, les programmes utilitaires, les programmes de mise au point et de vérification.

En fait, le software hybride a beaucoup de points communs avec un software numérique classique. Le langage de programmation, les programmes bibliothèque et une partie du système moniteur peuvent ne pas être spécifiques à l'ensemble hybride.

2 - LE SOFTWARE

Nous définissons ici la structure du système d'exploitation de plusieurs machines, c'est à dire les pro-

grammes de base à inclure pour faciliter les échanges et diminuer le nombre d'opérations à exécuter.

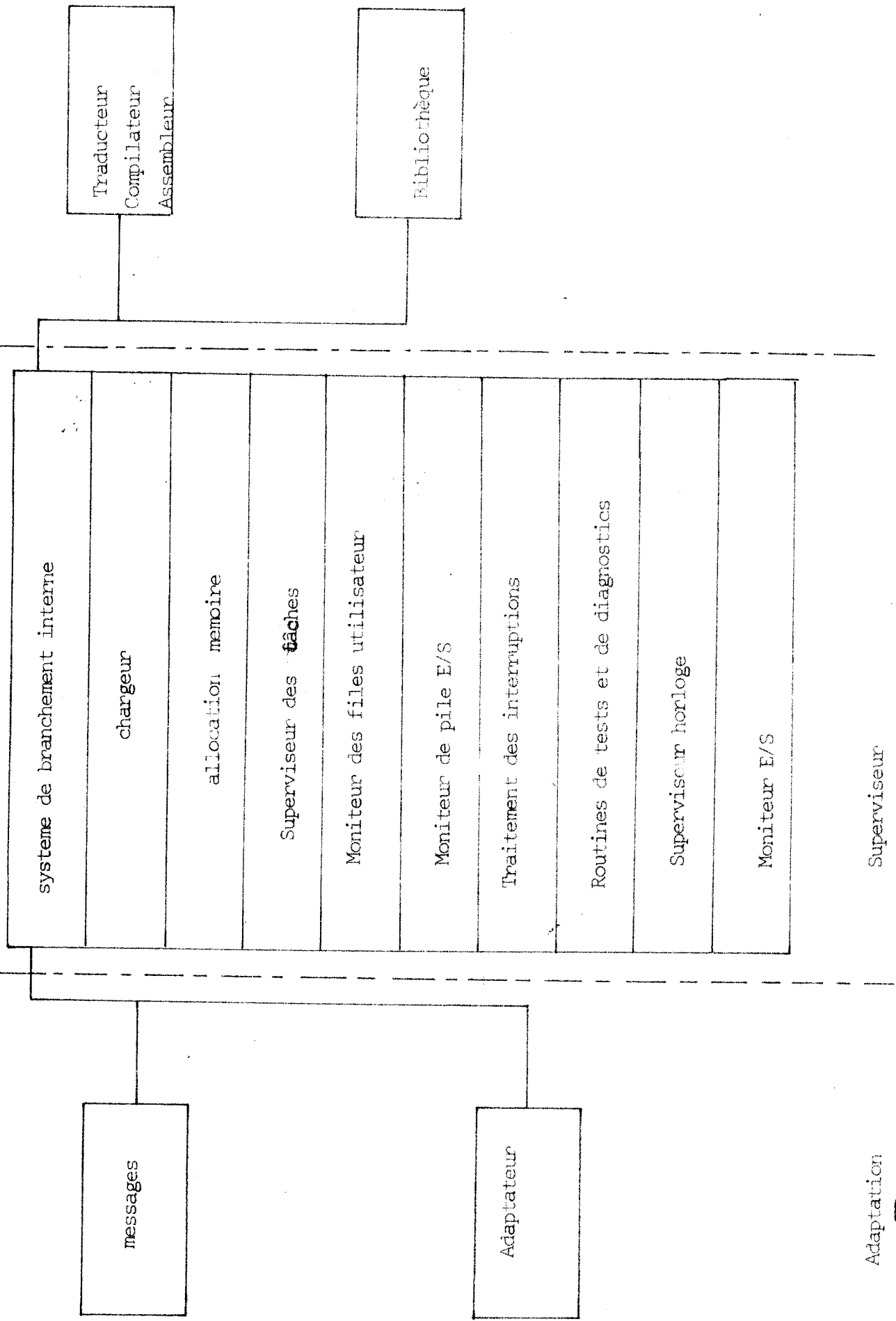
Les programmes doivent assurer la liaison des différentes unités physiques entre elles par l'intermédiaire du calculateur central. Ils comprennent les moniteurs, les programmes de tests, les traducteurs et permettent le traitement de terminaux en mode conversationnel. Ils sont complétés par des moniteurs temps réels, et des superviseurs contrôlant les transferts et traitant les interruptions externes.

2.1 - Les moniteurs

Chaque moniteur a un double but : d'une part fournir à l'utilisateur les programmes d'exploitation du système, d'autre part régler l'enchaînement des travaux. Il est en outre possible de lier plus étroitement le hardware et le software en contrôlant les erreurs de programmation, pour protéger le moniteur, aider à la mise au point des programmes, et les isoler les uns des autres (dans le cas d'un fonctionnement en temps partagé), tout en prévenant l'opérateur de l'état interne du système.

Nous pouvons diviser l'ensemble du système en trois zones suivant la figure 1 : les programmes annexes (édition de messages, lecteur, interpréteur...), les moniteurs et les programmes de traitement.

Plus particulièrement, le moniteur temps réel traite les échanges sur les périphériques rapides et s'intègre dans le système comme programme séparé, c'est à dire résident en mémoire sans le contrôle d'un superviseur. Il traite toutes les interruptions et permet de déterminer les priorités, l'initialisation, l'achèvement des travaux ainsi que les temporisations, les mises en attente etc....



Adaptation



FIG. 1 - Le noyau du systeme

2.2 - Programmes spécifiques aux unités physiques

Ces programmes sont généralement des moniteurs non résidents, appelés d'un disque ou de bandes magnétiques par le superviseur principal ou le moniteur temps réel. Ils se réduisent le plus souvent à des fonctions de lecture ou d'écriture, sur des supports externes, ou à quelques traitements particuliers sur une unité physique déterminée.

2.3 - Programmes de test

Cet ensemble est utilisé pour l'entretien et, dans un contexte hybride, pour contrôler automatiquement la calculatrice analogique. Il permet de mettre en évidence trois types d'opérations : la simulation du système, la détection des erreurs et de leur origine.

2.4 - Les traducteurs

Ils permettent de passer d'un langage haut niveau à l'assembleur ou au langage machine. Il opère en mode conversationnel ou analyse globalement un ensemble d'instructions formant un tout. Il est particulièrement intéressant de les utiliser sur un système hybride, notamment pour la sélection des différents modes analogiques, l'affichage des potentiomètres etc...

3 - METHODE DE PROGRAMMATION

Les calculateurs analogiques sont couramment utilisés pour la simulation des systèmes continus. Chaque type

de machine a ses qualités propres, la partie numérique ayant néanmoins l'inconvénient d'opérer en séquence, et de permettre difficilement le dialogue homme - machine, la partie analogique étant limitée en précision uniquement par la technologie employée. L'ensemble hybride combine dans la mesure du possible les avantages des deux machines.

Une simulation hybride peut se dérouler en cinq phases schématisées dans la table de la figure 2.

La première étape est valable pour tous les types de simulations et quelque soit le système ; la deuxième donne quelques indications sur la variété des décisions à prendre et la complexité qui peut en résulter, mettant à l'épreuve l'esprit de décision de l'utilisateur. En effet, au niveau de la réalisation d'une simulation, le programmeur doit surmonter un certain nombre de difficultés :

- Il doit choisir sa solution parmi plusieurs combinaisons dues aux différentes possibilités du calculateur. Il doit tenir compte des contraintes de précision nécessitées par sa propre étude et imposées par les calculateurs.

- Il doit comprendre et traiter les problèmes de synchronisation et de temporisation dans le cadre d'un fonctionnement en temps réel.

- Il doit traiter les procédures nécessaires à la préparation de son travail d'une façon plus importante que s'il opérait d'une façon purement numérique.

C'est ce troisième volet que les routines d'aide à la programmation tentent de réduire au maximum.

Différentes phases de résolution d'un problème

à partir d'un système hybride

Ceci ne représente qu'un essai de classification et la structure des différentes étapes n'a rien d'impératif.

1 - Définition du problème

- a - Equations du système
- b - Etude des paramètres
- c - Répartition des tâches analogique et digitale

2 - Préparation hors ligne

- a - Mise à l'échelle des équations
- b - Allocation des éléments sur le panneau analogique
- c - Détermination des liaisons avec l'interface
- d - Développement des programmes numériques.

3 - Préparation en ligne

- a - Cablage
- b - Chargement des programmes sur l'ordinateur
- c - Entrée des paramètres

4 - Déroulement de la simulation

- a - Evolution des paramètres
- b - Diagnostics.

5 - Résultats

- a - Evaluation du modèle
- b - Détermination du procédé de calcul
Retour en phase 2 ou 3 si nécessaire.

Aucun système hybride existant ne présente réunies toutes les qualités décrites précédemment et en ce sens n'est un "bon système". Cependant, l'évolution générale laisse prévoir des langages spécialisés de plus en plus performants ainsi qu'une certaine normalisation.

Après la description des caractéristiques essentielles d'un système hybride, il est possible de déterminer les éléments de base de notre ensemble. Celui-ci, dans son état actuel, ne présente évidemment pas toutes les caractéristiques souhaitables mais nous nous sommes efforcés lors de son élaboration de permettre un développement ultérieur.

4 - LE MONITEUR

4.1 - Description du moniteur

Le moniteur gère l'exécution de 12 travaux se déroulant en parallèle. A chaque travail, sont associées une priorité et une zone d'implantation mémoire, le moniteur utilise les temps morts lors des entrées - sorties avec les périphériques lents pour activer les différentes tâches. D'autre part, certaines fonctions spécifiques au moniteur sont traitées comme des programmes ordinaires, mais ont une priorité fixée une fois pour toute, généralement haute, leur permettant de modifier l'état du système.

1.a - Notion de travail

Un travail est caractérisé par quatre états à priori différents : actif, inactif, inhibé, en attente (d'un

périphérique, d'une fin d'échange, d'une condition etc...).

- Travail actif

A un instant donné, un travail et un seul est actif, c'est à dire qu'il occupe alors l'unité centrale en développant une série d'instructions (calculs, tabulation etc...) qui ne nécessitent aucun périphérique. Ce travail peut d'ailleurs être un des modules spécifiques au moniteur.

- Travail inactif

Un travail inactif occupe de la place en mémoire et peut être activé à tout moment sur demande de l'opérateur par exemple, ou d'un autre travail. Sa présence est connue du moniteur en ce sens que ses caractéristiques sont définies (priorité, emplacement mémoire....) et qu'il lui est donc impossible d'implanter un autre programme sur sa zone mémoire, qui est considérée comme "gelée".

- Travail inhibé

Tout travail peut être inhibé quelque soit son état. Dans ce cas, toute demande ou tout appel à ce travail ne sera pas satisfait et provoquera l'inhibition du travail demandeur, le déroulement normal de celui-ci étant perturbé. Il est impossible d'inhiber une tâche du moniteur. Un travail inhibé est gelé dans l'état précédent l'inhibition et sa validation permettra son déroulement à partir de cet état.

- Travail en attente

Lors d'un échange avec un périphérique, il existe nécessairement un temps de latence mis à profit par le calculateur pour exécuter d'autres tâches ou lancer des échanges sur d'autres périphériques. Une fois l'échange satisfait, le travail repart de son point d'interruption. Tout se passe comme si le programme avait été momentanément inhibé, la fin de l'échange sur le périphérique permettant sa validation.

- Abandon

Lorsqu'une tâche provoque une alarme, elle est abandonnée ; elle passe à l'état inactif puis est supprimée. Le moniteur libère alors les périphériques qu'elle utilise, permettant une remise à jour des affectations. La zone mémoire correspondante pouvant être réalouée, le moniteur ne tient plus compte de sa présence et ne la sauvegardera pas, et le travail en dépendant devient alors inexistant.

Les différents états que peut prendre un travail sont représentés par le schéma de la figure 3. Le travail le plus prioritaire est le travail de priorité 1 (partie du moniteur), le moins important a la priorité 18 (travail de fond).

A un instant donné, il existe toujours un travail activé. Si plusieurs travaux sont en attente et que l'unité centrale est libre, le moniteur provoque l'exécution d'un travail de fond, qui peut être une boucle sans fin. Ce travail ne peut être inhibé et a pour but d'éviter l'arrêt total du calculateur.

Légende de la figure 3

- 0 Point de départ activation du travail 9
- 1 Le travail 9 demande le travail 10
- 2 Attente de 9 (Entrée - Sortie) et déroulement de 10 pendant ce temps
- 3 Interruption de fin d'échange - reprise de 9, inhibition de 10 par le moniteur
- 4 9 demande 8 avec condition de retour - attente de 9
- 5 8 en attente d'Entrée - Sortie. Comme 9 est en attente de 8 on a l'exécution de 10.
- 6 Interruption E/S - exécution de 8
- 7 La condition qu'attend 9 est réalisée. 8 est terminé donc 9 reprend son exécution (validation)
- 8 9 est en attente d'E/S. Pendant cette attente 10 se déroule et inhibe 9.
- 9 L'interruption de fin d'échange de 9 est apparue mais inhibée, il ne peut s'exécuter.
- 10 Validation par 10 de 9. Demande de 8.
- 11 8 demande 6 qui travaille anormalement. Abandon de 6 et de ses satellites : en cascade on abandonne les travaux demandeurs, 8 et 9.
- 12 Il ne reste que 10 en cours.

Exemple de fonctionnement

1.b - Organisation dans la gestion des travaux

Chaque travail est repéré par son niveau de priorité, aucune ne pouvant être attribuée simultanément à plusieurs travaux.

Ceci peut présenter un inconvénient lors d'un changement de priorité puisqu'alors le travail change de nom. Mais cette opération n'étant réservée qu'à l'opérateur, celui-ci peut très bien suivre l'évolution des travaux.

- Gestion des priorités

La priorité porte sur un travail, consistant en un certain nombre de sous programmes spécialisés (programmes de calculs, opérations particulières etc...). Ces tâches peuvent ne pas recevoir de priorité en elle-même, à condition d'avoir été répertoriées sous un nom de travail, afin d'éviter leur effacement.

- Gestion des périphériques

La succession des entrées - sorties sur les différents périphériques nécessite une gestion afin d'optimiser les échanges. D'autre part, pour éviter des interférences entre des travaux de priorités différentes, le moniteur doit gérer le fonctionnement de tous les périphériques.

Afin d'augmenter la rapidité, il importe de faire tous les échanges en mode prioritaire.

Une fois activé, tout programme peut faire une demande de périphérique au moniteur. Celui-ci décide alors

s'il le lui accorde suivant l'état d'occupation de l'unité. Si ce périphérique est occupé, la demande est rejetée, le travail est mis alors dans la file d'attente du périphérique. Il est inhibé momentanément jusqu'à sa validation qui n'interviendra qu'à la libération du périphérique et suivant la priorité du travail. En effet si un périphérique est affecté à un travail, il est considéré comme occupé pour tout autre demande, et ne sera libéré qu'à la fin du travail l'utilisant ou sur abandon.

1.c - Fonctions du moniteur

Pour une bonne gestion de la mémoire, tout travail est déclaré au moniteur avec son adresse d'implantation initiale, et suit certaines règles permettant de gérer au mieux les échanges avec l'extérieur. Le travail est donc ponctué d'appels au moniteur, à son début (insertion dans la journée des travaux) et à son achèvement (libération des périphériques utilisés).

D'autre part, afin de faciliter la programmation et permettre à l'utilisateur l'organisation de son travail, certaines fonctions sont utilisables par appel au moniteur. Il est ainsi possible de réaliser la synchronisation de plusieurs tâches ou une segmentation. Dans ce cas, nous distinguons :

- la structure simple ne nécessitant le chargement que d'un seul module. Le travail est considéré d'un seul tenant.

- la structure avec recouvrement à priori ;
Le programme est décomposé en une racine, résidente, et des segments, pouvant occuper successivement le même emplacement en mémoire.

- structure séquentielle

Le programme a la même composition que précédemment, mais il peut commander le chargement d'un segment qui s'exécutera lors des temps d'attente du programme corps.

1.d - Gestion des interruptions

Le moniteur temps réel traite toutes les interruptions, c'est à dire les interruptions d'entrées - sorties, les interruptions de type programme, les appels au moniteur, ainsi que les interruptions externes.

Les interruptions de type programme ont trait aux opérandes et aux résultats (débordement, opérateur incorrect), les interruptions externes correspondent a des appels demandant un certain traitement sur un périphérique (par exemple : un appel à partir de la console correspond implicitement à la demande d'un travail).

Il existe aussi des interruptions de type horloge indiquant l'écoulement d'un laps de temps programmé auparavant.

4.2 - Le programme de couplage

Le programme de couplage est en liaison directe avec la technologie des machines, sur laquelle nous ne reviendrons pas. D'un point de vue software, l'élément principal est le coupleur permettant les échanges à partir du calculateur numérique. Les modes de fonctionnement de ce coupleur sont variés : il est, dans le cadre du moniteur temps réel, utilisé en mode programmé prioritaire ou en mode canal.

2.a - Les commandes programmées

Il est possible à l'utilisateur de traiter son problème en mode conversationnel directement sous le contrôle du moniteur, l'interprétation des commandes se faisant immédiatement. Une autre possibilité consiste à traiter le programme hybride comme un travail global ne laissant aucune initiative à l'opérateur. Dans ce cas, toute erreur entraîne l'abandon pur et simple du travail en cours.

2.b - Commande des modes analogiques

Nous avons distingué auparavant les différentes fonctions sur les deux machines dont nous disposons actuellement. Leur utilisation ne présente aucun problème, que ce soit à partir de la console ou par programme. Nous devons sélectionner les machines pouvant évoluer indépendamment ou simultanément (mode esclave).

L'utilisation se fait en appelant les différentes fonctions dont les noms correspondent aux notations de la machine elle-même. La nomenclature est la même que ce soit en mode conversationnel ou en programmé. L'exploitation se fait à partir de macro instructions pour le langage assembleur et par appel à des sous programmes en fortran.

2.c - Commande des potentiomètres

La commande des potentiomètres asservis nécessite la mise en repos analogique de la machine. Jusqu'à présent, nous avons fait une différence entre potentiomètres asservis et potentiomètres numériques ; du point de vue du software, elle ne s'impose plus, la syntaxe étant la même

dans les deux cas. Le moniteur fait la différence à partir de leur numérotation lors de la sélection.

La demande d'un potentiomètre non opérationnel (non existant) ou défectueux (précision de l'affichage insuffisante) implique une erreur pouvant provoquer l'arrêt du travail responsable. Il est cependant possible de demander à partir de la console, lors de la préparation hors ligne, le numéro des potentiomètres hors service.

2.d - Commande des intégrateurs

De même que pour les potentiomètres, il est possible de lire les valeurs des intégrateurs, mais dans ce cas, si la lecture d'un amplificateur inexistant entraîne une erreur, travailler sur un intégrateur défectueux ne pénalise pas le travail. Cette lecture se fait par appel au moniteur. Il est possible de commander individuellement les intégrateurs, repérés par un numéro correspondant à leur place sur le panneau analogique.

4.3 - Programmes bibliothèques

La bibliothèque consiste en une série de programmes d'utilisation courante. Certains sont spécifiques au couplage, tels que la génération de fonctions, certaines transformations souvent utilisées (FFT), et d'autres plus généraux tels que certaines conversions.

Le rôle du bibliothécaire est de rechercher les programmes sur disque. Il contrôle simultanément l'identité du demandeur, seul le moniteur ayant droit à toutes les fonctions.

5 - CONCLUSION

L'accroissement des capacités du calculateur numérique et les nouvelles possibilités de transmission de données imposent un ensemble dans lequel plusieurs problèmes peuvent opérer simultanément. Autrement dit, on aura un partage du calculateur entre des travaux travaillant en temps réel (foreground) et d'autres en série continue (batch en background).

On cherche à réaliser un grand nombre de fonctions plutôt par hardware que par software, afin de simplifier les différentes procédures.

A plus longue échéance, avec l'accroissement des vitesses de calcul et des capacités mémoire, un seul calculateur digital commandera un grand nombre de calculatrices analogiques, les simulations se déroulant en temps partagé.

Pour ce qui nous concerne, nous ne sommes pas encore au stade de multiprocesseur composé de plusieurs minicalculateurs. Nous cherchons à améliorer le contenu de la bibliothèque notamment au point de vue hybride.

Il sera par la suite possible de mettre au point un interpréteur hybride, permettant une utilisation complète des machines.

Nous aurons atteint la troisième étape définie au chapitre premier. Il sera alors possible d'améliorer les performances en se définissant un nouvel ensemble d'étapes convergent vers un système multiprocesseur, ou tout autre chose suivant l'évolution des techniques hybrides.

PARTIE IV

EXEMPLES

ORGANISATION D'UN PROBLEME

1 - Introduction

Ce chapitre a pour but de montrer sur quelques exemples les possibilités des deux machines couplées.

Nous abordons en première partie une étude générale, mettant en évidence quelques principes que nous appliquerons lors de la résolution de problèmes plus particuliers tels que la résolution d'équations récurrentes, ou la simulation appliquée à une identification.

1.1 - Généralités

A partir des équations qui définissent le phénomène à étudier, il convient de trouver un modèle mathématique exact ou approché qui puisse être pris en compte par la machine. Ce travail consiste à appliquer les méthodes de résolution spécifique au calcul hybride telle que les méthodes d'optimisation dynamique, la méthode des caractéristiques pour les équations aux dérivées partielles, la méthode d'approximation fonctionnelle, etc...

En général, les différents rôles des calculateurs découleront directement de la méthode choisie pour le résoudre.

Nous avons vu que les principales tâches dévolues au calculateur numérique sont la mémorisation de fonction, le calcul algébrique ou intégral, ou parallèle ou en série avec la machine analogique, la décision et la commande.

La partie logique du calculateur analogique, sera utilisée en général comme organe de synchronisation entre le programme numérique et le programme analogique ; de plus elle sert d'intermédiaire pour l'exécution sur la machine analogique de certaines décisions prises sur la machine numérique.

Notons au passage la différence essentielle entre les décisions logiques séquentielles prises par le calculateur numérique et celles parallèles prises par la logique hybride.

Le travail particulier sur chaque partie, numérique, logique, analogique nécessite soit un organigramme, soit un schéma particulier selon les méthodes propres à chaque calculateur.

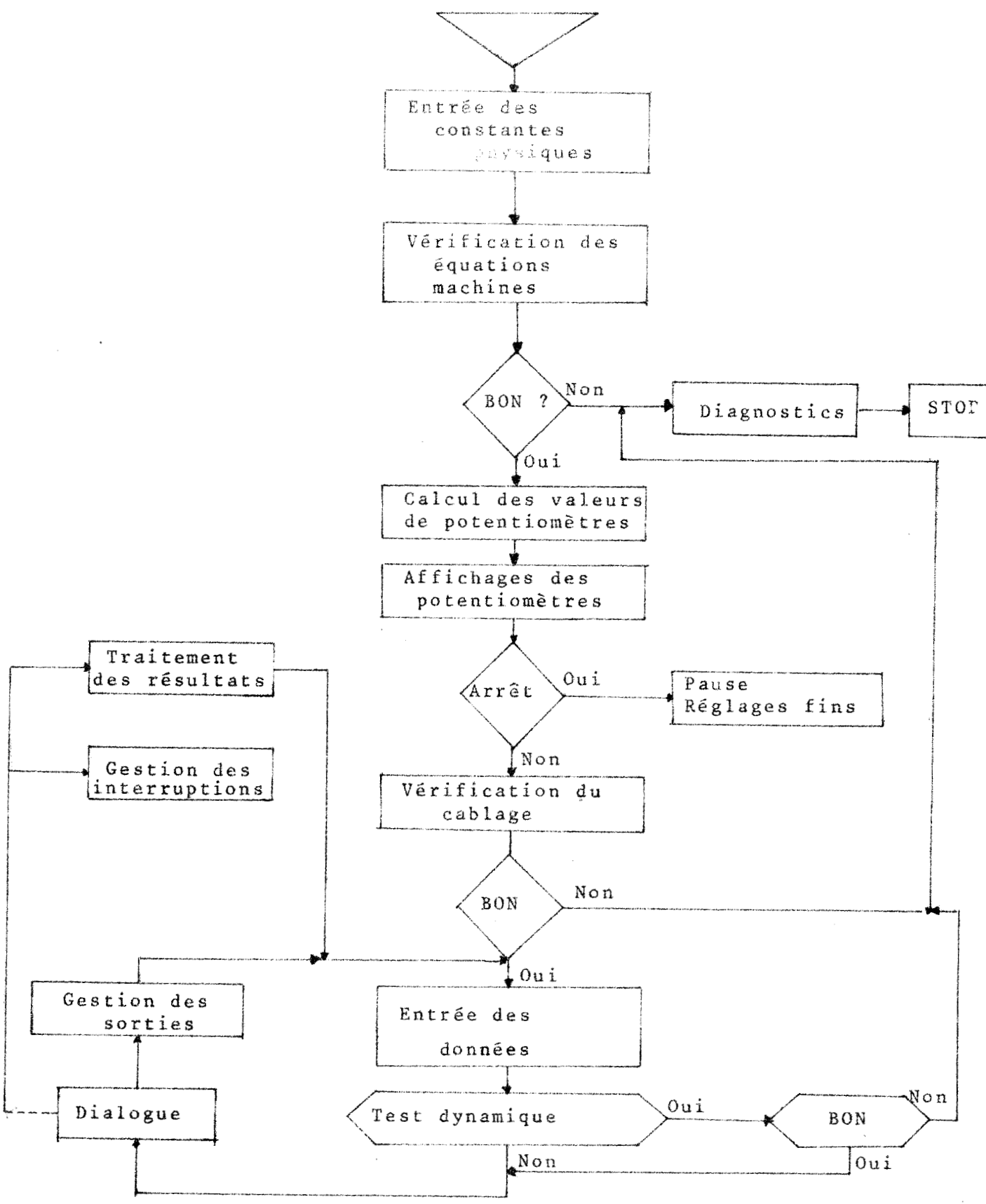
Il est cependant nécessaire de concevoir une organisation générale, un organigramme hybride (figure 1)

1.2 - Vérifications diverses et réglages

Cette étape permet de calculer et de régler les différentes valeurs des potentiomètres et d'effectuer certaines vérifications du modèle simulé. Les grandeurs de travail des physiciens, c'est-à-dire les paramètres physiques du problème, doivent être des grandeurs d'entrée du système.

Il est ensuite nécessaire de vérifier si les équations proposées à la machine correspondent bien aux équations physiques. Dans ce cas, on calcule les valeurs des potentiomètres et on les règle. Il est intéressant de prévoir un réglage éventuel à la main dans les cas délicats.

On vérifie ensuite que le câblage réalisé correspond bien aux équations données à la machine. C'est une vérification statique. La machine analogique est mise en mode "vérification statique". Les opérations réalisées en calcul analogique sont effectuées également en numérique et l'on compare les résultats. S'il y a erreur, les équations non



vérifiées et les amplificateurs correspondants sont notés sur la machine imprimante et le programme numérique passe en pause pour permettre éventuellement de modifier le câblage.

1.3 - Résolution

1.3.a - Calcul ou entrée des données dynamiques

On entend par entrées dynamiques les grandeurs de commande du système simulé. En d'autres termes, on veut en général connaître la réponse du système lorsqu'on lui applique un certain nombre de signaux d'entrée. Si l'on choisit pour la simulation d'un système multivariable une représentation discontinue d'une des variables, par exemple le temps, on doit connaître la valeur des grandeurs d'entrée pour chaque valeur du temps. Cela peut être soit une donnée, soit le résultat d'un calcul intermédiaire réalisé par le calculateur. De plus, l'état du système au temps initial doit être connu. Cela peut nécessiter un calcul préliminaire qui peut être lui-même hybride, notamment quand la connaissance des conditions initiales nécessite la convergence d'une série d'itération (cas des conditions finales connues).

1.3.b - Vérification dynamique

Il est très utile de pouvoir vérifier la logique du programme numérique ou programme de dialogue. Dans la phase de calcul où les calculateurs travaillent de concert, il est nécessaire de prendre un pas en "temps machine". Ce n'est que pour des valeurs discrètes de ce temps machine qu'il pourra y avoir conversation. Notons que ce temps machine est proportionnel à une variable quelconque du système. C'est suivant cette variable que se font les intégrations continues en calcul analogique, son choix ne dépendant que de la méthode de résolution retenue.

Pour vérifier la logique du programme numérique, il est donc nécessaire d'effectuer un pas en temps machine. On peut alors voir si les différents transferts ou opérations élémentaires se sont réalisées normalement. Certains programmes permettant cette vérification sans utiliser la machine analogique.

1.3.c - Programme de dialogue

Machine numérique et machine analogique doivent opérer ensemble ; il est donc nécessaire de synchroniser les deux programmes. Si en calcul analogique il est possible de garder une variable continue, pour le calcul numérique, toutes les grandeurs seront réduites à des valeurs discrètes. La résolution analogique supposera un état déterminé des grandeurs discrètes puis un nouveau calcul fera progresser les indices d'une unité. La synchronisation se fera donc à l'aide d'horloges propres ou connues à chaque calculateur, générées à partir de compteurs et permettant d'envoyer des interruptions sur le calculateur travaillant en mode esclave.

1.3.d - Programme d'interruption

Le programme de dialogue peut être aussi interrompu à tout moment par une décision du programme logique résultant en général d'un événement analogique. Par exemple, cela peut être le passage à un changement d'état survenant aléatoirement en nécessitant le calcul et le réglage de nouveaux paramètres.

1.3.e - Le programme de gestion des sorties

Il sera chargé de gérer toutes les sorties telles que l'impression sur l'imprimante l'envoi des résultats sur disques, exploitation de la table traçante, etc...

1.3.f - Traitement des résultats de calcul

Il s'agit ici d'un traitement en ligne, c'est-à-dire que les résultats du calcul sont analysés en vue de permettre un nouveau cycle, par exemple la recherche des conditions initiales en fonction des conditions finales.

2 - Traitement d'un exemple hybride

Le développement des techniques de commande amène à considérer de plus en plus des systèmes décrits par des relations de récurrence, tels que les systèmes échantillonnés à période fixe ou variable, les systèmes non conservatifs décrits par des équations différentielles linéaires à coefficient périodique, etc...

Le modèle mathématique se présente sous la forme d'une récurrence non linéaire.

(1) $\underline{V}_{n+1} = f(\underline{V}_n)$ où \underline{V}_n est le vecteur d'état du système à un instant t_n et \underline{V}_{n+1} le vecteur d'état à l'instant t_{n+1} .

Les propriétés des solutions de ce type d'équation sont d'une grande importance par l'analyse des systèmes qui leur correspondent. La connaissance de ces propriétés permet alors de mieux comprendre certains comportements particuliers de ces systèmes et de dégager des méthodes utiles d'analyse et de synthèse.

Nous allons montrer à travers un exemple, et à l'aide du calculateur hybride, comment il est possible d'étudier de manière systématique des propriétés telles que la recherche des états d'équilibre stables et instables, la détermination du domaine complet d'attraction d'un état d'équilibre stable, la détermination des régimes oscillants.

On peut montrer que l'étude de ces différentes propriétés sur la récurrence (1) se ramène à la détermination de tous les états d'équilibre stables ou instables, chacun de ces états d'équilibre étant donné par la résolution d'un système d'équations algébriques.

2.1 - Stabilité des systèmes modules en largeur

La difficulté d'étude de ces systèmes provient de leur non linéarité. Un échantillonneur à modulation d'amplitude classique, est essentiellement un organe linéaire dans le sens qu'entrée et sortie sont gouvernées par le principe de superposition. Cependant un échantillonneur à modulation de largeur doit être considéré comme un élément non linéaire, le principe de superposition entre la sortie et l'entrée ne s'appliquant pas.

Pour un échantillonneur à modulation de largeur, l'état du système à la fin de la période d'échantillonnage est la somme d'un vecteur dépendant de l'état du système au début de la période et d'un vecteur dont la longueur et la direction sont facteurs non linéaires de la largeur de l'impulsion.

Par une étude approfondie de la stabilité, le système à modulation de largeur des impulsions doit être traité comme un système essentiellement non linéaire, sans possibilité d'approximation linéaire.

2.1.a - Représentation du système

Nous considérons ici l'asservissement de position moteur commandé en modulation de largeur, avec une compensation tachymétrique (figure 2).

La commande s'effectue par un signal d'amplitude constant, du signe du signal d'erreur défini aux instants d'échantillonnage.

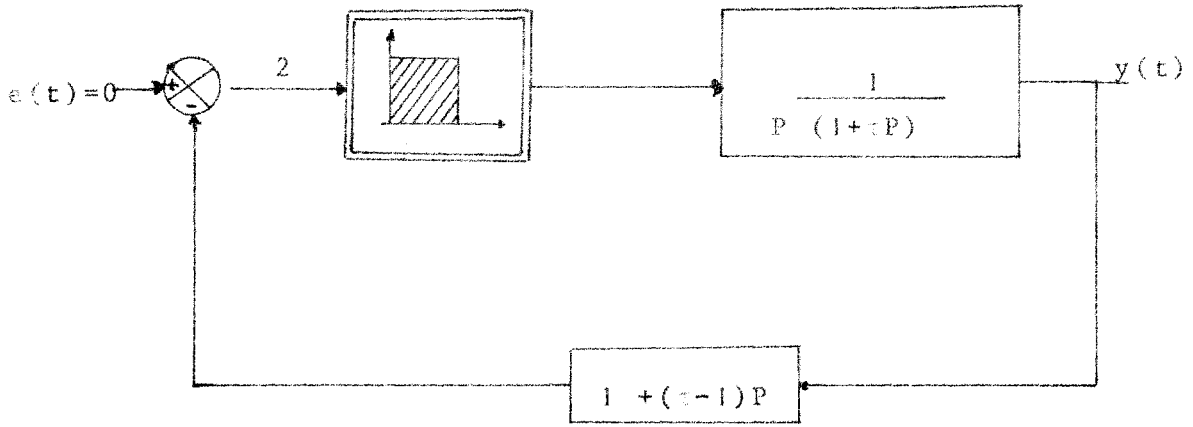


Figure 2
Asservissement

T désigne la période d'échantillonnage, $T_{in} = k|en|$ la durée de l'impulsion de commande en régime non saturé, λ la constante de temps du moteur.

λ caractérise l'effet de retour.

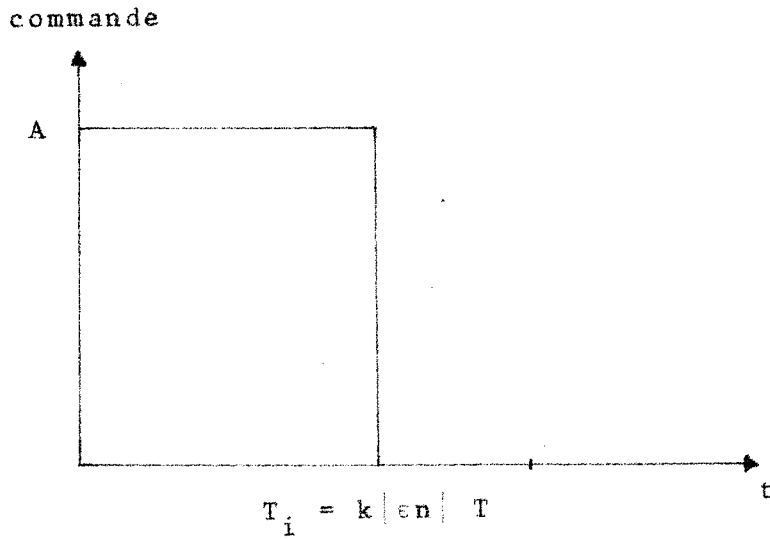


Figure 3
Modulation de largeur

2.1.b - Méthode de simulation

Il n'est pas dans notre intention d'établir toute la théorie relative à de tels systèmes, nous nous bornerons à mettre en évidence les possibilités de l'ensemble hybride.

Nous désirons effectuer un balayage paramétrique et étudier la stabilité d'un moteur, en régime saturé ou non nous ferons varier suivant différentes méthodes, les conditions initiales et déterminerons l'ensemble des points atteints dans le plan de phase (Y, Y') .

Nous étudierons l'influence du coefficient λ du retour tachymétrique ainsi que celle de paramètre $kA=RO$ (figure 3) sur la stabilité.

Le moteur ne présente pas en temps que système continu de difficultés lors de la simulation. Puisque nous considérons un système autonome ($e(t) = 0$), nous avons

$$\epsilon(t) = -Y_2(t) = - \left| 1 + (\tau - \lambda) p \right| Y$$

Cette valeur est envoyée à l'aide d'un convertisseur dans le calculateur numérique qui établit la commande à envoyer : $\epsilon_2 = \alpha A$ pendant un temps $T_i = k\epsilon n$

Nous avons alors le schéma de simulation de la figure 4.

Nous avons envisagé plusieurs possibilités sur la méthode de balayage dans le plan de phase Y, Y' . La première consiste à étudier systématiquement les différents points du plan, et à constater s'il y a stabilité ou non.

Mais cette méthode est extrêmement lourde, tout en utilisant un temps considérable ; si d'autre part nous voulons étudier la stabilité par rapport à certains paramètres, par exemple kA , elle se révèle inutilisable. Il est alors préférable de balayer le plan, y, y' , suivant une direction définie a priori au départ de la recherche et de modifier par la suite le contour dans le cas d'une instabilité, afin de déterminer le cycle limite.

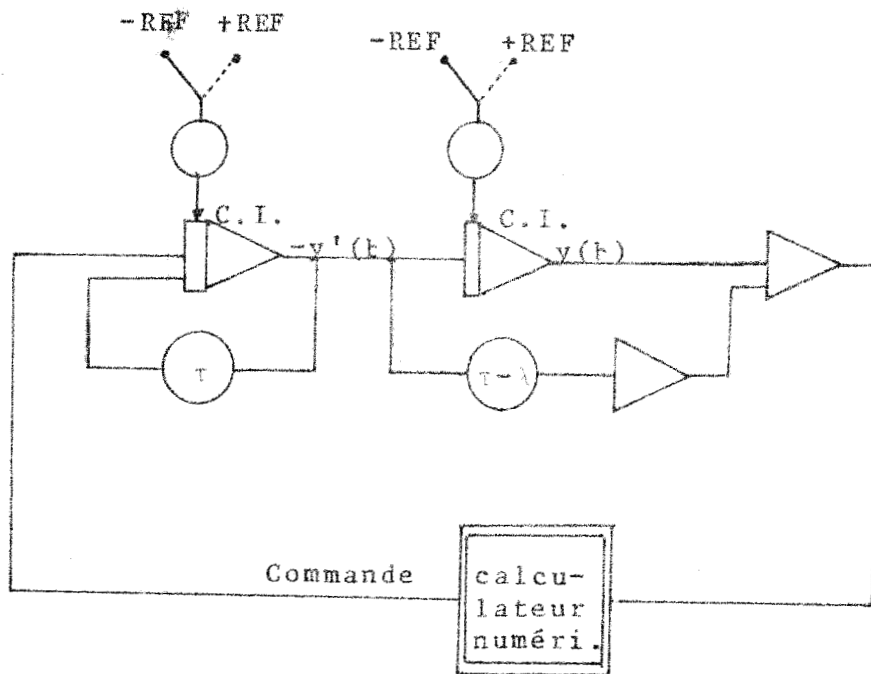


Figure 4
Schéma de simulation

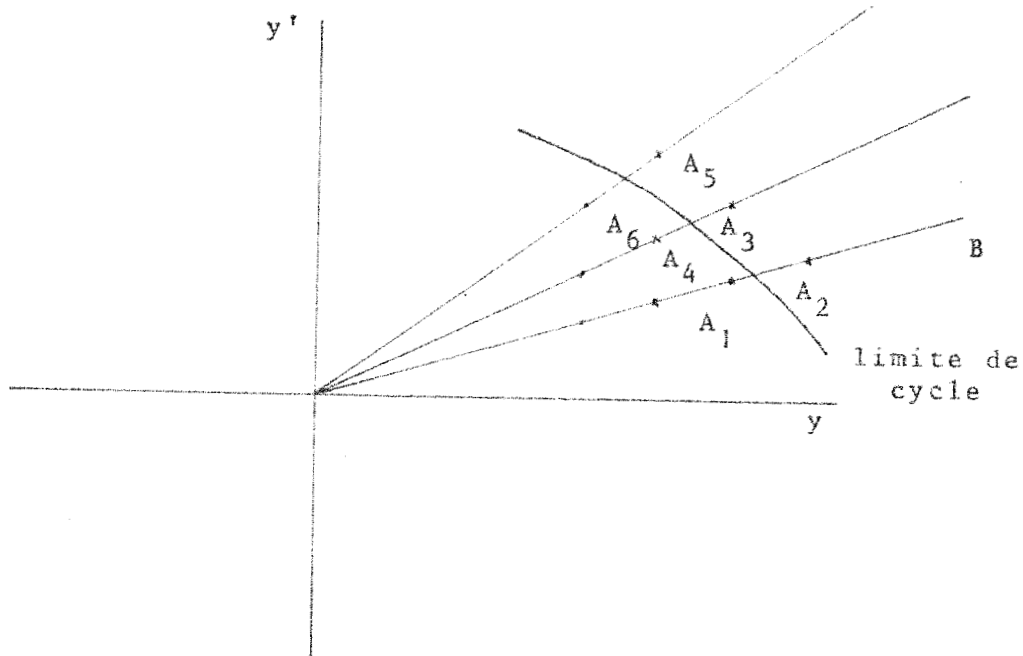


Figure 5
Balayage dans le plan y, y'



Le balayage en coordonnée polaire semble alors meilleur. Définissons une direction à priori AB, et faisons varier le point écrivons y, y' sur A B de A vers B; passons de A_1 à A_2 , nous observons une instabilité ; α est alors incrémenté et la recherche se fait entre A_3 et A_4 , le cycle se renouvelant nous obtenons alors la représentation du cycle limite avec une précision dépendant du pas choisi sur α et sur les couples de points consécutifs A_1 et A_2 . Pour déterminer le cycle limite nous pouvons utiliser une méthode du genre de celle de Little par la théorie des graphes.

2.2 Résultats

Nous avons quatre paramètres de discussion :

k
A
 λ
T

Dans certains domaines, notamment pour les valeurs relativement faibles de k et de A, nous pouvons raisonner en fonction de $kA = BO$, les résultats étant sensiblement les mêmes pour $BO = \text{constant}$; comme on peut le voir sur les enregistrements en fin de chapitre (figure A3)

Par $\lambda = 0,25$

$BO = 1$

$T = 1,$

nous avons concordance entre les résultats pratiques et les résultats théoriques. Le système est stable.

Pour $BO = 2,5$ nous avons des oscillations, que ce soit avec $h = 2,5$ $A = 1$ ou $A = 2,5$ et $h = 1$.

Dans un premier cas nous étudions la nature du cycle obtenu et l'évolution de la trajectoire dans l'espace des phases y, y' . Nous obtenons alors en régime saturé (k grand) les cycles de la figure A 5 (annexe).

Les notations sont les suivantes :

I : condition initiale sur y

I' : condition initiale sur y'

Nous avons étudié ensuite, en suivant l'évolution du paramètre λ , les types de trajectoires puis la stabilité illimitée. Par ce faire, nous faisons varier le paramètre T/... et relevons les valeurs de K indiquant un cycle. La liste des résultats est obtenue sur un télétype par laquelle on peut commander les nouvelles valeurs, ou laisser l'évolution automatique. Nous en déduisons l'évolution des points dans le plan BO plan BO, λ .

Ceci ne représente évidemment qu'une ébauche d'étude des oscillations limite d'un système de deuxième ordre, mais permet d'apercevoir la grande souplesse d'exploitation offerte par le système, pour commander l'évolution des paramètres.

Nous ne nous sommes pas attachés à la précision des résultats, les cycles limites n'étant en fait "perçus" par le calculateur qu'à partir d'un certain seuil, mais à l'allure générale du phénomène. Cependant une exploitation plus précise de cette simulation a permis de vérifier la concordance entre les études physiques et les conclusions de l'étude théorique

Le déroulement des opérations est représenté par l'organigramme de la figure A 6 (annexe).

Il est possible de balayer systématiquement le plan en faisant varier tel ou tel paramètre, ou même en introduisant les valeurs autour desquelles on désire travailler à partir du clavier de la télétype.

Nous envisageons les balayages suivant les différents paramètres en annexe.

Le système se caractérise par une simulation relativement aisée de par sa simplicité initiale (système du deuxième ordre), mais le mode d'utilisation hybride permet

une logique parallèle pratiquement inexistante ; notamment on n'a pas de compteur, d'horloge, de temporisation, etc...

Ce mode de fonctionnement est particulièrement intéressant dans notre cas, où seul l'occurrence d'un événement (présence de cycle, instabilité) peut être pris en compte. Ainsi le calculateur peut noter uniquement les cas de cycle et éditer dans un tel cas la valeur des différents paramètres, commander une table traçante, si bien qu'il sera possible d'éditer par la suite et de visualiser l'ensemble des conditions initiales donnant lieu à un cycle stable ou instable.

En conclusion, nous pouvons résumer les qualités des appareils en les repérant dans le tableau 6. Certains avantages du calcul hybride par rapport au calcul numérique sont en fait dus au fonctionnement en temps réel de l'ensemble hybride.

La solution par calculateur hybride apparaît la meilleure pour ce problème, par rapport aux solutions purement numériques ou analogiques.

	Calculateur scientifique numérique	Calculateur analogique	Calculateur hybride
Intégration	I	A	A
Interation	A	I	A
Non linéarité	A	I	A
Changement de paramètres	I	A	A
Mémorisation	A	I	A
Facteur d'échelles	A	I	I
Visualisation directe	I	A	A
Sortie des résultats	A	I	A
Paramètres et conditions initiales	A	I	A
Logique de recherche	A	I	A
Temps de calcul global	I	I	A

A Avantage

I Inconvénient

Figure 6
Etude comparative



3 - EXEMPLE 2

Nous allons étudier maintenant un problème simulé à la fois sur une calculatrice hybride EAI 580 et un calculateur 360 CSMP. Nous allons reprendre cet exemple, en l'adaptant au système hybride que nous avons réalisé ; nous pourrions de cette manière comparer les résultats de la publication (17), avec ceux que nous avons obtenus.

3.1 - Définition du problème

L'équation que nous envisageons, traite d'un écoulement de fluide suivant la loi de Poiseuille (17).

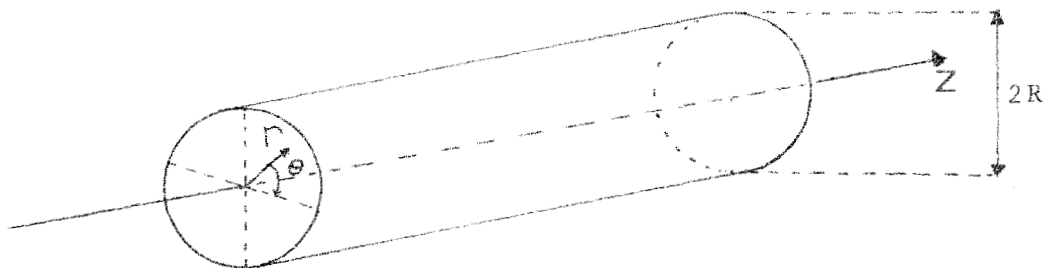


Figure 7

Fluide dans une canalisation - Notations

Pour un fluide circulant dans une canalisation, nous connaissons sa température externe T_1 et sa vitesse au niveau de l'enveloppe ($u = 0$).

Nous désirons connaître la vitesse et la température de ce fluide au centre de cet écoulement.

Les coefficients de viscosité (μ) et de conductivité thermique (k) sont des fonctions non linéaires. Nous

prendrons uniquement le cas où μ et k sont des fonctions de la température.

$$\mu = \mu_0 \left(\frac{T_0}{T}\right)^2$$

$$k = k_0 \frac{T_0}{T}$$

avec $T_0 = \frac{T_l + T_m}{2}$ T_m valeur maximale de la température.

La résolution de ce problème nécessite de partir de conditions initiales données et de vérifier que les valeurs finales correspondent aux valeurs théoriques à atteindre. Par un critère de notre choix, nous réajustons, à chaque itération, les conditions initiales du départ.

3.1.1 - Equations du système

Nous pouvons écrire les équations du système :

$$\text{Equation de continuité : } \frac{\partial u}{\partial z} + \frac{\partial v_r}{\partial r} + \frac{v_r}{r} = 0 \quad (\text{I})$$

$$\text{Equation du moment : } \rho u \frac{\partial u}{\partial z} + \rho v_r \frac{\partial u}{\partial r} + \frac{dp}{dz} = \frac{1}{r} \frac{\partial}{\partial r} \left(r \mu(T) \frac{\partial u}{\partial r} \right) \quad (\text{II})$$

$$\begin{aligned} \text{Equation d'énergie : } \rho u \frac{\partial h}{\partial z} + \rho v_r \frac{\partial h}{\partial r} - \frac{1}{r} \frac{\partial}{\partial r} \left(k(T) r \frac{\partial T}{\partial r} \right) \\ - \mu(T) \left(\frac{\partial u}{\partial r} \right)^2 - u \frac{dp}{dz} = 0 \quad (\text{III}) \end{aligned}$$

r et z sont les coordonnées cylindriques

v_r et u , sont les composantes de la vitesse suivant les directions r et z

ρ masse volumique du fluide.

Nous disposons d'un certain nombre de conditions :

* le flux est irrotationnel : $v_r = 0$

$$\frac{\partial u}{\partial z} = \frac{\partial T}{\partial z} = 0$$

* Le fluide est incompressible : $\rho = \text{const.}$

$$\approx dh = cdT + \frac{1}{\rho} dp$$

* $\frac{dp}{dz} = \text{constant} = C_1 < 0$

L'équation (I) est vérifiée par ces conditions.

L'équation (II) devient :

$$\frac{dp}{dz} = C_1 = \frac{1}{r} \frac{d}{dr} (r \mu (T) \frac{du}{dt}) \quad (\text{IV})$$

$$\frac{d}{dr} (r k (T) \frac{dt}{dr}) = - r \mu (T) \left(\frac{du}{dr}\right)^2 \quad (\text{V})$$

Les conditions de cet écoulement sont :

$$r = 0 \quad \frac{du}{dr} = 0 \quad \frac{dT}{dr} = 0$$

$$r = R \quad u = 0 \quad T = T_1$$

Le travail consiste à trouver les quantités $u(0)$ et $T(0)$, qui sont les conditions initiales du système.

$$\text{Or} \quad u(0) = u_m \quad \text{et} \quad T(0) = T_m$$

$$\text{Nous définissons donc :} \quad \bar{r} = \frac{r}{R} \quad \bar{T} = \frac{T}{T_1} \quad \bar{u} = \frac{u}{u_m}$$

En reportant dans les équations (IV) et (V), nous obtenons :

$$C_1 = \frac{1}{\bar{r}} R^2 \frac{d}{d\bar{r}} \left(\bar{r} \frac{\mu_0 T_0^2}{\bar{T}^2 T_1^2} u_m \frac{d\bar{u}}{d\bar{r}} \right)$$

$$\text{Soit } C_1 R^2 \frac{\bar{r}^2}{2} = \bar{r} \frac{\mu_0 T_0^2}{\bar{T}^2 T_1^2} u_m \frac{d\bar{u}}{d\bar{r}} + C_2$$

$$\bar{r} = 0 \quad = \quad C_2 = 0$$

d'où :

$$\frac{d\bar{u}}{d\bar{r}} = \frac{C_1 R^2}{2} \frac{T_1^2}{\mu_0 T_0^2} \frac{\bar{T}^2}{u_m} \bar{r}$$

$$\text{Posons } \Gamma = \frac{2 C_1^2 R^4}{\mu_0 k_0 T_1} \quad \lambda = \frac{T_m}{T_1} \quad u^* = u \frac{\mu_0}{k_0 T_1}$$

$$\frac{du^*}{d\bar{r}} = \frac{C_1 R^2}{2} \cdot \frac{1}{\mu_0 k_0 T_1} \cdot \frac{T_1^2}{T_0^2} \bar{r} \bar{T}^2$$

$$\text{Or } \Gamma^{1/2} = - \frac{2 C_1 R^2}{\mu_0 k_0 T_1} \quad (\text{puisque } C_1 < 0)$$

$$(\lambda + 1)^2 = \left(\frac{T_m + T_1}{T_1} \right)^2 = 4 \frac{T_0^2}{T_1^2}$$

L'équation se met alors sous la forme

$$\frac{du^*}{d\bar{r}} = - 2 \frac{\Gamma^{1/2}}{(\lambda + 1)^2} \bar{r} \bar{T}^2 \quad (\text{VIII})$$

De même :

$$\frac{d}{d\bar{r}} \left(\frac{\bar{r}}{\bar{T}} \frac{d\bar{T}}{d\bar{r}} \right) = - \frac{\Gamma}{(\lambda + 1)^3} \bar{r}^{-3} \bar{T}^2 \quad (\text{IX})$$

Les conditions sont :

$$\bar{r} = 0 \quad \frac{du^*}{d\bar{r}} = 0 \quad \frac{d\bar{T}}{d\bar{r}} = 0$$

$$\bar{r} = 1 \quad u^* = 0 \quad \bar{T} = 1$$

Nous devons donc rechercher les conditions initiales de ce système $u^*(0)$ et $\bar{T}(0) = \lambda$

Nous poserons pour la simulation $\bar{r} = Kt$

En prenant $K = 0,1$, la période de calcul sera de 10s.

On doit alors avoir $u^* = 0$ et $\bar{T} = 1$.

3.1.2 - Simulation hybride de Type I

A partir des équations VIII et IX, nous pouvons élaborer un schéma de simulation (Figure 3).

Nous constatons que nous devons diviser les quantités

$$- \frac{\bar{r}}{\bar{T}} \frac{d\bar{T}}{d\bar{r}} \quad \text{par} \quad \frac{\bar{r}}{\bar{T}}$$

Or, cette opération n'est pas définie par $\bar{r} = 0$

pour $\bar{r} = 0 \quad \frac{d\bar{T}}{d\bar{r}} = 0$

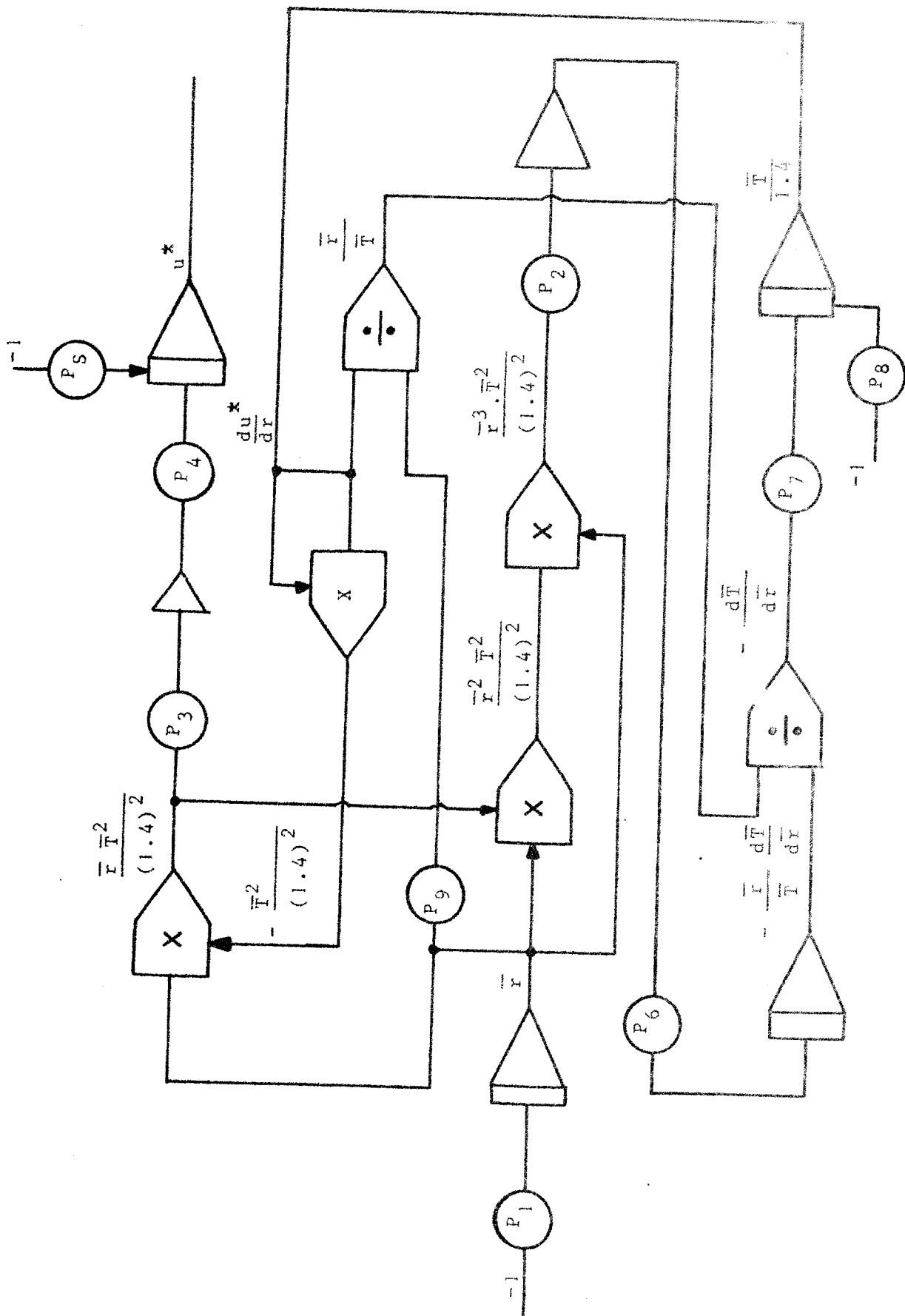


Figure 8
Simulation analogique



- LEGENDE -

$$P_1 = 0.1$$

$$P_2 = \frac{\Gamma}{25}$$

$$P_3 = \frac{\Gamma^{1/2}}{5}$$

$$P_4 = \frac{1.388}{(\lambda + 1)^2}$$

$$P_5 = u^* (0)$$

$$P_6 = \frac{4.9}{(1 + \lambda)^3}$$

$$P_7 = \frac{1}{14}$$

$$P_8 = \frac{\lambda}{1.4}$$

$$P_9 = \frac{1}{1.4}$$



Nous devons donc admettre que pour $0 < \bar{r} < r_0$,

$\frac{d\bar{T}}{d\bar{r}}$ reste constant et égal à zéro.

Il faut choisir r_0 de manière à n'apporter aucune perturbation au système.

On pourra choisir $r_0 = 0,1$

3.2 - Définition de la simulation en hybride de type II

3.2.1 - Recherche du schéma

Si nous considérons le montage de la figure nous constatons que le calculateur peut jouer le rôle de générateur de fonction en élaborant la quantité $\frac{d\bar{T}}{d\bar{r}}$ à partir des informations $-\frac{\bar{r}}{\bar{T}}$ $\frac{d\bar{T}}{d\bar{r}}$ et $\frac{\bar{r}}{\bar{T}}$

Pour $\bar{r} < r_0$ le calculateur par l'intermédiaire de sorties analogiques élabore une tension nulle.

Par $r_0 < \bar{r} < 1$, le calculateur fournit une tension :

$$v = \frac{d\bar{T}}{d\bar{r}} = \frac{\alpha}{\beta}$$

avec $\alpha = \frac{\bar{r}}{\bar{T}} \frac{d\bar{T}}{d\bar{r}}$ et $\beta = \frac{\bar{r}}{\bar{T}}$

Nous aurons alors le schéma de simulation donné par la figure 9

Le calculateur a deux tâches à accomplir :

a) Générer sur une période de 10s, le signal $\frac{d\bar{T}}{d\bar{r}}$

Pour cela, il lance une séquence de lecture qui lui définit

$$\bar{r} \quad , \quad \frac{\bar{r}}{\bar{T}} \quad , \quad \text{et} \quad - \frac{\bar{r}}{\bar{T}} \frac{d\bar{T}}{d\bar{r}}$$

Après calcul, il génère le signal $\frac{d\bar{T}}{d\bar{r}}$, puis recommence le cycle.

b) En fin de période, le calculateur "lit" les valeurs de u^* et de \bar{T} et doit, suivant ces valeurs, modifier les conditions initiales

$$u^*(0) \quad \text{et} \quad \bar{T}(0) = \lambda.$$

Cette tâche accomplie, un nouveau cycle de calculs recommence.

Cette séquence se déroulera jusqu'à l'atteinte des conditions finales, c'est à dire $u^* = 0 \quad \bar{T} = 1$ au bout de 10 secondes.

3.2.2 - Recherche du critère d'optimisation (ou de recherche)

Nous considérons les équations VIII et IX, nous constatons que si u^* dépend de \bar{T} , \bar{T} ne dépend pas de u^* .

Nous pouvons simplifier l'étude de la manière suivante :

Le paramètre " Γ " étant choisi, nous pouvons déterminer λ tel que $\bar{T}(1) = 1$.

Lorsque λ est trouvé, alors il nous reste à ajuster $u^*(0)$ tel que $u^*(0) = 0$.

Pour cela, il suffit de laisser évoluer le système sans la condition initiale $u^*(0)$.

La valeur de sortie sera l'opposé de la valeur initiale pour $u^*(0)$.

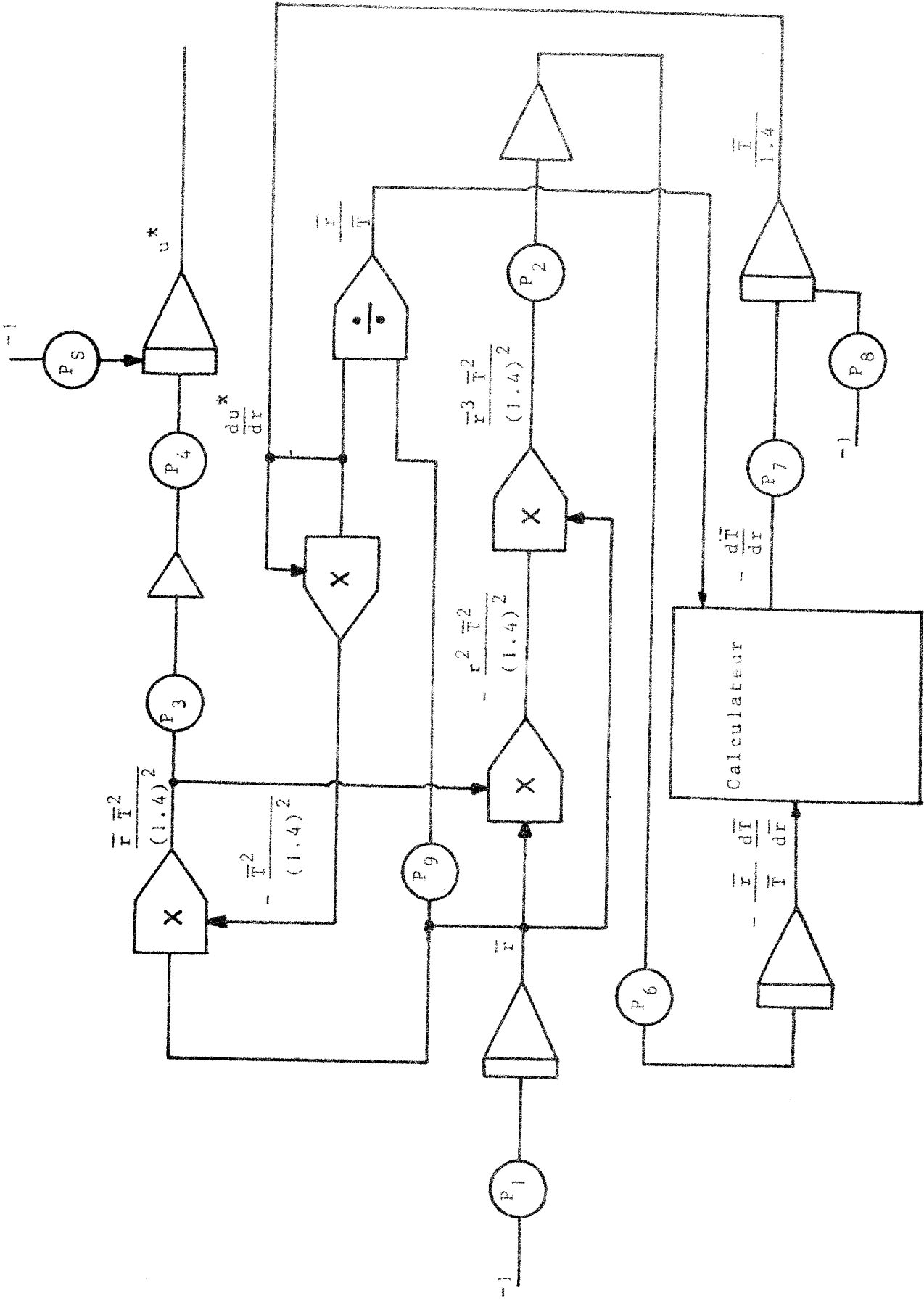


Figure 9
Simulation_hybride

3727
L314E

L'équation IX nous indique que \bar{T} croît avec λ . L'observation de \bar{T} fournira donc le critère de choix de (λ) . Les résultats et l'organigramme du système sont donnés dans l'annexe (D).

4 - EXEMPLE 3

Cet exemple a été réalisé surtout pour montrer la précision du couplage et mettre en évidence une propriété de l'ensemble hybride : la génération de fonctions. L'étude choisie est simple, mais est la mieux adaptée pour discuter du problème de la précision.

4.1 - Problème

Notre intention est de générer une sinusoïde.

Nous disposons sur la partie analogique du signal Y généré et de sa dérivée Y'.

Les signaux sont de la forme : $Y = a \sin (\omega t + b)$
 $Y' = a\omega \cos (\omega t + b)$

Nous allons réaliser le produit $Y.Y'$ de deux manières :

- d'une manière analogique

- d'une manière numérique, en lisant par la chaîne les valeurs Y et Y', en effectuant le produit numérique et en générant la valeur analogique correspondante.

Soit Z le signal donné par le multiplieur analogique et T le signal fourni par le convertisseur digital analogique du calculateur.

Nous observons chacun de ces signaux et la différence $T - Z$.

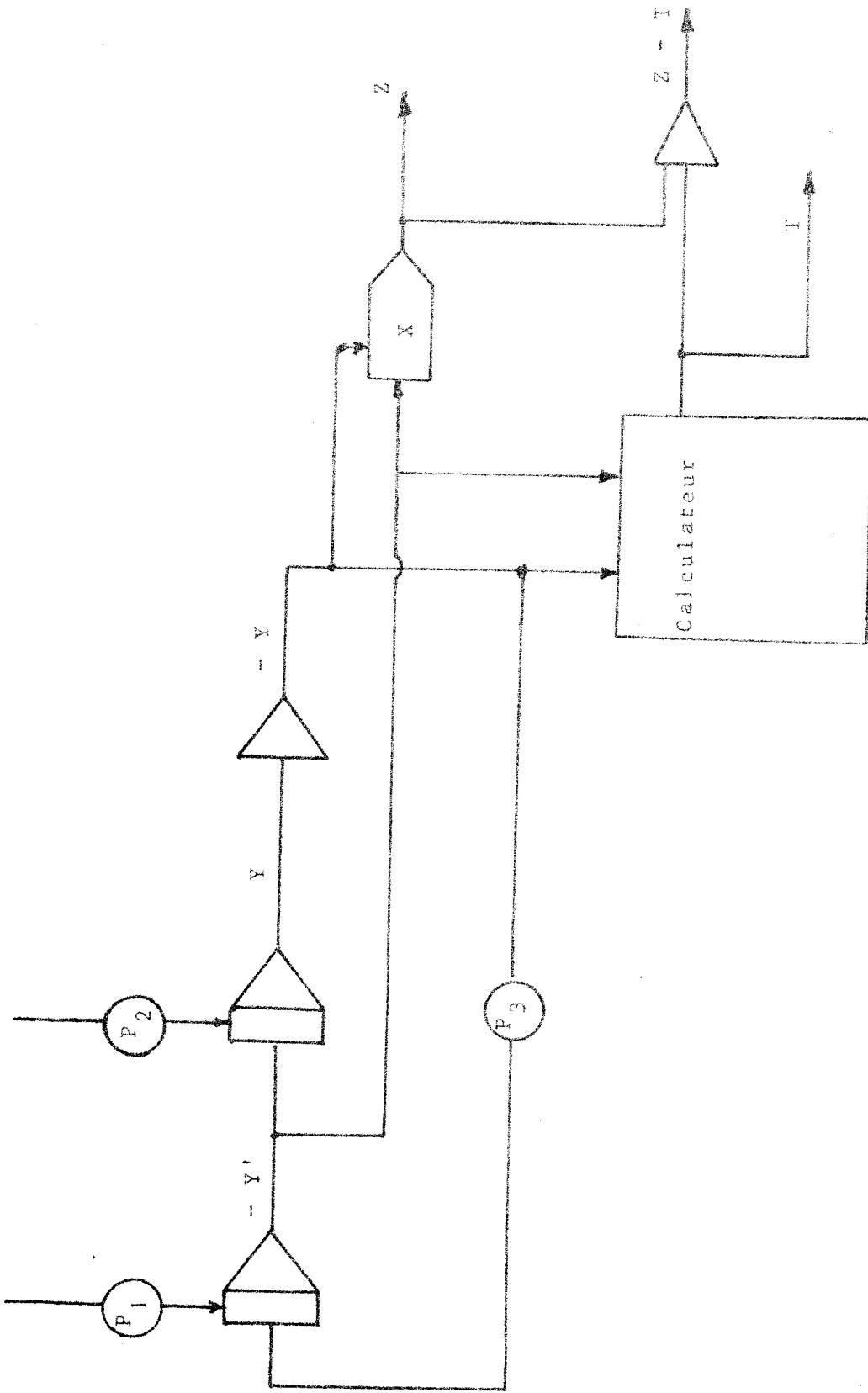


Figure 10

Simulation_hybride_-_Génération_de_fonctions

Le résultat est aussi une sinusoïde de période moitié

$$Y.Y' = a^2 \omega \sin(\omega t + b) \cdot \cos(\omega t + b) = \frac{a^2 \omega}{2} \sin 2(\omega t + b)$$

4.2 - Résultats

Les résultats sont donnés dans l'annexe (E).

Afin de pouvoir observer Z et T, nous sommes obligés de les considérer en opposition de phase sinon les sorties se correspondent.

Pour pouvoir observer l'erreur du système (T - Z), nous ajoutons un gain de 10 dans la chaîne.

Si l'on considère l'erreur absolue, elle est de ± 1 mV, compte tenu du bruit qu'apportent les amplificateurs, la table traçante

l'erreur relative est donc de $\frac{1}{2500} = 4 \cdot 10^{-4}$, c'est à dire de l'ordre de la précision du système.

CONCLUSION GENERALE

La plupart des études, au niveau du laboratoire ou lors d'une application industrielle, font appel à une simulation d'un modèle. Les différents modes de simulation prennent donc une grande importance et c'est de l'évolution du matériel que dépend l'application de nouvelles méthodes de recherche permettant une analyse des problèmes plus efficace et une solution plus rapide. En nous préoccupant de satisfaire les nécessités spécifiquement hybride, nous nous orientons vers l'élaboration d'algorithmes sans perdre de vue toutefois la liaison étroite existant entre le hardware et le software.

L'accroissement des capacités des calculateurs numériques et les nouvelles possibilités de transmission des données impose un ensemble ne répondant plus à un concept à structure fixe, mais permettant le déroulement de plusieurs problèmes simultanément : on aura un partage du calculateur avec d'autres travaux travaillant en temps réel ou en mode séquentiel (Batch).

Les langages spéciaux (CSSL) doivent réduire les efforts de programmation et contenir l'accroissement des prix du software. Une solution consiste à introduire des langages orientés vers une classe particulière de problèmes, ou dans un autre sens, en reportant dans la mesure du possible les fonctions programmées sur des fonctions câblées, à utiliser des minicalculateurs microprogrammés spécialisés.

Nous avons décrit dans ce mémoire le système hybride tel que nous l'avons réalisé et non tel qu'il devrait être. Nous sommes conscients de ses limites et songeons, à partir de l'ébauche ainsi réalisée, à élaborer un système plus performant ; nous ne devons pas perdre de vue que cet appareil est utilisé dans le cadre d'un laboratoire, et en ce sens doit répondre à certains critères : il n'est donc pas nécessaire d'en accroître la complexité à seule fin d'obtenir un appareil performant. Certains compléments sont donc prévus au point de vue software pour l'établissement d'algorithmes, et au point de vue hardware pour améliorer la fiabilité de la machine et les contrôles des différentes phases au niveau de l'interface.

A la fin de cette dernière étape, nous aurons atteint le but que nous nous sommes assignés et défini lors de la première partie, nous pourrons prévoir alors dans un autre contexte une nouvelle direction de recherche à partir des techniques hybrides ainsi mises au point.

ANNEXE A

RECHERCHE DES SEQUENCES DE TESTS.

Il n'est pas intéressant d'étudier la réponse du système à toutes les séquences d'entrées, cela entraîne une perte de temps considérable. Nous déterminons pour chacune des plaques à tester les séquences optimales d'entrée, pour détecter les différents modules défectueux.

A.1 - PLAQUE DE COMMANDE INDIVIDUELLE DES INTEGRATEURS

L'observation de la figure A.1 nous montre deux circuits indépendants. Pour chacun des modèles, nous déterminons les séquences permettant de détecter un collé à 1 ou à 0.

Nous allons montrer la méthode de recherche des séquences de test. Nous utilisons les algorithmes et notations de Roth (10). Nous construisons alors la table suivante :

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
6	d		0			d'											
	0		d			d'											
7		d		0			d'										
		0		d			d'										
8				d		0		d'									
				0		d		d'									
9			d				0		d'								
			0				d		d'								
10					d			1		d'							
					1			d		d'							
11					d						d'						
12						d			1			d'					
						1			d			d'					
13					d								d'				
14		d										1		d'			
		1										d		d'			
15			d										1		d'		
			1										d		d'		
16										d				1		d'	
										1				d		d'	
17															1		d'
															d		d'

Figure A.2 - Cube de propagation simple

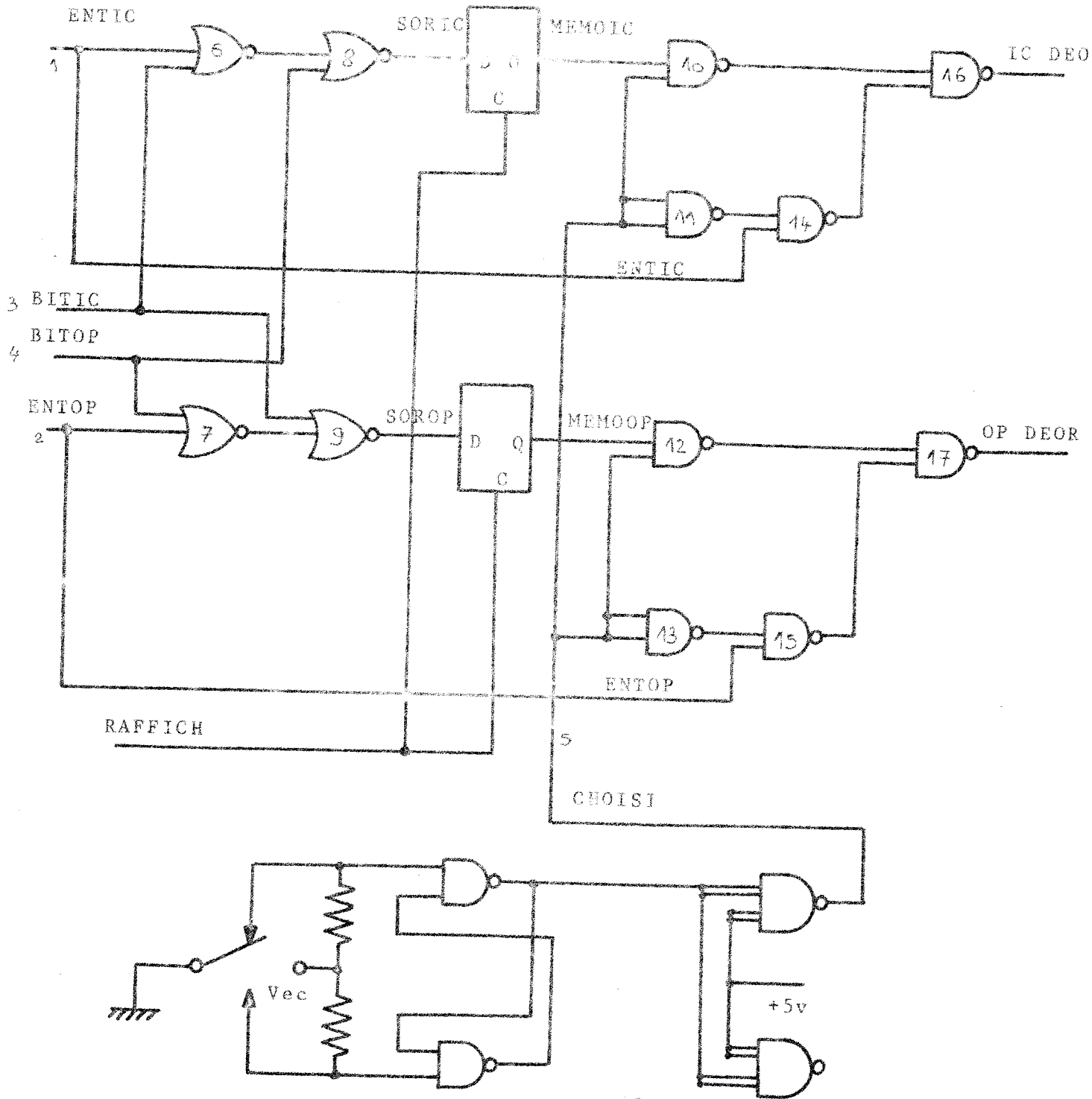


Figure A1

Schéma logique de la commande individuelle des intégrateurs



Nous cherchons les séquences de tests pour déterminer un collé à 1 ou 0 pour le circuit n° 8

		D. Intersection																	N°	Module d Sor	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17			
					0	0			d										t_c^0	8	10
0	10				0	1	0		d		d'								$t_c^{0,1}$	8,10	16
0,1	16				0	1	0		d		d'			1		d			$t_c^{0,1,2}$	8,10,16	∅

Figure A.3 - Cycle de test

					0	1	0		d		d'			1		d										
14a	0											x		1												
14b	x											0		1												
1,2	14b	x			0	1	0		d		d'	0		1		d										
					1							0														
1,2	14b	11	x		0	1	0		d		d'	0		1		d										
	6a		1	x		0													Séquences :	1	x	x	0	1		
	6b		x	1		0																x	x	1	0	1
1,2	14a		0		0	1	0		d		d'	x		1		d										
cas :																										
	x = 0																									
	x = 1																									
cas x=0			0		0	1	0		d		d'	0		1		d										
					1							0														
1,2	14a	11	0		0	1	0		d		d'	0		1		d			Séquence :	0	x	1	0	1		
	6a		1	x		0													6a ne peut être chois							
	6b		x	1		0																				
cas x=1																										
1,2	14a		0		0	1	0		d		d'	1		1		d			impossible :							
					0							1							contradiction							

Figure A.4 - Séquence de test

x peut être choisi de la manière désirée. Seules les informations fixées ne peuvent changer.

Il existe trois séquences pour détecter une défectuosité du circuit 8 ; elles sont $1 x x 0 1$, $x x 1 0 1$, $0 x 1 0 1$.

De la même manière, nous déterminons toutes les autres séquences. Nous donnons figure A.5, les différentes entrées de test des modules. La variable x , pouvant être choisie de la manière désirée, nous associons des séquences, pour détecter simultanément les erreurs des circuits communs.

Par exemple, dans la configuration, les circuits 8 et 9 jouent le même rôle. Ils sont soumis aux mêmes entrées.

8 est détecté par les séquences

$1 x x 0 1$
$x x 1 0 1$
$0 x 1 0 1$

9 est détecté par les séquences

$x 1 0 x 1$
$x x 0 1 1$
$x 0 0 1 1$

Nous remarquons que le choix de la séquence $1 1 0 0 1$, permet de déceler un collage à 0 ou 1 des circuits 8 ou 9.

Cette remarque s'applique pour tous les autres circuits. Nous avons répertorié dans la figure A.6, les différentes séquences de test, et les modules touchés par ces séquences. Le choix des séquences optimales se fait de la manière suivante :

Si l'on considère que les séquences sont les implicants premiers, et que les numéros de circuits sont les mintermes expansés, le choix des séquences optimales revient à rechercher les implicants premiers principaux.

N°	Séquences	N°	Séquences	Tests
6	0 x 0 0 1	7	x 0 0 0 1	0 0 0 1
8	1 x x 0 1 x x 1 0 1	9	x 1 0 x 1 x x 0 1 1	1 1 0 0 1
10	0 x 1 0 1	12	x 0 0 1 1	
11	1 x x 1 0 1 x 1 1 0 1 0 x 0 0 1 1 x x 0 1 x x 1 0	13	x 1 1 x 0 x 1 1 1 0 0 1 0 x 0 1 1 x x 0 x 1 1 x 0	1 1 1 1 0 1 1 x x 0
14	1 x x 1 0 1 x 1 1 0 1 x x 0 0 1 x 1 0 0 1 x x 1 0 1 x 1 1 0	15	x 1 1 x 0 x 1 1 1 0 x 1 0 x 0 x 1 0 1 0 x 1 1 x 0 x 1 1 1 0	1 1 1 1 0 1 1 0 0 0
16	0 x 0 x 1 0 x 0 1 1 1 x x 1 1 x x 1 1 1 0 x 0 x 1 0 x 0 1 1 0 x 1 1 1 0 x 0 x 0 0 x 0 1 0 0 x 1 0 0 0 x 0 1 0 0 x 1 1 0 0 x x 0 0	17	x 0 x 0 & x 0 1 0 1 x 1 1 x 1 x x 1 1 1 x 0 x 0 1 x 0 1 0 1 x 0 1 1 1 x 0 x 0 0 x 0 1 0 0 x 0 0 1 0 x 0 1 0 0 x 0 1 1 0 x 0 0 x 0	0 0 0 0 1 1 1 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 1 1 0

A.5 - Séquences de tests pour la plaque de
commande individuelle des intégrateurs.



	0000	1100	1110	1100	0000	1111	0011	0000	0010
6	1	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0
8	0	1	0	0	0	0	0	0	0
9	0	1	0	0	0	0	0	0	0
10	0	1	0	0	0	0	0	0	0
11	0	0	1	1	0	0	0	0	0
12	0	1	0	0	0	0	0	0	0
13	0	0	1	1	0	0	0	0	0
14	0	0	1	1	0	0	0	0	0
15	0	0	1	1	0	0	0	0	0
16	1	0	0	0	1	1	1	1	1
17	1	0	0	0	1	1	1	1	1

Figure A.6

Circuits touchés par les séquences de test.



Nous constatons qu'il nous suffit d'envoyer les séquences 0 0 0 0 1, puis 1 1 0 0 1, et 1 1 1 1 0, afin de détecter une panne. L'observation des sorties nous permet de détecter les erreurs et une analyse de ces valeurs nous indique les circuits défectueux. Nous donnons ci-après les organigrammes de tests et de détection des erreurs (annexe B.17 et B.18)

A.2 - PLAQUE DE COMMANDE PUPITRE

A.2.1 - Les fonctions analogiques

La figure A.4 permet d'élaborer les différentes séquences de tests.

	000	001	010	011	100	101	110	111
4	0	0	0	0	1		1	0
5	0	0	1	1	0		1	0
6	0	1	0	1	0		0	1
7	1	0	0	0	0		0	0
8	0	0	0	0	0		1	0
9	0	1	0	0	0		0	0
10	0	0	1	0	0		0	0
11	0	0	0	0	1		0	0
12	0	0	0	1	0		0	0
13	0	0	0	0	0		1	0
14	0	0	0	0	0		0	1
15	1	1	1	1	1		0	1
16	1	0	1	1	1		1	1
17	1	1	0	1	1		1	1
18	1	1	1	1	0		1	1
19	1	1	1	0	1		1	0
20	1	1	1	1	1		1	0
21	1	1	1	1	1		1	0

Figure A.5 - Circuits touchés par les séquences de tests.



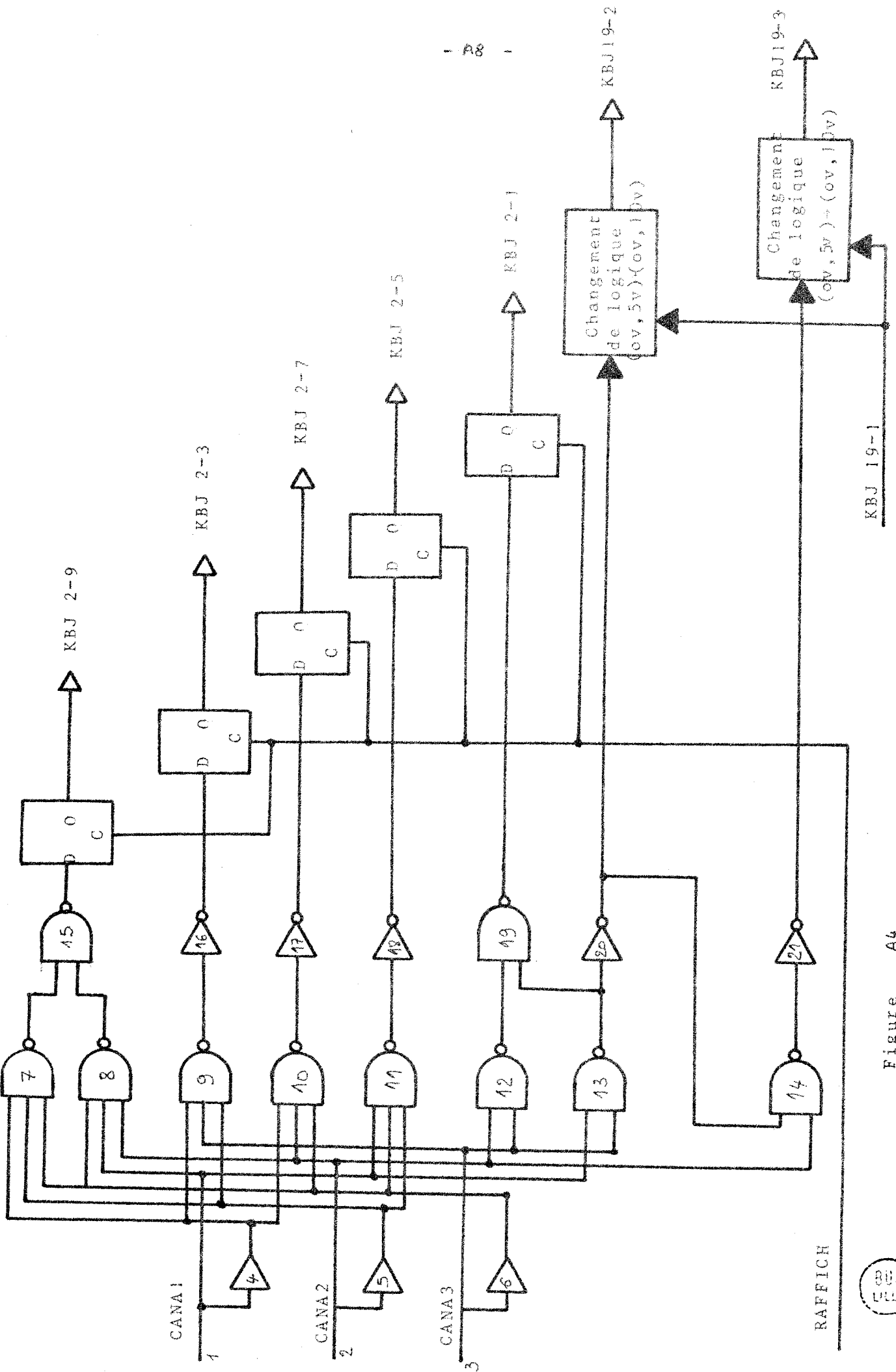


Figure A4
Schéma logique de la commande des fonctions analogiques.



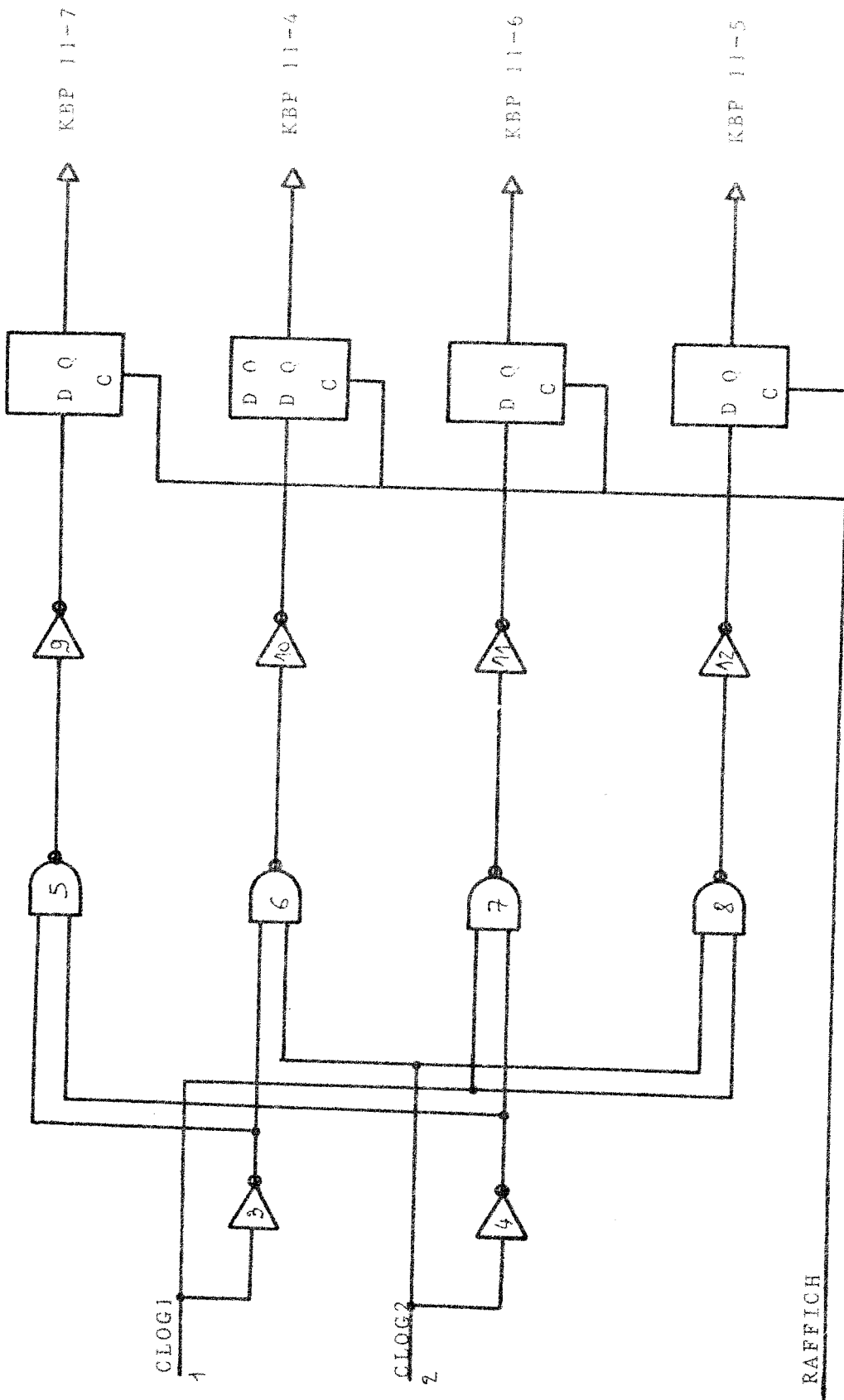


Figure A6

Schéma logique des commandes logiques

RAFFICH



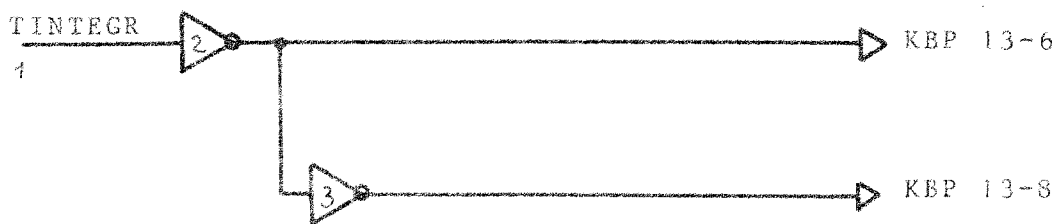


Figure A10

Schéma logique des constantes d'intégration

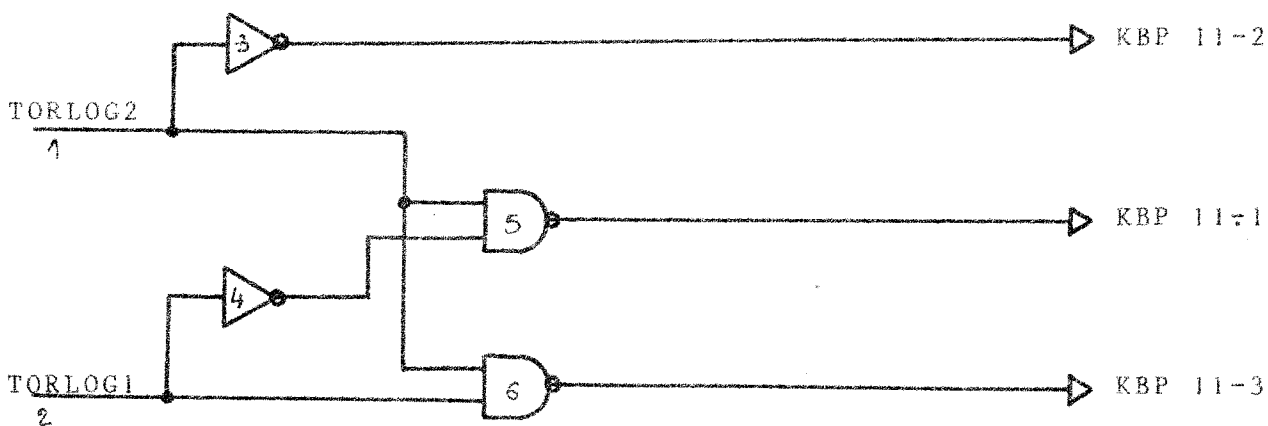


Figure A8

Schéma logique des horloges



On constate que les séquences de test à retenir sont :

110 - 001 - 000 - 010 - 100 - 011 - 111 -

Les organigrammes de détection des modules défectueux sont donnés annexe B (B.1 à B.9)

A.2.2 - Les fonctions logiques

L'analyse de la figure A.6 nous fournit le tableau de test suivant :

	00	01	10	11
3	0	0	1	1
4	0	1	0	1
5	1	0	0	0
6	0	1	0	0
7	0	0	1	0
8	0	0	0	1
9	0	1	1	1
10	1	0	1	1
11	1	1	0	1
12	1	1	1	0

Figure A.7 - Circuits touchés par les séquences de tests.

Les quatre séquences sont nécessaires pour détecter une erreur
Les diagrammes de détections sont donnés Annexe B (B.10 à B.14)

A.2.3. - Les horloges de la logique

La figure A.8 donne :

	00	01	11	10
3	0	0	1	1
4			1	
5			0	1
6			1	

Figure A.9 - Circuits touchés par les séquences de test

Les séquences retenues sont 11 et 10. La diagramme est en annexe B.16

A.2.4. - Temps d'intégration

Nous avons :

	1
2	1
3	1

Figure A.11 - Circuits touchés par les séquences de tests.

Le diagramme est donné Annexe B.16

A.3 - PLAQUES DE SELECTION DES POTENTIOMETRES

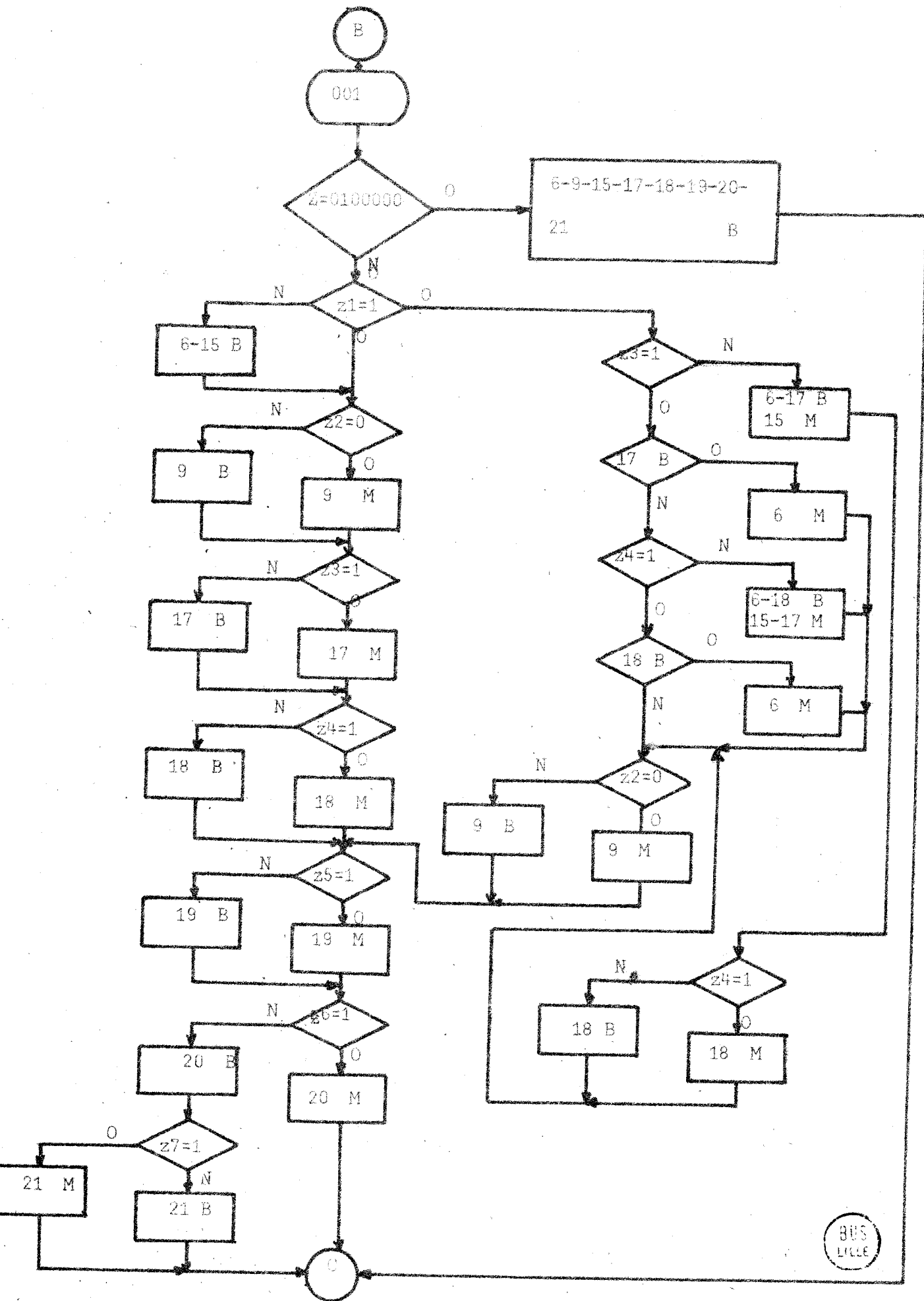
Nous devons essayer chacune des voies pour tester chacun des changements de logique. Les séquences seront donc successivement : 0 0 0 0, 0 0 0 1,, 1 0 0 1, et les informations de sortie seront :
1 0 0 . 0 0 0 . 0 0 0 0 , 0 1 0 . 0 0 0 . 0 0 0 . 0 ,
0 0 0 . 0 0 0 . 0 0 0 . 1.

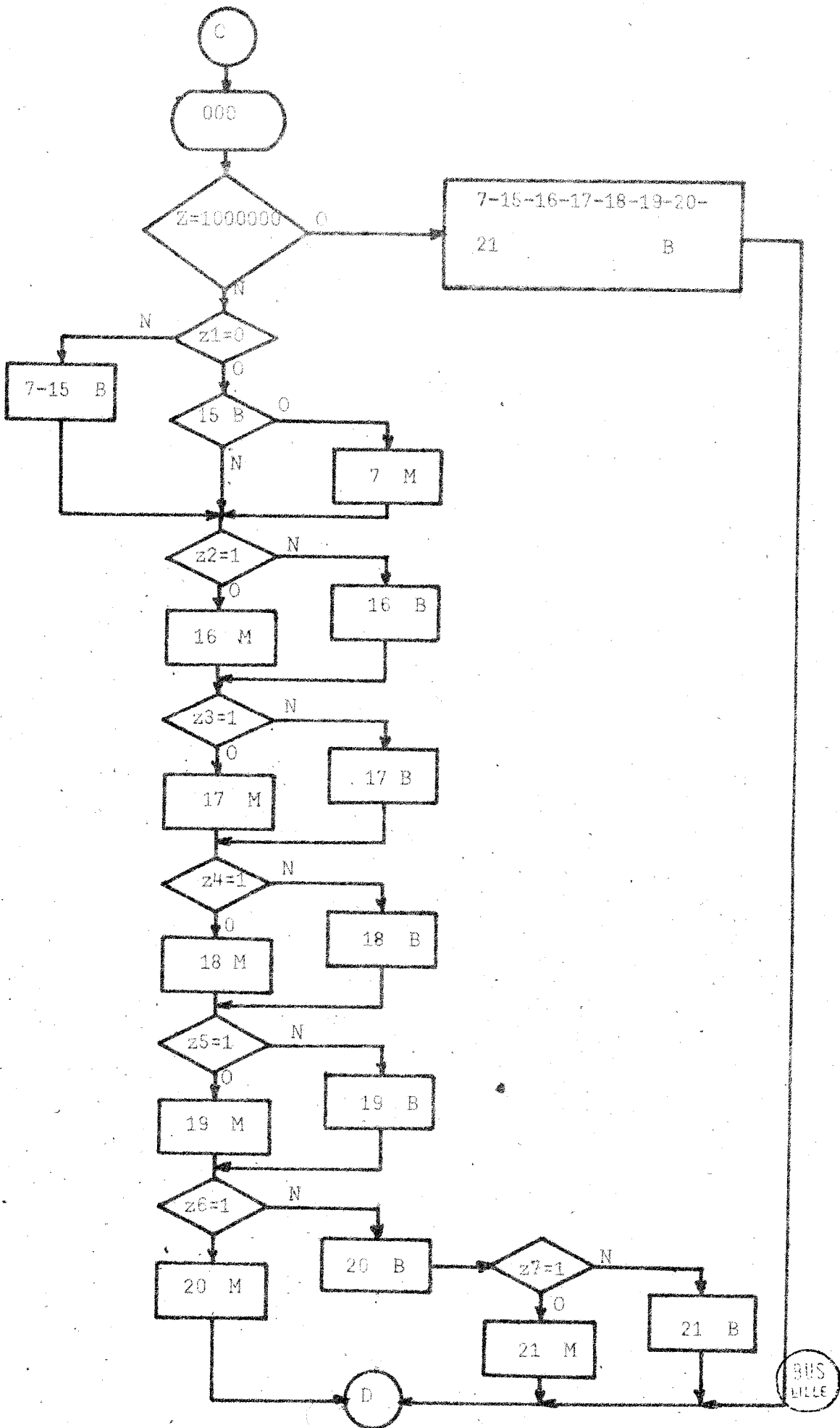
A.4 - PLAQUES DE SATURATION

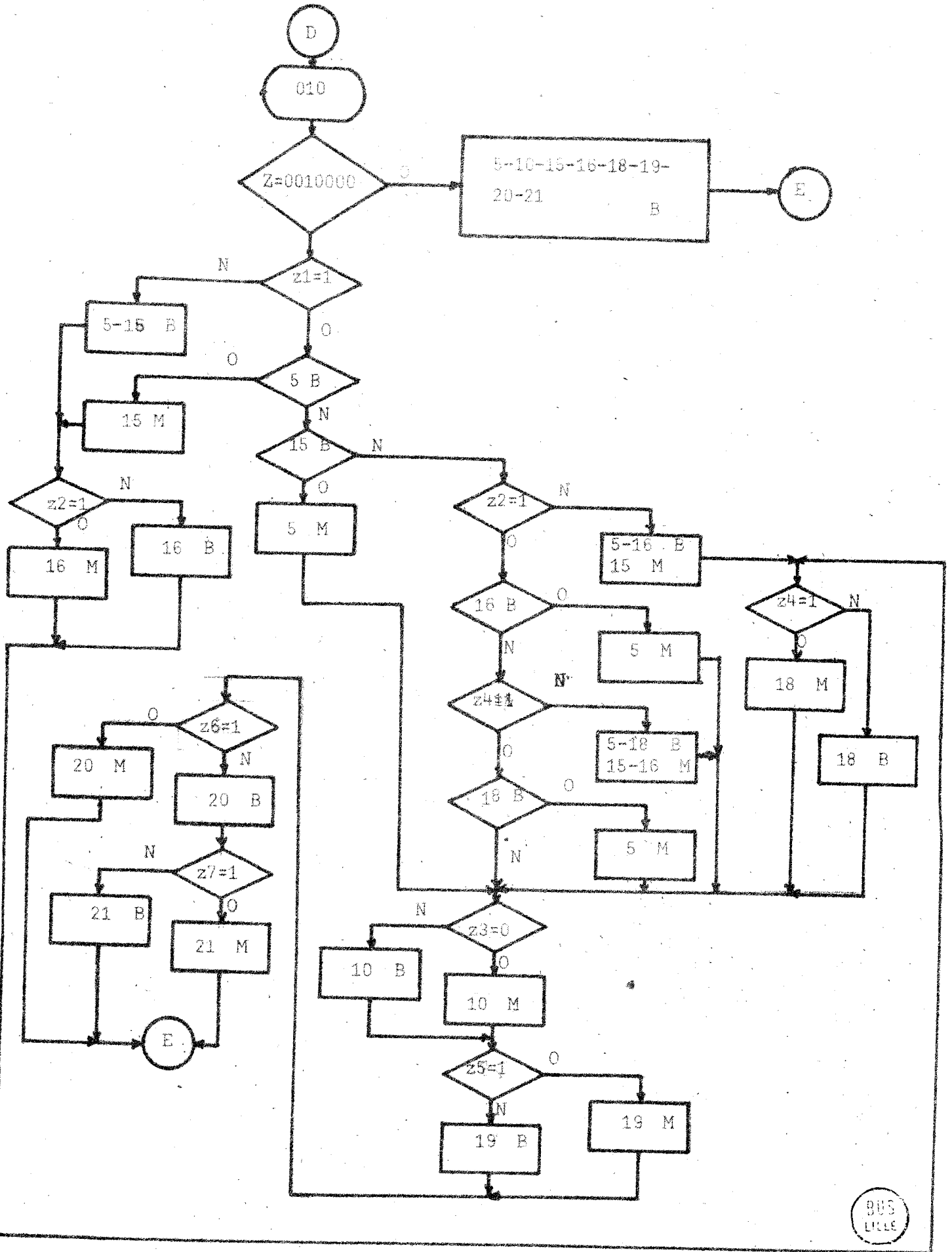
Ici encore, chacun des circuits de mise en forme du signal SATUR doit être testé. Ce test sera fait par groupe de cinq circuits afin de vérifier plus rapidement la plaque.

ANNEXE B

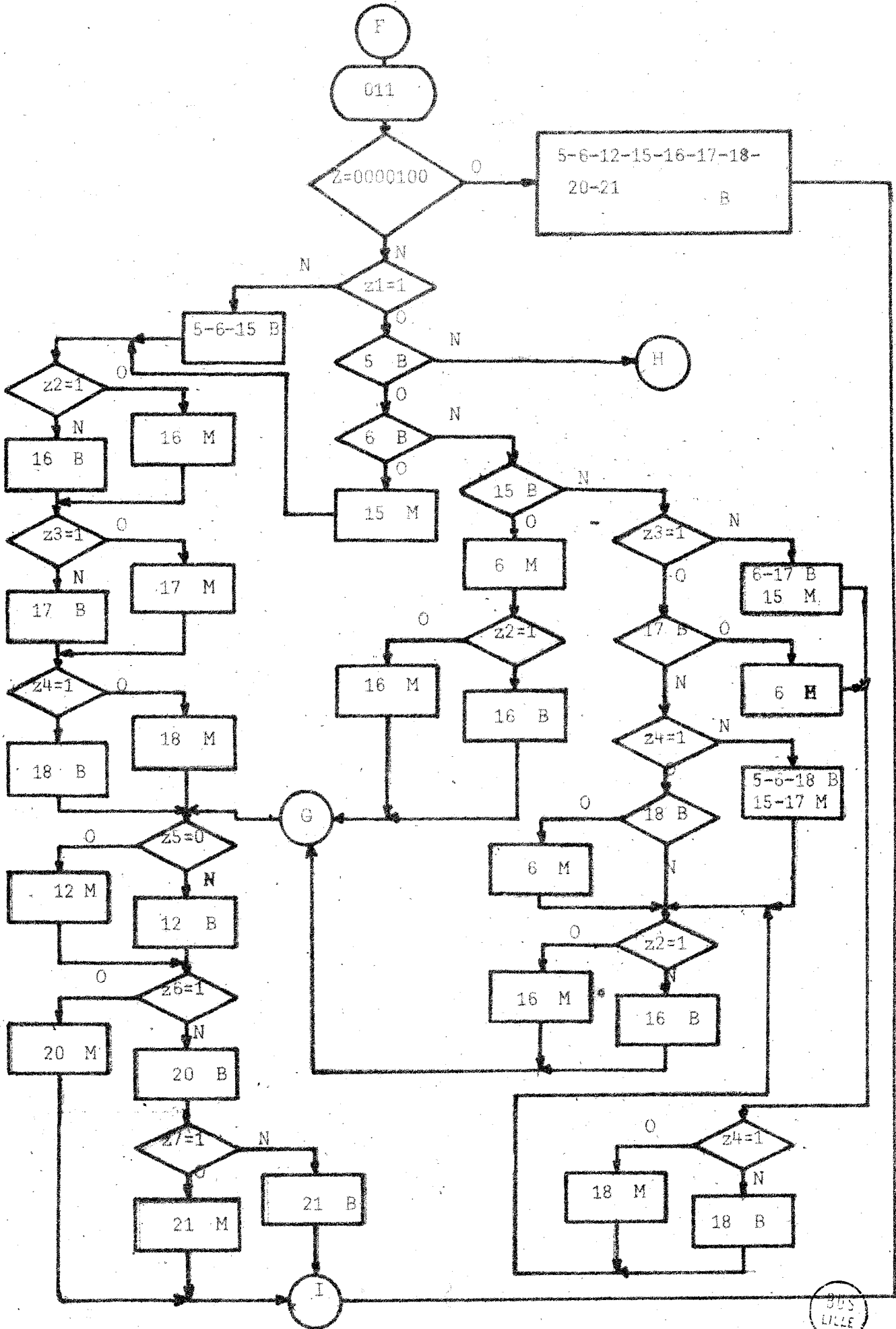
ORGANIGRAMMES DE DETECTION DES MODULES DEFECTUEUX.

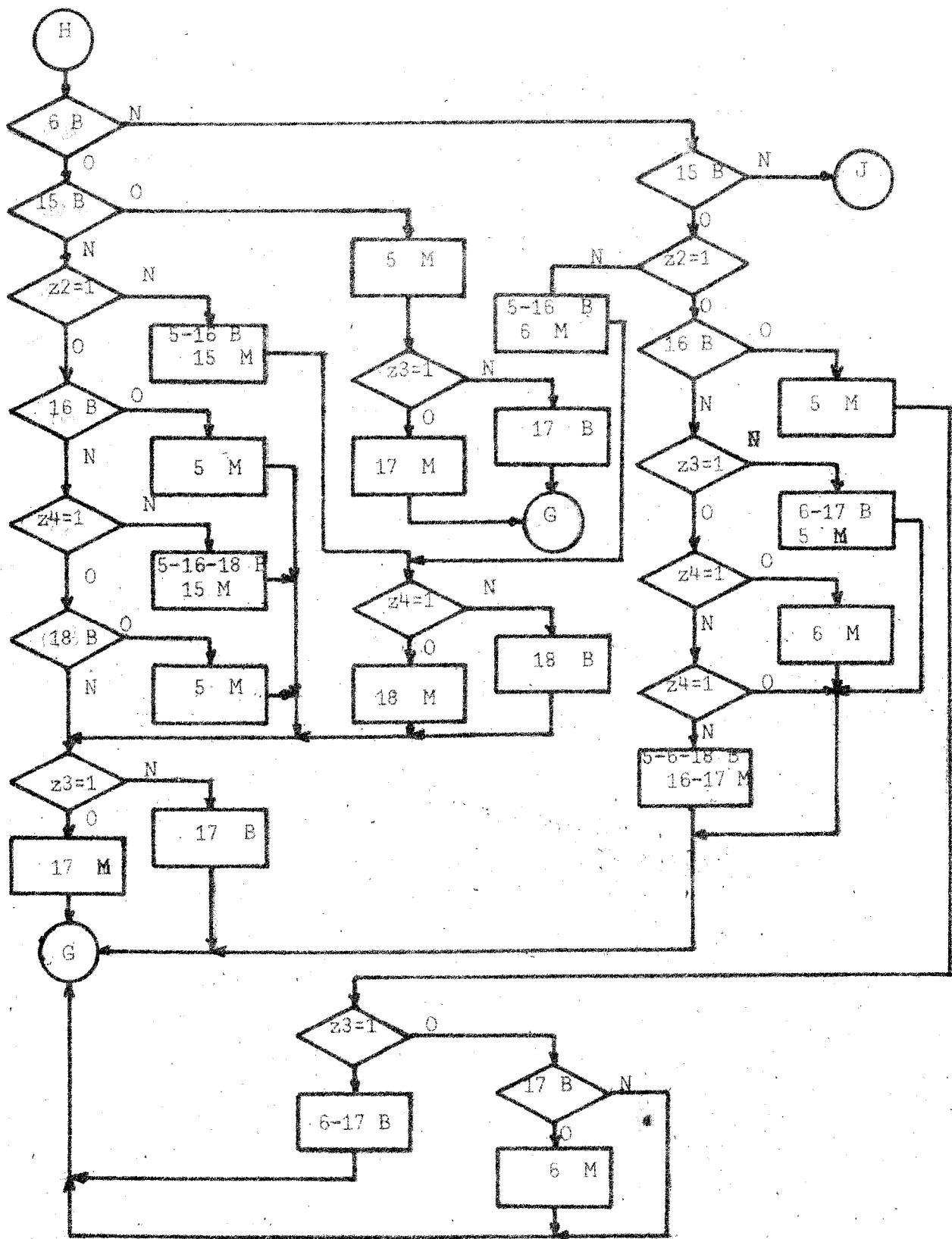


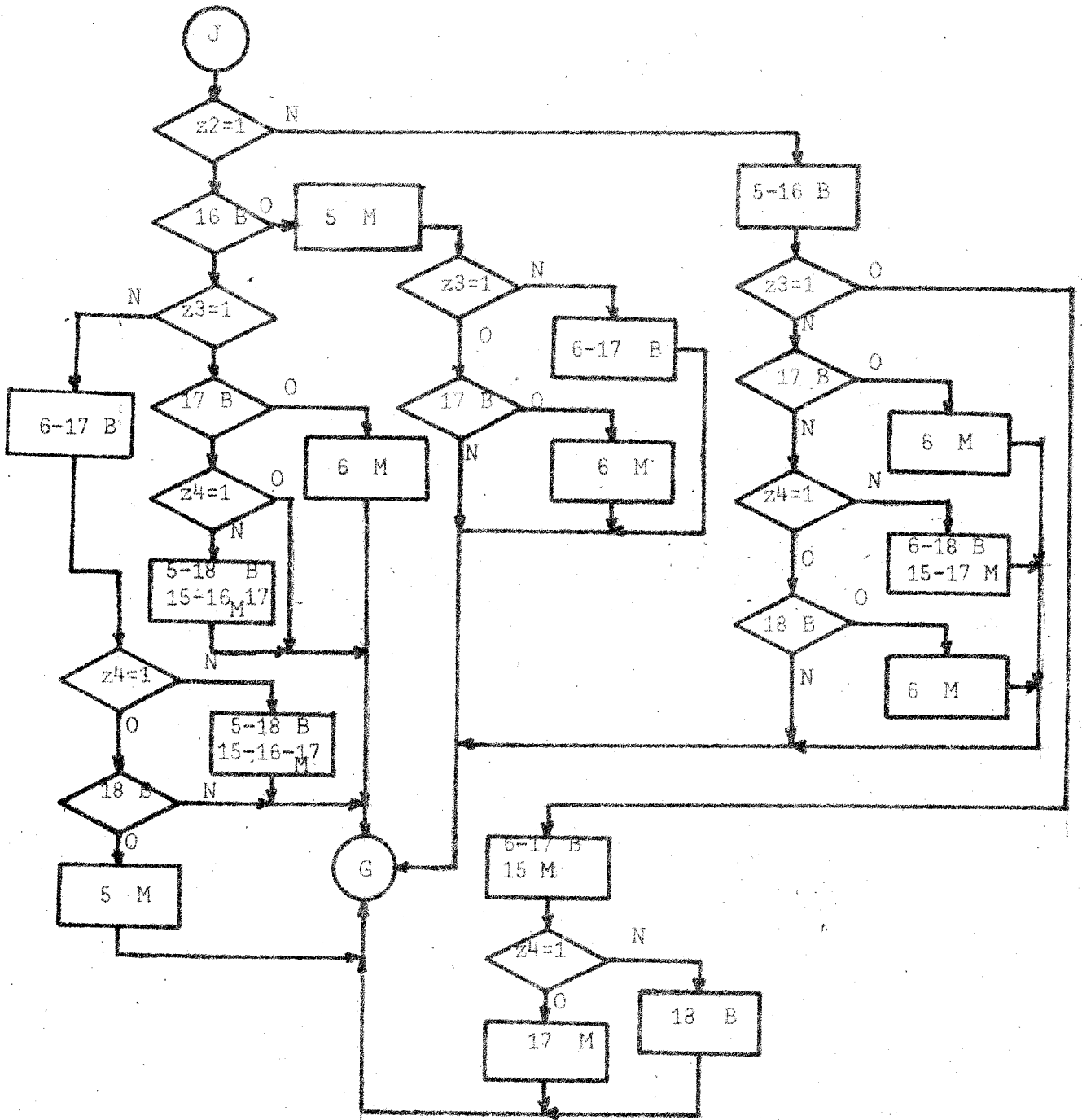


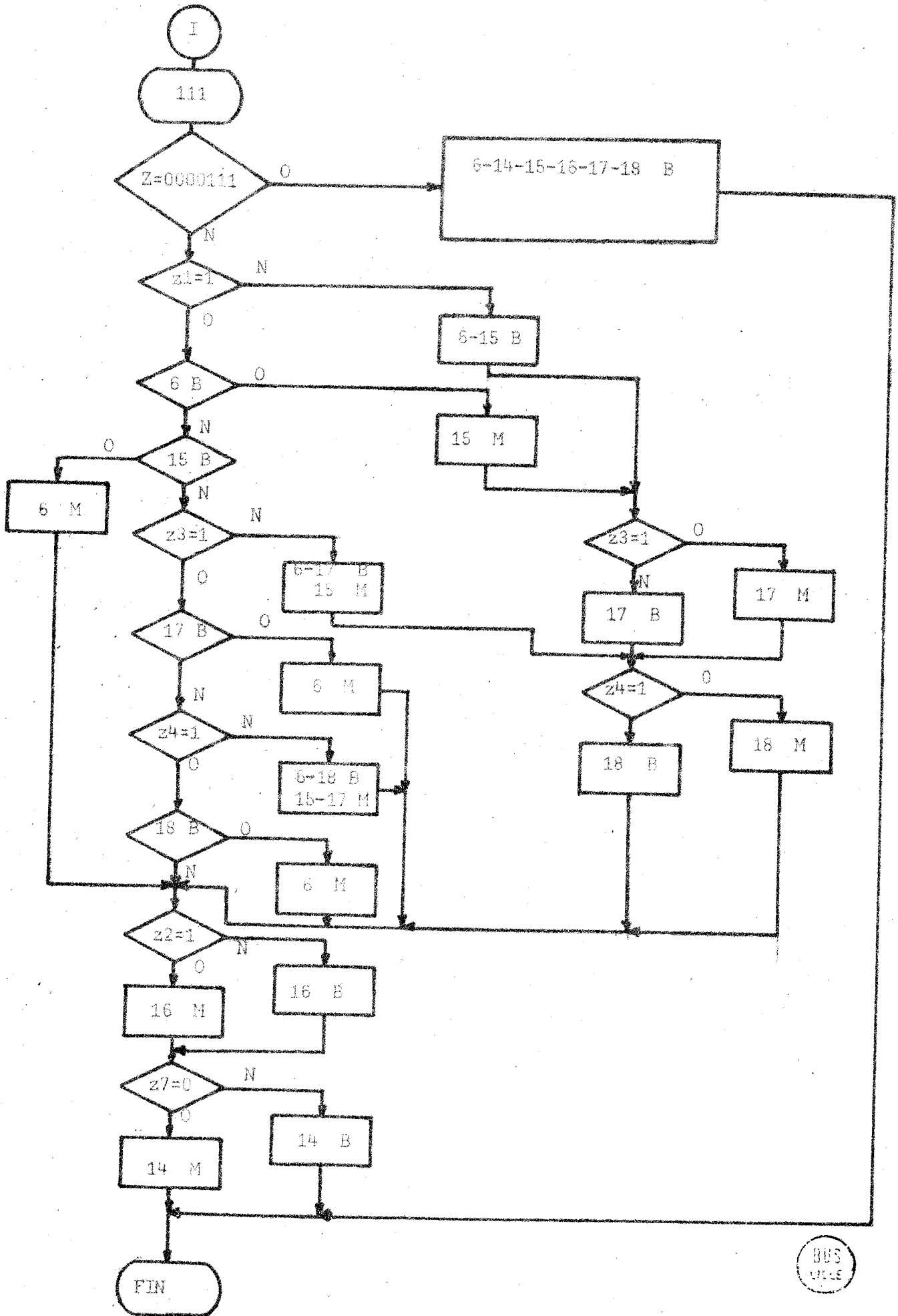


BUS LULE

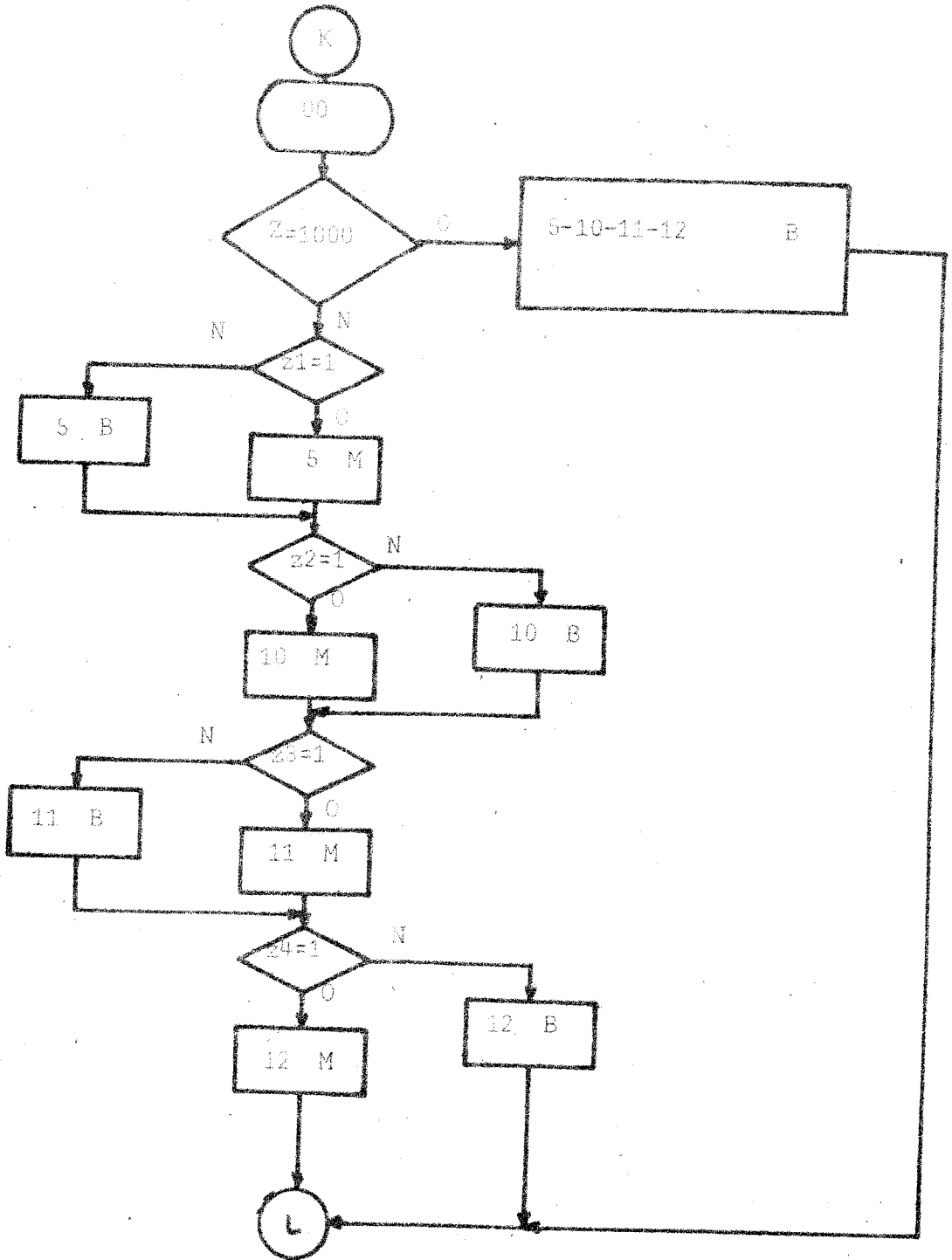




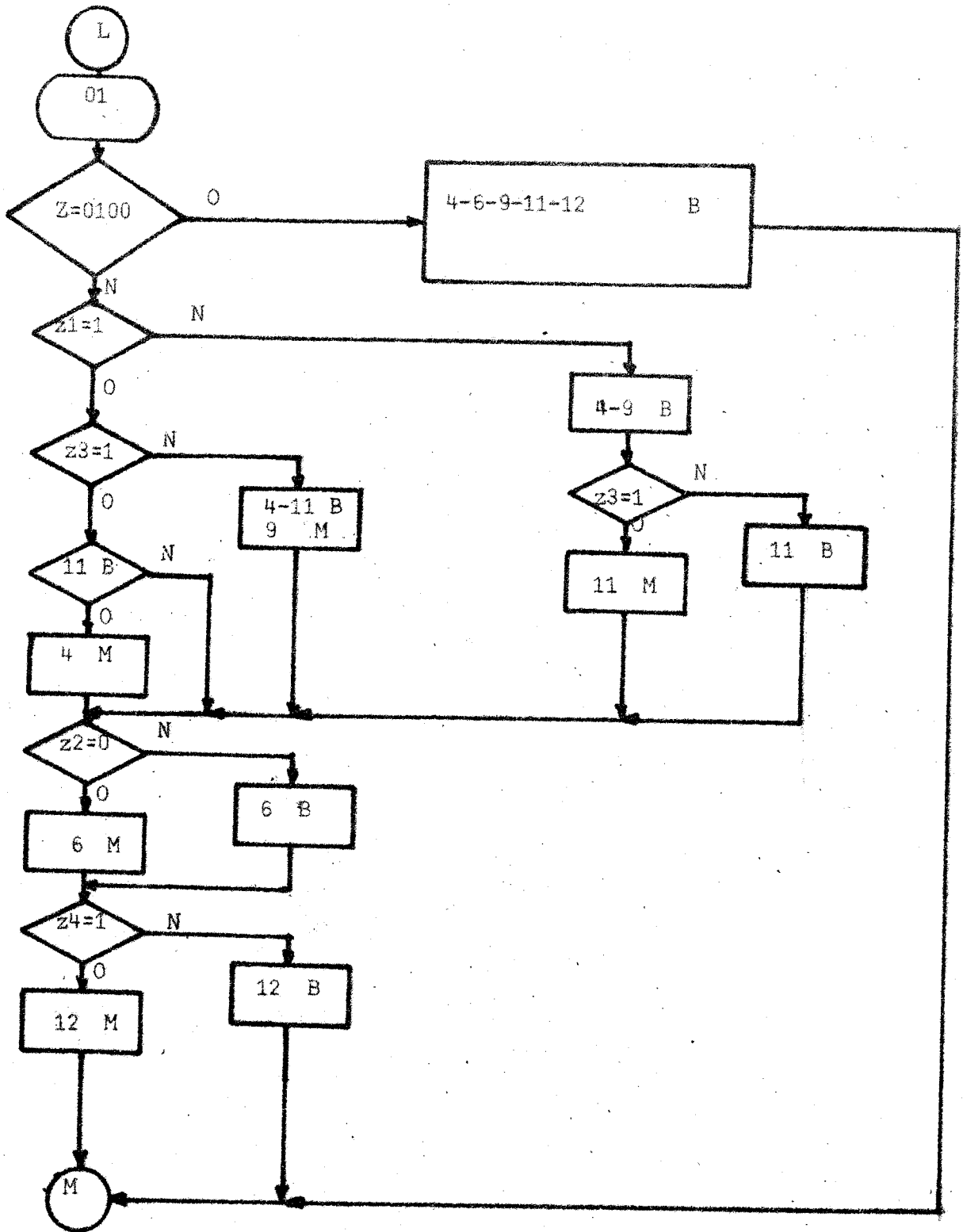




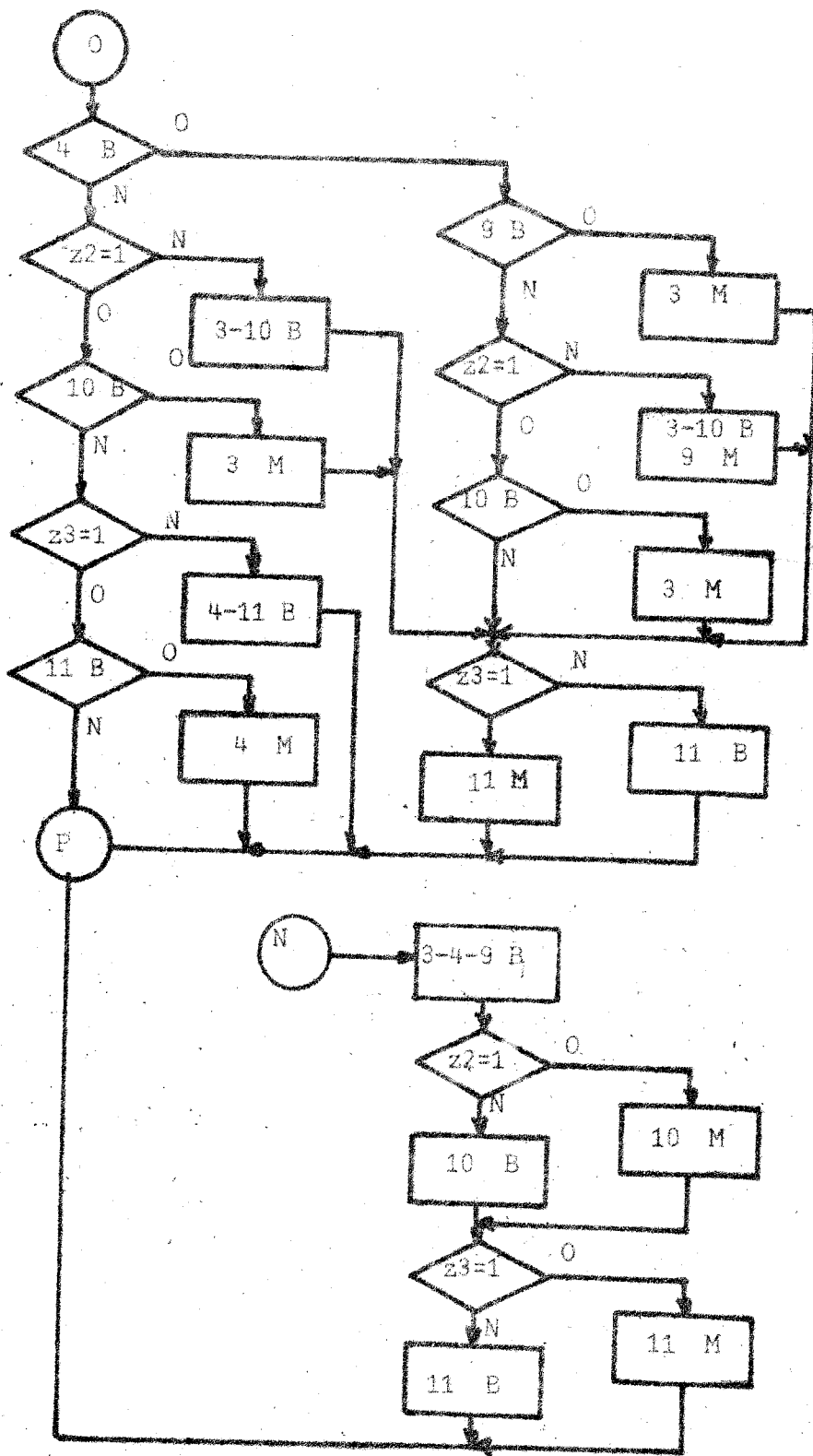
BUS
V. 1.5

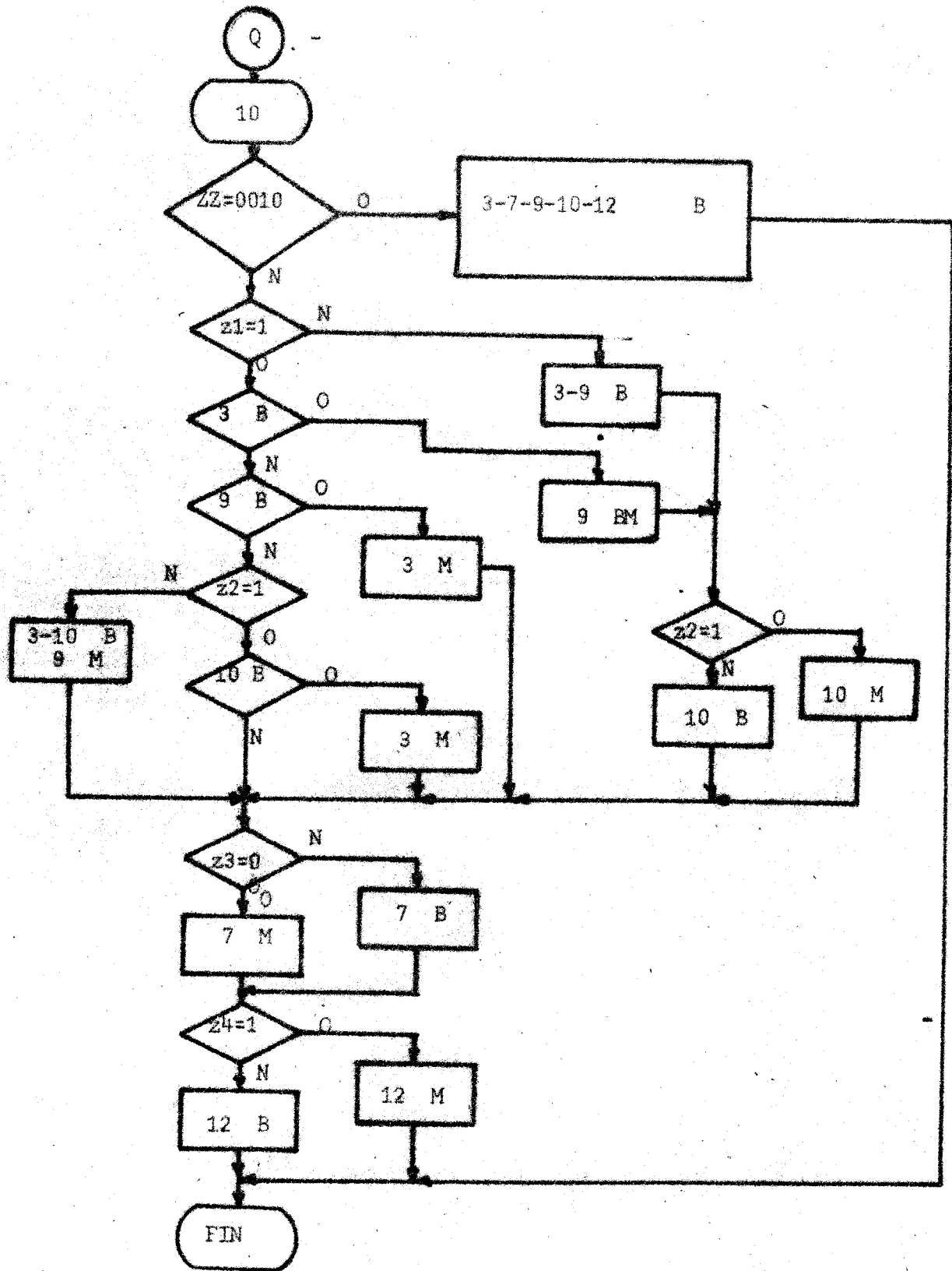


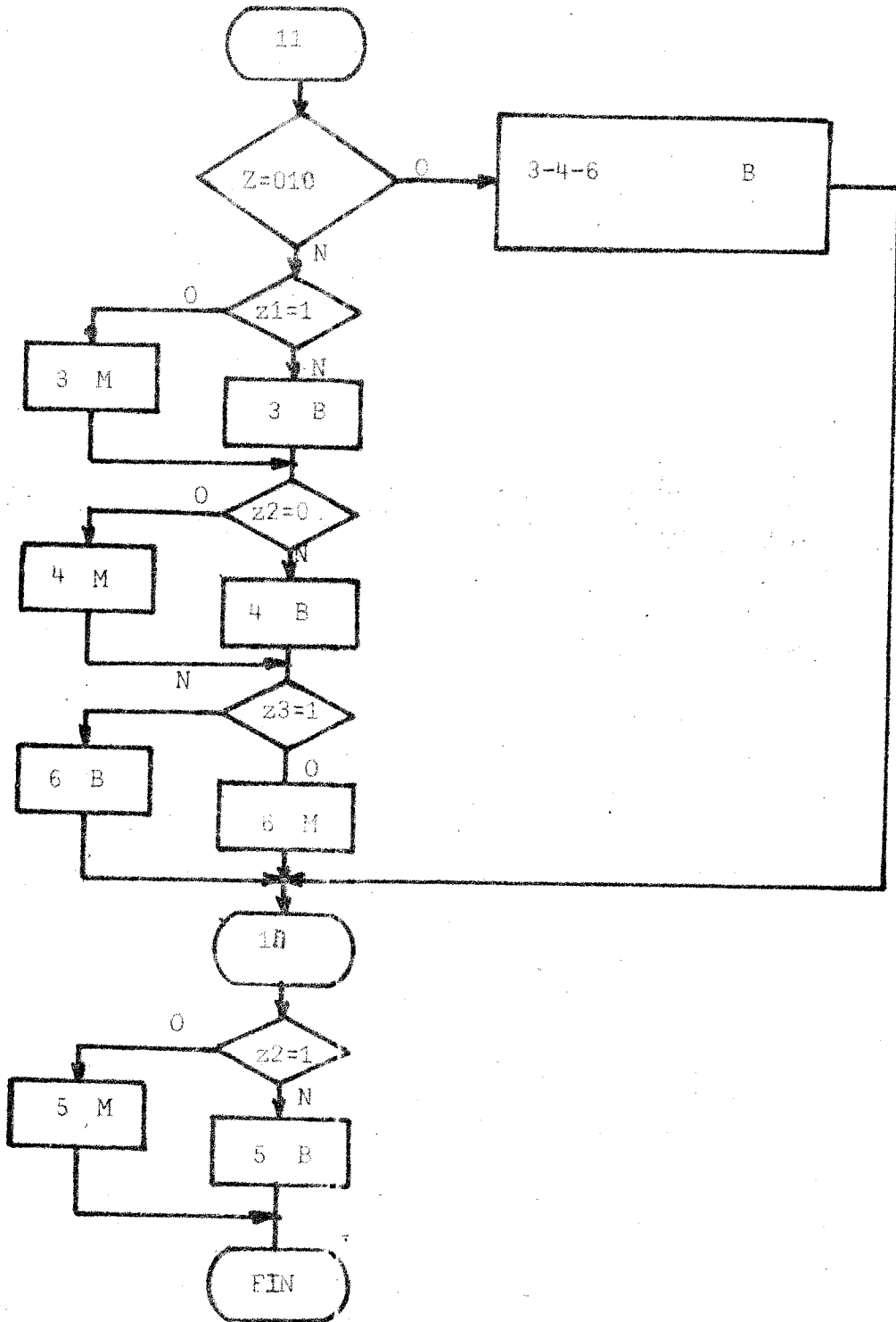
BUS
TITLE

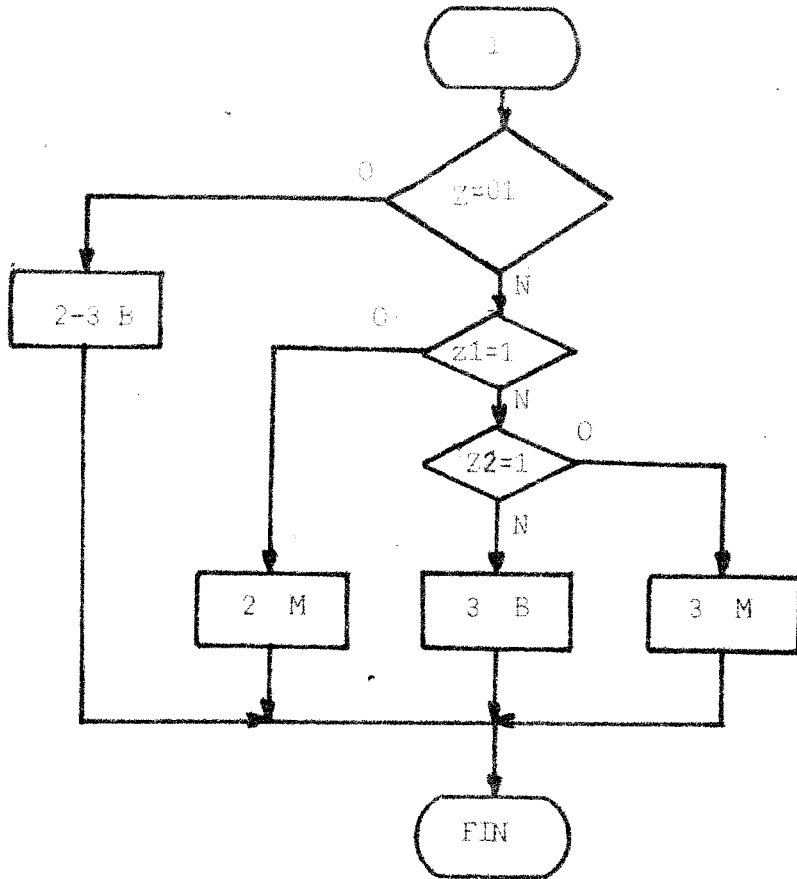


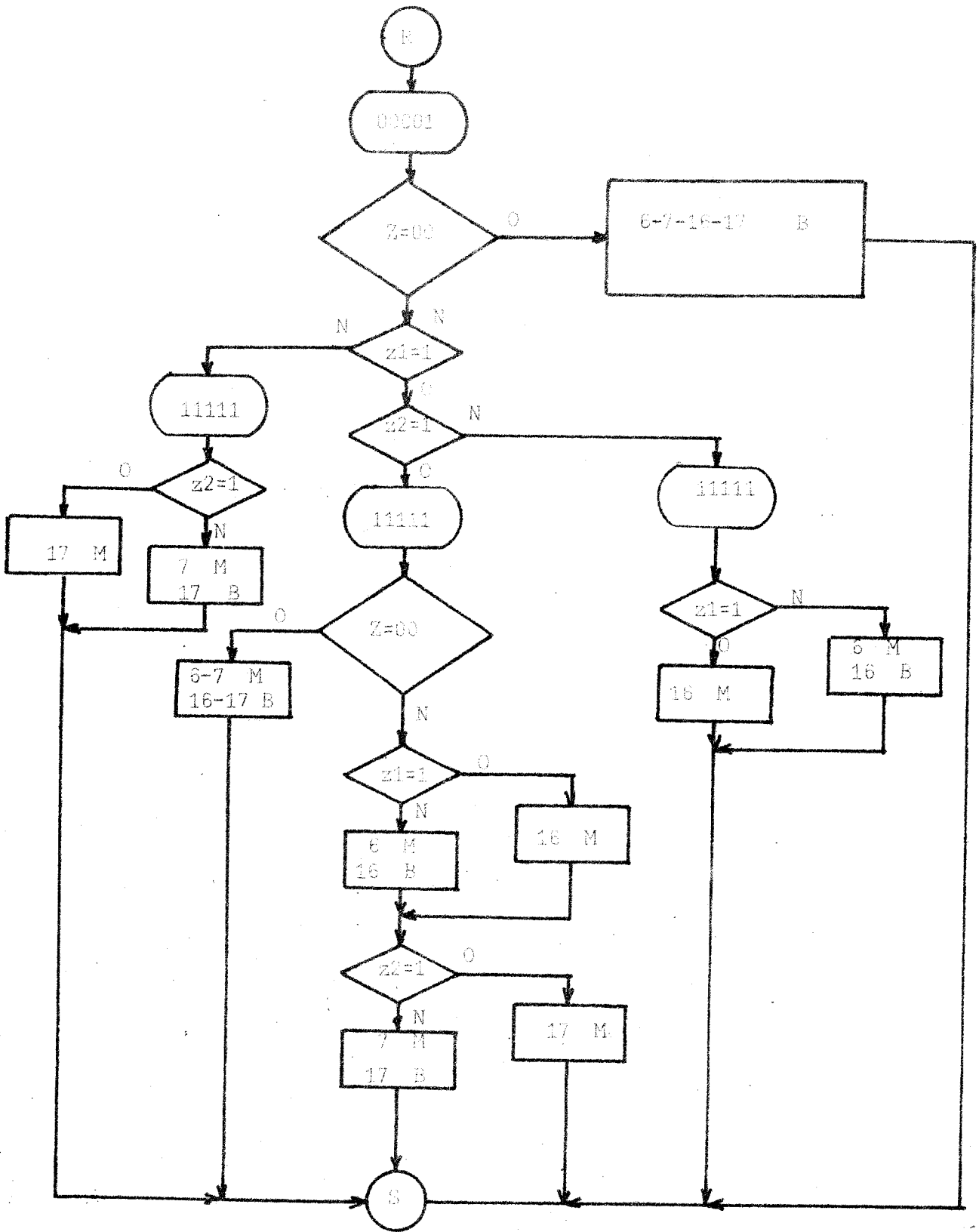
305
LILLE

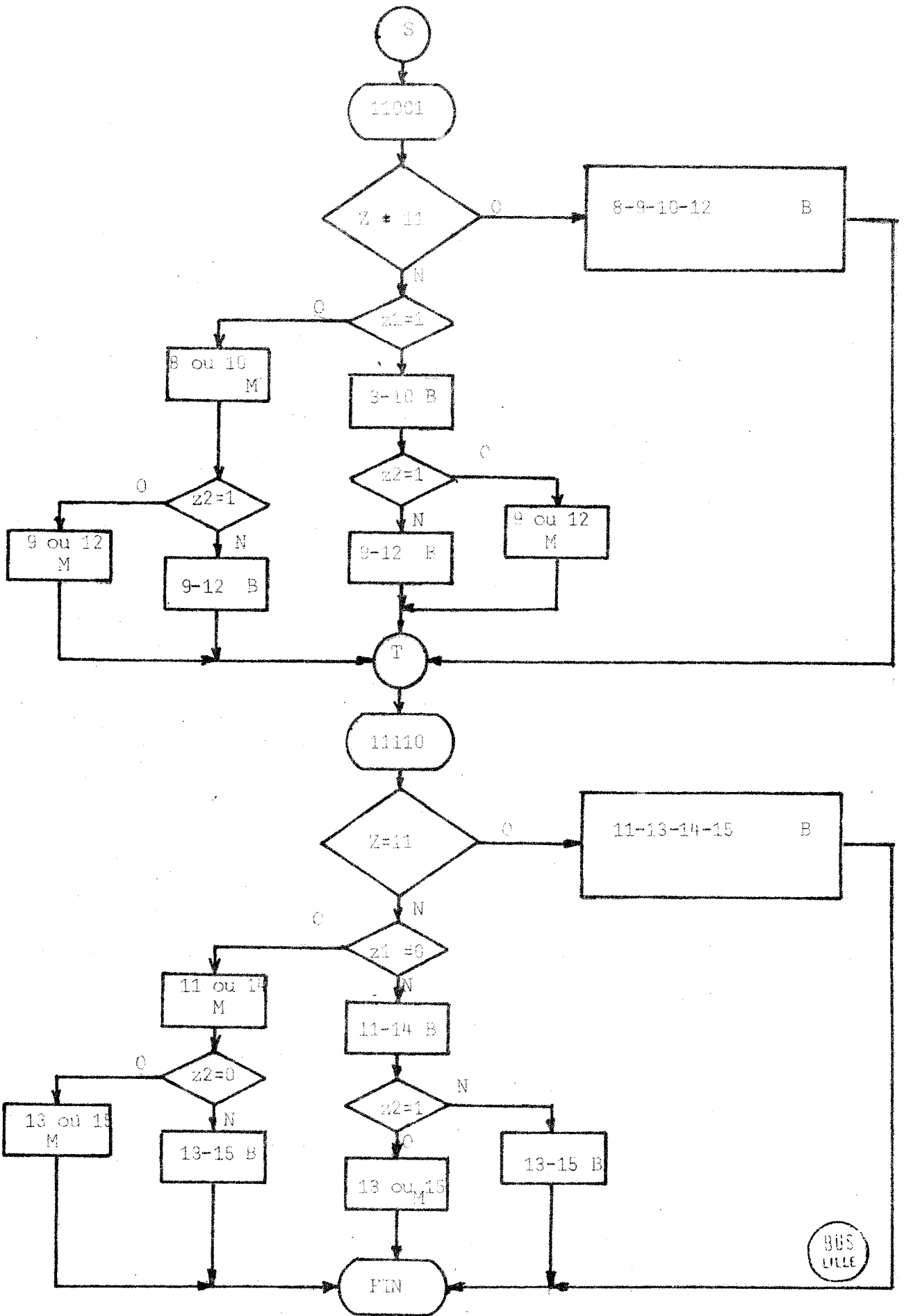






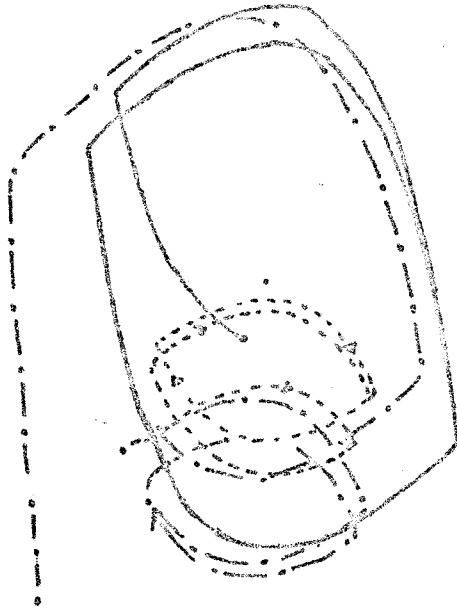






ANNEXE C

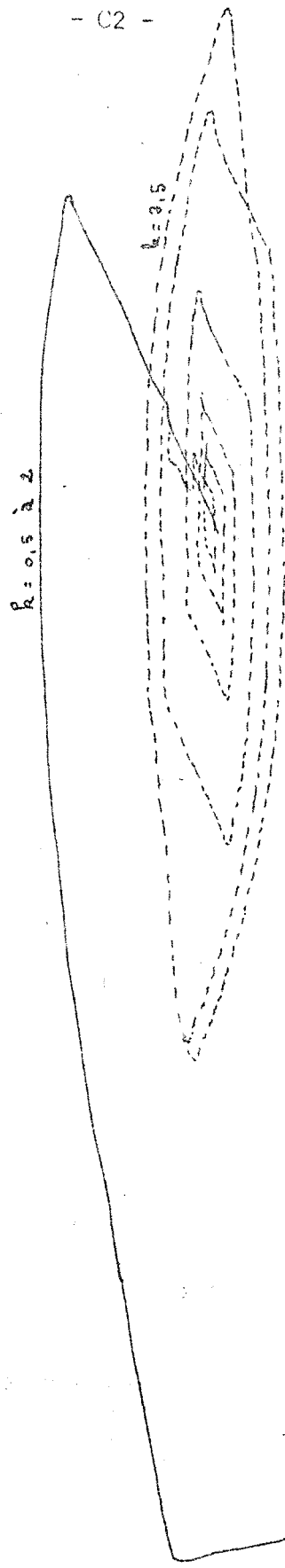
ORGANIGRAMME et COURBES DE L'EXEMPLE 1.



Regime saturé: évolution dans le plan de phase y, y'

suivant les conditions initiales.

FIG. A1.



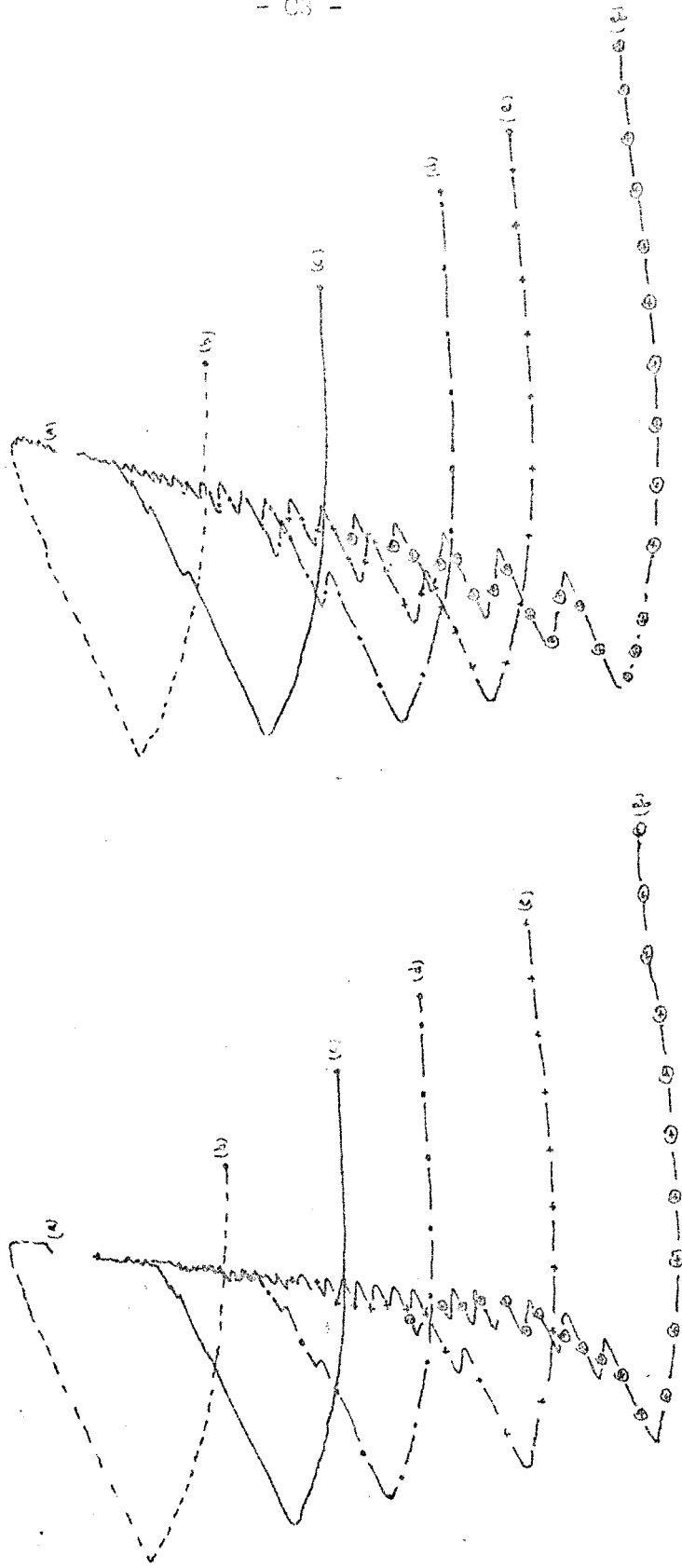
$A=1$
 $\lambda=0,46$
 $I=I'=1$
 Evolution en fonction de K
 Cycle pour $K=2,5$



FIG. A2.

K=0,125
A=1

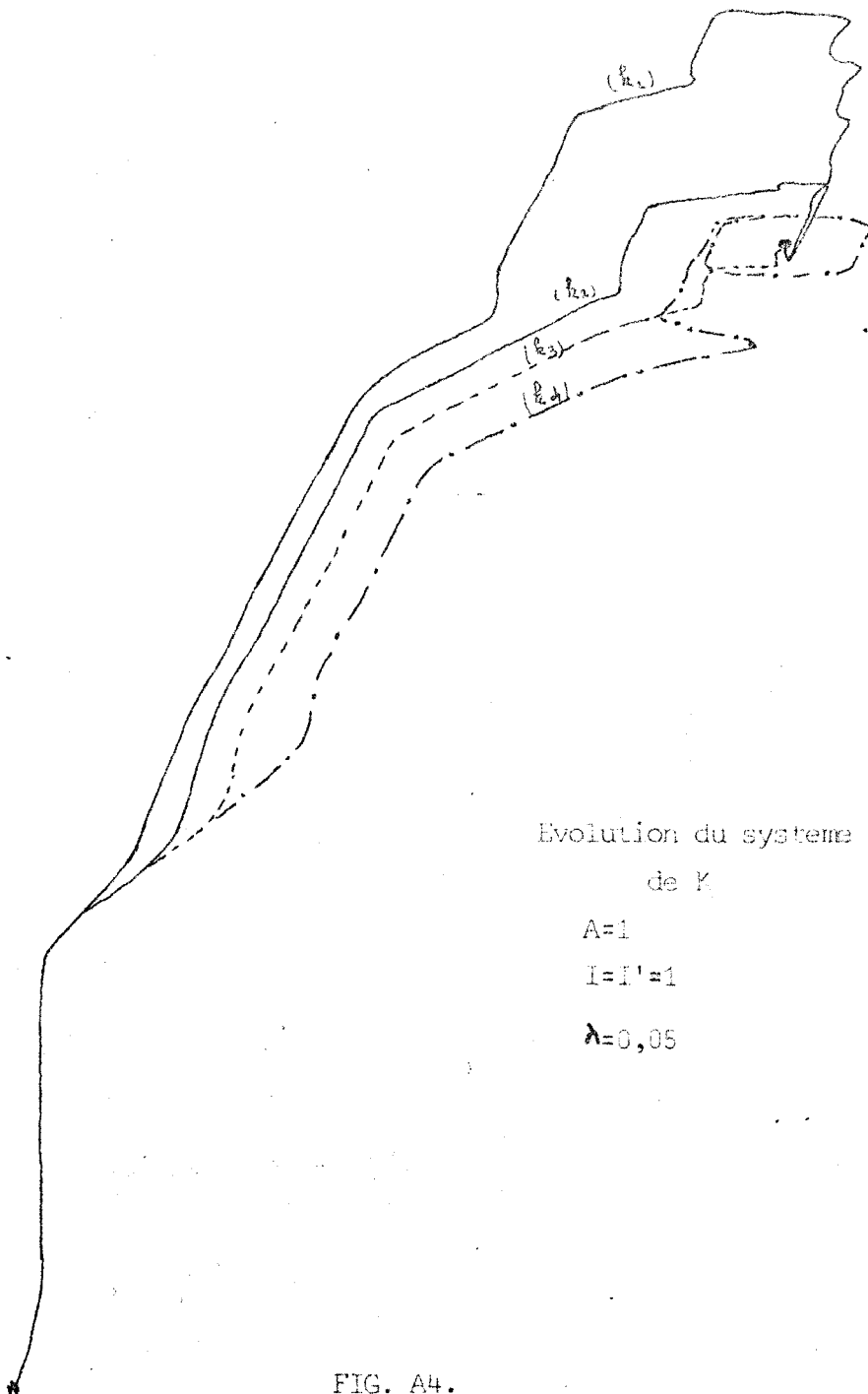
M=1,050
A=1,020



Composantes... du système à K, A-constante.
pour différentes conditions initiales.

FIG. A3.





Evolution du systeme en fonction
de K .

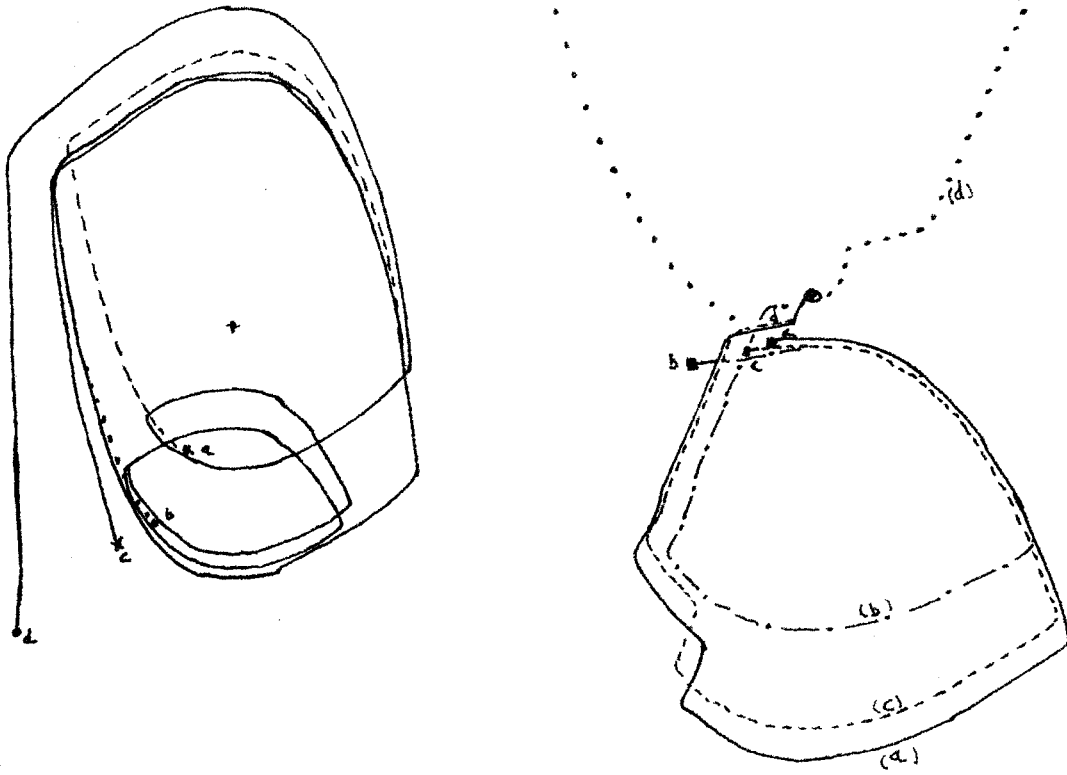
$$A=1$$

$$I=I'=1$$

$$\lambda=0,05$$

FIG. A4.





$$\dot{A} = 1,97$$

$$K = 0,8$$

$$\lambda = 0,29$$

$$T = 1$$

(A)

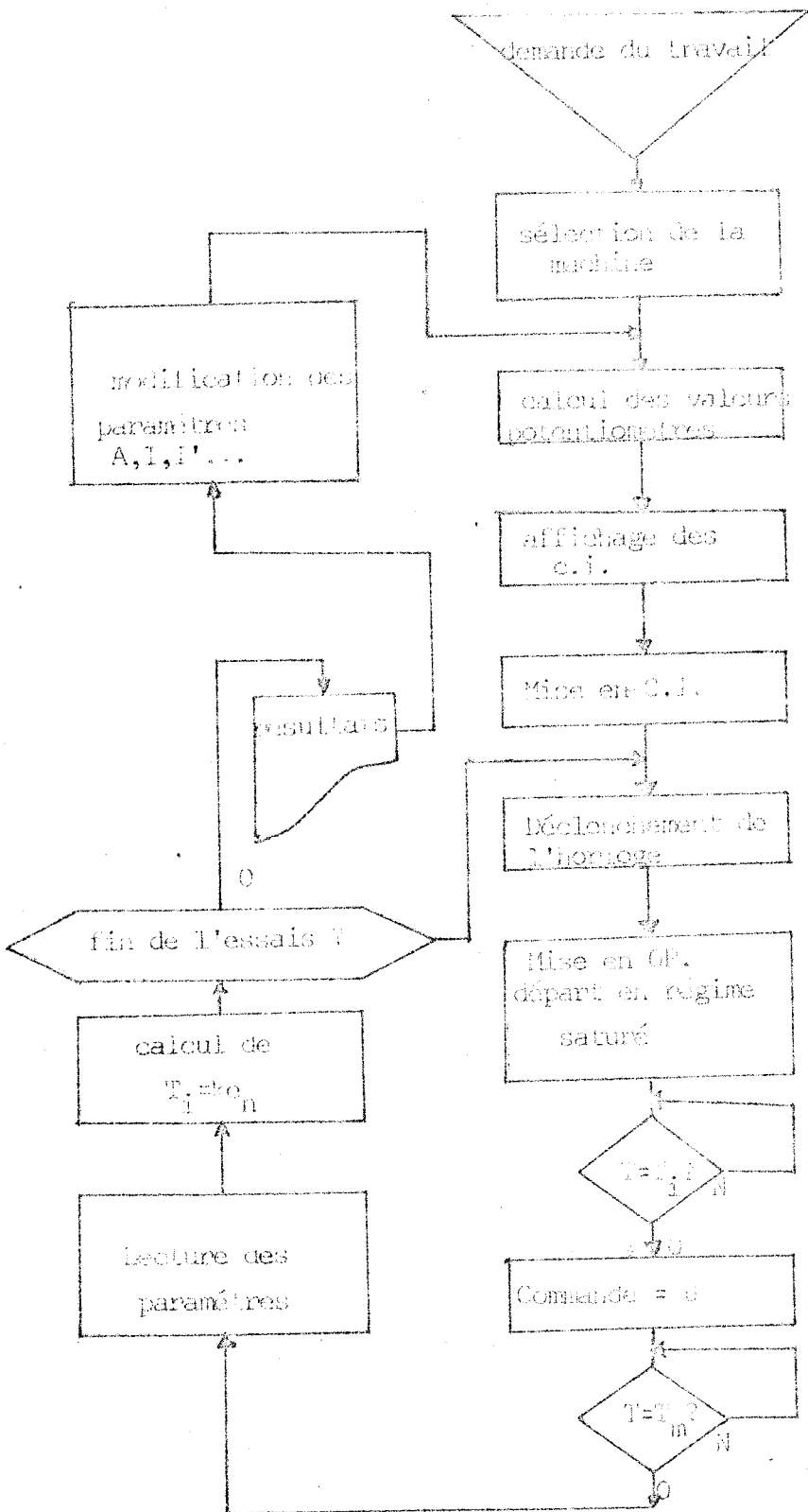
(B)

Evolution en fonction des conditions initiales
plan y, y'

(A) régime saturé type de cycles.

(B) régime non saturé.

Figure - A5.



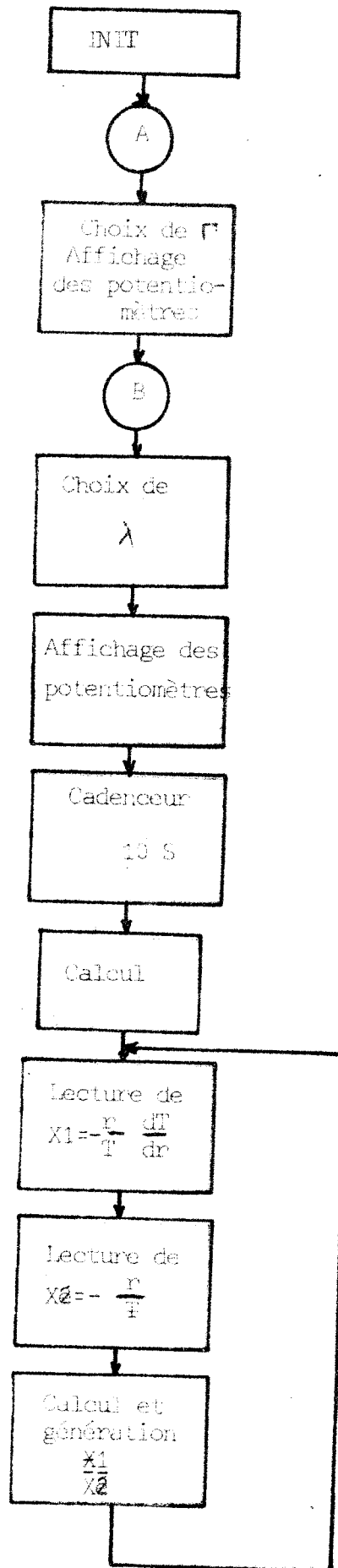
Organigramme de la simulation.

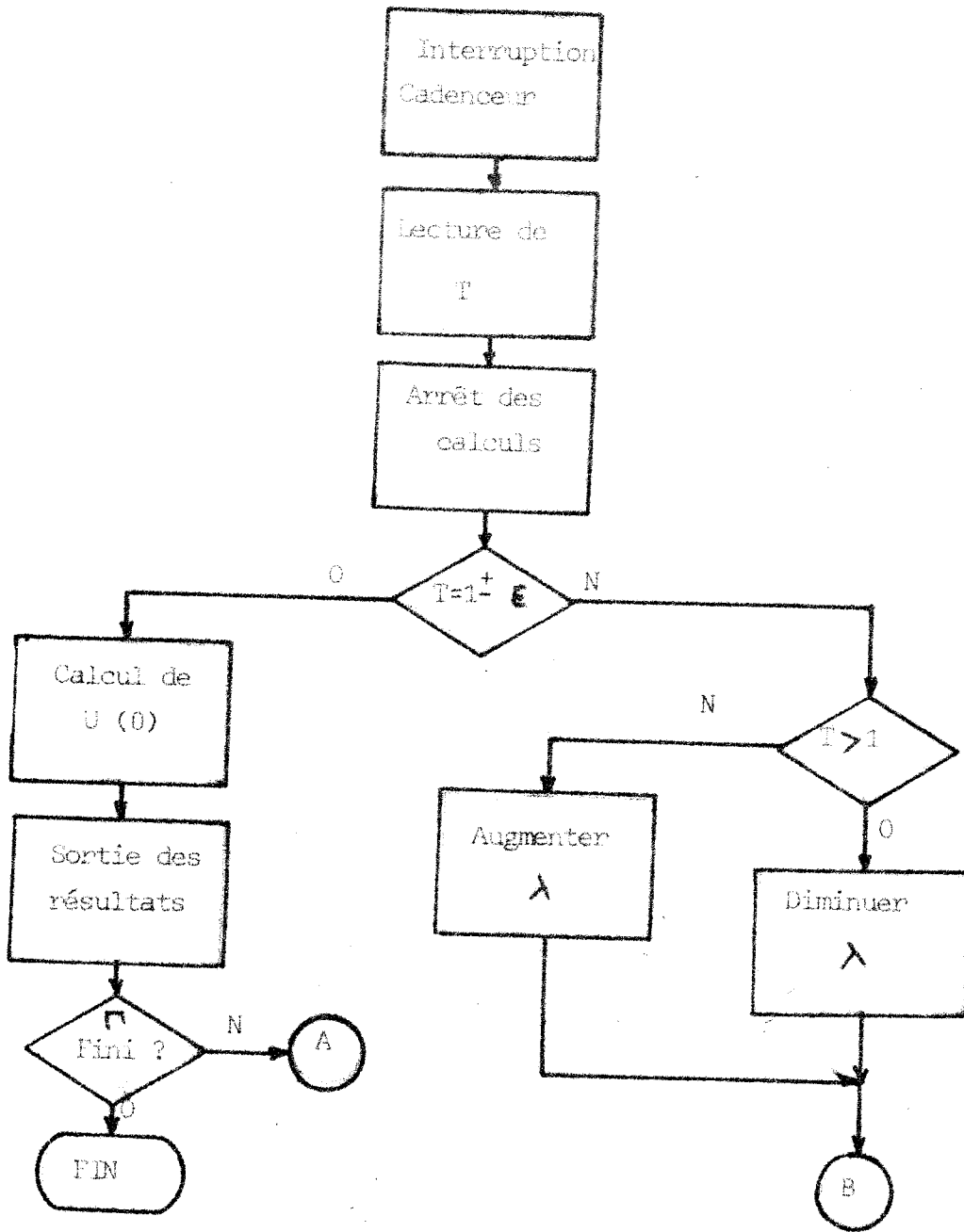
FIG. A6.

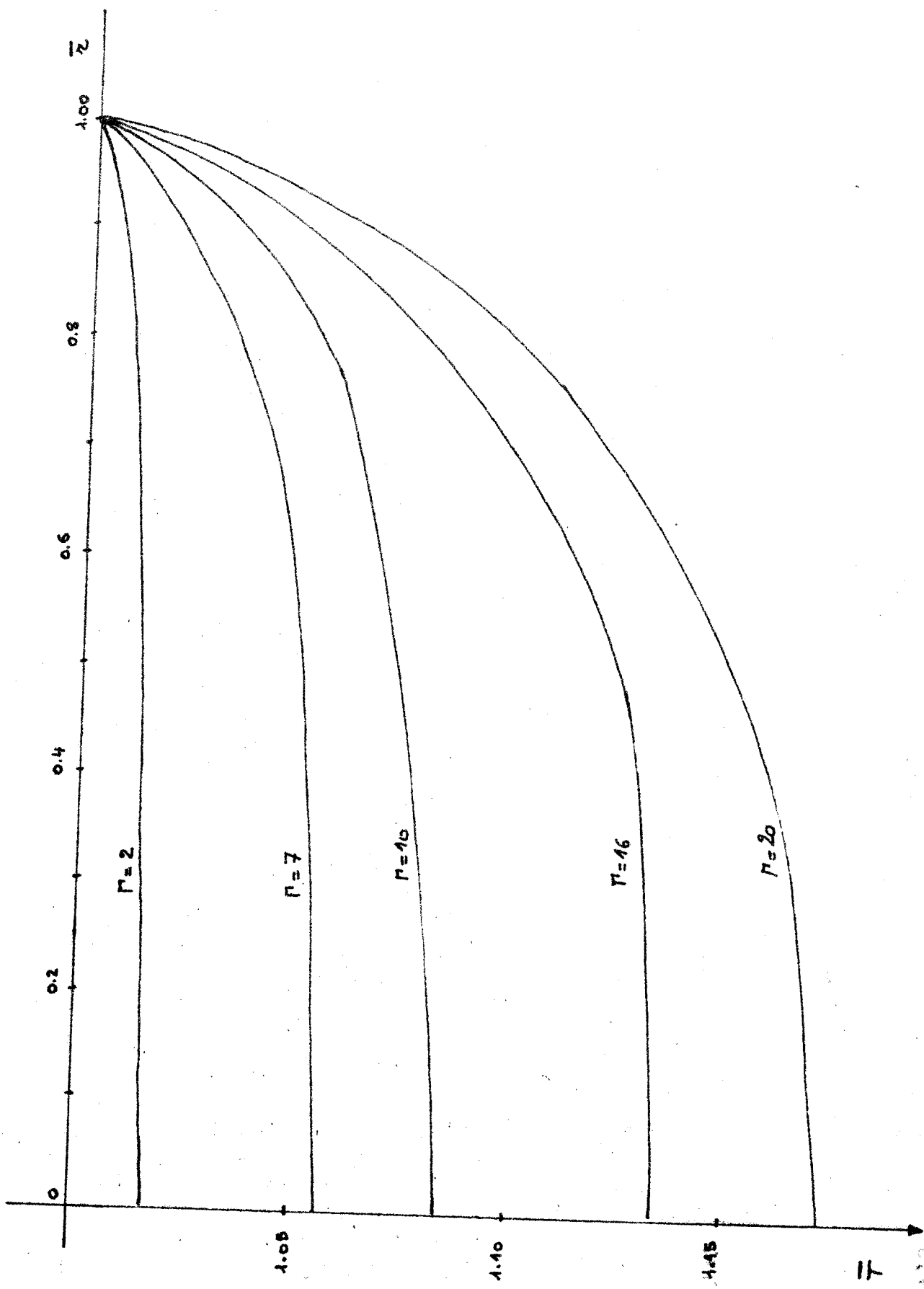


ANNEXE D

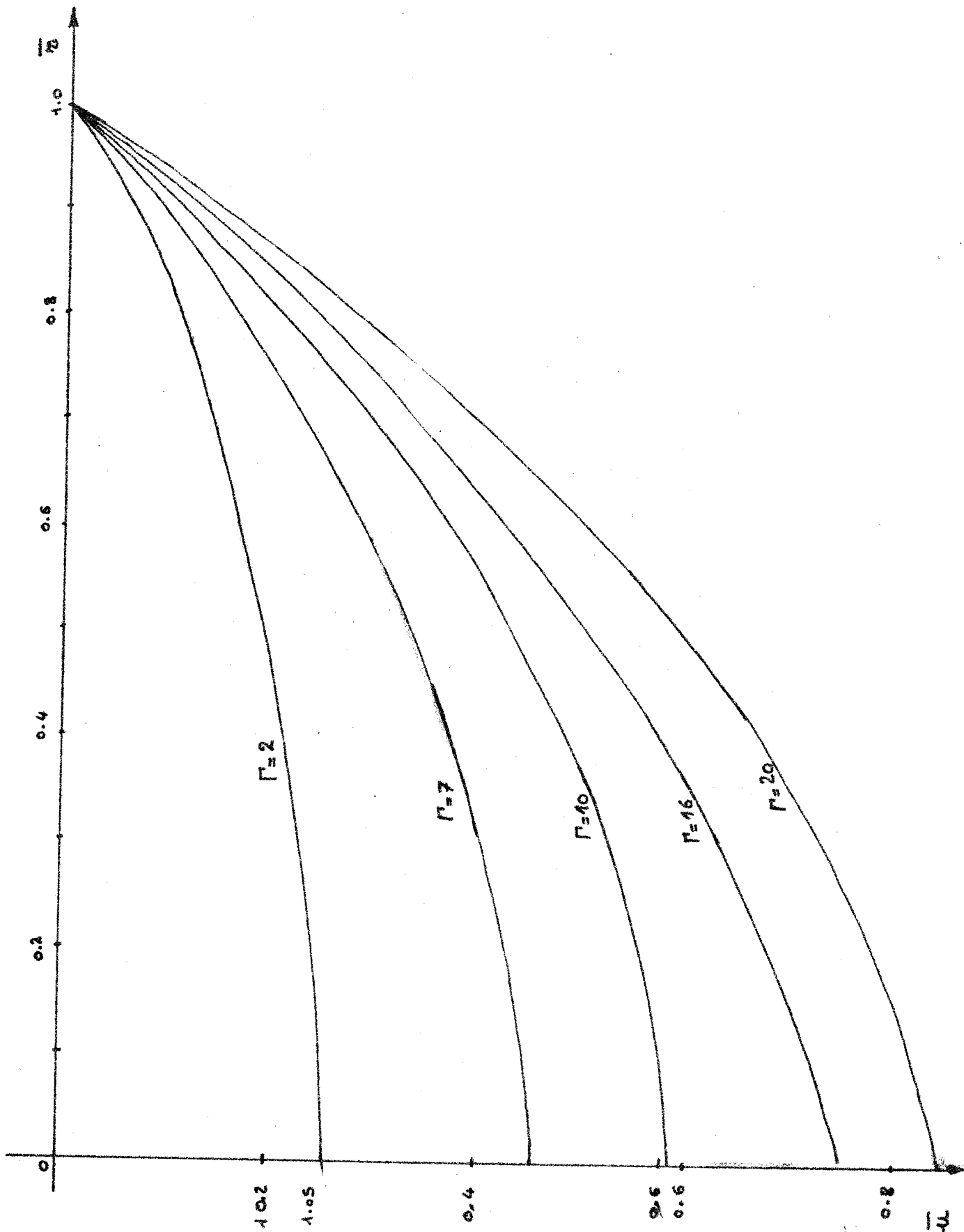
ORGANIGRAMME, PROGRAMME et COURBE de l'EXEMPLE 2.







1.20



```

SEC  ETUDE2;
INIT  ( )      <MEMOIRES RELAIS >;
HYBD  (1,SP,CL,T1,H6) <INIT DU SYSTEME >;
NIVAL  (9,CAPEN,8,TRAHOC F1* R0* );
NIVAL  (11,RDEF,9, ) ;
DEBUT: POT  (1,POTNUM,POTAB) <AFF DE P1 >;
      POT  (1,POTNUM+1,POTAB+1) <AFF DE P7 >;
      POT  (1,POTNUM+2,POTAB+2) <AFF DE P8 >;
      POT  (1,POTNUM+3,POTAB+2) <AFF DE P9 >;
HODEBU:CAV  +1;
      RA   GAMMA      <INIT DE GAMMA >;
ACAMA: MPV  +40;
      GNE  +4;
      RA   VALEUR;
      POT  (1,POTNUM+4,VALEUR) <AFF DE P2 >;
      CA   GAMMA;
      IRV  BACVF      <SF DE MACINE CABREE >;
      IRV  PEUPAS;
      MPV  +40;
      GNE  +4;
      RA   VALEUR;
      POT  (1,POTNUM+5,VALEUR) <AFF DE P3 >;
      RZ   INFLAM;
      CA   BONINF;
      RA   NOVLAM;
      RA   SUPLAM      <PREP DU CRITERE SUR LAMBDA >;
ALAMDA:RA  LAMRDA      <VALEUR DE T(O) >;
      AD   MILLE;
      RA   TOTO;
      MP   TOTO;
      GNE  +18;
      RA   LAMKRE;
      CA   TREIZ8;
      DNE  +18;
      DV   LAMKRE;
      RA   VALEUR;
      POT  (1,POTNUM+6,VALEUR);
      CA   LAMKRE;
      MP   TOTO;
      GNE  +18;
      RA   TOTO;
      CA   QUAT9;
      DNE  +18;
      DV   TOTO;
      RA   VALEUR;
      POT  (1,POTNUM+7,VALEUR ) ;
ENFIN: RZ   COMPT;
      RZ   VALEUR;
      CONVER (VALEUR) <CONVERSION N/A >;
      HYBD  (1,CI,CL,0,0) <MISE EN CI DU SYSTEME >;
DEPART: HORLOG (S10,VOIE2) <DEPART DE LA PERIODE DE CALCUL>;
      HYBD  (1,HOP,CL,0,0) <RESOLUTION >;
AVANT: CX   AIGUI;
      IR X  VAZY      <GENERE SI X=0,TRAVAU SI X=1 >;
GENERE: CHAINE (TABLOC) <ACQUISITION DE DONNEES >;
      CA   TABLEC+1 <VALEUR DE R >;
      MPV  +10;
      ADV  -1;

```



```

NRV AVANT;
CA TABLEC+4 <X1 >;
DNE +18;
DV TABLEC+3 <X2 >;
RA VALEUR;
CONVER (VALEUR) ;
IRV AVANT;
TRAHOG:CAV +1 <FIN DE TRAVAIL >;
RA AIGUI;
FINTRA ( ) <RETOUR AU PROGRAMME >;
TRAVAU:RZ AIGUI;
HYBD (1,SP,CL,0,0) <FIN DE PERIODE ET TESTS >;
HORLOG (SO,VOIE2) <ARRET DU CADENCEUR >;
CA COMPT;
PRV APRES <TRACES DE COURBES >;
CA TABLEC+2 <T/1,4 >;
DNE +4;
MPV +875;
ST BONINF;
NRV CPABON <T INF A 1 - LAMEDA TROP PETIT >;
ST BONSUP;
PRV CPUBON <T SUP A 1 - LAMEDA TROP GRAND >;
CERON: CA TABLEC+5 <CAL CUL DE U(C) >;
MPV +10;
GNE +7;
AD HUN;
RA HUZERO;
POT (1,POTNUM+8,HUZERO) <AFF DE PS >;
HYBD (1,CI,RU,0,0) ;
AVANT: IC COMPT <PREP POUR LES COURBES >;
CXV -10;
CAV 2070;
SORT (9) <SORTIE DE SONNERIES SUR TTY >;
AXV +1;
IRV AVANT+2;
WAIT (ATTENTE DE LANCEMENT) ;
WAIT ( ) ;
IRV DEPART;
APRES: CA COMPT;
ADV -2;
PRV AVANT <CE N'EST PAS FINI >;
CA GAMMA;
AD ECART;
RA GAMMA;
ADV -25;
NRV AGAMMA <GAMMA NON TERMINE >;
FINTRA ( ) <FIN DE TRAVAIL >;
CPUBON:CA NOVLAM;
RA SUPLAM;
CA INFLAM;
AD SUPLAM;
DNE +1;
RA NOVLAM;
IRV ALAMDA;
CPABON:CA NOVLAM;
RA INFLAM;
IRV CPUBON+2;
PEUPAS:CAV "?";
SORT (9) <NOMBRE INF A 0 - RACINE IMPOSSIBLE>;
FINTRA ( ) ;
VAZY: GENERE BI* BO;
TRAV

```

VAZY: GENEHE BI* BO;
TRAVAU BI* FO*;
GAMMA: ;
AIGUI: ;
VALEUR;
INFLAN:;
SUPLAK;
NOVLAM:;
LAMKRE:;
TOTO: ;
HUZERO:;
COMPT: ;
BONINF:+9900#;
BONSUP:+200#;
HUN: +10000#;
TREIZ8:+1388#;
QUAT9: +4900#;
ECART: +4;
MILLE: +1000#;
POTNUM:+10;
+16;
+17;
+18;
+11;
+12;
+13;
+15;
+14;
POTAR: +1000#;
+71#;
+714#;
TABLOC:TABLEC BO*;
TABLEC:TAB +8;
FIN;



```

COM          <SP DE RACINE CARREE >;
SEC  SHACVF;
HACVF: RT  PTRT;
NRV  TILT;
NE   NBNE;
ZRV  TILT-1;
RA   NN;
CA   NBNE;
ITV  +1;
ZRV  **+7;
CA   NN;
DLS  +1;
RA   NN;
MP   NP1;
AD   NP2;
IRV**+4;
CA   NN;
MP   NI1;
AD   NI2;
CXV  -3;
IRV  ITRAC+7;
ITRAC: CA  NN;
DV   RACIN;
DLS  +1;
RA   MANVR;
CA   RACIN;
DLS  +1;
AD   MANVR;
RA   RACIN;
MPV;
AXV  +1;
IRV  ITRAC;
CAV  +18;
AD   NBNE;
DLS  +1;
UNV  0400;
RA   DECDNE;
CA   RACIN;
DECDNE:;
IC   PTRT;
TILT: IR   PTRT;
PTRT: ;
NBNE: ;
NN:   ;
RACIN: ;
MANVR: ;
NP1:  3240 474;
NP2:  1140;
NI1:  2220;
NI2:  1560;
FIN;

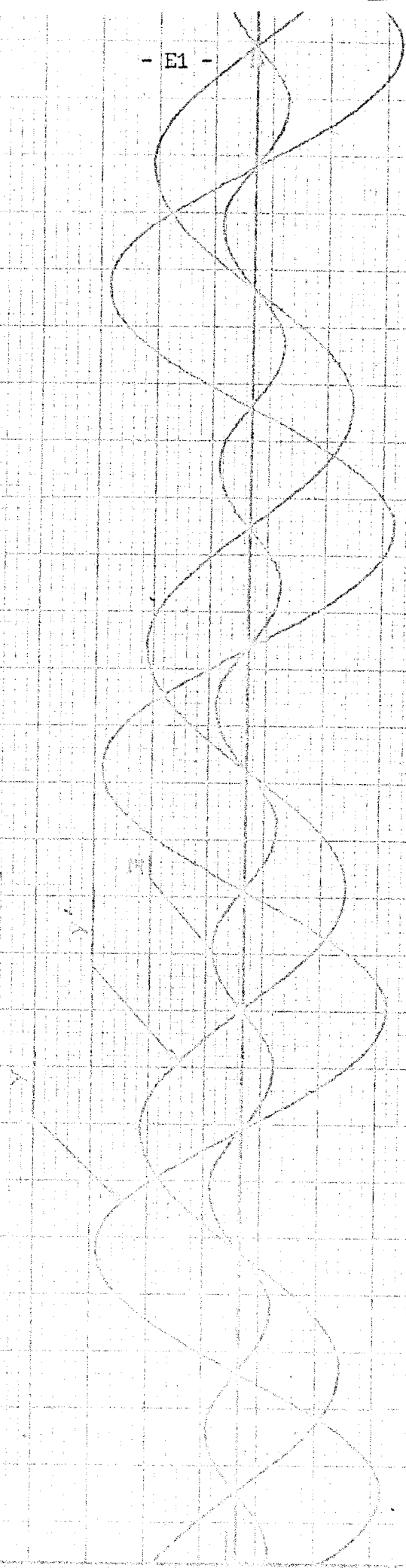
```



ANNEXE E

VERIFICATION GRAPHIQUE ET EVALUATION DE LA PRECISION

EXEMPLE 3



Echelle : 0,5 mV / Cm

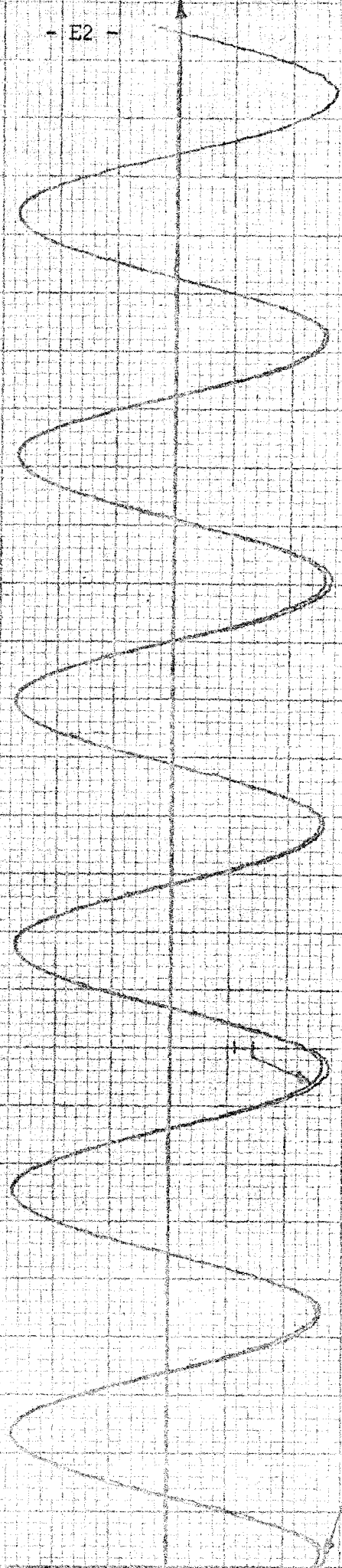
$$E = Y \cdot Y'$$

Figure F-1 : Elaboration des sinusoides par une simulation analogique



Scale : 0.5 mV/cm

- E2 -



Echelle: 0.5 mV/cm

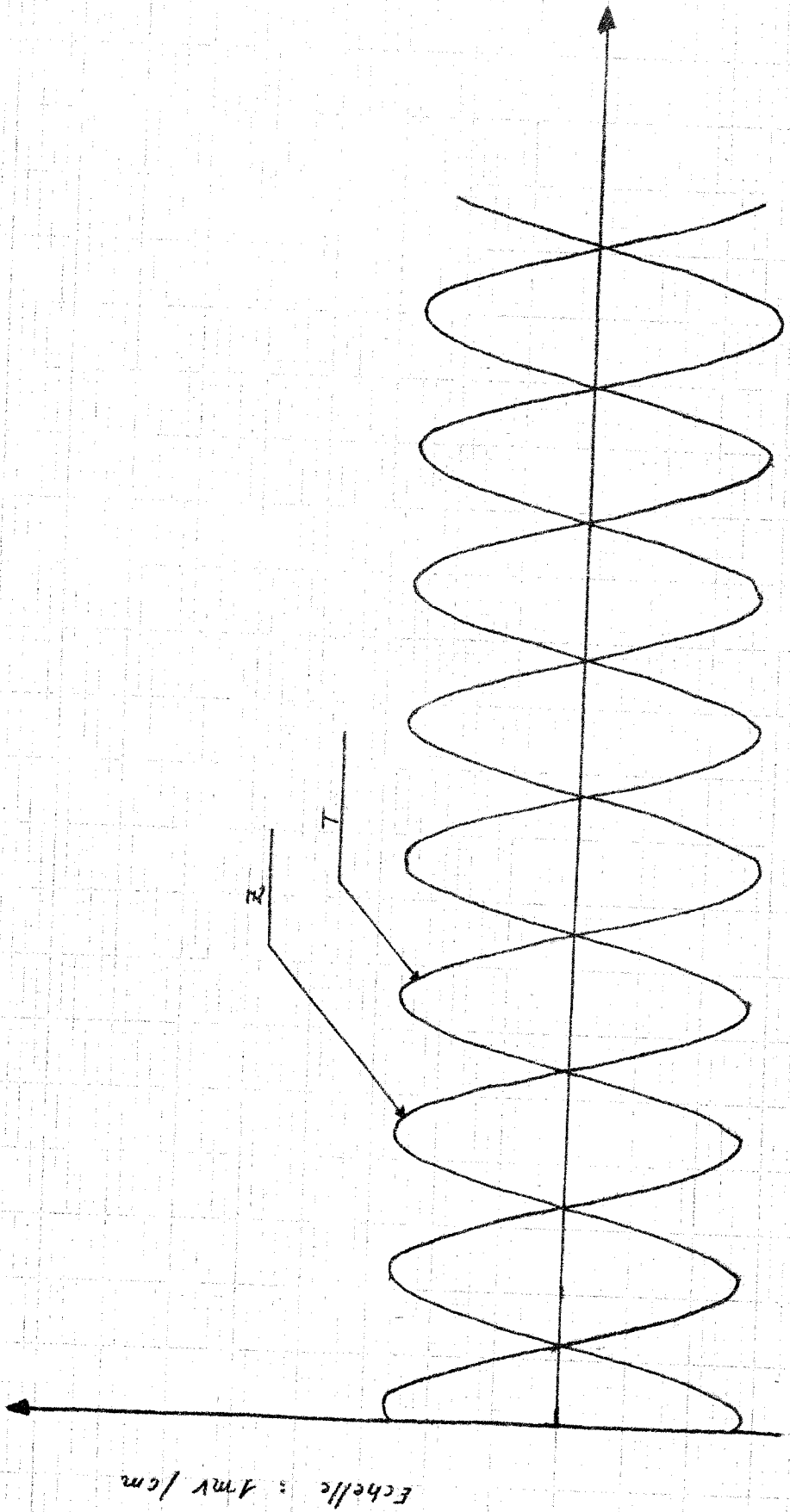
2 Signal venant du multiplicateur analogique

1 Signal venant du calculateur numérique

Figure 6.2 : Signaux de ser



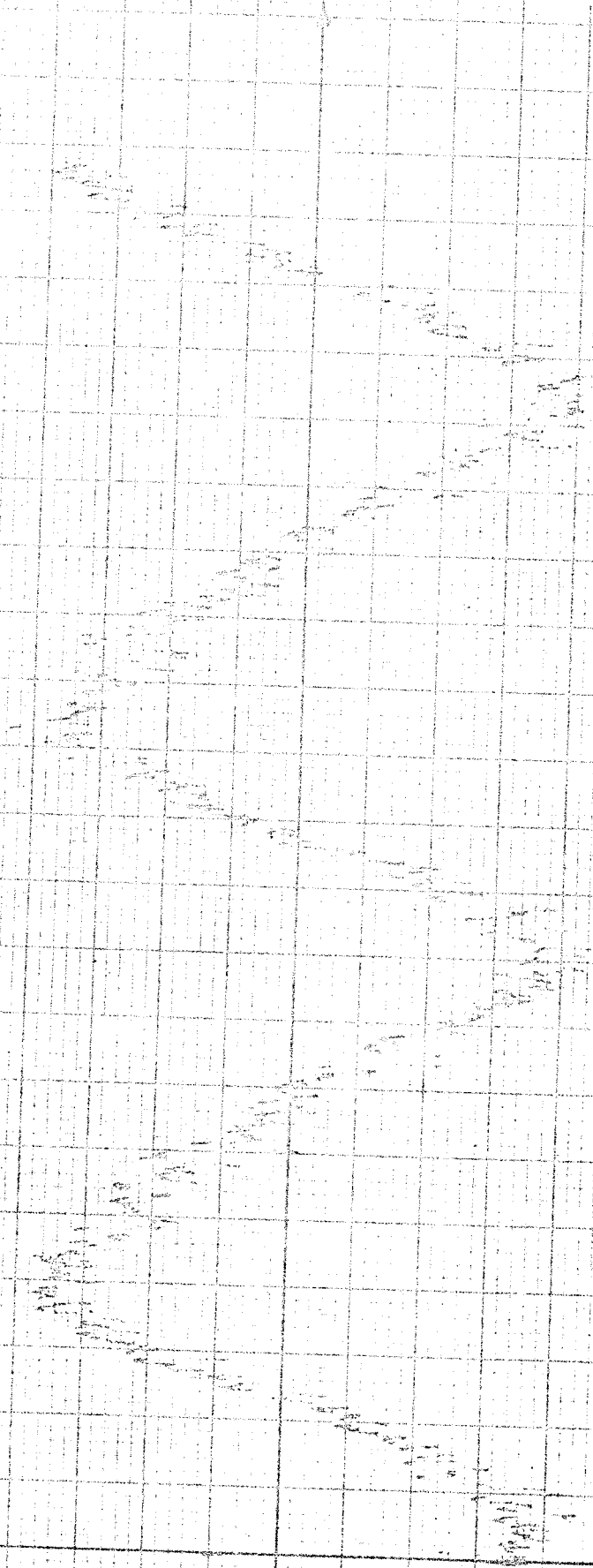
Echelle: 0.5 mV/cm



Echelle : 0.5mV/cm

Figure E.3 : Sortie des signaux en opposition de phase.

Echelle: 0.25 V/cm



Echelle: 0.5mV/cm

Figure 2a: Signal I-2 avec gain de 10 en entrée



BIBLIOGRAPHIE

- 1 GRANINO A.KORN, THERESA M.KORN
"Electronic - Analog and Hybrid Computers".
- 2 J.P. NANTET
"Ordinateurs en temps réel - Applications industrielles".
- 3 SAUL STIMLER
"Real time data - processing systems."
- 4 S. FIFER
"Analog Computation ". .
- 5 "EAI - 590 - Hybrid Computing Systems"
- 6 "EAI 640 - Digital Computing Systems".
- 7 "Totality new hybrid computing Systems".
Applied Dynamics.
- 8 J. GAUDFERNAU
"Introduction au traitement de l'information en temps réel".
- 9 J. ARSAC
"Les systèmes de conduite des ordinateurs".

- 10 H.Y. CHANG, E.G. MANNING, G.METZE
"Fault diagnosis of digital systems".
- 11 W.J.KARPLUS, W.W. SOROKA
"Analog Methods".
- 12 W.S. KARPLUS
"Analog Simulation".
- 13 C. MELIN - J.M. TOULOTTE
"Les machines hybrides".
- 14 C.V. FEUVRIER
"La simulation des systèmes".
- 15 J.C. ANGUE
"Contribution à l'étude des systèmes discrets non linéaires
décrits par un modèle recurrent".
Thèse de 3ème cycle - Juillet 1973
- 16 P. VIDAL
"Systèmes échantillonnés non linéaires".
Gorden and Breach - 1968
- 17 S.C. CHANG, L.G. BIRTON, A. KUMER
"Porseuille flow with variable physical properties - a simulation
approach".
Septembre 1972.

PLAN du MEMOIRE

<i>Généralités</i>	3
<u>Ière PARTIE</u> - DEFINITION DU CALCUL HYBRIDE	5
I.1 - Structure d'un calculateur hybride	7
I.2 - Fonction des organes constitutifs d'un ensemble hybride : partage des tâches	15
I.3 - Conclusion	
<u>IIème PARTIE</u> - LE COUPLAGE	25
II - Les calculateurs	26
II.1 - L'ordinateur de processus	26
II.2 - Le calculateur hybride	32
II.3 - Conclusion	34
III - Interface	35
III.1 - Commande du pupitre	36
III.2 - Commande individuelle des intégrateurs	54
III.3 - Affichage des potentiomètres asservis	57
III.4 - Les potentiomètres numériques	67
III.5 - Acquisition de données	75

III.6 - Etude de la saturation des amplificateurs	77
III.7 - Conclusions	83
IV - Les Tests du système	84
IV.1 - Test du calculateur et des périphériques	85
IV.2 - Test de l'interface	85
IV.3 - Test des calculatrices hybrides	89
IV.4 - Conclusion	94

IIIème PARTIE - LANGAGE ET PROGRAMMATION

V.1 - Introduction	1
V.2 - Le Software	3
V.3 - Méthode de Programmation	6
V.4 - Le Moniteur	9
V.5 - Conclusion	19

IVème PARTIE - EXEMPLES

1 - Introduction	1
2 - Traitement d'un exemple hybride	6
3 - Exemple 2 - Equations de Poiseuille	15
4 - Exemple 3 - Génération de fonctions	25

Conclusion Générale

Annexe A - Recherche des séquences de tests

Annexe B - Organigrammes de détections des modules défectueux

Annexe C - Organigrammes et courbes de l'exemple 1

Annexe D - Organigramme, programme et courbe de l'exemple 2

Annexe E - Vérification graphique et évaluation de la précision-Exemple 3

Bibliographie

