

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

201

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE  
« HOUARI BOUMEDIENNE »

ECOLE NATIONALE POLYTECHNIQUE

Département d'Electronique



**SUJET**

**SYSTEME DE REGULATION  
DE CARREFOURS**

Sujet proposé par :

Mr KACPRZYNSKI

Etudié par :

Melle TEBBIKH Raja

Mr MEDJELDI Tayeb

الجمهورية الجزائرية الديمقراطية الشعبية  
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE  
« HOUARI BOUMEDIENNE »

**ECOLE NATIONALE POLYTECHNIQUE**

**Département d'Electronique**

—————»O«—————

PROJET DE FIN D'ETUDES

**SUJET**

**SYSTEME DE REGULATION  
DE CARREFOURS**

Sujet proposé par :

**Mr KACPRZYNSKI**

Etudié par :

**Melle TEBBIKH Raja**

**Mr MEDJELDI Tayeb**

Promotion Juin 1983

-Nous saisissons cette occasion pour exprimer toute notre gratitude à tous les professeurs qui ont contribué à notre formation qu'ils soient à l'école polytechnique , à Annaba ou à Guelma .

-Nous remercions également monsieur KACPRZYNSKI pour toute l'aide qu'il nous a procurée durant la préparation de ce projet .

-Nous remercions aussi monsieur BOUDALI pour L'aide précieuse qu'il nous a fournie en nous procurant une documentation qui nous a été tres utile.

- Nous remercions aussi monsieur Abdelkader pour avoir voulu nous aider à faire la frappe

- Nous n'ometons pas de remercier biensur tous nos amis(es) qui nous ont encouragé durant tout ce semestre;



Je dédie ce travail à :

- Mon Père et Ma Mère qui ont donné le meilleur d'eux même pour que j'arrive à ce stade
- Mes Frères et Soeurs (surtout à Salim)
- Mes Grands Parents
- Mes Oncles
- La mémoire de mon Frère Larbi et à celle de la grandnère
- Toute la Famille
- Tous les amis(es)

Abdelghani Tayeb



Je dédie ce modeste travail à:

- Mon Père
- Ma Mère
- Mes Grands Parents
- Mes Frères: Mohaned, Youcef, Hichen  
et Khaled
- Toute la Famille
- A toutes mes amies (is)

Raja

# SOMMAIRE

## Introduction

### Chap 1: Etude générale des problèmes de régulation

- I.1 - Généralités
- I.2 - Formulation des problèmes de trafic
- I.3 - Conception générale d'un carrefour isolé
- I.4 - Exemples de carrefours et stratégies correspondantes
  - Description d'un croisement de deux routes
- I.5 - Conception mathématique
  - I.5.1 - Cas d'un carrefour isolé
  - I.5.2 - Cas de deux carrefours adjacents
  - I.5.3 - Cas d'un réseau de carrefours
- I.6 - Simulation de carrefour
  - I.6.1 - Microrégulation
  - I.6.2 - Macrorégulation

### Chap 2: Partie statique

- 2.1 - Introduction
- 2.2 - Système de régulation
  - 2.2.1 - Propriétés
  - 2.2.2 - Modes de fonctionnement
- 2.3 - Detecteurs
  - 2.3.1 - Propriétés
  - 2.3.2 - Principe de fonctionnement
- 2.4 - Organisation fonctionnelle du système
  - 2.4.1 - Carte microprocesseur (MPU)
  - 2.4.2 - Système d'alimentation et bloc de secours
  - 2.4.3 - Carte de commande en puissance des feux
  - 2.4.4 - Détection des pannes des lampes
  - 2.4.5 - Horloge temps réel

### Chap 3: Partie dynamique

- 3.1 - Partie logicielle
- 3.2 - Programmation
  - 3.2.1 - Procédés de programmation
  - 3.2.2 - Notions sur le module d'interface et son programme de gestion

### 3.2.4 - Exemples de programmation de gestion d'un carrefour

#### a)- Exemple 1 de stratégie

- Algorithme

#### b)- Exemple 2 de stratégie appliquée à un carrefour de deux routes

- Commande à fournir aux feux

- Initialisation des P.I.A

- Table de transition des états

- Organigramme

- Programme

- Temporisation

- conclusion

### Chap 4: Coordination des carrefours (modélisation par macro-régulation)

4.1 - Modélisation d'un carrefour par macrorégulation

4.2 - Organisation générale de la simulation d'un réseau  
, de carrefours

### Conclusion finale

Bibliographie



INTRODUCTION

Il semble que de tout temps l'offre, au moins en infrastructure ait du mal à s'adapter à la demande de transport puisqu'il paraît qu'on connaissait les embarras des grandes villes dès le 18<sup>ème</sup> siècle.

Cependant le développement explosif de l'utilisation de l'automobile individuelle pose depuis déjà des années à Alger à l'égard des autres grandes villes, le problème avec une acuité telle qu'on a été amené à chercher les moyens de tirer le meilleur parti possible de l'infrastructure existante. C'est ainsi que des premières études de régulation de trafic ont été entreprises sur le réseau urbain d'abord, sur les réseaux suburbains et interurbains ensuite. Sur le plan théorique, la détermination des stratégies de régulation de trafic sur un réseau maillé urbain est extrêmement complexe et il n'est pas surprenant que plusieurs modèles de simulation aient été établis dans différents pays afin d'expérimenter des politiques de régulation. Et il est incontestable que des progrès importants ont été faits dans la détermination des méthodes de calcul des plans de coordination. Ces progrès ont été les plus importants dans les pays où on s'est rendu compte qu'il convenait d'effectuer conjointement les études théoriques et les réalisations expérimentales. On peut regretter qu'en Algérie, aucune expérience en vraie grandeur de régulation centralisée d'un réseau urbain n'ait été encore réalisée; Ceci s'explique par le manque de moyens dont dispose notre si jeune pays en industrie électronique.

L'Algérie a pris un retard certain en matière de régulation du trafic urbain aussi bien au niveau du savoir faire qu'au niveau de la conception des matériels. Il semble clair que ce retard ne pourra être comblé que si des moyens suffisants sont rassemblés pour mener une action de recherches coordonnées s'appuyant sur une action de réalisation sur le terrain ayant une certaine envergure et procurant un changement de structures adéquat dans les villes qui serait apte à satisfaire les besoins allant avec la rapide expansion de celles-ci.

Cependant il ne faut pas oublier de souligner l'aspect pédagogique de la régulation des feux qui fait que l'utilisation de ces derniers sans l'intervention d'un agent nécessiterait un respect strict de la part du conducteur et du piéton. Nous savons que pour arriver à cette fin, le processus est assez long comme le montrent les expériences des pays occidentaux. Ça risque de durer toute la durée d'une génération ; mais, après, heureusement d'ailleurs, le résultat est des plus satisfaisant ce qui entraînerait une nette baisse du taux des accidents, d'où la nécessité de ces feux remplaçant définitivement l'agent qui, lui travaillait dans des conditions qui ne peuvent qu'être nuisibles pour sa santé et son bien être.

Pour ce qui est de notre projet, nous avons essayé de concevoir un système moderne, que nous espérons adaptable aux problèmes actuels de la circulation qui se posent aux niveaux des différents carrefours, sans trop tenir compte des détails techniques.



ETUDE GENERALE DES PROBLEMES DE REGULATION

I - 1 - GENERALITES :

Le système utilisé jusqu'à présent pour la commande des feux de circulation en Algérie est à cadence fixe, il consiste à provoquer le basculement des feux d'une route à l'autre suivant un cycle de durée constante, la division de ce cycle en phase rouge, orange et verte sur chaque route étant également constante. Si ce type de régulation assure convenablement la sécurité aux points de conflits, il devient insuffisant lorsque le volume de trafic s'accroît vu qu'il faut satisfaire 2 objectifs de régulation qui sont de types contradictoires :

- L'une consiste à donner du feu vert sur un ou plusieurs itinéraires.
- L'autre à minimiser le temps d'attente aux feux rouges.

Le système de contrôle qu'on se propose d'étudier doit s'adapter à l'état du carrefour, c'est ce qu'on appelle système adaptatif. Indépendamment du rôle qu'il joue dans le domaine de sécurité des piétons et des véhicules, ce contrôleur de circulation doit remplir les objectifs suivants :

- Obtenir le maximum de capacité de débit
- Diminuer la somme des temps de parcours
- Augmenter le confort de l'utilisateur, en particulier en diminuant le nombre d'arrêts.

Les moyens acquis par la théorie actuelle de contrôle optimal du trafic urbain sont :

- Changement du cycle suivant l'état du trafic produisant une augmentation de la capacité de débit disponible.
- Réglage du décalage des carrefours les uns par rapport aux autres.
- Création des tronçons privilégiant du trafic rapide .
- Réglage de la répartition du temps de passage disponible sur chaque carrefour entre les diverses voies.

On remarque que suivant le but envisagé, l'action à entreprendre est différente : l'augmentation de la capacité de débit que produit l'allongement du cycle est quelquefois favorable au confort de conduite mais augmente le temps de parcours, maintenir la vitesse constante sur un axe est favorable au confort mais coûte en général du débit et de l'attente sur les transversales. Delà découle la nécessité d'une régulation adaptative de manière à adopter l'offre à la demande si c'est possible.



En effet ce type de régulation comprend plusieurs points en interaction si on ne veut pas se limiter à des solutions partielles. Pour ce genre de problème qui apparaît très compliqué, on envisage une solution qui consiste à construire un modèle traitant le système réel comme une certaine fonction de transfert donnant pour chaque condition de marche certaines sorties auxquelles on s'intéresse. Chaque solution de ce contrôle doit prendre en considération plusieurs phénomènes qui limitent le nombre de degrés de libertés C.a.d qui acquièrent une certaine fiabilité.

## I ← 2 - FORMULATION DES PROBLEMES DU TRAFIC :

Les différentes stratégies pour changement de feux appliquées dans des carrefours adjacents ou plus généralement dans un réseau de carrefours et le comportement dynamique des véhicules dans chaque tronçon reliant deux carrefours sont les paramètres qui affectent le flux de circulation.

La méthode imaginée pour maximaliser le flux consiste à faire une corrélation entre les stratégies s'adaptant aux états des circulations : circulation lente ou rapide de façon à s'approcher de l'optimum de régulation. Ainsi une coordination optimale nécessite un système de ramassage de plusieurs données ce qui conduit non seulement à une augmentation du coût mais aussi à une complexité de la résolution du problème.

Afin d'obtenir une estimation aussi exacte que possible s'adaptant aux perturbations qu'on pourrait rencontrer et qui pourraient déformer les caractéristiques du processus et par la suite la caractéristique optimale ainsi déterminée on se propose de construire un modèle simulable au système réel. Si ce dernier peut s'interpréter comme une fonction de transfert donnant pour chaque condition de marche certaines sorties auxquelles on s'intéresse on pourra dire qu'on a un modèle chaque fois qu'on a pu construire un système ayant une fonction de transfert voisine de celle du système à étudier. Le modèle permettra l'étude du système et la détermination des solutions remplaçant celles qu'on peut en déduire par observation sur l'endroit même.

Rappelons que le but poursuivi est de produire un outil capable de reproduire les phénomènes réels de circulation avec une fidélité suffisante pour que les diverses politiques de réglage de feux de la circulation puissent être comparées sur le modèle, pour mettre en évidence comment il est possible d'appliquer les méthodes d'optimisation au système de carrefours comportant des inerties importantes.

Le modèle que nous proposons consiste à définir les carrefours comme des noeuds internes du réseau et les tronçons comme des mailles. Il y ajoute les noeuds périphériques qui n'ont pas de réalité physique mais qui permettent d'identifier les entrées et les sorties du réseau, de cette manière le modèle général de ce dernier sera défini d'une façon globale en rassemblant tous les modèles de simulation des carrefours composant le réseau en tenant compte des retards subis par les véhicules lorsqu'ils parcourent les tronçons conduisant à une interaction entre les modèles élémentaires. Les grandeurs agissant sur cette interaction sont donc :

- Le temps de cycle optimal du parcours.
- La vitesse de circulation.
- Les décalages de synchronisation correspondant à la solution optimale.

Avant d'aborder la méthode de régulation optimale dans une agglomération de carrefours on va étudier tout d'abord le problème de contrôle dans un carrefour sans tenir compte des autres qui l'entourent afin de permettre la mise au point des différentes stratégies de régulation de trafic que nous pouvons rencontrer pour faciliter l'étude globale du réseau.

### I - 3 - CONCEPTION GENERALE D'UN CARREFOUR ISOLE :

Le modèle de simulation d'un carrefour est considéré comme un instrument de mesure qui permet d'évaluer l'effet des différentes politiques de régulation. Il faudra donc qu'il soit un bon instrument de mesure : juste, sensible et fidèle. En ce qui concerne la justesse, pratiquement il est impossible que le modèle reproduise la réalité avec exactitude, mais ceci n'est pas très grave ce qui compte c'est que le modèle soit sensible - qu'il réponde bien aux variations, des conditions extérieures, et surtout fidèle - les résultats qui nous intéressent varient de la même manière dans le modèle que dans la réalité.

De ce fait la description qui décrit le comportement des véhicules pas à pas à travers tout le réseau est très compliquée, pour construire un modèle on définit un certain nombre de points critiques et on cherche à reproduire les temps de passage en ces divers points, en essayant de retrouver ces temps de manière la plus efficace, autrement dit pour cette simulation dite macroscopique on doit définir les états discrets du système pour savoir en déduire le modèle général.



PARAMETRES UTILES POUR LA SIMULATION D'UN CARREFOUR

La connaissance d'un certain nombre, assez limité, de paramètres est nécessaire pour l'utilisation d'un modèle de simulation d'un carrefour, les paramètres sont :

a) Paramètres géométriques (description des carrefours, nombre de files à chaque entrée, capacité de files, les temps de traversée des carrefours etc ...)

Ces paramètres sont déduits par observations directes sur terrain et seront par la suite intégrés au programme de simulation du carrefour.

La distribution des temps de parcours tronçon pour les poids lourds et les véhicules légers.

b) Matrice origine - destination :

Les données de comptage sur un carrefour sont insuffisantes pour reconstituer une matrice origine - destination pour une route d'une seule voie emmenant à trois directions : les allant tout droit, les T.A. et les T.A.D. C'est pourquoi il est préférable d'utiliser des routes de 3 voies pour faciliter la détection de la direction de chaque véhicule permettant l'application de la stratégie convenable.

c) Gêne apportée par les piétons :

On suppose que les piétons commencent à bloquer une voie (pour les T.A.D) lorsque le feu passe au vert ; la durée de ce blocage est calculée suivant la longueur de la route

d) Caractéristique de démarrage des véhicules :

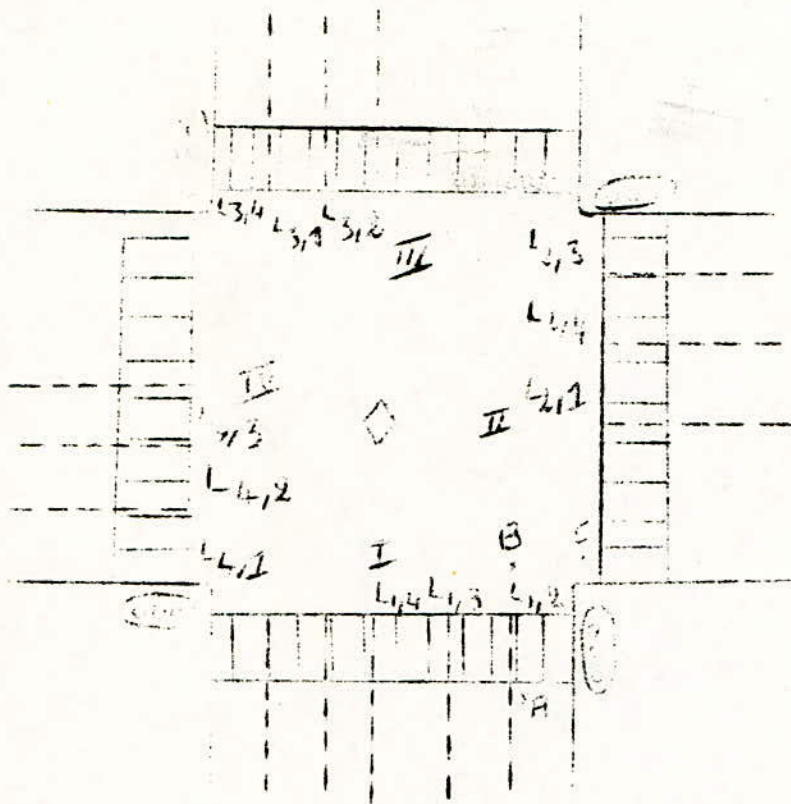
L'intervalle de temps séparant le démarrage de 2 véhicules successifs est très important parce qu'il conditionne le débit des carrefours.

Les mesures faites sur places montrent que pour une file, le premier véhicule léger part en moyenne de 3 à 4s après le passage au vert du feu, tandis-que les véhicules successifs se suivent à 2s les uns des autres.

Pour les poids lourds ces temps sont multipliés par deux, on tient compte du nombre de files à l'entrée de chaque carrefour en divisant les écarts entre véhicules par le nombre de files.



Envisageons le cas du carrefour suivant sur lequel notre étude va se baser dans la partie programmation qui suit :



DESCRIPTION DU CARREFOUR :

Le carrefour est situé en l'encontre de 4 routes, dans chaque route on distingue 3 courants de véhicules :

- Le courant direct
- Le tourne à gauche
- Et le tourne à droite

Pour les mouvements tournant un certain nombre de facteurs influent leurs écoulements. Pour les T.A.D on a :

- Le rayon de giration (temps de freinage)
- Le débit des piétons entrant en conflit avec le mouvement tournant (temps d'attente).
- Stockage
- Créneau dans le trafic opposé

Ce qui conduit à un temps d'attente considérable. Ainsi un véhicule pourra être servi ou non selon l'état du carrefour :

- Couleur du feu à franchir
- Existence d'un créneau sur la voie antagoniste (Pour les T.A.G)
- Existence d'un créneau dans le flot des piétons (Pour les T.A.D)

Décrire l'état du carrefour reviendra donc à décrire :

- L'état des différentes files d'attentes
- L'état des feux
- L'état des paramètres non contrôlables (densités des flots de piétons, accidents etc ...)

Stratégies appliquées :

Dans le cas où on a des files infinies dans 4 routes, on fixera la stratégie de changement de feux de telle sorte que les véhicules n'attendent pas 2 fois le vert. Et ceci en utilisant toujours des capteurs pouvant déterminer le nombre de véhicules attendant le feu vert dès que le rouge est allumé.

Si les 3 voies d'une même route ne sont pas de même densité, la durée du vert sera déterminée de telle sorte à évacuer tous les véhicules de la voie à circulation la plus intense.

La stratégie qu'on peut appliquer est  $V_I, V_{II}, R_{II}, R_{IV}$  et si l'une des routes I ou II sera évacuée avant l'autre on tombera sur un autre type de stratégie V,R,R,R. A la fin de cette phase le même phénomène se répétera pour les 2 routes I et II et ainsi de suite.

La stratégie V,R,R,R est la plus simple et permet une évacuation plus rapide, il reste donc à soulever les problèmes d'application de la stratégie V,R,V,R.

Voie de T.A.D :

Pour les T.A.D le seul gênant est la passage des piétons, après l'atteinte de la vitesse maximale (autorisée) de franchissement du carrefour les véhicules freinent pour laisser passer les piétons pendant un temps dépendant de la largeur de la route, puis ils redémarrent.

Voie directe :

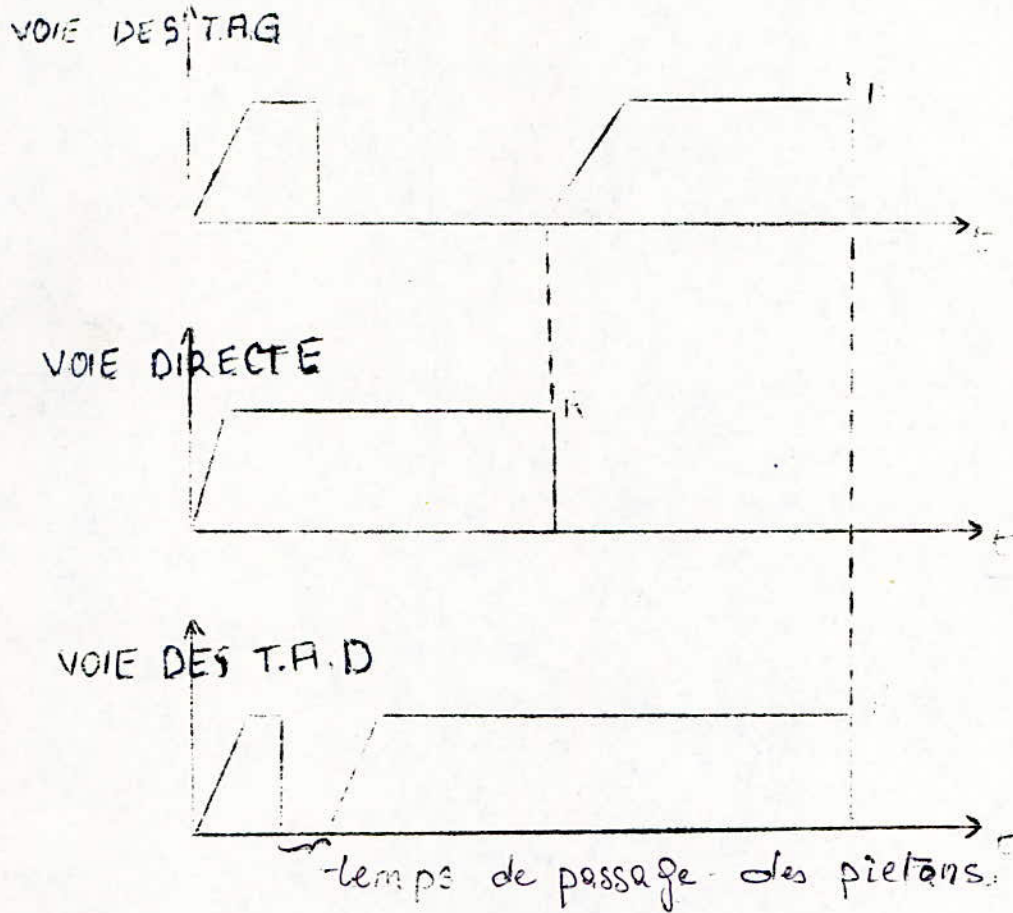
Les itinéraires des voies directes sont toujours libres, le démarrage des véhicules est plus rapide, par conséquent l'évacuation de la voie l'est aussi.

Voie des T.A.G :

Pour les T.A.G le mode de circulation n'est pas le même que pour les T.A.D. Le véhicule atteignant la vitesse maximale de passage par le carrefour peut franchir ce dernier comme il peut être arrêté par les véhicules de la voie directe de la route antagoniste. De ce fait le démarrage des T.A.G est plus long et l'évacuation de la voie correspondante dépend du trafic de la voie directe antagoniste et par conséquent elle est trop lente. C'est pour cela que nous avons conçu la solution qui consiste à introduire un créneau en plus permettant l'arrêt des véhicules de la voie directe avant les deux autres (T.A.D ; T.A.G) et par conséquent l'évacuation de l'itinéraire au T.A.G. La durée maximale donc de la phase verte dépend de l'état du trafic dans les 2 voies T.A.D et T.A.G.



Donc pour une circulation dense où les files sont infinies on aboutit au diagramme suivant d'évacuation des véhicules des trois voies,

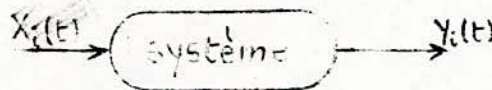


I - 5 - CONCEPTION MATHÉMATIQUE :

I - 5 - 1. Cas d'un carrefour isolé :

Comme cité précédemment nous avons dit que nous pouvons réguler un carrefour sans se préoccuper des carrefours voisins.

Soit alors un carrefour à plusieurs voies (entrant et sortant), nous pouvons le simuler à un système à une entrée  $x_i(t)$  et une sortie  $y_i(t)$



La contrainte majeure réside dans le fait qu'on doit avoir autant de véhicules entrant au carrefour que de véhicules sortant de ce dernier

donc

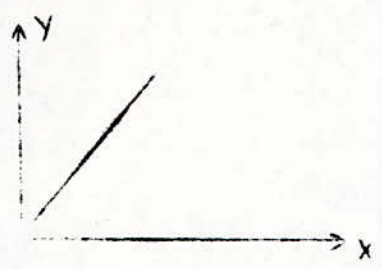
$$\int_{t-\Delta t}^{t+\Delta t} \sum_i x_i(t) dt = \int_{t-\Delta t}^{t+\Delta t} \sum_i y_i(t) dt$$



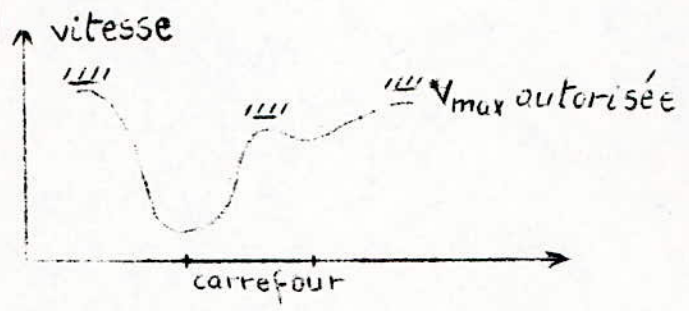
2  $\Delta t$  étant le temps nécessaire pour un véhicule pour franchir le carrefour  
Donc nous pouvons dire que la description d'un tel carrefour est présentée  
à l'aide d'un opérateur  $tq$  :

$$y_i(t) = F[x_i(t)]$$

$f$  ; Peut être une fonction statique et linéaire de pente dépendante de la stratégie de changement de feux choisie (fig a), autrement dit l'opérateur  $f$  peut être réglé suivant l'influence des voies convergeant vers le carrefour et le comportement dynamique des véhicules sachant que ceux-ci doivent à des instants freiner et à d'autres accélérer suivant l'état du carrefour (fig b).



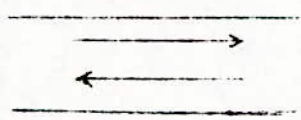
(fig a)



(fig b)

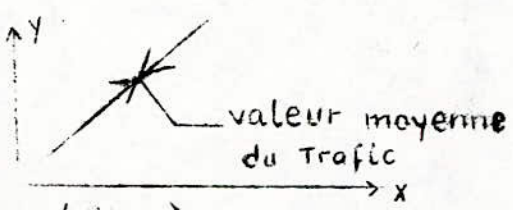
En passant par le carrefour la vitesse max de passage autorisée par le code doit être respectée.

Alors posons nous la question : Quelle est l'influence du phénomène dynamique du carrefour sur cette fonction ? Imaginons le cas suivant :

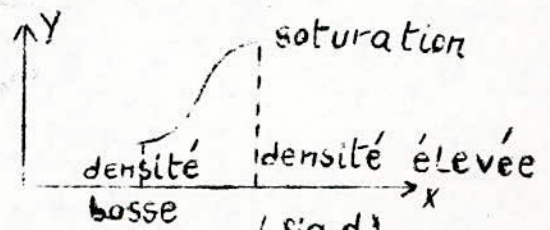


L'opérateur  $f$  est une fonction linéaire autour de la valeur moyenne du trafic (fig c).

Mais vu que les lois régissant le trafic ne sont pas stables, cette fonction  $f$  peut être localement convexe ou localement concave respectivement en cas des véhicules lourds dont la vitesse est basse ou pour une évacuation dense en cas de véhicules rapides (fig d)



(fig c)



(fig d)

Si l'augmentation de véhicules croit on tombera sur le cas limite de saturation où on aura embouteillage et où la fonction  $f$  changera de direction en sens de décroissance.

Problème de contrôle optimal :

Si on fixe le nombre de véhicules se présentant à l'entrée du carrefour à

$$X = \int_0^T x(t) dt$$

où  $T$  : Période du signal du changement de feux.

Le nombre de véhicules évacués par le carrefour sera :

$$Y = \int_0^T y(t) dt = X$$

Sachant que  $y_i(t) = f[x_i(t)]$

Le problème à poser consiste à maximaliser le trafic à la sortie et ceci en répondant aux contraintes suivantes :

Sachant que  $x(t), y(t) \geq 0$  ;  $\Delta t \in [0, T]$

1.  $\int_0^T x(t) dt = X \quad \forall t \in [0, T]$

2.  $\int_0^T x(t) dt = \int_0^T y(t) dt = \bar{y} \cdot \Delta t$

Le terme  $\bar{y} \cdot \Delta t$  est négligeable et a pour sens physique le nombre de voitures au sein de chaque carrefour.

Avec  $x(t) \leq \bar{x} + \epsilon$

$y(t) \leq \bar{y} + \epsilon'$

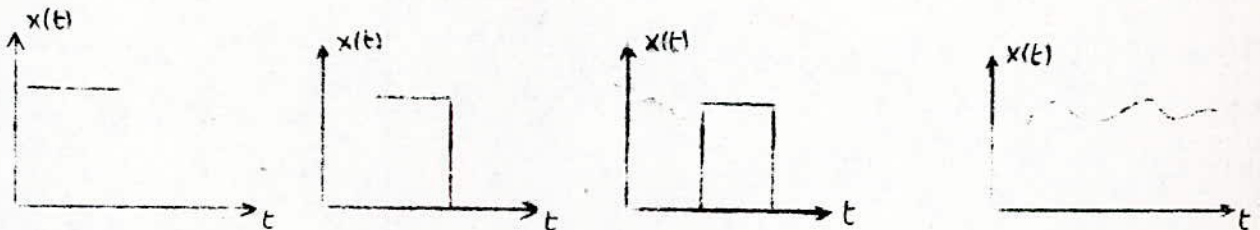
où  $\epsilon$  et  $\epsilon'$  dépendent du trafic instantané.

et  $\bar{x}$ ,  $\bar{y}$  sont fonction des vitesses max autorisées par le code.

Donc on cherche comme solution au problème posé une fonction :

$t \rightarrow x(t) \quad t_0: c \leq x(t) \leq \bar{x}$

Cette fonction peut prendre n'importe quelle forme vérifiant les deux conditions précédentes. (fig - e -)



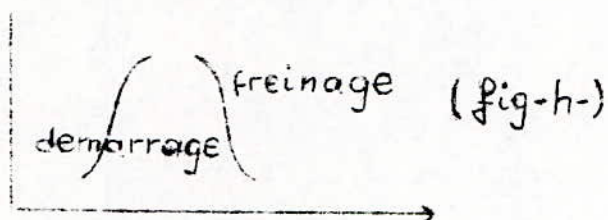
( fig - e - )



La solution de telles formes définit la stratégie des feux du carrefour considéré. Mais vu que à la limite de saturation la formation tend à décroître, cette inertie de véhicules rend la solution (stratégie) plus compliquée.

La loi de variation de vitesse du premier véhicule suit un processus de poisson (fig. f.)

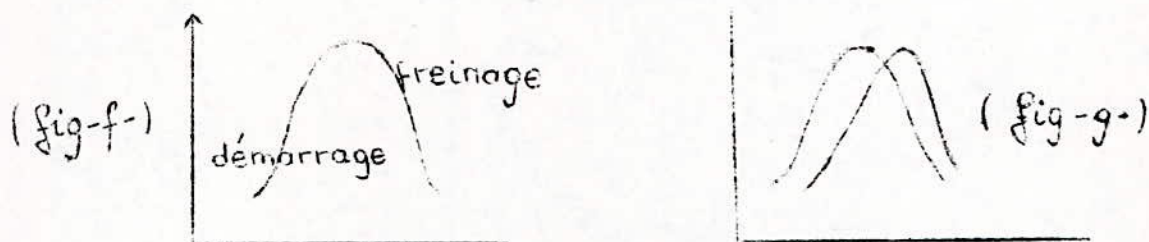
Les véhicules qui suivent font d'autant mais avec un certain retard, on aura donc presque les mêmes variations dépendantes des distances libres entre véhicules (fig. g)



En plus, l'homme lui même intervient dans le système avec ses réactions psychologiques propres, voyant un feu rouge devant lui - ou le véhicule précédent freiner - l'aut mobiliste peut choisir de garder sa vitesse constante le plus longtemps possible ou au contraire de ralentir assez à l'avance. Dans la simulation on suppose qu'il adopte la première attitude mais en réalité la plupart des automobilistes ralentissent très progressivement.

Ceci explique que lorsque le réseau est chargé les conséquences de petites variations dans la génération sont importants.

La valeur moyenne de génération de véhicules déterminée par observation sur chantier suit une loi Gaussienne. (fig. h.)



Au point de vue mathématique, on cherche la solution dans l'espace de fonctions continues relativement lisses.

La précision du problème de linéarité d'optimisation qu'on a proposé auparavant (comme solution théorique) n'est pas nécessaire à traiter, il suffit donc d'analyser seulement le caractère de la fonction solution.

1.  $f$  est localement **concave** (circulation intense) la solution est de la forme :

$$X(t) = X_0 + \varepsilon(t)$$

où  $\varepsilon(t)$  suit une loi de Gauss  $\varepsilon(t) \in N(0, \sigma)$

Par hypothèse :  $\int_0^T (X_0 + \varepsilon(t)) dt = X = X_0 T$

$F = \alpha X + r$  (terme indiquant que  $f$  est localement convexe)

d'où  $y(t) = (\alpha X + r)(X_0 + \varepsilon(t))$

$\int_0^T y(t) = \text{const} + F [\sigma, \text{phénomène (convexité ou concavité)}]$

L'étude approfondie (voir annexe I) de cette solution  $F$  a donné que  $F$  augmente lorsque  $\sigma$  diminue.

2.  $f$  est localement **convexe** (densité de circulation basse) par le même principe que précédemment on aboutit au résultat que  $F$  augmente avec  $\sigma$

Donc :

Pour une  $f$  **concave** :  $F \uparrow$  quand  $\sigma \downarrow$

" " **convexe** :  $F \uparrow$  "  $\sigma \uparrow$

De ce résultat on peut répondre à la question de maximalisation de trafic à la sortie du carrefour comme suit :

En présence d'un trafic de densité moyenne conduisant à une  $f$  localement concave il faut augmenter la variation de la fonction solution VAR  $x(t)$  tant que possible sans dépasser la valeur maximale admissible.

$$0 \leq \text{VAR } X(t) \leq \text{VAR } X(t)_{\max}$$

$$\text{où } \text{VAR } X(t) = \lim_{n \rightarrow \infty} \sum_{i=0}^{n-1} |X(t_{i+1}) - X(t_i)|$$

$$\max_i |t_{i+1} - t_i| \rightarrow 0$$

Donc la solution optimale peut être obtenue en utilisant un changement de feux très dense. Mais dans ce cas là on ne peut augmenter indéfiniment la fréquence du signal de ce changement, cette dernière est limitée à une certaine  $f_{\max}$  tenant compte du temps maximum de passage des piétons et de celui des poids lourds franchissant le carrefour avec un temps supérieur à celui des poids légers. On peut admettre que cette  $f_{\max}$  admissible détermine la solution optimale pour un trafic de densité faible.

De ce fait on peut deviner qu'en cas contraire où le trafic se présentant pour franchir le carrefour est de densité assez élevée (à la limite elle tend vers une constante) on adopte la solution avec la variation minimale possible. Mais ceci est impossible vu qu'il faut aussi laisser passer les piétons, en plus on risquerait de tomber sur un temps d'attente du vert assez considérable.



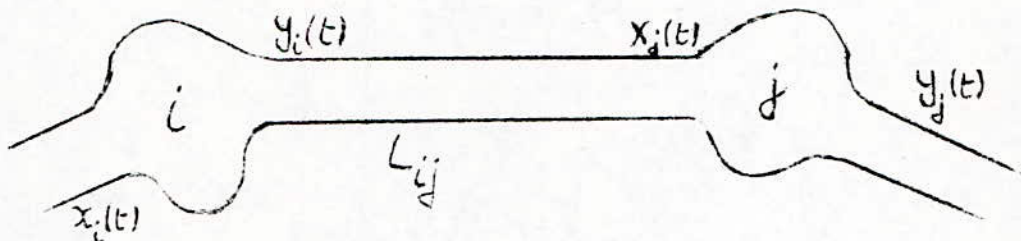
C'est pourquoi dans ce cas là on est obligé d'appliquer une solution  $f$  constante par morceaux (voir l'exemple de description de carrefour qu'on a traité auparavant). Donc il faut changer les feux le plus rarement possible.

CONSEQUENCE:

La solution du problème d'optimisation pour une densité de trafic faible est caractérisée par une fréquence de changement de feux grande proche de la limite et inversement pour une densité élevée celle-ci doit être petite proche de la limite. Cette constatation détermine le cadre de toutes les informations de contraintes que l'on peut mettre pour décrire le fonctionnement d'un carrefour isolé

I - 5 - 2. CAS DE DEUX CARREFOURS ADJACENTS :

Soient deux carrefours  $i$  et  $j$  reliés par un Tronçon.



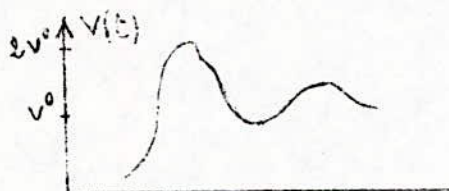
$L_{ij}$  : Longueur du Tronçon

$L_v$  : Longueur d'un véhicule.

Le maximum de véhicules pouvant occuper tout le Tronçon :

$$2 v_{ij}^0 = L_{ij} / L_v$$

À  $t \cong 0$  le nombre de véhicules attendant le passage dépend des circonstances de circulation.



Avec

$$0 \leq v(t) \leq 2v^0$$

En effet on s'intéresse surtout à la simulation des carrefours, les Tronçons sont assimilés à des dispositifs de retard.

De ce fait on peut rendre les carrefours fonctionnant indépendamment l'un de l'autre en tenant compte du retard existant entre les 2 carrefours. Plus précisément si le véhicule le plus rapide met 10s par exemple pour aller du carrefour  $i$  au carrefour  $j$  l'influence de  $i$  sur  $j$  ne se fera sentir qu'avec un décalage supérieur ou égal à 10s.

C'est-à-dire si  $\lambda$  désigne le temps minimal d'interaction entre les deux carrefours, on peut effectuer sur le carrefour  $i$  entre  $T$  et  $T + \lambda$  une simulation qui ne tiennne pas compte des événements qui ont lieu à l'autre carrefour  $j$ .

Supposons qu'à l'instant  $t_0 = 0$ ,  $V(t) = V^0$

$$V(t) = V^0 + \int_0^t y_i(\tau) d\tau - \int_0^t x_j(\tau) d\tau$$

$$V(t) = V^0 + \int_0^t [y_i(\tau) - x_j(\tau)] d\tau$$

$$0 \leq V(t) \leq 2V^0$$

D'où

$$\max_{t \in [0, T]} \left| \int_0^t [y_i(\tau) - x_j(\tau)] d\tau \right| = f(x, y) \leq V^0$$

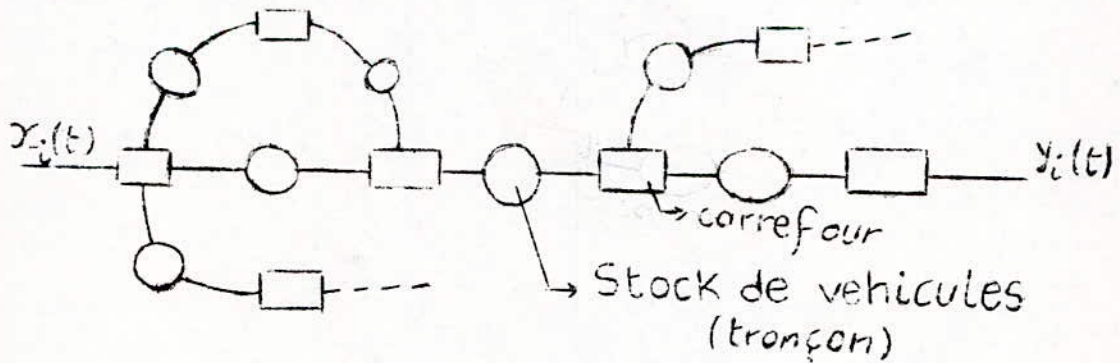
La coordination des fets caractérisant l'évacuation de véhicules dans les deux carrefours conduit à des stratégies appartenant à l'espace de fonctions possible que l'on peut utiliser, déterminées par les conditions de circulation imposées.

En effet si on fixe la stratégie en  $i$  c'est-à-dire si on fixe  $y_i(t)$  l'ensemble de stratégies qu'on peut appliquer en  $j$  est défini par une boule de centre  $y_i(t)$  et de rayon  $V^0$  c'est-à-dire :

Ayant  $y_i(t)$  fixé  $\rightarrow x_j(t) \in B(y_i(t), V^0)$

Et ayant  $x_j(t)$  fixé  $\rightarrow y_i(t) \in B(x_j(t), V^0)$

### I - 5 - 3. GAS D'UN RESEAU DE CARREFOURS :



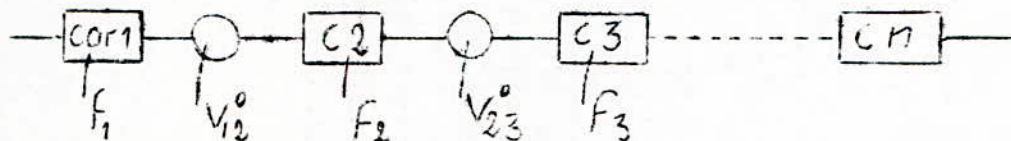
Les Tronçons sont assimilés par des modèles de stockage de véhicules.

Le modèle de chaque Tronçon définit le nombre de véhicules stockés.

Le modèle de chaque carrefour définit la fonction  $f$  caractérisant l'évacuation ( $f$  est localement convexe ou concave) et par la suite le type de stratégie à utiliser,



Le problème de contrôle pour l'agglomération de carrefours toute entière devient assez complexe. Pour maximaliser la somme de trafic passant par tous les carrefours dans un intervalle de temps, ou pour chaque temps on pourrait admettre que la solution à la sortie de n'importe quel carrefour dépend de l'entourage le plus proche possible, mais ceci n'est pas suffisant pour une solution optimale. Il faut simplifier le problème en disant qu'on est intéressé par la solution optimale instantanée pour un réseau bien déterminé débutant et finissant par des carrefours bien déterminés.



On sait qu'ayant la stratégie dans un carrefour fixée nous pouvons en déduire celle du carrefour adjacent

$$y_1(t) \text{ fixé} \Rightarrow x_2(t) \in B(y_1(t), V_{12}^0)$$

$$x_3(t) \text{ fixé} \Rightarrow y_2(t) \in B(x_3(t), V_{23}^0)$$

De cette manière on peut résoudre plusieurs problèmes d'optimisation locale en procédant comme suit :

On fixe les stratégies dans les carrefours pairs afin d'endéduire celles dans les carrefours impairs, puis on fait l'inverse. C'est-à-dire on règle les carrefours de numéros impairs en parallèle ensuite ceux de numéros pairs.

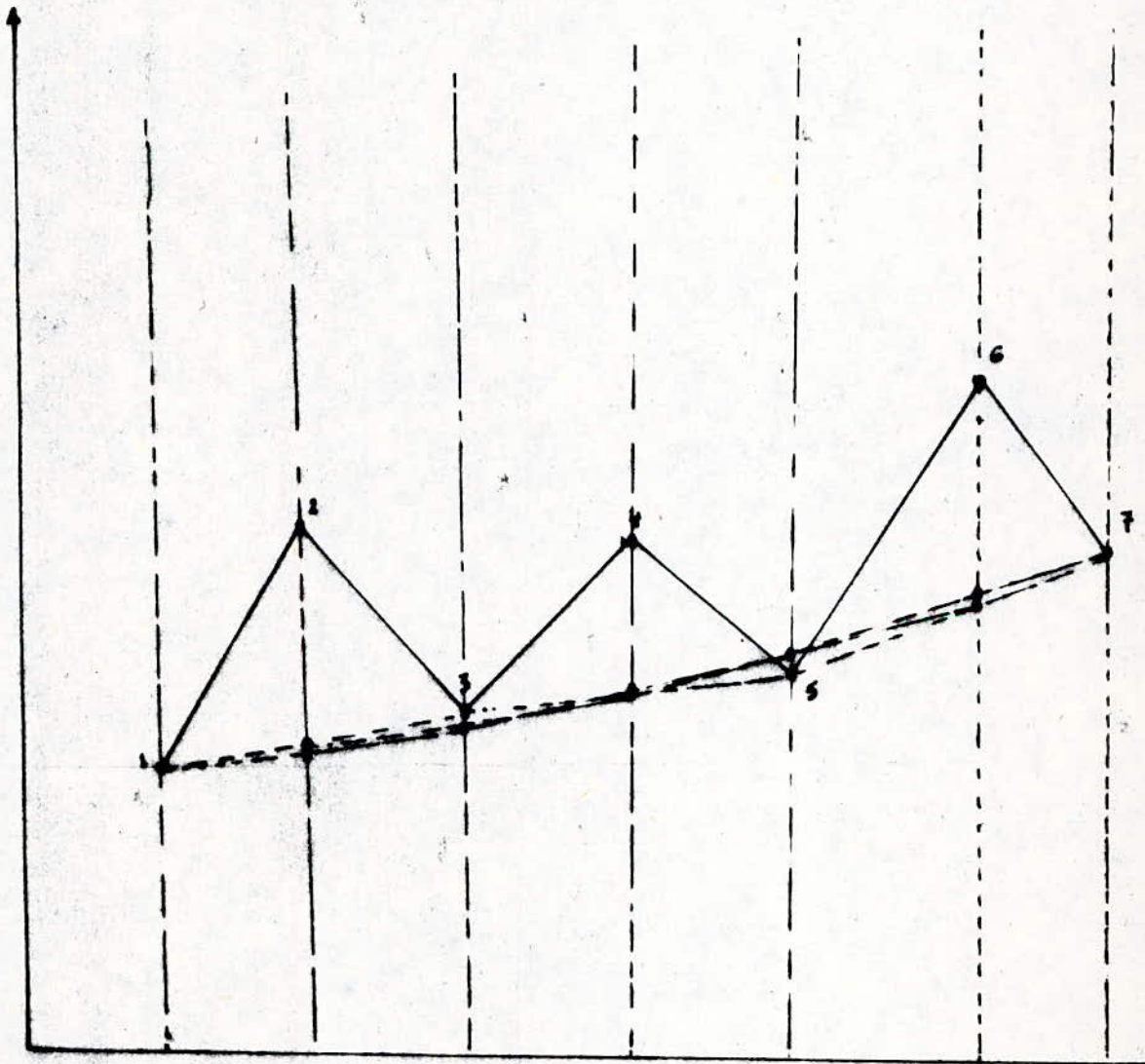
Alors chaque système peut être décomposé en plusieurs sous-systèmes pour lesquelles on peut utiliser cette méthode qu'on vient de décrire, pour résoudre les problèmes d'optimisation pour chaque sous-système.

Pour un réseau de carrefours répartis d'une façon quelconque, le principe d'optimisation que nous proposons d'appliquer est le suivant :

Les carrefours sont assimilés à des points répartis sur un plan d'une façon aléatoire, reliés par des lignes constituant une ligne brisée. Les noeuds de cette ligne brisée obéissent à une certaine contrainte qui consiste à les faire déplacer seulement selon les **verticales** passant par ces derniers. En reliant les points impairs, on peut déplacer les pairs, suivant la vertical toujours, sur les lignes de liaison. Et en faisant le même pour les points pairs obtenus, on constate que la ligne brisée tend clairement à devenir droite (voir fig. I).

Cette idée peut nous servir de solution optimale de carrefour : il faut trouver un rythme de changement de feux pour donner à chaque carrefour la solution optimale. Vu le phénomène aléatoire de densité de trafic cette solution est assez compliquée mais elle peut être applicable dans le cas où on a des dispositi

(fig-I-A-)





qui déterminent l'état des stocks à chaque instant. Mais dans notre projet nous sommes incapables d'envisager cette solution optimale car ceci exige des expériences assez compliquées sur ordinateur de grande taille. Donc ce que nous venons de traiter n'est alors qu'un petit aperçu sur l'optimisation de la régulation.

## I - 6 - SIMULATION DE CARREFOURS:

La simulation de carrefours qu'on utilisera s'effectue par microprocesseur en mémoire centrale en programmant toutes les informations nécessaires et propres aux carrefours considérés.

Lorsqu'on passe de la simulation d'un carrefour à une autre il faut effectuer des transferts sur les supports périphériques contenant les données relatives au carrefour (ces supports sont de capacité pratiquement illimitée).

Ce sont ces transferts qui consomment le plus le temps? Ainsi il est important d'avoir une méthode qui minimise le nombre de ces transferts. On avait d'abord imaginé une méthode où les carrefours sont traités en séquence, et tous pendant le même temps. On a mis au point un processus d'appel des carrefours tels que ceux-ci peuvent être appelés pendant des périodes différents mais, pour chaque carrefour, ces périodes sont constantes au cours du temps. La seule précaution qu'il convient d'observer est qu'entre carrefours adjacents, les décalages dans les temps de traitement ne dépassent pas le temps de propagation de l'information d'un carrefour à l'autre c'est-à-dire le temps de parcours du véhicule le plus rapide. Dans ce cas là, les carrefours peuvent être considérés comme fonctionnant indépendamment les uns des autres.

Comme systèmes de solution on envisage deux types :

- Le système le plus simple dispose d'un contrôleur de trafic pour chaque groupe d'intersections réalisant une régulation locale - microrégulation - tout en respectant le temps de décalage entre deux carrefours adjacents.

- Un système plus complexe laisse à un ordinateur situé au poste central le soin de prendre toutes les décisions de régulation, le matériel installé sur les carrefours ne jouant qu'un rôle d'exécution - macrorégulation -

Le processus de régulation est alors le suivant :

- Un programme moniteur détermine quel carrefour (disons C) doit être traité et pendant combien de temps (disons de T à T') - on va dans ce qui suit voir comment déterminer ce temps de traitement.

- Les informations relatives au carrefour C sont entrées en mémoire centrale ainsi que celles relatives aux véhicules sortant des carrefours adjacents à C.

- On vide les voies sorties des carrefours adjacents à C dans la voie entrée du carrefour C (simulation du Traçon)

- On engendre les véhicules entrant dans le système au carrefour C entre T et T'.

- On met l'horloge à T et on procède à la simulation du carrefour jusqu'à ce que l'horloge marque T'.

- Les informations relatives au carrefour C sont envoyées sur le support périphérique.

#### SIMULATION DU TRONÇON :

La simulation du tronçon reliant le carrefour i à celui i + 1 consiste à faire sauter les véhicules des voies sortie du carrefour i aux voies entrées du carrefour i + 1 en leur affectant un certain retard. Ce retard est évalué à partir des temps de parcours mesurés, en fait, il est tiré d'une distribution dont les paramètres sont ajustés sur les résultats des mesures. Ces derniers sont insuffisants pour qu'on tienne compte de l'effet de l'encombrement sur le temps de parcours.

#### I - 6 - 1. MICROREGULATION :

##### PRINCIPES DE REGULATION : (l'intervalle critique)

Le cycle est divisé en phases (2 ou 3) qui passent au vert successivement autorisant certains mouvements. Entre chaque phase il y a une courte période de rouge intégral.

- Lorsqu'une phase passe au vert elle y demeure au moins pendant un temps dit (temps de vert minimum -  $V_{min}$ ).

- Lorsque le temps de vert minimum sur une phase est écoulé le signal passe au rouge et le cycle suivant passe au vert (après le temps de rouge intégral) si les 2 conditions suivantes sont réalisées :

a) Il existe au moins un véhicule en attente sur une voie dont le feu est au rouge.

b) L'intervalle de temps séparant le passage de deux véhicules successifs sur la ligne des feux des voies dont le feu est au vert est supérieur à un intervalle critique  $I_c$ .

Si au bout d'un temps  $V_{r1}$  (temps du vert max) les deux conditions ci-dessus n'ont pas été réalisées le feu passe au rouge et le cycle suivant passe au vert (après un temps de rouge intégral). Et si certaines entrées sont très chargées par rapport à celle dont le feu est au vert on favorise l'accès sur lequel on a décelé une longue file d'attente :



- En diminuant l'intervalle critique  $I_c$  pour profiter d'un trou plus petit pour interrompre une phase antagoniste. C'est la raison pour laquelle on utilise un intervalle critique variable

### INTERVALLE CRITIQUE :

Soit  $I$  l'intervalle de temps qui sépare l'arrière d'un véhicule de l'avant de celui qui le suit. Nous avons constaté que cette durée a pour valeur moyenne de 2s en ce qui concerne une file de véhicules qui démarre après un feu rouge. Cette valeur peut varier en fonction des conditions locales, pente et la largeur de la voie, présence de stationnement, etc, ...

Mais pour une voie donnée et une fois que le régime permanent est établi, cette valeur s'écarte peu de sa valeur moyenne. En revanche, même si dans un écoulement libre, la moyenne des intervalles a le même ordre de grandeur, la dispersion de valeurs individuelles est plus forte.

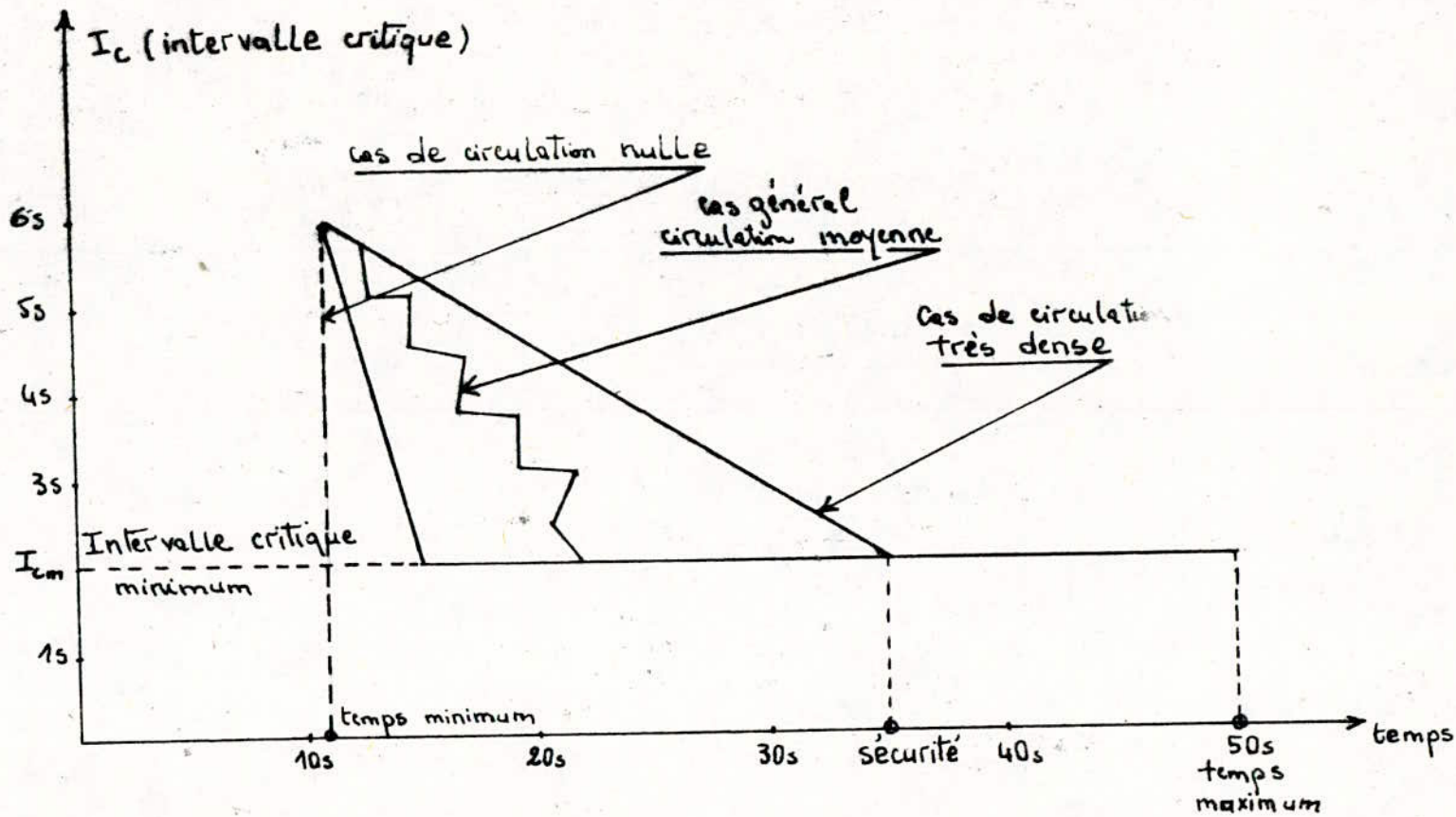
On appelle  $I_c$  (l'intervalle critique) la valeur telle que :

Si  $I \leq I_c$  on maintient le feu au vert

Si  $I > I_c$  on coupe cette phase

On obtient alors le résultat suivant ; tous les véhicules qui ont été arrêtés devant le feu se présentent avec un  $I < I_c$  et passent donc au premier feu vert ; en revanche, les véhicules arrivant librement se présentent en général avec  $I > I_c$  et sont donc obligés de s'arrêter devant le feu rouge. Tout véhicule est donc amené à s'arrêter une fois devant le feu et une fois seulement s'il n'existe pas d'autres voies plus intenses qui conduisent à un basculement des feux sans qu'il y ait un dépassement de  $I_c$ .

En plus de cette raison pour laquelle on utilise un intervalle critique variable on en rencontre une autre qui consiste au danger pouvant se présenter en utilisant cette méthode de régulation qu'on vient de décrire : il est en effet toujours possible de trouver des  $I$  supérieurs à  $I_c$  au début d'une phase (démarage lent) et il est indispensable de ne pas couper la phase dans cette circonstance. La figure 2 montre comment varie  $I_c$  en fonction du temps et du trafic. Sa valeur initiale est égale à trois sa valeur finale et sa décroissance est d'autant plus rapide que les intervalles anormaux dans le trafic sont nombreux et importants.



(Fig 2) Variation de  $I_c$  au cours d'une phase de vert



I - 6 - 2. MACROREGULATION :

CHOIX DES PARAMETRES :

Le choix du paramètre qui représentera l'intensité de circulation est un élément essentiel à la qualité de la régulation obtenue. Reportons-nous à l'objectif de régulation qui consiste à adapter la capacité de débit à la demande. Il est donc nécessaire de mesurer la demande de débit, cette demande est égale au nombre de véhicules qui se présentent pour entrer dans la zone régulée pendant l'unité de temps. Supposons que les conducteurs désirent rouler à 60 Km/h ; nous pouvons dire que tous les conducteurs situés dans une zone de 1 Km avant la zone régulée désirent y entrer en cours de la minute qui suit (puisque à 60 Km/h, 1 Km est parcouru en une minute). La demande de circulation est donc égale au nombre, de véhicules contenus dans le kilomètre de voie précédant la zone à réguler, ce dernier paramètre est également appelé (concentration). Le problème donc est de mesurer cette concentration. L'idée la plus simple consiste à compter les véhicules qui entrent dans la zone de 1 Km et ceux qui en sortent et de tenir à jour le compte de ceux qui s'y trouvent. Mais cette mesure directe est impraticable en ville pour diverses raisons :

- Il est généralement impossible de définir une zone de mesure de cette longueur.

- L'éloignement des capteurs hors de la zone régulée en augmente considérablement le prix.

- La précision des comptages : Dans un tel système les erreurs se cumuleraient

Il est donc nécessaire de déduire la concentration de mesures faites sur un tel détecteur. Le paramètre le plus facile à mesurer est le volume c'est-à-dire le nombre de véhicules par unité de temps.

A l'endroit où se trouve le capteur, 3 types de circulations sont possibles

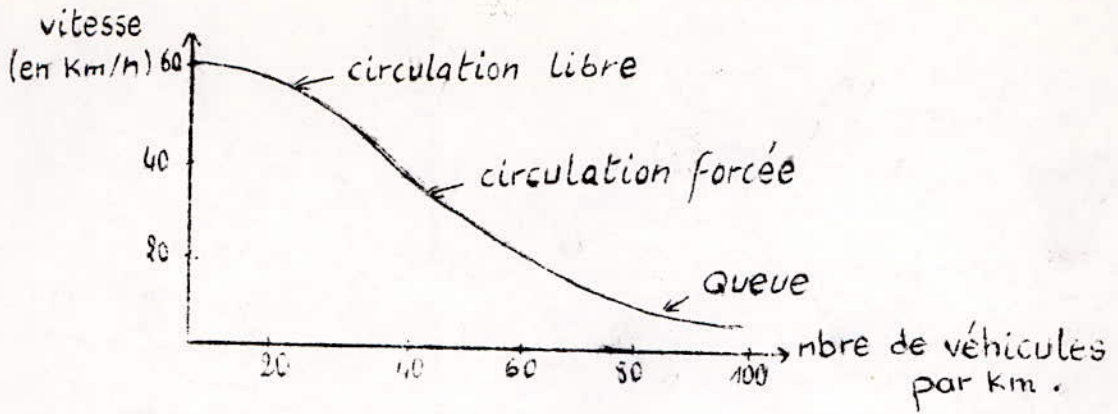
a) Circulation libre :

Chaque conducteur roule à la vitesse qu'il désire sans être gêné par les autres véhicules. Nous avons supposé que la vitesse désirée étant de 60 Km/h.

b) Circulation forcée sans arrêt :

Lorsque le niveau de circulation augmente, les véhicules se gênent entre eux et la vitesse diminue.

c) Queue : Comme il est souvent impossible de situer le capteur suffisamment loin du premier carrefour, il arrive que la queue repose sur celui-ci. La fig.3 donnant la variation de la vitesse en fonction de la concentration illustre ces phénomènes.

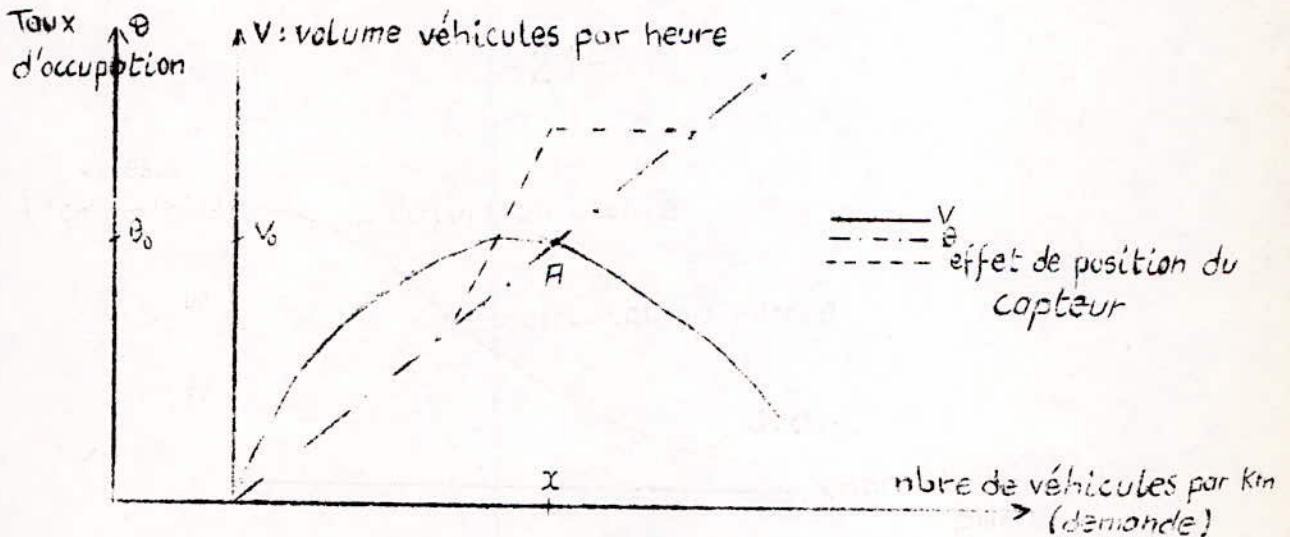


(fig.3.) Vitesse en fonction de la concentration.

Sur la (fig.4) nous avons reporté le volume en fonction de la concentration. Nous constatons que la relation entre ces deux paramètres n'est pas biunivoque, le volume n'est donc pas un paramètre satisfaisant. On introduit alors un autre paramètre qui est le taux d'occupation défini par rapport  $\frac{\sum t_i}{T}$

où  $t_i$  : La durée du passage d'un véhicule

T : durée de mesure



(fig 4) volume et taux d'occupation en fet de la concentration

Si on suppose que la vitesse est homogène sur un kilomètre, le taux d'occupation sera proportionnel à la concentration. On voit que lorsque la demande commence à croître, le débit augmente plus vite que le taux d'occupation, ce qui permet de mettre rapidement en évidence une demande de trafic plus importante. Par contre lorsque le niveau de saturation est atteint, l'information occupation prend tout son intérêt. En général le taux d'occupation ne dépasse pas 50 %.



Les 2 courbes se coupent en A donnant un rapport de  $C = \frac{V_0}{e_0}$  entre les échelles. Il suffit donc de faire la lecture du volume toutes les  $3600/c$  secondes.

Le choix du cycle dépend de la valeur de seuil obtenue, et se fait par programmation, ce qui conduit à un changement du plan de feux selon l'état du carrefour.

Ce changement peut causer des perturbations sur le trafic, c'est pourquoi il faut limiter le nombre de changements, cette solution a l'inconvénient de retarder la mise en place d'un plan de feux en cas de pointe rapide au trafic.

Nous proposons la solution qui consiste à mettre tout de suite en place un nouveau plan de feux demandé par accroissement du trafic par exemple et d'attendre trois confirmations en cas de diminution du trafic (demande de cycle plus court).

CHAPITRE    II

—o— PARTIE    -    STATIQUE    —o—



II - 1. INTRODUCTION :

Après cette étude de régulation on conçoit que la solution par micro-ordinateur peut être mieux adoptée que celle par circuit séquentiel câblé par ce genre de problème assez complexe et ceci pour plusieurs raisons :

- Souplesse de la logique programmable :

Il est à réaliser au cours du cycle de l'automatisme est simplement réalisé par programme, ce qui nous procure une grande souplesse d'utilisation. Tandis-qu'en logique câblée la configuration fonctionnelle est rigide, toute modification ou extension n'est pas aisée, elle entraîne généralement une modification complète du système comprenant plusieurs circuits intégrés qui sont probablement irrécupérables.

Par contre si le système a été conçu en logique programmée, tout se fait par programme sans modification du matériel. De cette manière si on désire modifier la stratégie des feux on aura qu'à changer de programme.

- La fiabilité de mise au point :

Dans la conception d'un système logique, il y a toujours une part d'imprévisible ou d'erreur dès-que le système est un peu complexe, et aussi un petit nombre d'erreurs qu'il faudra détecter lors de la mise au point. Et là, le concepteur de systèmes logiques câblés est livré à sa propre expérience car il n'y a pas de méthodes générales de mise au point. Par contre, en logique programmable le concepteur peut se permettre d'un micro-processeur. Il disposera de programmes de détection des erreurs qu'il lui sera possible d'utiliser grâce à ce micro-ordinateur spécialisé.

- La consommation et l'encombrement réduits.

Malgré tous les avantages de la logique programmable elle ne peut se substituer complètement à la logique câblée. La logique programmable bipolaire et la logique programmable MOS sont respectivement 10 et 100 fois moins rapides que la logique câblée TTL et ne peuvent remplacer cette dernière dans des applications où le critère du choix est la vitesse. Mais dans notre cas, et vu que la durée d'exécution des instructions de programmation est négligeable (de l'ordre de  $\mu s$ ) par rapport à la durée des feux, nous avons préféré que notre système soit conçu à base de la logique programmable permettant la régulation de carrefours pour n'importe quelles densités de circulation.

II - 2. SYSTEME DE REGULATION :

II - 2. 1 - PROPRIETES :

Le but principal de celui-ci est de gérer la circulation des véhicules de la traversée des piétons au niveau des intersections simples, mais ayant des propriétés universelles, il peut être utilisé indifféremment à des carrefours isolés ou dans les réseaux de commande coordonnée.

Exploitant tous les avantages procurés par la technique moderne des micro-ordinateurs, notre système peut remplir plusieurs modes de fonctionnement, ces derniers sont déterminés grâce au logiciel de base c'est-à-dire au programme figé dans la mémoire morte offrant une grande variété de fonctions afférentes à la technique du trafic. Ces fonctions peuvent être activées par programmation correspondante de la phase mémoire disponible dans la mémoire des données. Toutes les données et tous les paramètres du trafic spécifiques au carrefour, ainsi que la logique des fonctions commandées par le trafic sont introduites en mémoires, les paramètres spécifiant le trafic au cas d'utilisation en présence sont déterminés grâce à des détecteurs à boucles inductives placées sur la chaussée de chaque voie au dessous de son niveau de quelques centimètres.

II - 2 - 2. MODES DE FONCTIONNEMENT :

a) Autonome :

Fonctionnement normal dans un carrefour, comme c'est le cas actuellement à ALGER. Il se fait à cadence fixe avec un seul plan de feux considéré dans les conditions maximales de saturation.

Fonctionnement adaptatif dont la durée des phases est fonction des paramètres enregistrés par les détecteurs, autrement dit, selon la densité du trafic le lancement des différentes phases est retardé ou accéléré.

b) Coordonné :

La commande coordonnée consiste en la commande des feux de deux ou plusieurs carrefours voisins sur synchronisme réciproque. En cas normal, cette coordination a lieu par câble à partir d'une commande centrale. Pour un cas simple on peut utiliser une coordination sans fils (commande par horloge).

c) Clignotant :

En cas de panne du système de contrôle ou pendant la nuit où la circulation est moins intense les feux passent au Jaune-clignotant.



Dans la partie programmation de notre étude nous nous limitons au fonctionnement d'un contrôleur de trafic dans les carrefours isolés en tenant compte du temps de simulation du tronçon entre deux carrefours adjacents, vu que notre pays avec les moyens qu'il a ne peut pas se permettre d'avoir une commande centralisée.

### II - 3. DETECTEURS :

Pour l'enregistrement des données techniques du trafic routier, on se sert aujourd'hui de détecteurs à boucle inductive. Ils se distinguent avant tout par une surface exactement définie, dans laquelle ils enregistrent les corps métalliques. Cette surface de la chaussée dans laquelle la boucle inductive est coulée peut être définie comme poste de mesure de trafic. Lors de la présence d'un véhicule, le détecteur envoie une impulsion au moyen de laquelle les véhicules peuvent être annoncés ou comptés. Les valeurs numériques servent par exemple à déterminer l'importance du trafic (véhicules /h) et constituent ainsi la base d'une statistique de la circulation et de sa planification. A partir de la durée de l'impulsion ou d'une suite d'impulsions, il est possible de calculer certaines valeurs qui jouent - en particulier pour la commande du trafic - un rôle important ; ainsi par exemple : le taux d'occupation (qui représente une évacuation de la densité du trafic, ou encore l'intervalle entre véhicules. En combinant plusieurs détecteurs, il est possible d'exploiter d'autres informations sur la direction, le genre de véhicules, leur vitesse et la longueur des embouteillages. Celles-ci servent par exemple à influencer le trafic sur les voies rapides ou au contrôle de trafic dans les tunnels.

Le détecteur M qu'on utilisera (fig.5) fourni par Siemens est le premier détecteur de trafic au monde dans lequel les fonctions sont exécutées par un micro-ordinateur. Celui-ci permet l'emploi d'un nouveau système d'exploitation digitale, dont les qualités de mesure sont particulièrement remarquables. Le détecteur M peut être utilisé de façon universelle car tous les paramètres sont réglables au moyen de commutateurs. Ainsi le détecteur M est adaptable à chaque situation de façon précise et rapide, ainsi il est applicable pour n'importe quelle stratégie utilisée.

#### II - 3 - 1. PROPRIETES :

- Le détecteur possède 2 modes d'exploitation :
  - "Puls" (pulsation)
  - "Anwesenheit" (Présence)
- 6 degrés de sensibilité  $\Delta L/L = 4.10^{-2} \dots\dots 5.10^{-5}$



- Durée de maintien de l'information réglable jusqu'à 26 ms (13 positions)
- 8 fréquences de travail. (40 à 110 KHZ)
- Auto-contrôle intégré de la boucle
- Protection intégrée contre la surtensions sur la boucle;

Modes de fonctionnement : (fig. 6)

Sur le mode d'exploitation "Puls" l'information disparaît lorsque la durée réglable de l'impulsion (de 100 à 500 ms) est écoulee. Si un véhicule reste arrêté sur la boucle il sera oublié après un temps de répit de 1,5s , de manière à ce que les véhicules suivants puissent être enregistrés de nouveau normalement. Sur le mode d'exploitation "Anwesenheit", l'information disparaît lorsque le véhicule a quitté la boucle et dans tous les cas lorsque la durée réglable de maintien de l'information est dépassée. Grâce à un phénomène hystérésique (fin de prise en compte de l'information lorsque la valeur du signal est égal à la valeur moitié de celle du seuil de prise en compte de l'information), il est possible de générer dans les cas limites des signaux de sortie stables.

L'inductance de la boucle et sa ligne d'alimentation, observés dans le temps subit des variations dues aux influences du milieu ambiant, mais la nouvelle fréquence de la boucle qui en résulte se corrige elle-même. Ainsi aucun réajustement n'est nécessaire.

II - 3 -2. PRINCIPE DE FONCTIONNEMENT :

Le principe repose sur le changement d'inductance d'une bobine dans le champ de laquelle se met un corps métallique. Une boucle conductrice coulee dans la chaussée constitue la bobine. Quand un véhicule se trouve sur la boucle, l'inductance est diminuée et la fréquence du générateur d'alimentation est augmentée. Le microprocesseur mesure la fréquence de la boucle dans des cycles de mesure successifs. Dans chaque cycle de mesure on compte le nombre de pulsations d'un générateur de référence à quartz , pendant une durée définie à partir de la fréquence de la boucle.

Le nombre ainsi obtenu pendant cet intervalle est inversement proportionnel à la fréquence de la boucle. En faisant la différence de ces nombres de pulsations successifs, il est possible d'enregistrer le changement de fréquence de la boucle. Cette différence est employée pour détecter la présence des véhicules. La sensibilité du seuil de détection est réglable.



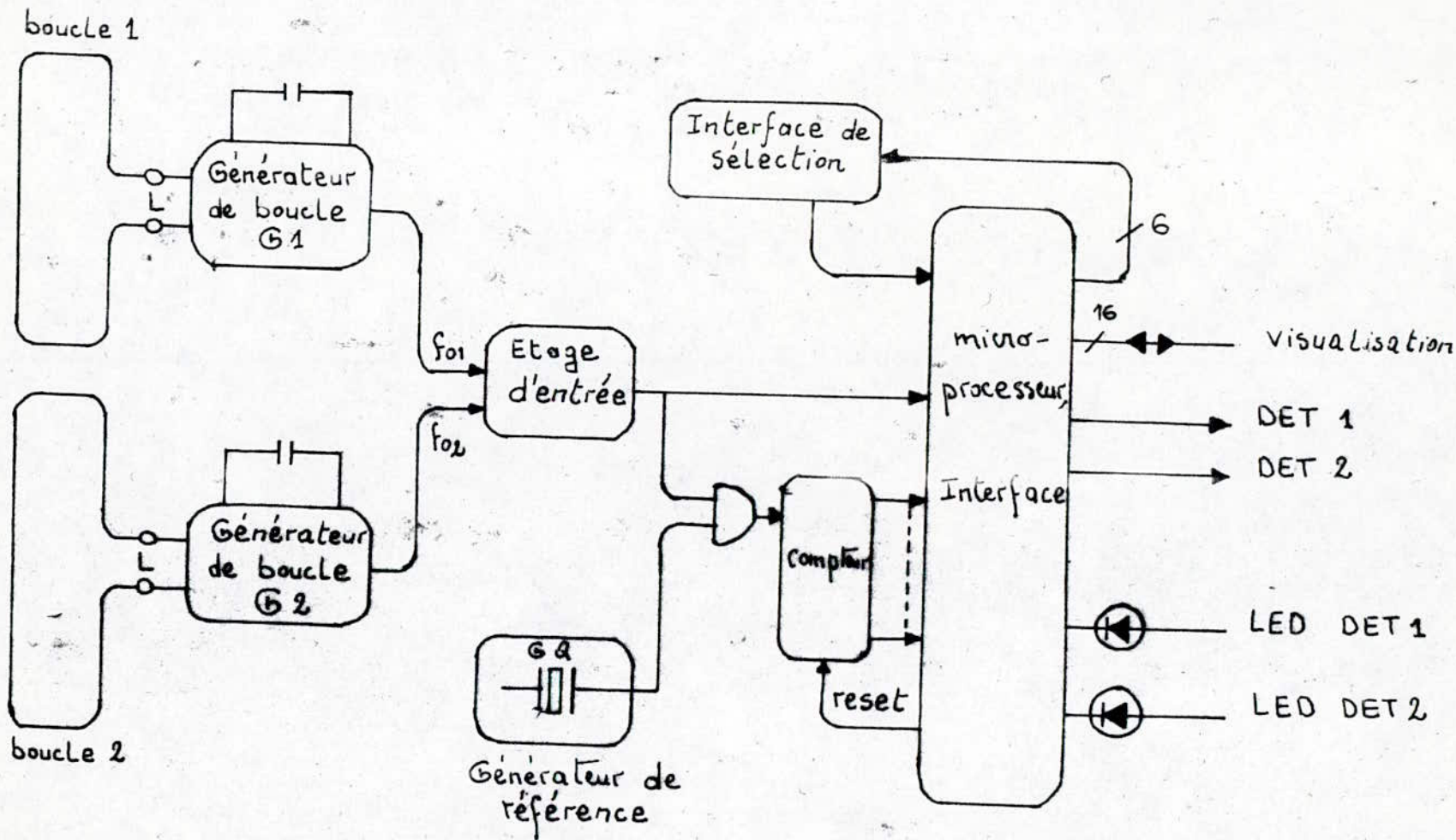
Chaque carte de détecteur comporte deux blocs de détection. Chaque boucle est alimentée par son propre générateur de fréquence, respectivement G1 et G2. Pour éviter toute inter influence, on utilise des fréquences différentes ( $f_{01}$  et  $f_{02}$ ). Ces fréquences sont réglables numériquement.

La mesure des fréquences respectivement  $f_{01}$  de G1 et  $f_{02}$  de G2 se fait tour à tour sous forme numérique. Selon la séquence de mesure, le microprocesseur dirige le signal de mesure de  $f_{01}$  ou de  $f_{02}$  vers l'étage d'entrée dont le signal de sortie forme le signal d'entrée du microprocesseur. La durée de mesure est définie par celle d'un certain nombre de périodes à la fréquence de la boucle ce nombre, lui, dépend de la sensibilité du seuil de détection.

Pendant cet intervalle de mesure, on compte les pulsations du générateur de référence. Le nombre de pulsations ainsi obtenu par cycle de mesure est inversement proportionnel à la valeur de la fréquence de la boucle correspondante.

Ces valeurs constituent les grandeurs de base pour la suite de l'exploitation.

A partir de ces grandeurs de base, le microprocesseur génère sur son interface de sortie les signaux de détection DET1 et DET2 et interroge l'interface de sélection sur le choix du mode d'exploitation, sensibilité de seuil de détection, les temps limites et, si nécessaire, exécute l'échange de données entre le détecteur et un appareil de visualisation servant à afficher le mode de travail du détecteur.

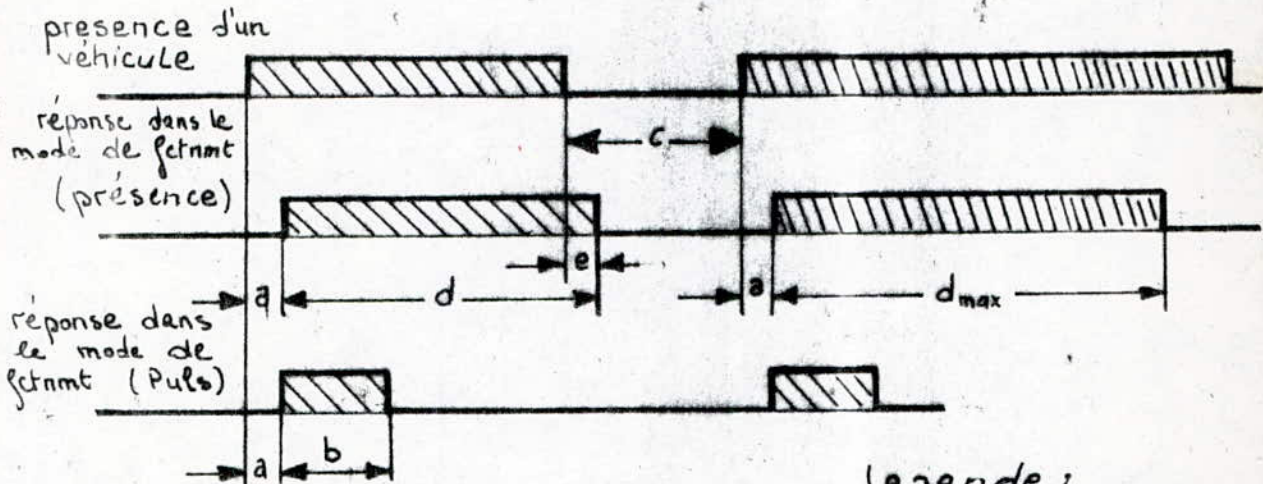


(fig 5)

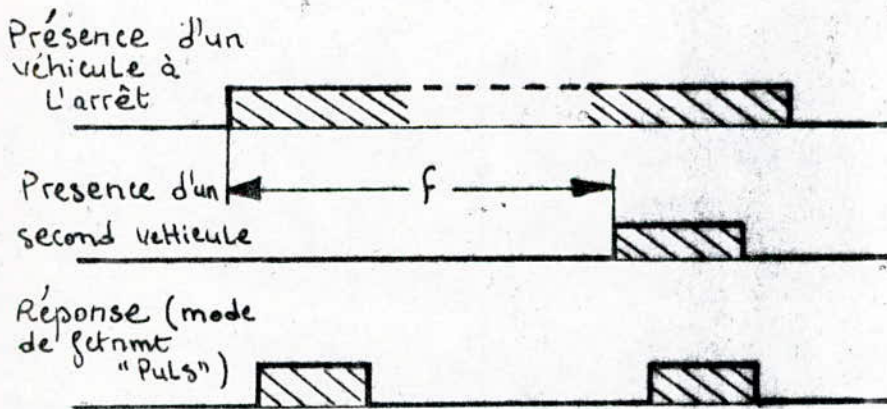
Schéma de principe du détecteur



Cas normal : 1 véhicule sur la boucle



Cas particulier : 2 véhicules sont côte à côte sur la boucle



Legende :

a: Temps de réaction lorsque le véhicule arrive sur la boucle.

b: durée d'impulsion réglable

c: intervalle entre véhicules.

d: durée de présence du véhicule.

d<sub>max</sub>: durée max. de maintien de l'inf. réglable

e: temps de réaction lorsque le véhicule quitte la boucle

f: temps nécessaire au repérage d'un deuxième véhicule sur la boucle.

(fig 6) Mode de réponse du détecteur

#### III - 4 - ORGANISATION FONCTIONNELLE DU SYSTEME :

L'unité centrale de contrôle est bâtie autour d'un microprocesseur type MC 6800 est comprend la carte microprocesseur plus la carte de commande de puissance de feux.

##### III - 4 - 1. CARTE MICROPROCESSEUR MPU :

Le microprocesseur à mettre en oeuvre nécessite :

- Un microprocesseur gérant le programme de contrôle de l'automatisme.
- Une mémoire du système (ROM) contenant les programmes de gestion du système
- 1024 X 8 bits -
- Une mémoire RAM contenant les données non figées spécifiques aux carrefours
- 128 X 8 bits -

On peut même utiliser plusieurs mémoires en cas de nécessité d'un espace mémoire assez grand.

Les mémoires contiennent le programme de contrôle qui peut être modifié selon le type de carrefour à réguler et suivant le type de stratégie adoptée.

Les mémoires mortes sont protégées par batterie pour ne pas perdre leurs informations en cas d'absence d'alimentation.

#### INTERFACES DE GESTION :

Le microprocesseur doit pouvoir dialoguer avec la périphérie : (claviers, capteurs, circuits de commande de feux .... ), le dialogue se fait par l'intermédiaire de coupleurs programmables, (c'est-à-dire que le rôle que va jouer le coupleur est défini par programmation). Les coupleurs utilisés pour permettre la connection sur le bus de données du microprocesseur et les périphériques indiqués ci-dessus sont du type parallèle PIA - périphéral interface adapter. on utilise :

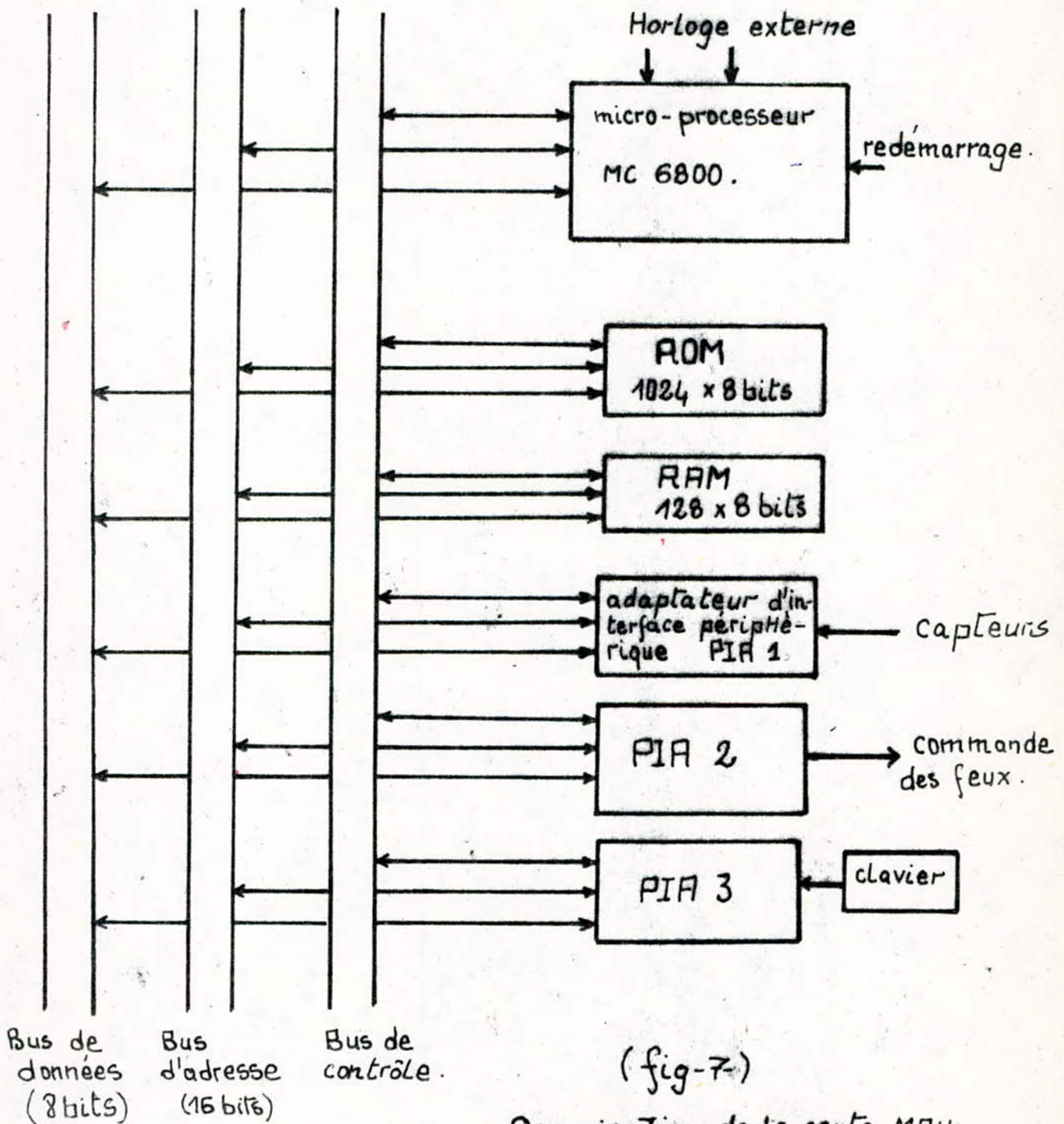
- 1 PIA : servant d'interface avec le clavier.
- 1 ou plusieurs PIA suivant la complexité du carrefour à réguler permettant au microprocesseur d'envoyer des signaux de commandes des feux, et de recevoir les signaux provenant des détecteurs.

Pour l'exemple de carrefour qu'on va étudier dans ce qui suit on aura besoin que de deux PIA (fig 7)

Pour le microprocesseur M 6800 le coupleur parallèle (PIA) utilisé est le M 6820 comprenant deux ports A et B. Chaque port comprend :

- 8 lignes programmables individuellement en sortie ou en entrée.
- 2 lignes de contrôle dont l'une est programmable en entrée et l'autre est programmable en entrée ou en sortie.





(fig-7)

Organisation de la carte MPU

En ce qui concerne la constitution interne chaque port comprend :

- Un registre de données DRA ou DRB
- Un registre de direction des données DDRA ou DDRB
- Un registre de controle du PORT CRA ou CRB.

Il s'ensuit qu'on aura besoin de 2 bits pour l'adressage des registres d'un PIA (voir partie Annexes - Description complète du PIA)

SELECTION DES MEMOIRES ET DES PERIPHERIQUES

		A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
	En decimal	32768	76384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
ROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1023	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
RAM	1024	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	4151	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1
Coupleur 1	32768	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	32771	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Coupleur 2	33792	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	33795	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1

SELECTION DES BOITIERS : ROM - Ram et coupleurs

Le bit A<sub>15</sub> est utilisé pour différentier les mémoires (RAM et ROM) des coupleurs.

Le bit A<sub>10</sub> est utilisé pour différentier la ROM de la RAM et le coupleur 1 du coupleur 2.

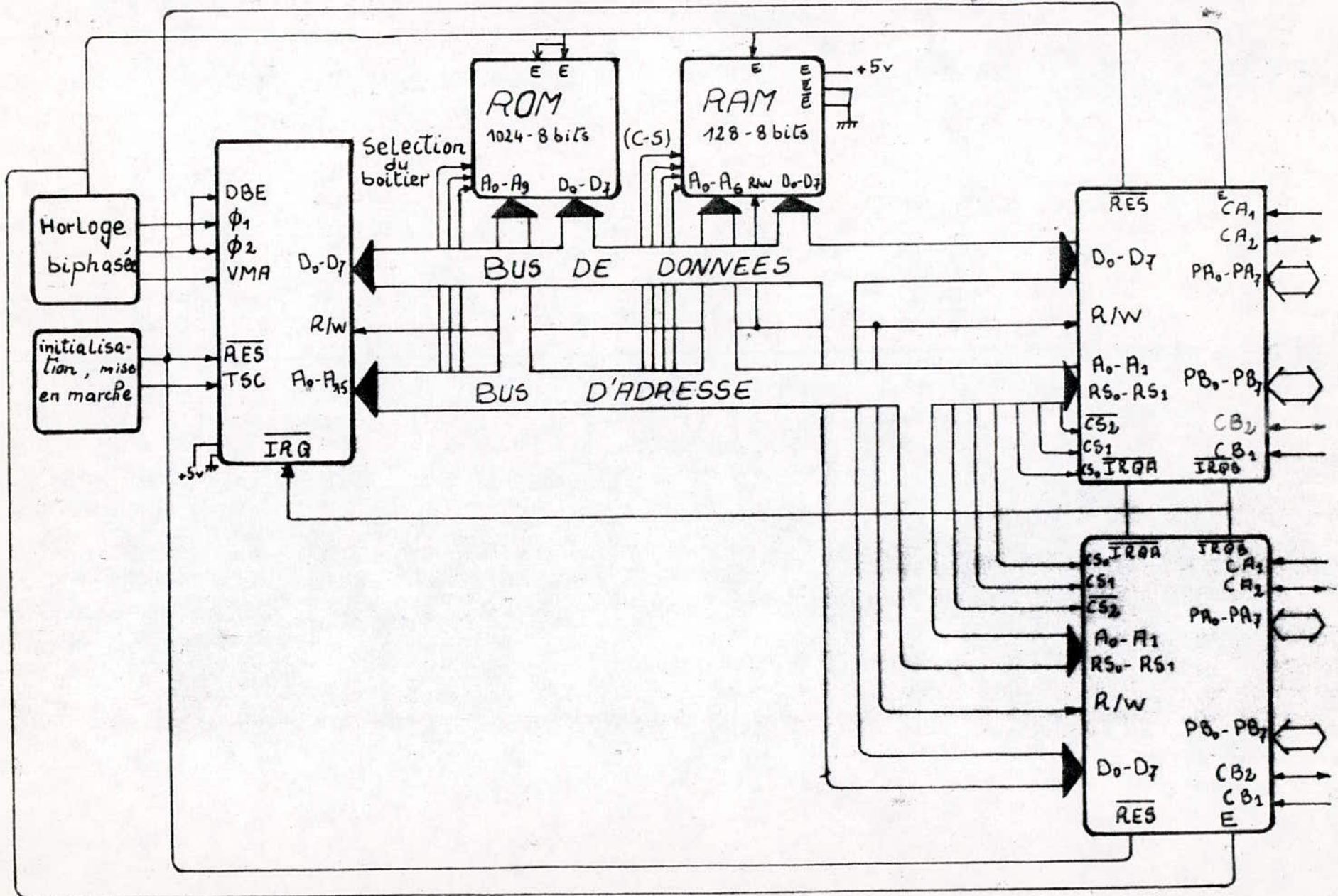
LIGNES DE SELECTION :

Chaque coupleur dispose de 3 lignes de selection (Chip select) :

CS<sub>0</sub> , CS<sub>1</sub> , CS<sub>2</sub>.



(Fig 8) Système minimum conçu autour de la famille 6800



Par conséquent les bits utilisés pour la sélection sont :

$A_{10}$  ,  $A_{15}$  plus la masse.

Les mémoires ROM et RAM :

La ROM dispose de 3 lignes de sélection: 2  $\overline{CS}$  , 1  $\overline{CS}$

La RAM dispose de 4 lignes de sélection : 1  $\overline{CS}$  et 3  $\overline{CS}$

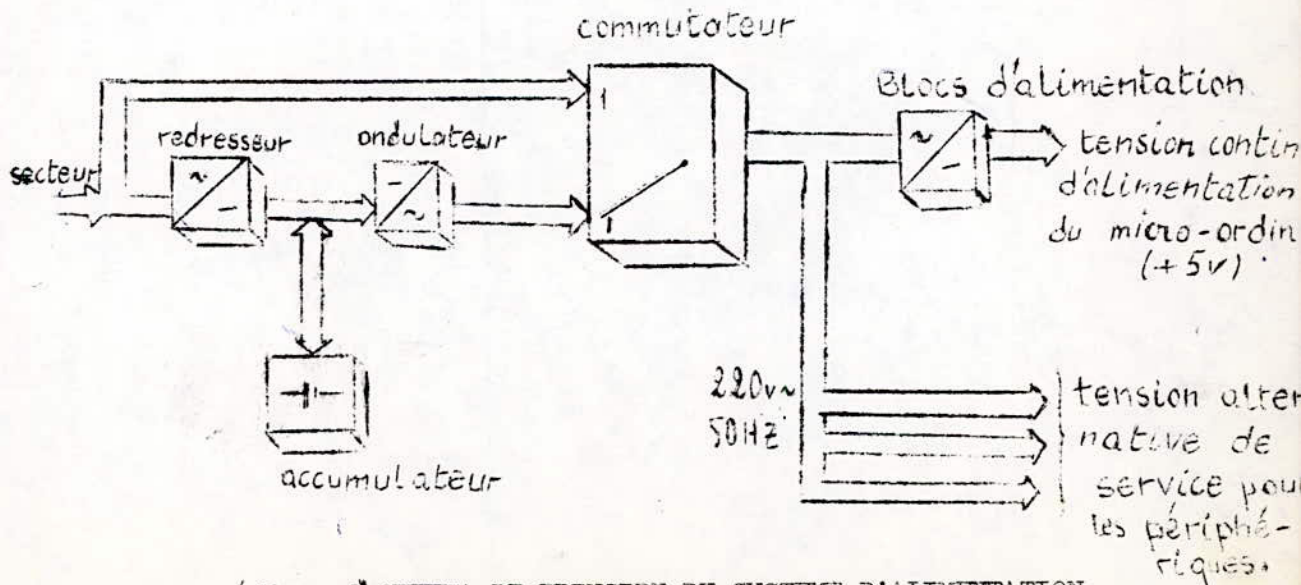
## II - 4 - 2. SYSTEME D'ALIMENTATION ET BLOC DE SECOURS : (fig.9).

Afin de garantir la continuité du service en cas de panne secteur, nous avons proposé que notre micro-ordinateur soit branché de préférence sur un bloc d'alimentation de secours en réserve active.

Un redresseur alimenté par le secteur délivre la tension du circuit intermédiaire disponible en permanence grâce à un accumulateur. Cette tension du circuit intermédiaire est transformée par un onduleur en tension alternative qui assure le fonctionnement de l'ordinateur et les périphériques.

Ce système d'alimentation assure :

- La production de la tension utilisateur régulée par tous les groupes fonctionnels.
- Le maintien de la tension indispensable à l'ordinateur et aux périphériques ainsi que de la fréquence de la tension alternative.
- La commutation immédiate sur une source d'alimentation de secours (accumulateur) en cas de panne de secteur.
- La commutation immédiate sur le secteur en cas de défaillance du système d'alimentation (par commutation électronique)



(fig - 9) SCHEMA DE PRINCIPE DU SYSTEME D'ALIMENTATION



II - 4 - 3. CARTE DE COMMANDE (en puissance) des feux : (Triac)

Les feux sont alimentés en 220 V alternatif, donc il faut un composant qui puisse être commandé par une faible tension et qui supporte 220 V vu que les signaux de commande générés par la carte MPU sont de bas niveau car la logique à microprocesseur et toute la logique qui l'entoure sont commandés par une tension d'alimentation de 5V continu.

Pour cela on utilise des cartes Triac permettant la commande en puissance des 4 phases indépendantes des feux : rouge - vert - jaune, jaune clignotant.

2.4.3 CIRCUIT DE COMMANDE EN PUISSANCE DES FEUX (fig 10)

Pour alimenter les différents étages du circuit nous avons utilisé une alimentation stabilisée de  $-12V$  appliquée à chaque émetteur des transistor utilisés comme commutateurs .

Pour chaque phase des feux , le signal de commande provenant de l'interface -P.I.A- ( $+5V$ ) attaque l'étage commutateur (transistor 2N2222A) et enclenche ce dernier c'est à dire le transistor sera dans l'état saturé ceci provoque l'amorçage du triac et par conséquent l'allumage de la lampe considérée .

En absence du signal de commande le transistor est bloqué le triac se comporte donc comme un interrupteur ouvert et le feu correspondant est alors éteint .

CALCUL DES COMPOSANTSCalcul de  $R_c$ 

La résistance  $R_c$  fixe le courant de déclenchement du triac

Pour mieux commander le triac nous lui appliquons une tension négative entre la gachette et l'anode I ; Nous prenons :

$V_{GT}$  : tension de déclenchement du triac  $\approx -2V$

$I_{GT}$  : courant de déclenchement du triac  $\approx 50mA$

$$V_{CE\text{ sat}} = 0,3V$$

$$V_{GT} = -12 + V_{CE\text{ sat}} + R_c I_{GT}$$

$$R_c I_{GT} = 12 + V_{GT} - V_{CE\text{ sat}} = 12 - 2 - 0,3 = 9,7V$$

$$\text{donc } R_c = \frac{9,7}{50} \cdot 10 = 194\text{ohm nous prenons } R_c = 200\text{ ohm}$$

Calcul de  $R_b$ 

La tension de commande  $V_c = 5V$

$$V_{BE\text{ sat}} = 0,7V$$

le gain  $B = 100$

$$V_d = 0,6V \text{ (de la diode)}$$

$$V_c = -12 + V_{BE\text{ sat}} + R_B I_B^+ + V_d$$

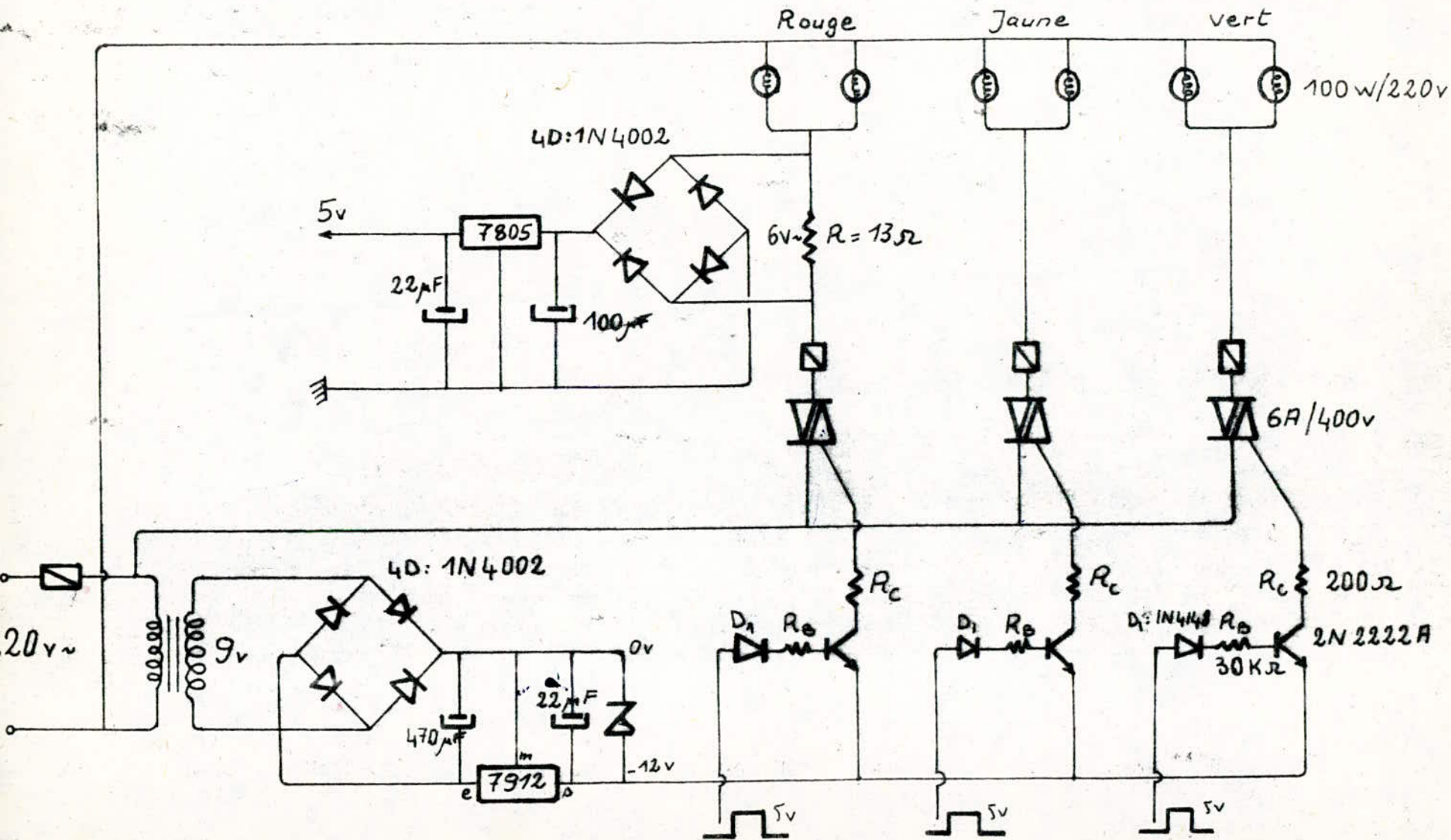
$$R_B I_B^+ = 12 - V_{BE\text{ sat}} + V_c - V_d = 15,7V$$

$$R_B = \frac{15,7}{I_{GT}} \cdot B = 31,4 \text{ K Ohm nous prenons } R_B = 30 \text{ KOhm}$$

Lampes utilisées : 100W/220V - Triac : 6A/400V



(fig 10) Circuit de commande des feux



2.4.4

DETECTION DES PANNES DES LAMPES

Seule la panne des feux rouges des vehicules nous interesse car il suffit que le conducteur sache quand est ce que qu'il doit s'arreter en arrivant à proximité d'un carrefour.

En cas d'une panne d'un feu vert ,si le rouge est intact il y aurait pas tant de problemes que pour la panne du rouge car en trouvant le rouge il sait qu'il doit s'arreter ,si ce dernier est éteint ça équivaut au vert ; Mais dans le cas contraire ça risque de créer de vraies catastrophes .

C'est pour cette raison que nous nous sommes seulement intéresser à la panne du feu rouge des vehicules .

Nous avons essayer de concevoir notre système tel qu'à chaque panne d'un feu rouge (lampe grillée) il déclenche le jaune clignotant sur toutes les voies .

Pour détecter la panne nous prélevons la tension aux bornes de R en série avec la lampe(fig 10); Si la lampe correspondante est grillée ,il n'y aurait pas de tension aux bornes de R mais si la lampe est intacte nous avons une ddp à ses bornes

Le système de détection de panne consiste à envoyer un signal de +5V (tension régulée) au P.I.A chaque fois qu'il y a une panne.

Pour cela la tension la tension prélevée aux bornes de R doit être proche de 5V pour être par la suite régulée à +5V

Les signaux de détection de pannes de toutes les lampes concerné sont reliés par une porte NAND pour donner à la sortie de cette dernière un signal de haut niveau(5V),à chaque fois qu'il y a une panne dans les lampes, au P.I.A .



On voit que d'une heure à l'autre, pour une même politique, les résultats peuvent être très diversifiés. Cela tient à ce que pour certains carrefours on est proche de la saturation et que certaines files d'attente peuvent croître assez largement, ainsi d'ailleurs qu'on peut l'observer sur le terrain, lorsque pour une période de durée limitée la génération dépasse sa valeur moyenne. Alors, par programmation, nous pouvons changer de stratégie plusieurs fois par jour car aux horaires de pointe la circulation est plus intense et demande une stratégie qui sera différente de celle des autres heures, alors qu'à la tombée de la nuit les feux tricolores seront inutiles, on utilise donc le jaune clignotant. Tout ce changement de stratégies est facilité par l'utilisation d'une horloge temps réel dont nous relierons l'afficheur des heures (et celui des jours si nous voulons changer les stratégies selon les jours de semaine) à notre P.I.A et à l'aide du programme nous indiquons au micro les heures (ou les jours) du changement de stratégies.

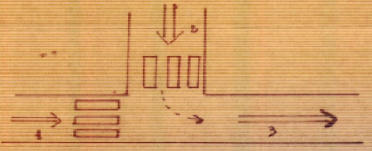
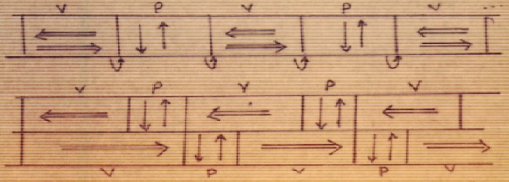
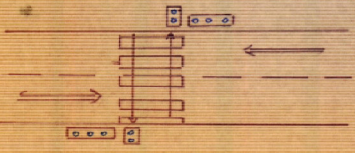
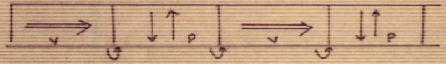
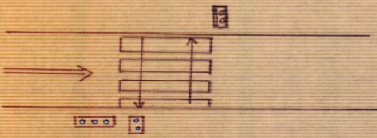
#### SYNOPTIQUE GENERALE DE LA MONTRE UTILISEE (fig II)

L'oscillateur piloté par quartz délivre une fréquence parfaitement stable qui est divisée un certain <sup>nombre</sup> de fois pour produire un signal à un hertz, c'est à dire un système ayant une période de une seconde. Ce signal est appliqué à un compteur de dix qui indiquera les unités de secondes. Les sorties de ce compteur aboutissent sur l'afficheur des secondes mais aussi sur un deuxième compteur par six celui là, qui va compter les dizaines de secondes. Ce dernier commande lui aussi un afficheur qui est celui des dizaines de secondes et attaque à son tour un troisième compteur par 10 qui sera celui des minutes et ainsi de suite jusqu'aux dizaines d'heures. Ce dernier est un compteur par douze ou par vingt quatre selon la façon dont il a été réalisé.

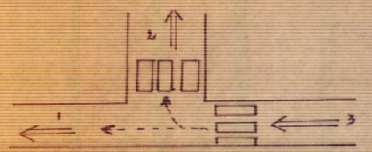


Carrefours

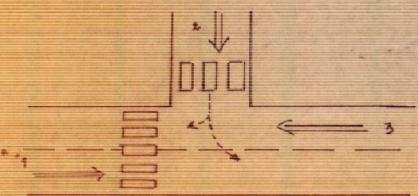
strategies



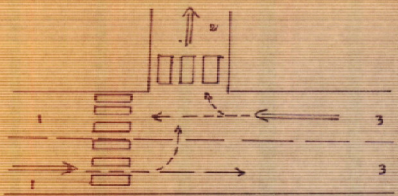
$V_1, V_2$	$P_1, P_2$	$V_1, V_2$	$P_1, P_2$	...
$V_1$	$P_1$	$V_1$	$P_1$	
$P_2$	$V_2$	$P_2$	$V_2$	



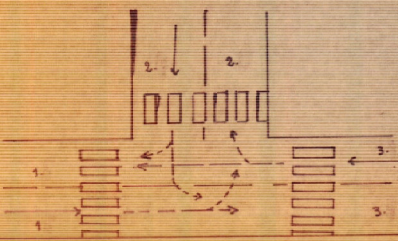
$V_2, V_3$	$P_1, P_3$	$V_2, V_3$	$P_1, P_3$
$V_2$			
$P_3$			



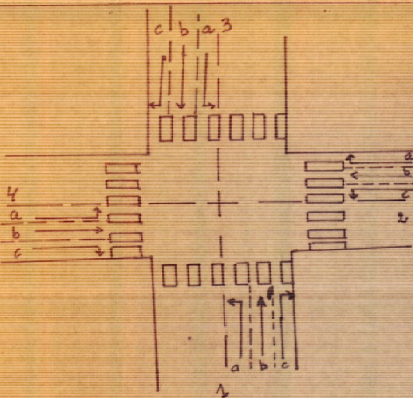
$V_1, V_3$	$V_2, V_1$	$P_1, P_2$	$V_1, V_3$	$V_2, V_1$
$P_2$			$P_2$	



$V_{31}$	$V_{12}$	$P, P_2$	$V_{31}$	$V_{12}$
$V_{13}$	$V_{32}$		$V_{13}$	$V_{32}$



$V_{13}, V_{12}$ $V_{32}, V_{31}$	$V_{31}, V_{32}$ $V_{21}, V_{13}$	$V_{23}, V_{21}$ $V_{13}, V_{32}$	$V_{12}, V_{13}$ $V_{32}, V_{21}$
$P, P_2, P_3$			$P_1, P_2, P_3$



$V_{a1}, V_{a3}$	$V_{b1}, V_{b3}$ $V_{c1}, V_{c3}$	$V_{c1}, V_{c3}$ $V_{a1}, V_{a3}$	$V_{b2}, V_{b4}$ $V_{c2}, V_{c4}$	$V_{c2}, V_{c4}$ $V_{a2}, V_{a4}$	$V_{b1}, V_{b3}$
$P_1, P_4$			$P_1, P_3$		$P_2, P_4$

Exemples de carrefours  
et strategies correspondantes



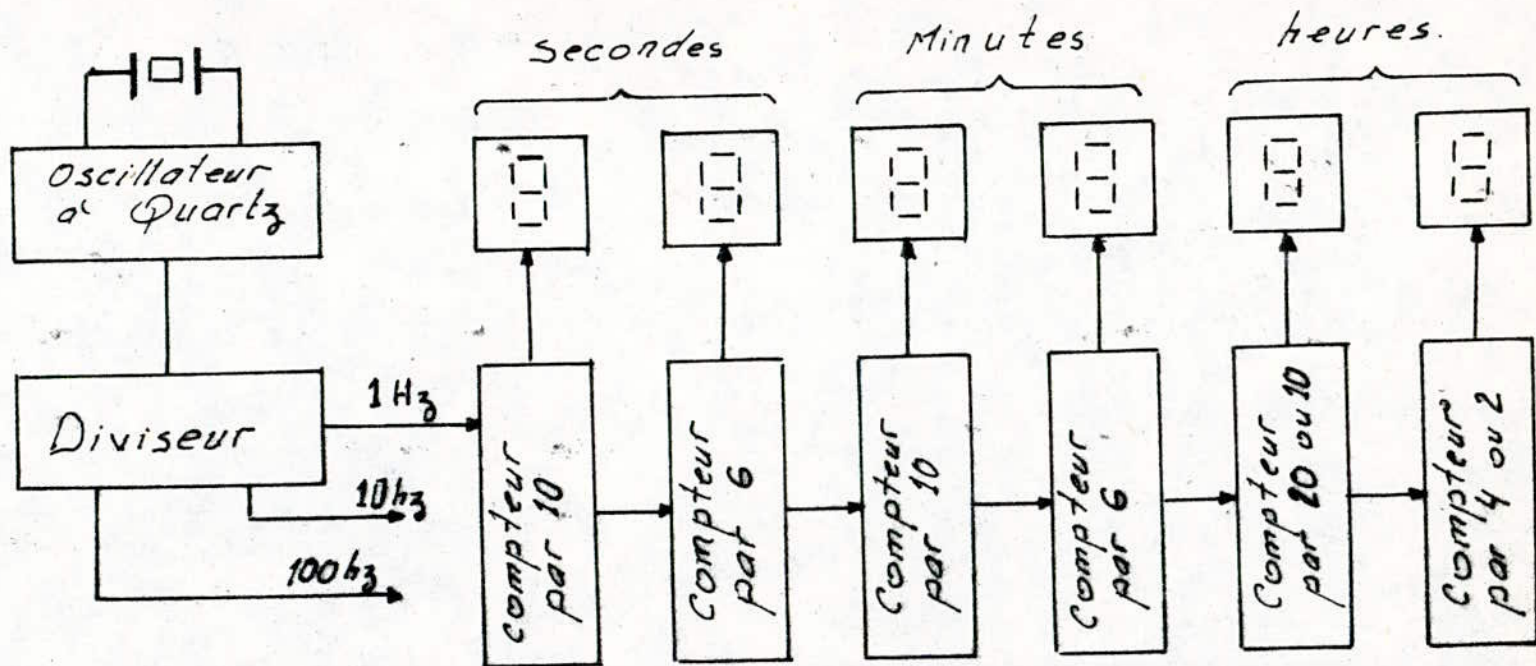


fig 11: Synoptique Générale d'une montre digitale  
- pilotée par Quartz -

Chapitre III : Partie dynamique  
(Logiciel et programmation)



Notons que les performances de notre système dépendent essentiellement des programmes disponibles, programmes où seront consignés une foule de fonctions internes et externes.

Mais le développement de tels programmes presuppose un vaste savoir faire en informatique, ce qui aurait procuré une vaste bibliothèque de programmes élaborés qui seront incorporés au système d'exploitation. Logiciel et matériel seront alors parfaitement harmonisés, ce qui permet d'adapter l'éventail des ressources de notre système aux problèmes spécifiques que pose la régulation du trafic d'une ville. Tous les programmes doivent être présentés en langage assembleur. L'ensemble des instructions de service et les éditions sont, comme tout le monde le sait, sous forme d'abréviations en clair ou d'expressions numériques. Une foule de tests formels (sur différents états et différentes données) garantit une grande souplesse d'exploitation et une parfaite fiabilité. L'ensemble de ces programmes permettent d'exécuter un vaste éventail de tâches se rapportant à la régulation du trafic, à la gestion et au traitement des données de la circulation. Le produit programme devrait comprendre:

- le programme de gestion ou d'exploitation
- Les programmes de régulation

Le système d'exploitation : sert au lancement, à la gestion, à la commande du déroulement et à la coordination des modules de programmes; Ce système assure en outre, l'ordonnancement et le stockage en mémoire des données échangées entre l'unité centrale et les périphériques ou l'élément de signalisation.

Les programmes de régulation: regroupent toutes les fonctions inhérentes à la régulation de la circulation dans un réseau urbain comme:

L'exécution des fonctions de commutation selon l'heure, le jour et le calendrier. Les heures, le jour et les activités à exécuter sont, par le biais d'instructions, indiquées en clair par des

abreviations en cours d'exploitation.

L'exécution, le plus brièvement possible, le changement des plans de feux initialisés par des instructions de service. Ce programme réalise automatiquement une transition durant laquelle les groupes de feux qui seraient au vert dans l'ancien plan de feux et au rouge dans le nouveau, sont effectivement bloqués jusqu'à écoulement du temps minimum de vert. Après écoulement du temps minimum de securite, les groupes de feux, au rouge dans le cadre de l'ancien plan de feux et au vert dans celui du nouveau, sont finalement au vert. Pour ces groupes de feux la fin du temps minimum de vert correspond avec l'achèvement du changement du plan de feux, le nouveau plan peut alors être lancé avec une coordination parfaite. En general, cette transition s'effectue simultanément dans tous les carrefours concernés.

Interrogation des detecteurs et obtention des mesures concernant

- L'occupation (durée de passage des vehicules sur les boucles
- Les comptages (volume du trafic)

La selection du plan de feux en fonction de la circulation (exploitation en fonction de l'heure ou du trafic)

Il y a aussi le programme qui sert à optimiser le décalage temporel des carrefours coordonnés. Cette optimisation est déterminée par la circulation tout en agissant directement sur elle. Comme criteres d'optimisation qu'on a déterminé auparavant et qu'on doit en tenir compte dans cette programmation il y a:

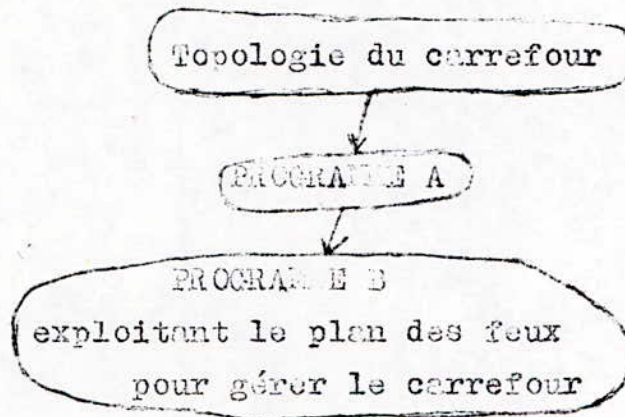
- Les attentes
- Le nombre d'arrets.



3.1

PARTIE LOGICIELLE

Notre systeme de controle est composé d'une partie matérielle qui a été décrite auparavant, constituée d'un microprocesseur (6800) ,des mémoires de programmation et des interfaces;et d'une partie logicielle servant à assurer le sequencement des feux. Cette derniere est la plus importante,elle conditionne la fiabilité du systeme et permet d'adapter n'importe quelle strategie de regulation de changement des feux.



Le programme A est prévu pour pouvoir utiliser le systeme de commande des feux pour n'importe quel type de carrefour. Il devra alors à partir de la topologie du carrefour (une serie de points representant des voies) trouver les voies qui sont en conflit et generer le plan de feux adéquat.

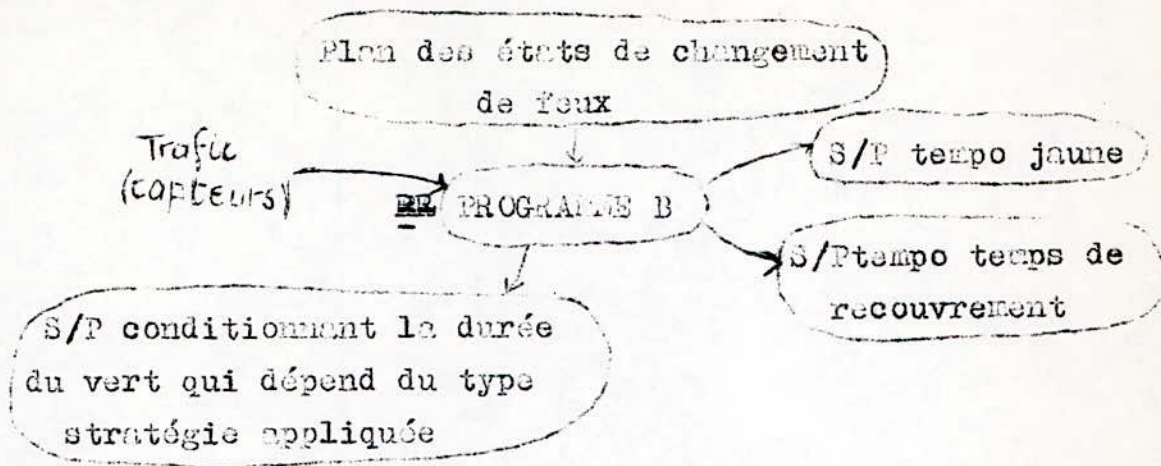
Le programme B lui se charge à partir de ce dernier de realiser le sequencement des feux conditionné par le trafic en y apportant les temporisations necessaires.

TEMPORISATION

Pour effectuer une temporisation sur toute machine programmable il suffit de boucler un certain nombre de fois sur les instructions mais auparavant il faudra connaître la durée de cycle d'horloge et le nombre de cycles que mettent certaines instructions pour s'exécuter. L'horloge du micro 6800 a un cycle de 1 us

La temporisation du temps de recouvrement dépend du type de carrefour à réguler car ce temps doit permettre l'évacuation complete de la chaussée du carrefour avant le basculement des

des feux, il s'ensuit que cette durée définit le temps nécessaire à un véhicule pour franchir le carrefour.



### Sous programme conditionnant la durée du vert

La durée du feu vert est conditionnée suivant certains paramètres qui sont:

-Le temps de réaction  $t_r$  du conducteur lors du démarrage c'est pourquoi on temporise un vert min avant de faire appel au test du trafic. Et en cas où la stratégie se base sur le test de l'intervalle de temps entr 2 véhicules, ce dernier est choisi au démarrage un peu plus élevé, vu qu'on doit tenir compte du temps de réaction du conducteur.

-Temps de franchissement du carrefour

-La complexité de l'itinéraire de la voie dont le feu est au vert. Ce qui conduit à utiliser un vert maximum indifféremment d'une voie à une autre suivant le type de l'itinéraire correspondant.

-Le nombre de véhicules présents en attente du vert.



## PROGRAMMATION

### 1) Procédés de programmation :

La programmation est particulièrement simplifiée par un clavier qu'on pourrait coupler à un interface permettant de loger des données spécifiques aux carrefours à régler. Pour les exemples de programmation qui suivent nous utiliserons la microregulation donc les données nécessaires sont fixes et déterminées par observations, nous n'aurons pas besoin du clavier puisque le programme de gestion sera logé directement en EPROM.

Pour programmer une stratégie nous devons procéder comme suit:

- 1°) Etablir le diagramme des feux souhaités
- 2°) Decouper le diagramme obtenu en sequences (une sequence est la durée pendant laquelle un ou plusieurs courants de circulation sont admis simultanément.) Elle est delimitée par un ou plusieurs debuts de vert.

#### 3°) Parametres des sequences:

- Temps maximum de la sequence
- Temps minimum de la sequence
- Intervalle critique
- Temps de jaune et de rouge integral par sequence

#### 4°) Parametres des lignes:

Etat des lignes: vehicules, pietons, fleches etc...

#### 5°) Affectation des detecteurs

6°) Enchainement des sequences: à cet emplacement mémoire on définit l'enchainement des sequences en durée fixe.

7°) Programmation des fonctions particulieres (applicables aux sequences).

-Escamotage: une sequence peut être escamotée s'il y a pas d'appel sur les detecteurs commandant la sequence.

-Report: la sequence spécifiée comme étant sequence de report peut être prolongée au delà de son temps maximum si la sequence dite de reference a été escamotée

-Antisaturation: une sequence peut être forcée au vert à la fin de son temps maximum par action d'un detecteur spécifique

-Blocage: augmentation du rouge integral entre deux sequences par action d'un detecteur spécifique programmable

- 8°) Coordination: outre l'entrée impulsienne, le coordinateur possede des entrées de coordination programmables
- 9°) Programmation du mode de fonctionnement: plusieurs stratégies peuvent etre mises en oeuvre: automatique à temps; automatique prioritaire, automatique dernier appel, automatique alterné et rouge integral.
- 10°) Programmation des heures de changement de programme; plusieurs heures de changement de programme peuvent etre programmées. A chaque programme un cycle différent peut etre affecté.

#### MODIFICATION DE CERTAINS PARAMETRES DU DIAGRAMME

Le diagramme étant organisé en séquences, il est particulièrement facile de modifier le temps d'une séquence et par là, de modifier le temps d'une ligne sans avoir à retoucher tout le programme.

#### FONCTIONNEMENT

Après avoir programmer notre système, celui-ci est mis en fonctionnement après une phase de démarrage dite aussi phase d'initialisation. Après cette phase et suivant la stratégie choisie la séquence correspondant à cette dernière est générée. Le contrôleur tout en tenant compte des séquences durée fixe, cherche la séquence optimum en fonction des appels enregistrés sur les capteurs; C'est à dire cherche la voie la plus prioritaire



Le module d'interface implanté sur le data bus, jouant le rôle d'intermédiaire direct le microprocesseur et l'extérieur est appelé interface d'entrée/sortie.

Cette interface comporte essentiellement :

- 4 buffers d'interface où sont logés provisoirement les données transférées.

- Une porte d'entrée/sortie, donnant accès à ce buffer et validée par les commandes du micro (lecture ou écriture)

Cette interface comporte notamment une logique de contrôle recevant des commandes aussi bien du micro que des périphériques. Le microprocesseur dispose d'instructions spéciales pour contrôler cette interface par programme (d'où le terme d'interface programmable).

#### PROGRAMME DE GESTION DE PERIPHERIQUE

Une fois l'interface matériellement réalisée entre le micro et les périphériques, il reste à fournir au micro un programme lui indiquant comment assurer les échanges avec ces périphériques.

Ce programme doit contrôler les fonctions suivantes :

- Lors d'une opération d'entrée, il s'agit pour le micro d'interpréter correctement les signaux générés.

- Lors d'une opération de sortie le micro doit assurer la mise en forme convenable de l'information suivant la nature du périphérique.

Le programme de gestion est mémorisé sous forme d'un sous programme auquel se branche le micro au moment d'une entrée/sortie.

Les opérations de contrôle effectuées par le micro sont :

- Lire le contenu du buffer d'interface (instruction LDAA)

- Ecrire une valeur dans ce buffer (instruction STAA)

- Programmer chaque bit de ce buffer soit comme bit d'entrée soit comme bit de sortie.

3.2.3 CRITERE DE FONCTIONNEMENT DES FEUX

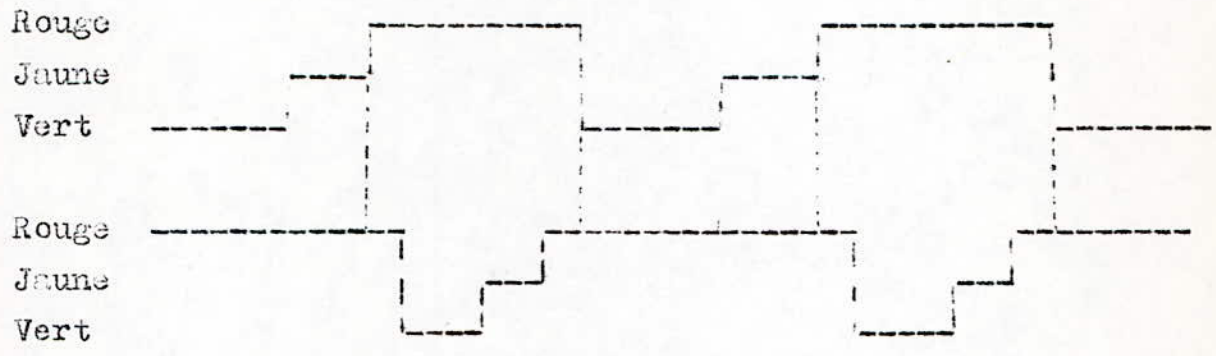
Dans un croisement ,on ne peut avoir simultanement ni deux feux verts,ni deux feux jaunes(sauf dans le cas d fonctionnement de nuit où les deux feux jaunes fonctionnent simultanement en clignotant).Mais on peut avoir simultanement deux feux rouges pour permettre aux vehicules engages sur une voie d'evacuer l'intersection avant d'ouvrir l'autre voie.

La durée pendant laquelle les feux seront rouges sur les deux voies sera réglables et fonction du type du carrefour.

Nous pouvons ajuster à notre guise la durée du feu jaune suivant le reflexe des conducteurs et de celui des pietons.

La durée du feu rouge sera directement fonction de celle du feu vert à laquelle on ajoutera celle du "recouvrement"

DIAGRAMME ILLUSTRANT LES DIFFERENTES PHASES



3.2.4 EXEMPLES DE PROGRAMMATION DE GESTION D'UN CARREFOUR

Exemple I La stratégie appliquée consiste à temporiser la durée du feu vert suivant le nombre des vehicules attendant le vert, l'enchainement des opérations se fait de la manière suivante:

- Saisie des variables identifiant le trafic pour tout capteur.
- Déterminer la voie la plus prioritaire.
- Déterminer la valeur de temporisation du vert:X
- Envoyer la combinaison complète de la table des états des feux.
- Temporiser suivant la valeur X.
- Envoyer la combinaison suivante.



- Temporiser suivant la valeur du jaune
- Envoyer la combinaison suivante
- Temporiser suivant la valeur du rouge de recouvrement
- Retour à la première étape

Pour déterminer la voie la plus prioritaire nous nous sommes à déterminer le temps d'attente du vert du premier véhicule de la file.

Si la table de changement de feux est fixée; seules les combinaisons où existe la phase verte dépendent des états des capteurs les autres sont déterminées par temporisations bien définies. Supposons que nous soyons dans le cas général où après chaque combinaison du rouge integral nous avons  $n$  voies dont le feu est au vert, supposons que  $X_1, X_2, \dots, X_n$  sont les durées des temps d'évacuation des voies  $1, 2, \dots, n$  respectivement

ALGORITHME (PROGRAMME B)

A  $J=1$

B Lire mot d'indice  $J$

Envoyer ce mot

Appel s/p qui calcule  $X_1, \dots, X_n$  et déterminer le  $\max X_j$

Appel s/p tempo vert  $X_j$

$J=J+1$

Envoyer mot d'indice  $J$

Appel s/p tempo jaune

$J=J+1$

Envoyer mot d'indice  $J$

Appel s/p tempo rouge de recouvrement

$J=J+1$

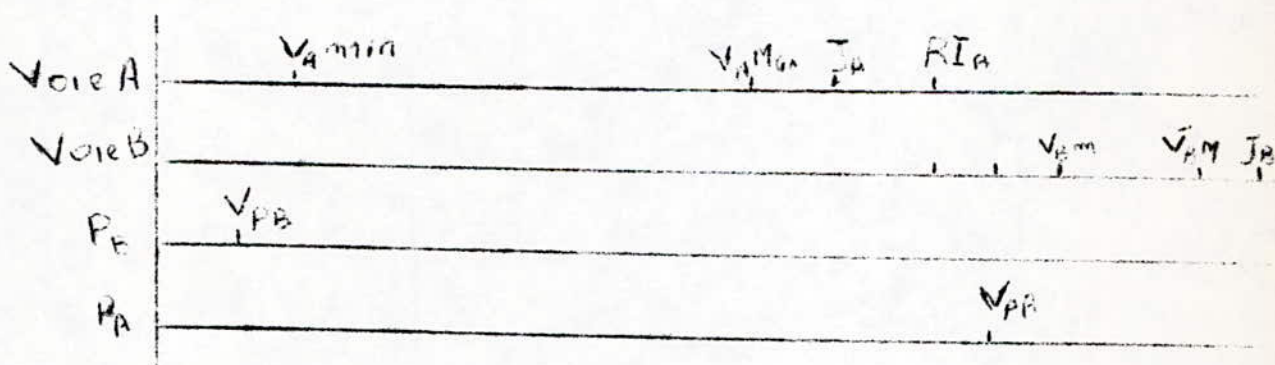
Si  $J$  supérieur à la dimension de la table alors aller à A sinon aller en B

REMARQUE SE RAPPORTANT A L'EXEMPLE I

Vu que la détermination du nombre de vehicules en attente du vert nécessite assez de capteurs pour une circulation intense ceci aurait augmenter le cout d'une façon exorbitante, alors nous avons décider d'utiliser d'autres stratégies dont le cout est moins élevé c'est à dire dans lesquelles nous utilisons un seul capteur par voie et qui peuvent etre utilisées pour n'importe quel type de carrefour et n'importe quel état du trafic à réguler.

Si nous avons à notre disposition un ordinateur de grande taille nous aurions utilisé alors deux capteurs par voie, l'un juste avant le passage des piétons, l'autre loin en avant tel que la distance entre celui ci et le premier cité pourra contenir le nombre de vehicules que l'on pourrait faire passer pendant la durée du vert maximum proposée. Alors, à l'aide de ces deux capteurs nous pouvons compter le nombre de voitures se trouvant entre eux en faisant la différence entre le nombre de vehicules entrant en  $d_2$  et ceux sortant de  $d_1$ .

b) EXEMPLE 2 PROGRAMMATION D'UNE STRATEGIE



Soit à programmer ce diagramme

Les feux piétons fonctionnent en opposition des feux vehicules.  
Les valeurs que nous devons programmer sont les suivantes:

- $V_{Amin} = 10s$  vert min sur voie A -  $V_{Bmin} = 10s$  sur voie B
- $V_{Amax} = 70s$  vert max sur voie A -  $V_{Bmax} = 70s$  sur voie B
- $I_{CA} = 4s$  intervalle critique sur A -  $I_{CB} = 4s$  sur voie B
- $J_A = 3s$  jaune sur voie A -  $J_B = 3s$  jaune sur voie B
- $R_{IA} = 4$  rouge integral sur A -  $R_{IB} = 4s$  sur voie B



Après avoir introduit le programme nécessaire dans le système en ayant choisi le mode de fonctionnement désiré (adaptatif) (toujours par programme); les détecteurs entrent alors en jeu. Après une période d'initialisation composée de :

- 5s de jaune clignotant voies véhicules (feux piétons éteints)
- 5s de jaune fixe voies véhicules (feux piétons éteints)
- 5s de rouge fixe voies véhicules (rouge fixe voies piétons)

Le vert sera alors donné à une des deux voies (on choisit la voie qui est en général prioritaire, voie A pour notre cas) ainsi qu'aux piétons de la voie opposée (PB)

Dès qu'un appel sera enregistré sur la voie A, le système décompte automatiquement le temps de vert sur la voie A plus ou moins rapidement selon l'arrivée des véhicules sur cette voie. Le décompte du temps du vert de A durera jusqu'à l'obtention du vert maximum. Dès que cette valeur est atteinte le décompte de la voie A est arrêté. La voie A passera ensuite jaune puis au rouge intégral, le vert sera alors donné à la voie B.

Le même principe de décompte de temps est appliqué à la voie B. Toutefois en absence totale de détections sur les deux voies le vert persistera sur chacune des voies durant  $V_{min}$  puis se déplacera sur l'autre voie et ainsi de suite.

En cas où la voie B est plus prioritaire que A et si un appel se présente sur la voie B alors que la voie A est en cours de décompte, la voie B passera au vert au bout d'un temps  $t$  tel que  $t = V_{Amin} - t_{ap}$

où  $t_{ap}$  = temps de vert de la voie A écoulé jusqu'à l'appel de B ;

Au cas où on a pas de voies prioritaires, le décompte du temps se fait de la même façon et la durée de vert de chaque voie varie entre un  $le_{min}$  et le  $max$  en fonction de l'intervalle véhiculaire comparé en permanence avec l'intervalle critique ou en fonction de la présence ou non d'un véhicule au cours du temps critique  $I_c$  (temps de prolongement de vert permettant au véhicule en présence franchir le carrefour).

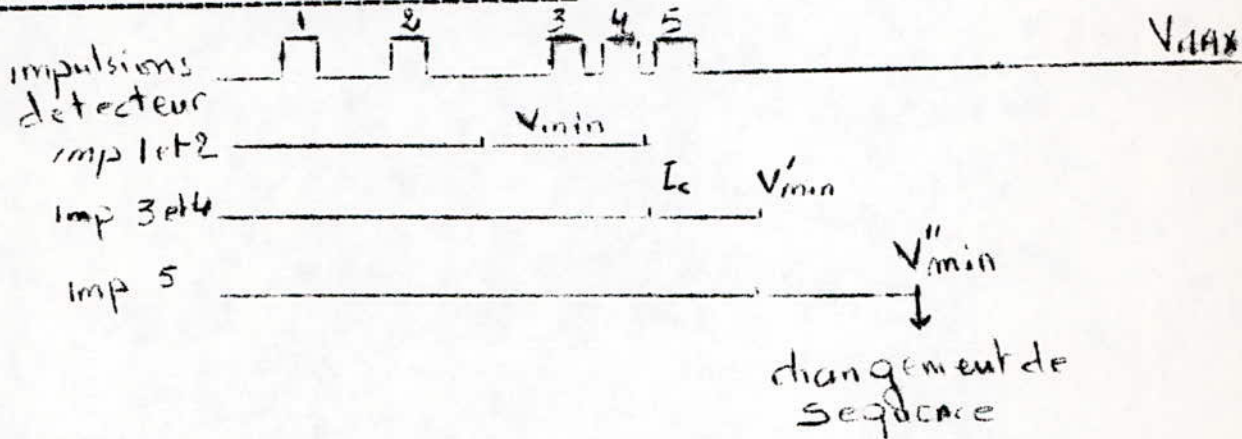
STRATEGIE DU DECOMPTE DES TEMPS UTILISES

Au début d'une séquence, le système donne aux véhicules un temps de vert minimum ( $V_{min}$ ); Si pendant ce temps  $min$ , les détecteurs affectés à cette séquence n'ont enregistré aucun passage de véhicules, le vert est alors coupé et transféré sur une autre séquence. Si au contraire le détecteur est informé du passage d'un véhicule la durée du vert est prolongée d'un temps  $I_c$  à condition que la présence du véhicule intervienne à un temps  $t$  tel que  $t$  supérieur ou égal à  $V_{min} - I_c$ .

$I_c$  est comme nous l'avons signalé le temps permettant à un véhicule de franchir le carrefour.

Le temps  $I_c$  est rajouté autant de fois qu'il y a présence de véhicules dans l'intervalle de temps  $t$ , sans toutefois dépasser le temps  $V_{max}$ .

EXEMPLE DE DECOMPTE DE TEMPS I



Impulsion 1: Cette impulsion intervenant avant  $t$  ( $t = V_{min} - I_c$ ) le système ne rajoute pas de  $I_c$ .

Impulsion 2: même chose que pour la première

Impulsion 3: Cette impulsion prolongera le vert  $min$  d'un temps  $I_c$  le nouveau vert  $min$  est alors  $V'_{min} = V_{min} + I_c$

Impulsion 4: même chose que la précédente

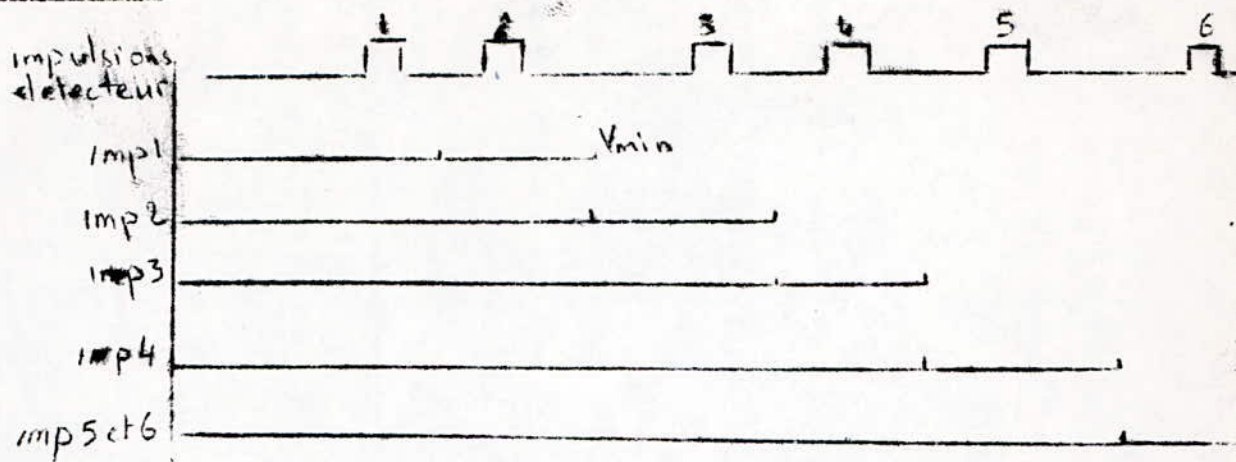
Impulsion 5: Cette impulsion prolongera le vert  $min$  d'un temps  $I_c$  le nouveau vert  $min$  est égal à  $V''_{min} = V'_{min} + I_c$

Si aucune impulsion n'intervient après la cinquième impulsion le changement de séquence aura lieu à l'instant  $t = V'_{min} + I_c$



EXEMPLE 2

-47-



Impulsion 1 : Cette impulsion intervenant avant  $t = V_{min} - I_c$ , le système ne rajoutera pas d'intervalle de temps de vert .

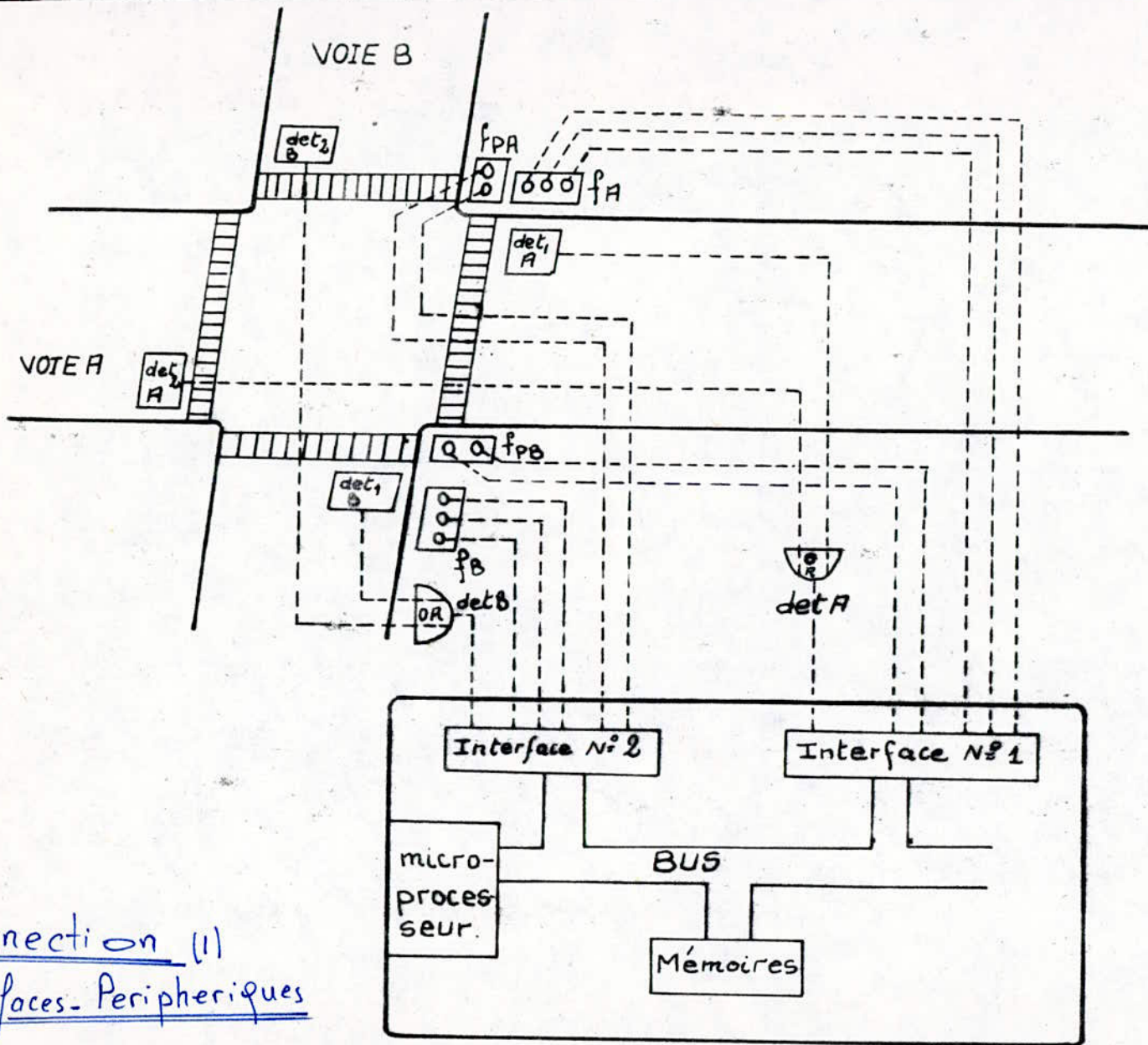
Impulsion 2 : Le nouveau temps de vert sera  $V_{min} = V_{min} + I_c$

Impulsion 3 : Le nouveau temps de vert sera  $V_{2min} = V_{min} + I_c$

Impulsion 4 : Le nouveau temps de vert sera  $V_{3min} = V_{2min} + I_c$

Impulsion 5 : Le nouveau temps de vert sera  $V_{4min} = V_{3min} + I_c$

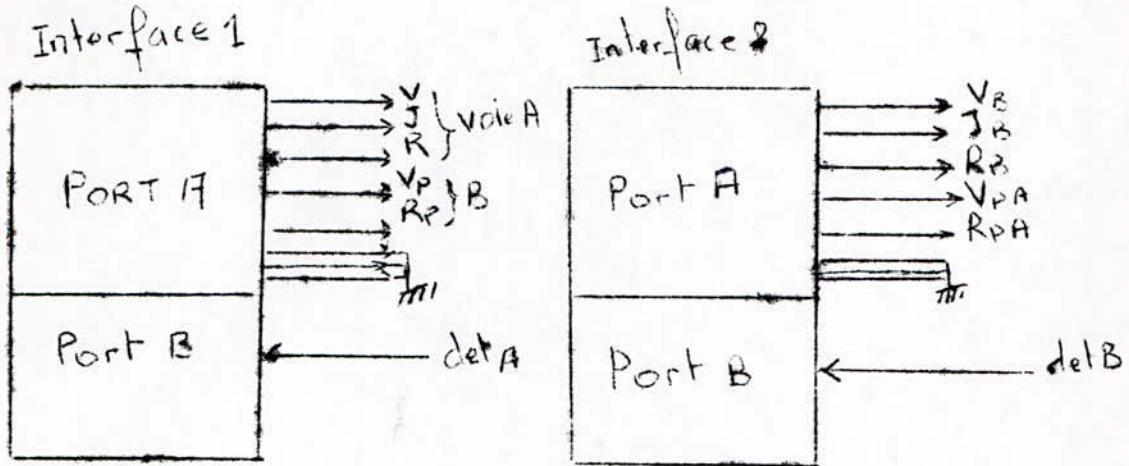
Impulsion 6 : Cette impulsion ne prolongera pas le temps de vert car le vert maximum est atteint.



Connection (1)  
Interfaces-Peripheriques



Le microprocesseur commande directement chaque feu par l'un des interfaces I ou 2 de la voie A ou la voie B . Un bit à zero correspond à un feu éteint et un bit à UN correspond à un feu allumé .



Les valeurs à charger dans les buffers d'interfaces seront donc:

- # 1 pour commander un feu vert de chaque voie
- # 2 pour commander un feu jaune de chaque voie
- # 4 pour commander un feu rouge de chaque voie
- # 8 pour commander un feu vert piéton de chaque voie
- /# 16 pour commander un feu rouge piéton de chaque voie

(S : signifie que nous programmons en hexadécimal)

Si le microprocesseur fonctionne en logique négative, il faudra programmer les compléments à deux de ces nombres .

Avant de passer à la programmation proprement dite, apportons une précision sur les ports A et B de chaque interface qui fonctionne comme suit : Chaque port peut fonctionner soit en entrée (les bits de chaque registre de données du port B à zero) soit en sortie comme les bits (0,1,2,3 et 4) des ports A

Les voies d'entrée et de sortie sont indépendantes .

Une instruction d'écriture WRR n'affecte que les bits mis à UN (programmés en sortie)

Une instruction de lecture RDR ne prend en compte que les bits mis à ZERO (programmés en entrée )

- 1) Les états des feux sont commandés par des instructions d'écriture sur les interfaces appropriés
- 2) le test du détecteur est effectué par une lecture du buffer N° 1 et N° 2 suivi d'un test de l'accumulateur, tant que celui ci est à 1 (présence de véhicules) le micro effectue une boucle d'attente pour que le véhicule en présence puisse avoir le temps pour franchir le carrefour .

INITIALISATION DU P.I.A 1

LDAA # 0  
 STAA /\$ 8001  
 STAA /\$ 8000  
 STAA /\$ 8003  
 STAA /\$ 8002  
 LDAA # 4  
 STAA /\$ 8001  
 STAA /\$ 8003

Pour le P.I.A 2, on procède de la même façon en remplaçant /\$ 8000 par /\$ 8400, /\$ 8001 par /\$ 8401, /\$ 8002 par /\$ 8402 et /\$ 8003 par /\$ 8403

Nous affectons les feux aux interfaces de la manière suivante:

- V<sub>A</sub> au bit B<sub>0</sub> de la mémoire /\$ 8000
- J<sub>A</sub> AU BIT B<sub>1</sub> de la mémoire /\$ 8000
- R<sub>A</sub> au bit B<sub>2</sub> de la mémoire /\$ 8000
- V<sub>PB</sub> au bit B<sub>3</sub> de la mémoire /\$ 8000
- R<sub>PB</sub> au bit B<sub>4</sub> de la mémoire /\$ 8000
- Détecteur de la voie A au bit B<sub>0</sub> de la mémoire /\$ 8002
- V<sub>B</sub> au bit B<sub>0</sub> de la mémoire /\$ 8400
- J<sub>B</sub> " B<sub>1</sub> " " "
- R<sub>B</sub> " B<sub>2</sub> " " "
- V<sub>PA</sub> " B<sub>3</sub> " " "
- R<sub>PA</sub> " B<sub>4</sub> " " "
- Détecteur de la voie B au bit B<sub>0</sub> de la mémoire /\$ 8402

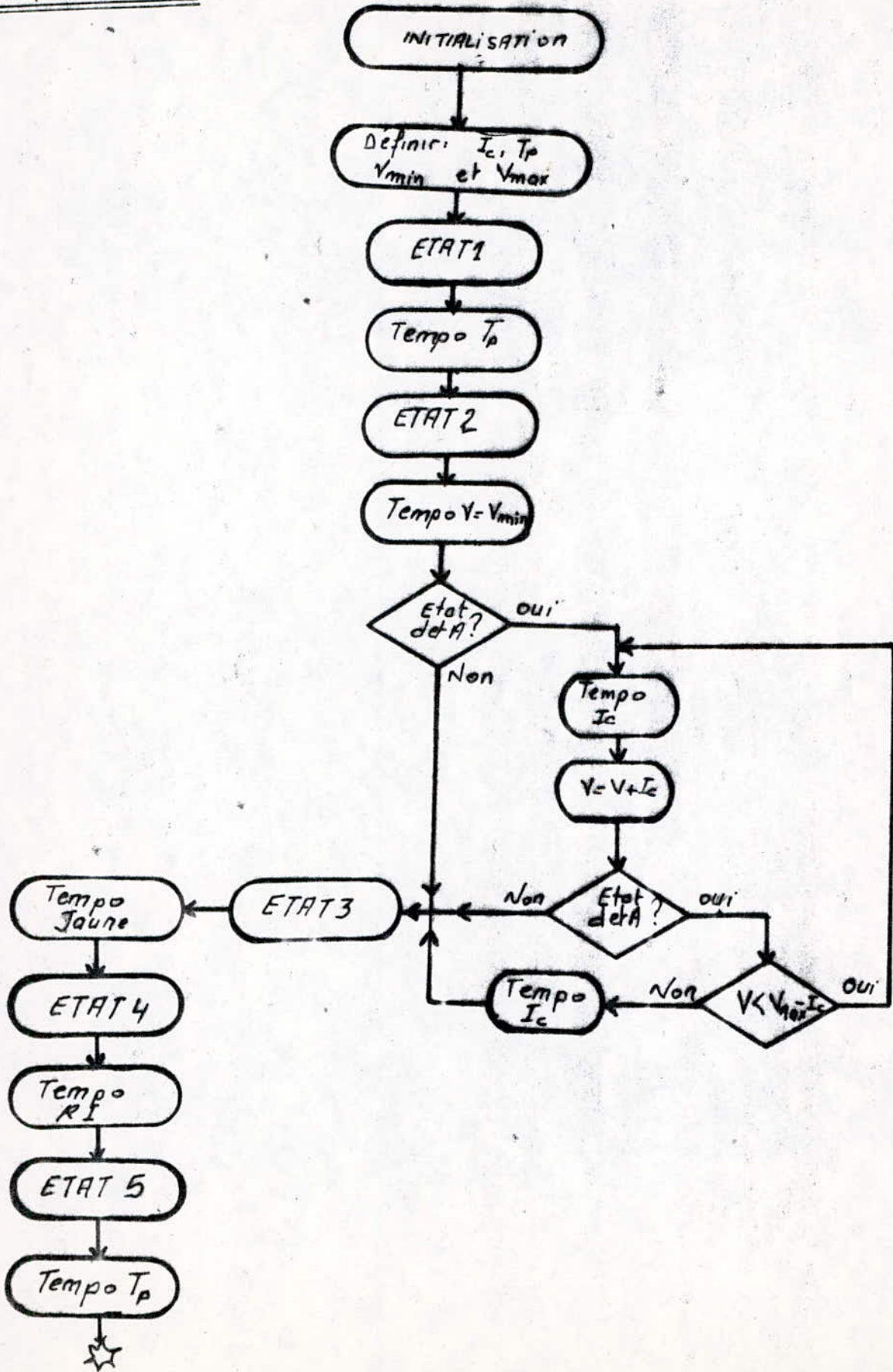


Etats	Voie A	Voie B	Pietons P <sub>B</sub>	Pietons P <sub>A</sub>	Temporisa- tion
1	V	R	V	R	T <sub>p</sub>
2	V	R	R	R	
3	J	R	R	R	Jaune
4	R	R	R	R	Rouge int
5	R	V	R	V	T <sub>p</sub>
6	R	V	R	R	
7	R	J	R	R	Jaune
8	R	R	R	R	RI

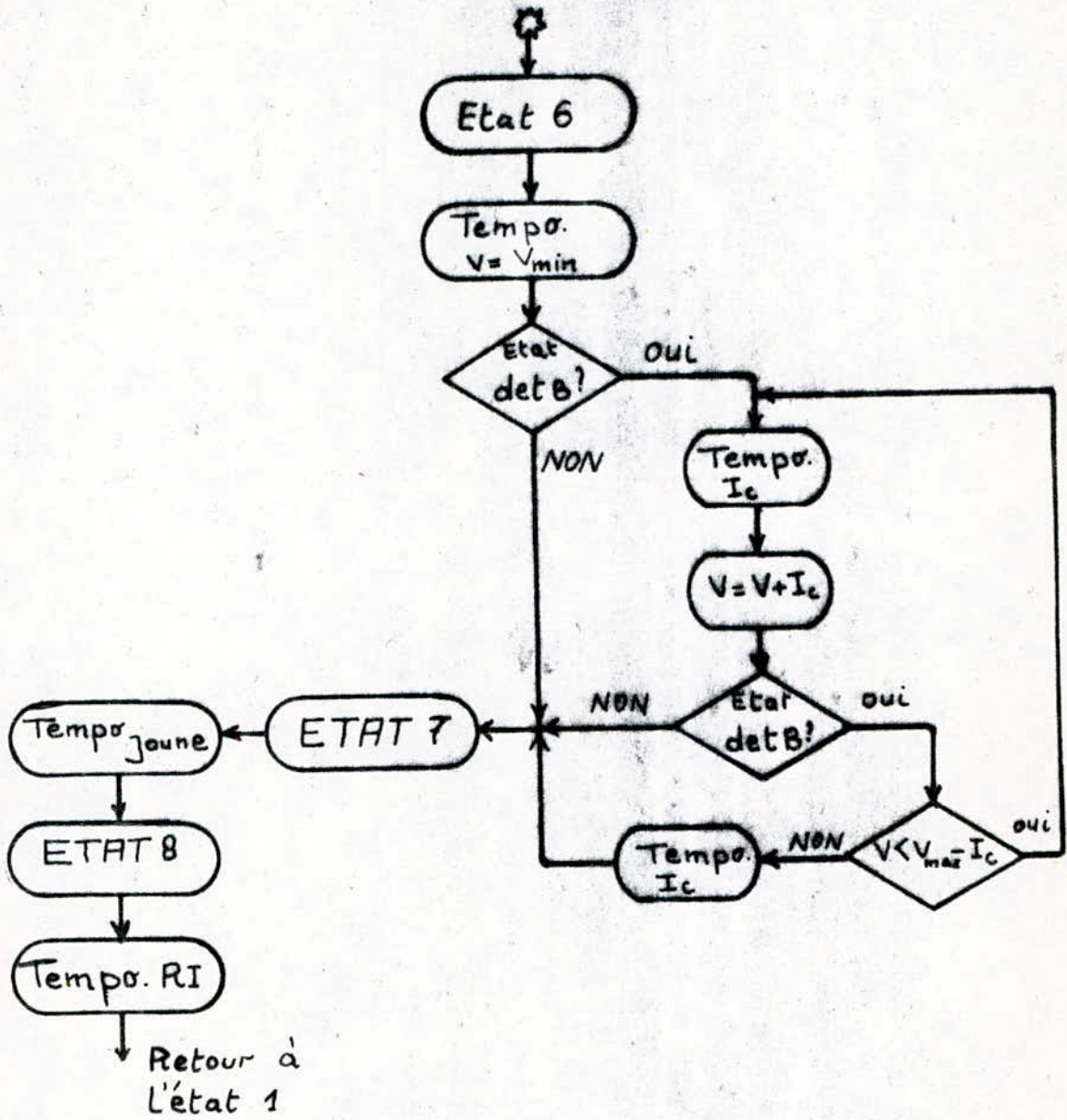
↳ Retour à l'état 1

Tableau des transitions des états

ORGANIGRAMME







```

LDAA ## val de Ic
STAA § 250
LDAA ## 20
STAA § 8400
LP9 LDAA ## 9
STAA § 8000
JSR LP1
LDAA ## I7
STAA § 8000
JSR LP2
TEST § 8002
BEQ LP3
LP4 JSR LP5
LDAB ## val de Vmin
ADDB § 250
TST § 8002
BEQ LP3
LDAA ## val de (Vmax-Ic)
CBA
BHI LP4
JSR LP5
LP3 LDAA ## I8
STAA § 8000
JSR LP6
LDAA ## 20
STAA § 8000
JSR LP5
LDAA ## 9
STAA § 8400
JSR LP1
LDAA ## I7
STAA § 8400
JSR LP2
TST § 8402
BEQ LP7

```



$L_{P8}$  JSR  $L_{P5}$   
LDAA ~~\*/~~ val de  $V_{min}$   
ADDB  $\$$  250  
TST  $\$$  8402  
BEQ  $L_{P7}$   
LDAA ~~\*/~~ val( $V_{max} - I_c$ )  
GBA  
BHI  $L_{P8}$   
JSR  $L_{P5}$   
 $L_{P7}$  LDAA ~~\*/~~ I8  
STAA  $\$$  8400  
JSR  $L_{P6}$   
LDAA ~~\*/~~ 20  
STAA  $\$$  8400  
JSR  $L_{P5}$   
BRA  $L_{P9}$   
 $L_{P1}$  Tempo  $T_p$   
RTS  
 $L_{P2}$  Tempo  $V_{min}$   
RTS  
 $L_{P5}$  Tempo  $\otimes I_c$  ou RI  
RTS  
 $L_{P6}$  Tempo Jaune  
RTS

- Remarquons que le programme que nous venons de faire ne pourrait être appliqué dans le cas où il y ait des véhicules T.A.G car ceux-ci vont être bloqués par les véhicules allant tout droit et venant de la voie d'en face et risquent de rester là bloqués et n'auront pas le temps de traverser le carrefour avant la fin du vert si la voie de face est trop dense. C'est pour cela que nous avons songé à réaliser une stratégie pouvant résoudre ce problème (stratégie suivante) celle-ci consiste à éteindre le vert des "allant tout droit" avant de celui des T.A.G et T.A.D .  
Remarque: Ces types de stratégies nécessitent que l'information de présence d'un véhicule persiste pendant un temps égal à  $I_c$  ( temps de franchissement du carrefour)

TEMPORISATIONS

Une temporisation n'est qu'un sousprogramme qui pourra retarder le systeme et le laissera figé dans l'état que nous souhaitons . Il suffit donc pour réaliser une temporisation de boucler un certain nombre de fois sur des instructions, mais auparavant il faut connaitre la durée du cycle d'horloge et le nombre de cycles de chaque instruction .

L'horloge du 6800 à un cycle de 1 microseconde

S/P Temporisation du Jaune(3s)

Cette temporisation est faite en tenant compte des reflexes des conducteurs

```

LDAA # 6
LP2 LDX # 37000
LPI DEX
      BNE LPI
      DECA
      BNE LP2

```

S/P Temporisation I<sub>C</sub> ou RI (4s)

Elle du meme type que la première avec une durée un peu plus longue

```

LDAA # 8
LP2 LDX # 37000
LPI DEX
      BNE LPI
      DECA
      BNE LP2

```

S/P Temporisation Vmin ( IOS)

```

LDAA # 17
LP2 LDX # 37000
LPI DEX
      BNE LPI
      DECA
      BNE LP2

```

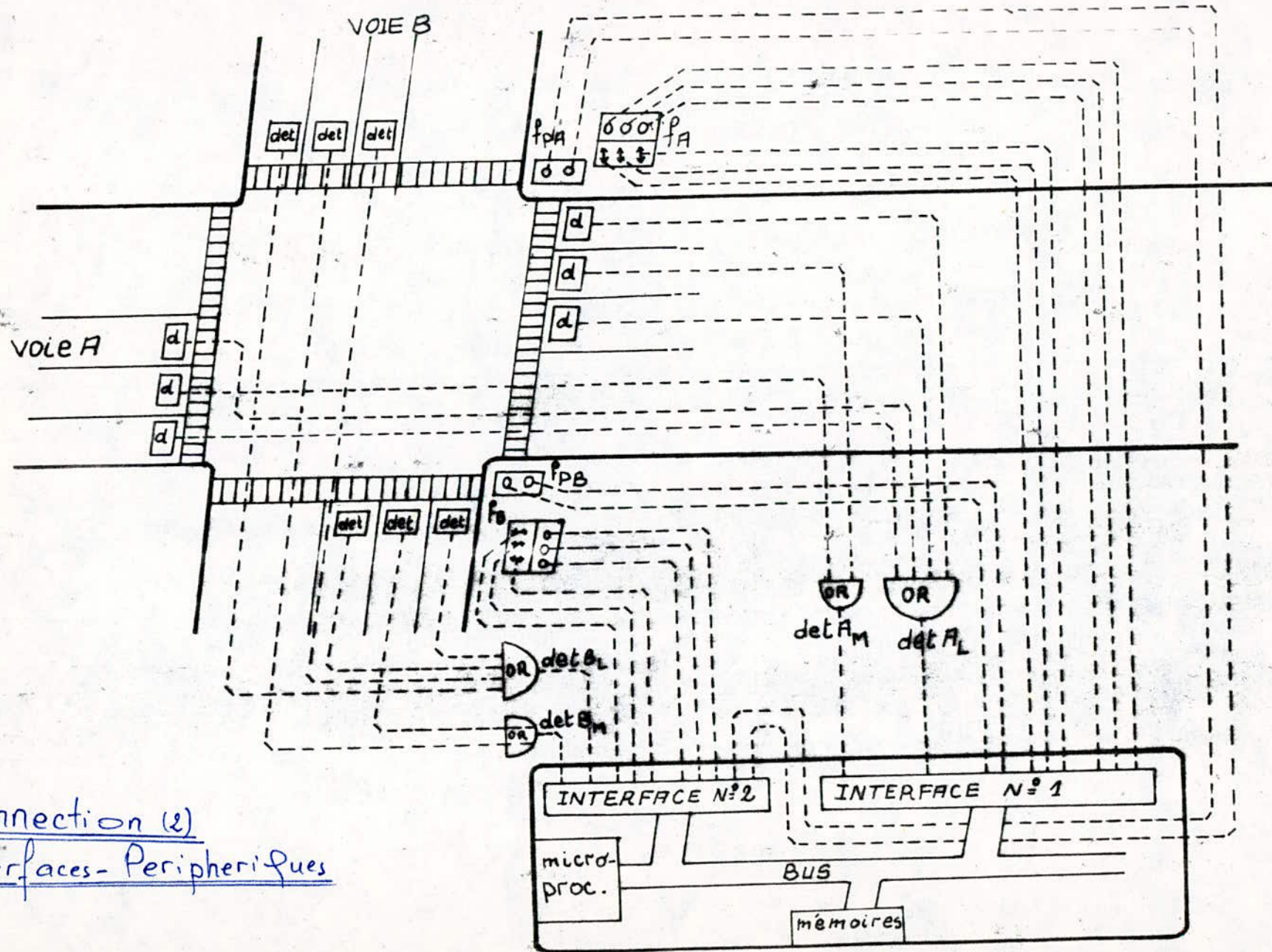
S/P Temporisation 0.5s

```

LDX # 62500
LP BEX
      BNE LP

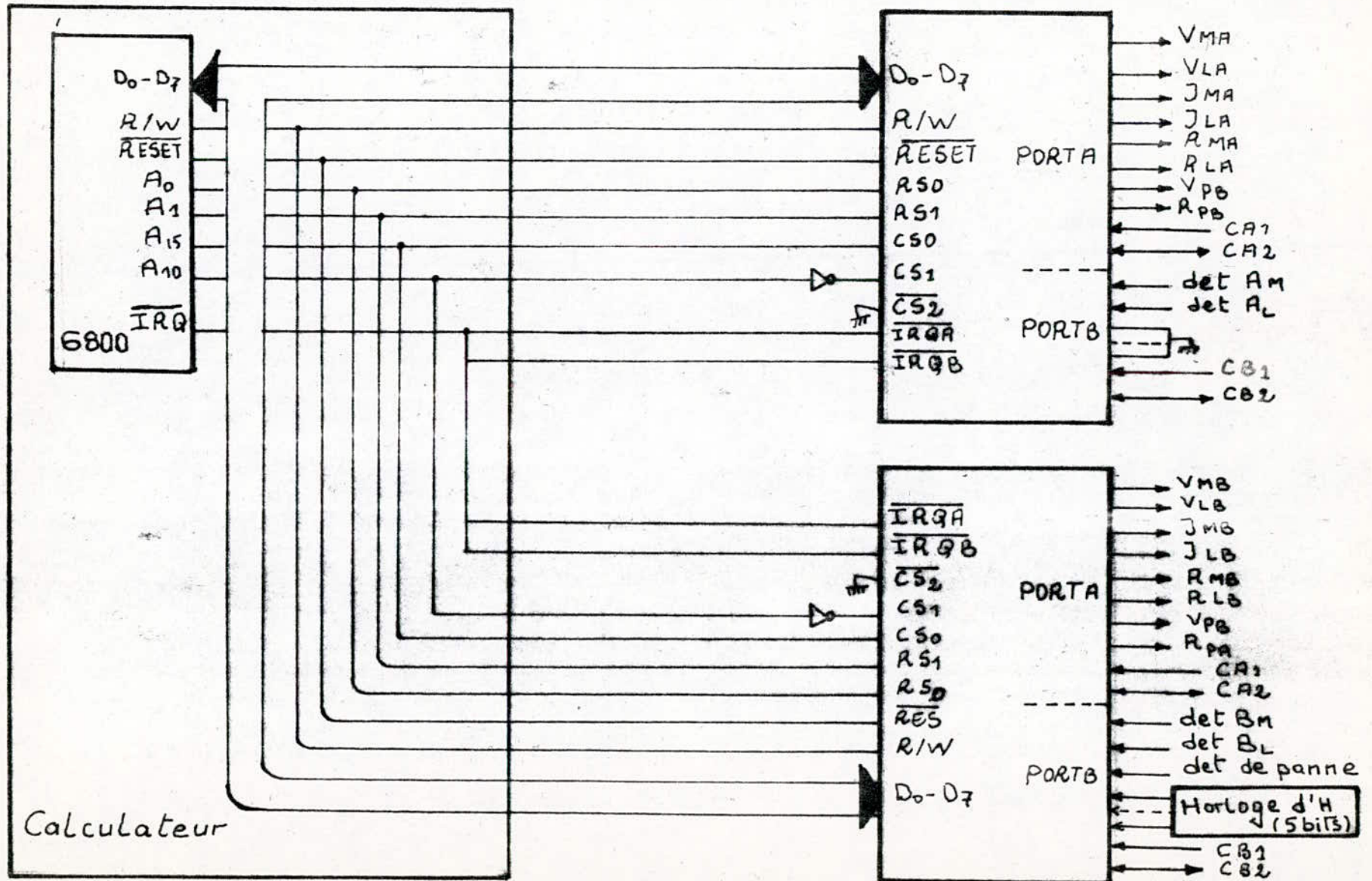
```





Connection (2)  
Interfaces- Périphériques

Présentation des connexions : micro-processeur-interfaces  
Interfaces - extérieur.





-55-

PROGRAMMATION DE LA 2<sup>ème</sup> STRATEGIE

Nous affectons les feux et les détecteurs aux interfaces de la manière suivante:

V <sub>MA</sub>	le vert de la voie milieu de A au bit	B <sub>0</sub>	de	8000
V <sub>LA</sub>	" " latérales "	B <sub>1</sub>	" "	"
J <sub>MA</sub>	le jaune de la voie milieu de A "	B <sub>2</sub>	" "	"
J <sub>LA</sub>	" " latérales "	B <sub>3</sub>	" "	"
R <sub>MA</sub>	le rouge de la voie milieu de A au bit	B <sub>4</sub>	"	"
R <sub>LA</sub>	" " latérales " "	B <sub>5</sub>	"	"
V <sub>PB</sub>	le vert des piétons de B au bit	B <sub>6</sub>	"	"
R <sub>PB</sub>	le rouge " " "	B <sub>7</sub>	"	"
V <sub>MB</sub>	le vert de la voie milieu de B au bit	B <sub>0</sub>	de	8400
V <sub>LB</sub>	" " latérales " "	B <sub>1</sub>	"	"
J <sub>MB</sub>	le jaune de la voie milieu de B "	B <sub>2</sub>	"	"
J <sub>LB</sub>	" " latérales " "	B <sub>3</sub>	"	"
R <sub>MB</sub>	le rouge de la voie milieu de B "	B <sub>4</sub>	"	"
R <sub>LB</sub>	" " latérales " "	B <sub>5</sub>	"	"
V <sub>PA</sub>	le vert des piétons de A au bit	B <sub>6</sub>	"	"
R <sub>PB</sub>	le rouge " " "	B <sub>7</sub>	"	"

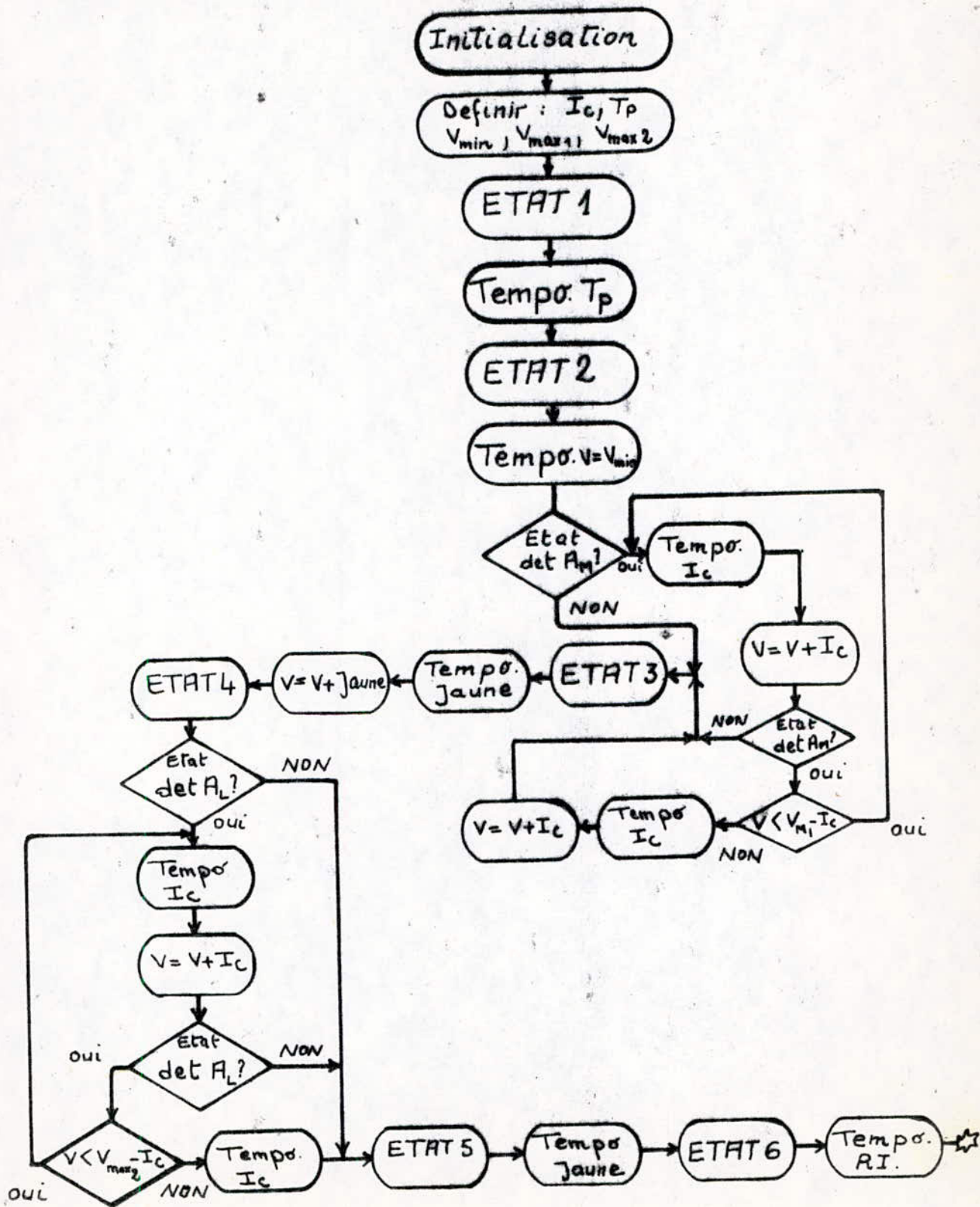
Le détecteur de la voie milieu de A au bit B<sub>0</sub> de 8002  
 Le détecteur des voies latérales de A " B<sub>1</sub> "  
 Le détecteur de la voie milieu de B au bit B<sub>0</sub> de 8402  
 Les détecteurs des voies latérales de B " B<sub>1</sub> "

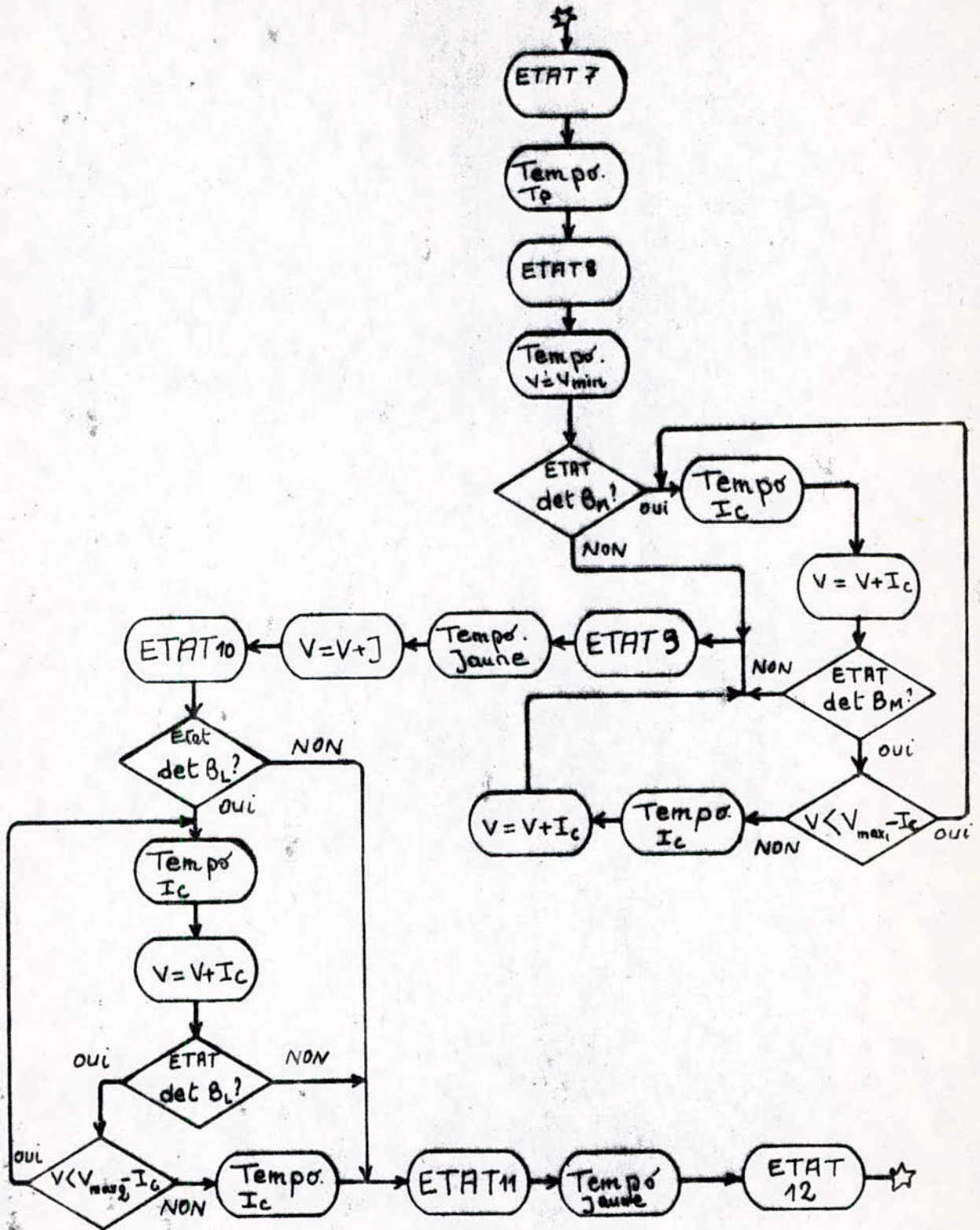
États	Voie A		Voie B		Piétons $P_B$	Piétons $P_A$	Temps
	TAG-TAD	Tout droit	TAG-TAD	tout droit			
1	V	V	R	R	V	R	$T_p$
2	V	V	R	R	R	R	,
3	V	J	R	R	R	R	
4	V	R	R	R	R	R	
5	J	R	R	R	R	R	Jaune
6	R	R	R	R	R	R	RI
7	R	R	V	V	R	V	$T_p$
8	R	R	V	V	R	R	
9	R	R	V	J	R	R	
10	R	R	V	R	R	R	
11	R	R	J	R	R	R	Jaune
12	R	R	R	R	R	R	RI
1	V	V	R	R	V	R	$T_p$

Tableau des transitions des états

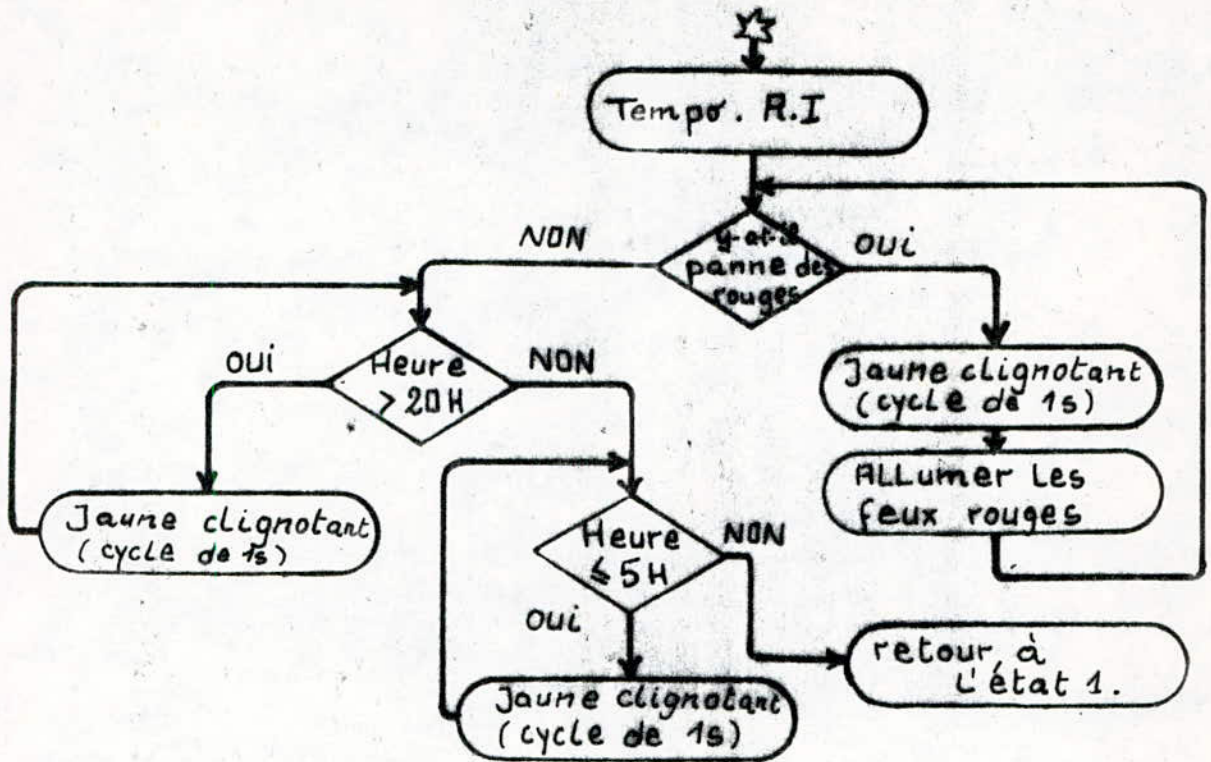


Organigramme:









Legende:

- $T_p$ : temps de passage des piétons.
- $I_c$ : temps de franchissement du carrefour.
- det  $A_M$  (ou  $B_M$ ): détecteurs des voies A (ou B) milieu - voies directes de la même route -
- $V_{max1}$ : vert max pris pour les voies directes
- $V_{max2}$ : vert max pris pour les T.A.D et T.A.G
- det  $A_L$  (ou  $B_L$ ): détecteurs des voies Latérales-T.A.D et T.A.G - de la route A (ou B)

PROGRAMME

```

LDAB ## val de Vmin
ldac ## val de Ic
STAA § 250
LDAA ## I76
STAA § 8400
LPI3 LDAA ## 67
STAA § 8000
JSR LPI
LDAA ## I31
STAA § 8000
JSR LP2
LDAA § 8002
STAA § 8002
ANDA ## I
BEQ LP3
LP5 JSR LP4
ADDB § 250
LDAA § 8002
STAA § 8002
ANDA ## I
BEQ LP3
LDAA ## val (VmaxI - Ic)
CBA
BHI LP5
JSR LP4
ADDB § 250
LP3 LDAA ## I34
STAA § 8000
JSR LP8
LDAA ## val de Vmin
AAB
LDAA ## I46
STAA § 8000
LDAA § 8002
ANDA ## 2

```



BEQ L<sub>P6</sub>  
L<sub>P7</sub> JSR L<sub>P4</sub>  
ADDB \$ 250  
LDAA \$ 8002  
ANDA ~~##~~ 2  
BEQ L<sub>P6</sub>  
LDAA ~~##~~ val (Vmax<sub>2</sub>-I<sub>c</sub>)  
CBA  
BHI L<sub>P7</sub>  
JSR L<sub>P4</sub>  
L<sub>P6</sub> LDAA ~~##~~ I52  
STAA \$ 8000  
JSR L<sub>P8</sub>  
LDAA ~~##~~ I76  
STAA \$ 8000  
JSR L<sub>P4</sub>  
LDAA ~~##~~ 67  
STAA \$ 8400  
JSR L<sub>PI</sub>  
LDAA ~~##~~ I3I  
STAA \$ 8400  
JSR L<sub>P2</sub>  
LDAB ~~##~~ val Vmin  
LDAA \$ 8402  
ANDA ~~##~~ I  
L<sub>PIO</sub> BEQ L<sub>P9</sub>  
JSR L<sub>P4</sub>  
ADDB \$ 250  
LDAA \$ 8402  
ANDA ~~##~~ I  
BEQ L<sub>P9</sub>  
LDAA ~~##~~ val (Vmax<sub>I</sub>-I<sub>c</sub>)  
CBA  
BHI L<sub>PIO</sub>  
JSR L<sub>P2</sub>  
ADDB \$ 250

```

LPG  LDAA ## I34
      STAA § 8400
      JSR  LP8
      LDAA ## val(jaune)
      AAB
      LDAA ## I46
      STAA § 8400
      LDAA § 8402
      ANDA ## 2
      BEQ  LPII
LPI2 JSR  LP4
      ADDB § 250
      LDAA § 8402
      ANDA ## 2
      BEQ  LPII
      LDAA ## val (Vmax2-Ic)
      CBA
      BHI LPI2
      JSR  LP4
LPII LDAA ## I52
      STAA § 8400
      JSR  LP8
      LDAA ## I76
      STAA § 8400
      JSR  LP4
LPI4 LDAA § 8402
      ANDA ## 4
      BEQ  LPI6
      JSR  LPI8
      LDAA ## I76
      STAA § 8000
      STAA §8400
      BRA  LPI4
      LDAA ## 20
      STAA § 254

```



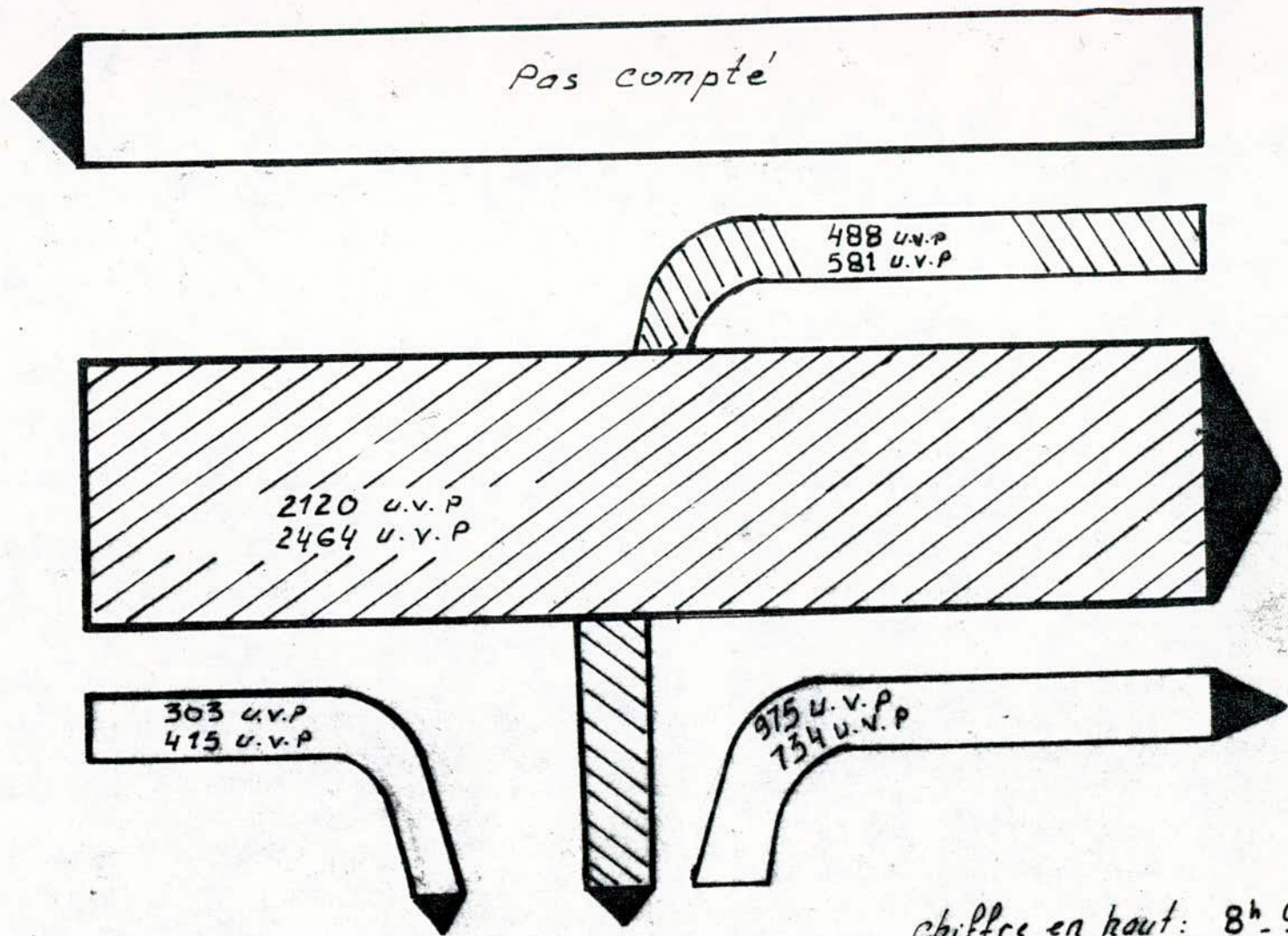
LDAA  $\S$  8402  
CMPA  $\S$  254  
BIS L<sub>PI7</sub>  
JSR L<sub>PI8</sub>  
BRA L<sub>PI6</sub>  
L<sub>PI7</sub> LDAA ~~5~~  
STAA  $\S$  254  
LDAA  $\S$  8402  
CMPA  $\S$  254  
BHI L<sub>PI3</sub>  
JSR L<sub>PI8</sub>  
BRA L<sub>PI7</sub>  
L<sub>PI1</sub> Tempo de T<sub>p</sub> (pour piétons)  
RTS  
L<sub>P2</sub> Tempo V<sub>min</sub>  
RTS  
L<sub>P4</sub> Tempo I<sub>c</sub> ou RI  
RTS  
L<sub>P8</sub> Tempo Jaune  
RTS  
L<sub>PI5</sub> Tempo de  $\frac{1}{2}$ s (de jaune)  
RTS  
L<sub>PI8</sub> LDAA ~~I2~~  
STAA  $\S$  8000  
STAA  $\S$  8400  
JSR L<sub>PI5</sub>  
LDAA ~~0~~  
STAA  $\S$  8000  
STAA  $\S$  8400  
JSR L<sub>PI5</sub>  
RTS

Le programme que nous venons de faire n'est qu'un exemple de programmation, nous pouvons changer ce programme suivant l'état du trafic et adapter la stratégie convenable.

Vu que suivant l'heure, la circulation peut varier au cours de la journée ; Nous pouvons alors changer de stratégies, ou pour la même stratégie que nous venons de programmer, nous pouvons faire varier les paramètres qui la caractérisent ; Par exemple nous pouvons programmer un  $V_{max}$  d'une même route variable selon l'heure de façon à avoir dans le même croisement le temps du vert Max de la route la plus dense supérieur à celui de celle la moins dense.

Ceci devra être très utile car comme le montre l'exemple suivant que nous avons compté au niveau du pont de Carroubier, les densités de circulation au cours d'une même journée sont très diversifiées.





(fig-11-a-)

chiffre en haut: 8<sup>h</sup>-9<sup>h</sup>

chiffre en bas: 11<sup>h</sup>30-12<sup>h</sup>30

Diagramme de circulation  
établi au pont de Carroubier

CHAPITRE IV

COORDINATION DE CARREFOURS  
(MODELISATION PAR MACROREGULATION )



La régulation microscopique de plusieurs carrefours adjacents comme nous l'avons déjà vu consiste à traiter les carrefours séparément (régulation véhicule par véhicule) en tenant compte des décalages entre carrefours, ce qui est réalisable avec un simple microordinateur puisque le capteur que nous avons utilisé permet de donner tous les paramètres, non directement mesurables, nécessaires pour la programmation de gestion de chaque carrefour. Mais au cas où nous voulons réguler un réseau de carrefours plus complexe, cette méthode devient insuffisante vu que les tronçons d'entrée (ou de sortie) peuvent provenir (ou conduire à) de plusieurs carrefours voisins. Alors la circulation devient de plus en plus aléatoire dans les tronçons et nous ne pourrions plus fixer le temps de traitement sans qu'il y ait interaction entre carrefours, ce temps de traitement devient donc dépendant des états aléatoires des tronçons origines et destinations.

Dans ces conditions si complexes, la solution avec un microordinateur n'est plus possible, la microrégulation n'étant plus favorable et nous aurons plutôt à utiliser un ordinateur avec lequel nous pouvons appliquer la macrorégulation qui est souple et permet la coordination de carrefours.

Dans ce cas là le microordinateur sera utilisé simplement comme moyen de commande dans l'endroit même du carrefour et c'est avec une commande centrale par ordinateur que le système de carrefours est géré;

Avec un ordinateur nous pouvons disposer d'un programme général applicable pour n'importe quel carrefour dans lequel on ne fait appel qu'aux paramètres utiles et spécifiques aux carrefours. Ces paramètres sont déterminés par programmation en chaque instant et à chaque état de circulation en parallèle avec le programme principal. Ces paramètres peuvent être logés en mémoire ou de préférence dans des périphériques dont la capacité mémoire est plus élevée.

Il s'ensuit qu'avec le programme principal nous pouvons gérer un certain nombre de carrefours puis passer à d'autres sous contrôle du moniteur. (exemple de régulation des carrefours de rang pair



puis ceux de rang impair et ainsi desuite)

Pour la coordination de carrefours, la description géométrique de chaque carrefour fixera seulement les différentes possibilités d'évacuation des voies en parallèle dans un réseau carrefour. La stratégie appliquée au carrefour pris en considération doit répondre aux conditions de génération optimales des tronçons origines et destination. De cette manière la stratégie elle même appliquée ne sera pas définitivement fixée pour un tel carrefour et tel temps, mais elle sera changeable suivant l'état de circulation aléatoire c'est à dire suivant les paramètres utiles pour la macrorégulation que nous avons défini, pour cela, nous disposons de quatre sous ensembles de sous programmes ou procédures :

- Un modèle de simulation des tronçons
- Un modèle de simulation des carrefours
- Les procédures d'entrée et de sortie de véhicules dans le réseau
- Un moniteur pour gérer les différents travaux .

#### IVI MODELISATION D'UN CARREFOUR PAR MACROREGULATION

Pour chaque carrefour la mémoire de modélisation se compose des différentes valeurs numériques stockées dans différents répertoires

- Connaissance des tronçons :origines et destinations
- Stockage des demandes de transfert dans une liste de demandes
- Paramètres caractérisant les tronçons demandeurs et recepteurs
- Indicateur des différentes regles de gestion du carrefour.

Les regles sont constituées par:

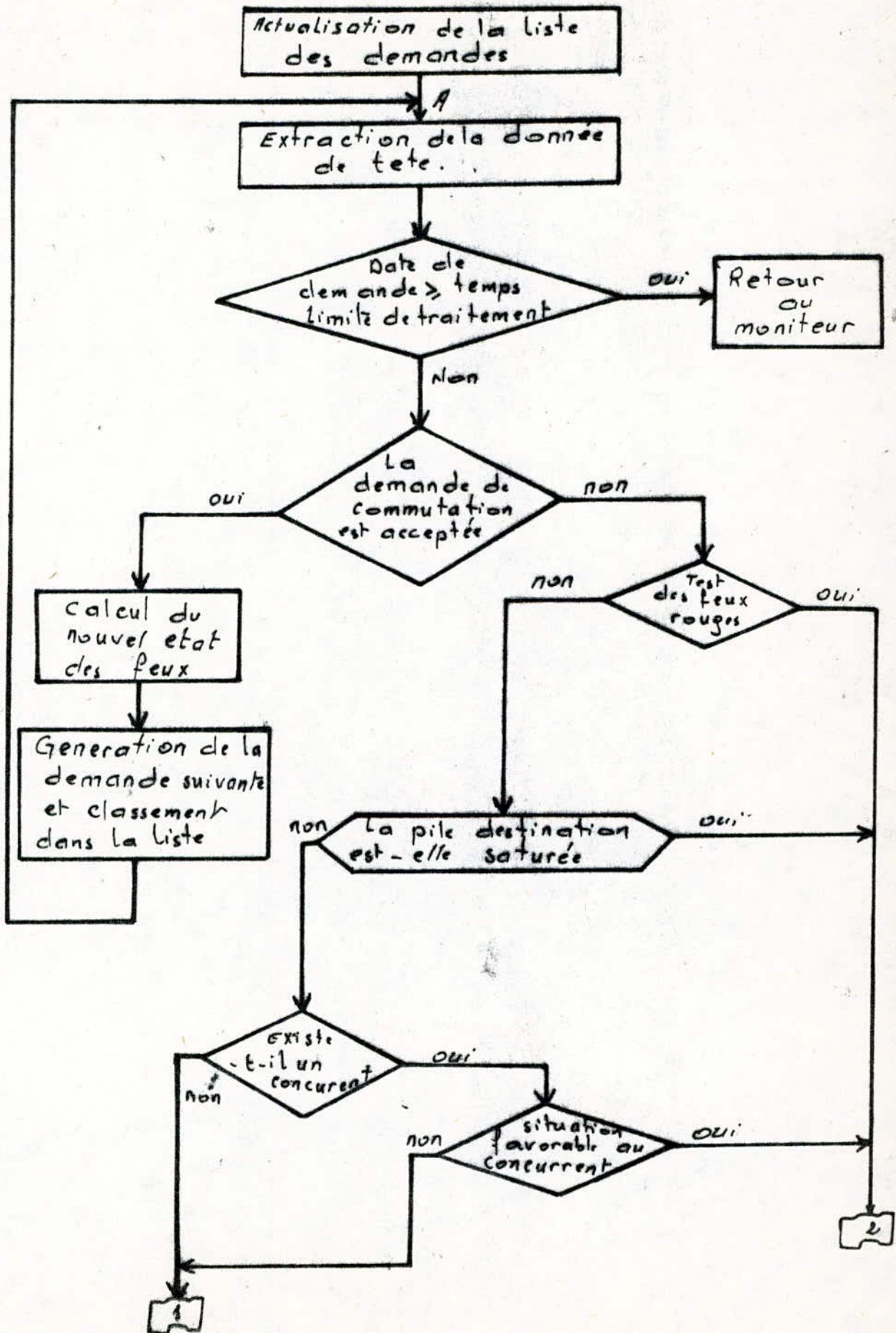
-Les tests d'interaction des itinéraires et d'interaction avec les autres déplacements élémentaires-piétons-et les décisions qui en découlent.

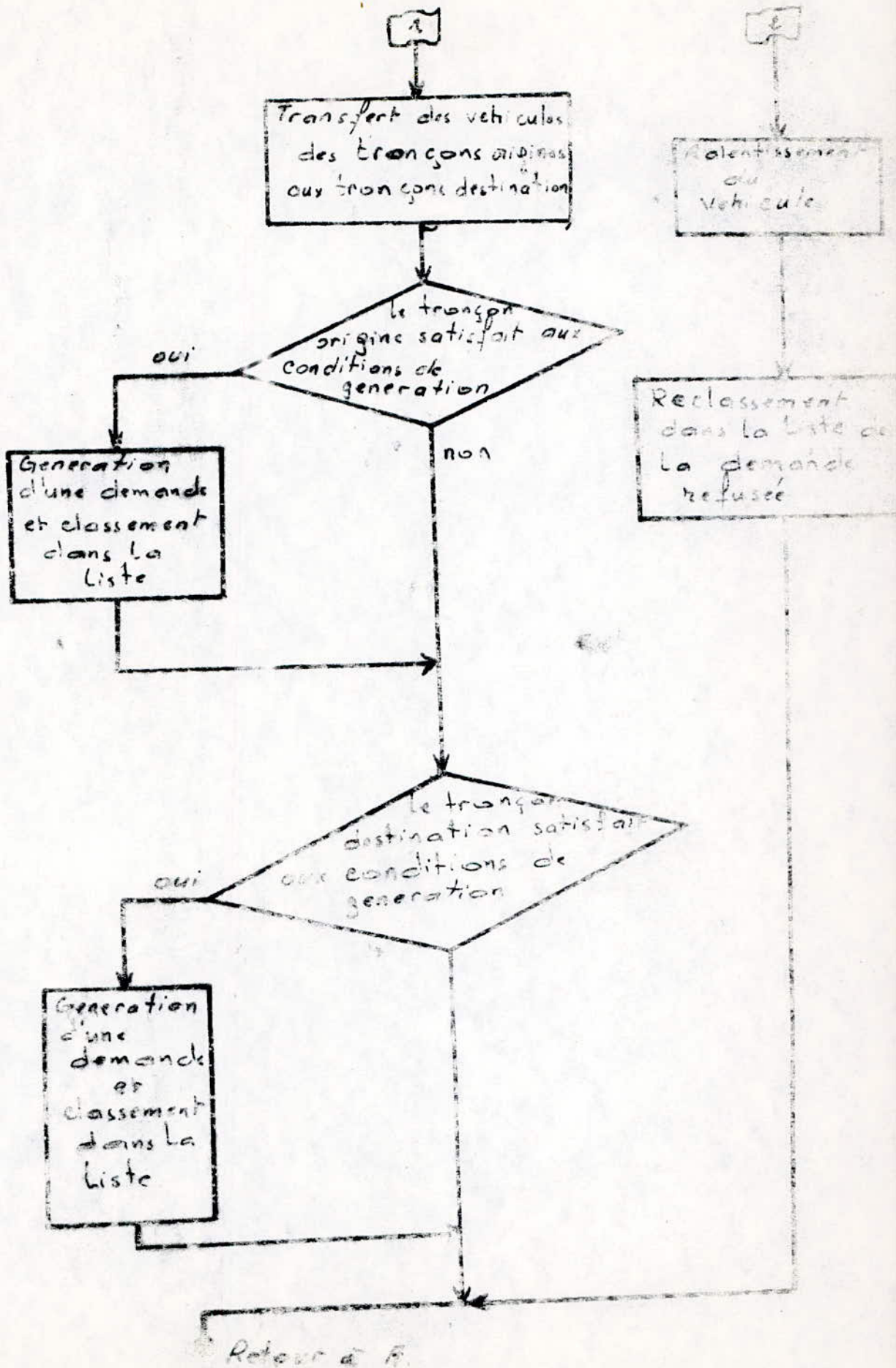
- Les calculs numériques permettant d'évaluer les temps de transfert ou les ralentissements .

Les demandes de transfert sont extraites les unes apres les autres et subissent un traitement qui est fonction des valeurs stockées dans la mémoire de modélisation .

-La logique du modèle est schématisée par l'organigramme de la figure en face . Pour que le fonctionnement soit possible la liste des demandes doit contenir une seule demande de commutation autorisant le mouvement des véhicules des tronçons entrés ou









sorties dont la demande de commutation a été acceptée .

Si la demande a été refusée on passe de nouveau au test des différentes conditions de circulation en présence pour reclasser les demandes de transfert s'il en existe.

-Lorsqu'un tronçon incident ne peut plus fournir de véhicules il se produit une absence dans la liste des demandes ; De même pour les tronçons récepteurs.

-Le nombre moyen de véhicules que nous supposons laisser dans un tronçon est fonction de la circulation en présence dans le réseau entier. C'est relativement à ce nombre de véhicules permanent que nous déterminons que le tronçon origine ou destination satisfait aux conditions de génération ou non pour pouvoir générer une demande .

En cas où les conditions favorables pour la génération persistent jusqu'à un temps limite de traitement , nous revenons au moniteur pour la gestion du réseau de carrefours (Hiérarchisation des carrefours auxquels il faut passer par la suite).

#### IV 2 ORGANISATION GENERALE DE LA SIMULATION D'UN RESEAU DE CARREFOURS

La simulation à une structure modulaire est organisée autour d'un moniteur , ensemble logique dans lequel on fait appel aux différents modules . On construit une logique basée sur une liste d'attente où sont classés, en fonction des priorités de traitement, tous les carrefours du réseau. Le traitement du carrefour le plus prioritaire entraîne l'évolution de sa situation et de celle de son voisinage , ce qui a pour conséquence le remaniement de la liste d'attente.

Le réseau contient quatre types d'éléments, chacun d'eux exigeant un traitement spécial. Ce sont comme nous l'avons vu , les carrefours, les tronçons, les entrées et les sorties du réseau .

Si on n'était pas limité en occupation mémoire, on pourrait traiter tous les événements dans l'ordre chronologique. Cela n'est pas réalisable car il est nécessaire de disposer d'une mémoire périphérique pour stocker les informations relatives aux carrefours. Le traitement se fait donc élément par élément, la simulation d'un élément carrefour nécessitant l'introduction en mémoire centrale des valeurs numériques de cet élément.



Les transferts mémoire centrale- mémoire périphérique sont très coûteux en temps d'exécution, nous devons donc faire en sorte qu'ils soient les plus rares possible, conséquence immédiate: la durée en temps réel de traitement d'un élément carrefour résident en mémoire centrale doit être la plus grande possible ?

Il s'ensuit que tous les éléments géographiques seront tour à tour, soit en retard soit en avance les uns par rapport aux autres ce qui interdit d'avoir des décalages trop importants. Donc l'une des attributions du moniteur est de doser correctement ces décalages et, puisque la principale contrainte vient des carrefours le problème revient à maximaliser, pour chacun d'eux, la durée en (temps réel) de simulation. Pour minimaliser le nombre d'appels périphériques nous traiterons en priorité le carrefour ayant la période de simulation la plus grande. Le choix des autres éléments à traiter devient alors automatique, nous réalisons, après traitement d'un carrefour, la mise à l'heure de son voisinage. Le traitement de cette mise à l'heure réagissent évidemment sur les priorités, ce qui permet de boucler le système.

Pratiquement cette notion de priorité est matérialisée et mémorisée en constituant une liste d'attente où tous les carrefours sont classés dans l'ordre

le moniteur est représenté par la figure suivante (en face)

#### MISE A L'HEURE DU VOISINAGE

Le carrefour étant considéré comme l'élément principal, le traitement des autres éléments en dépend.

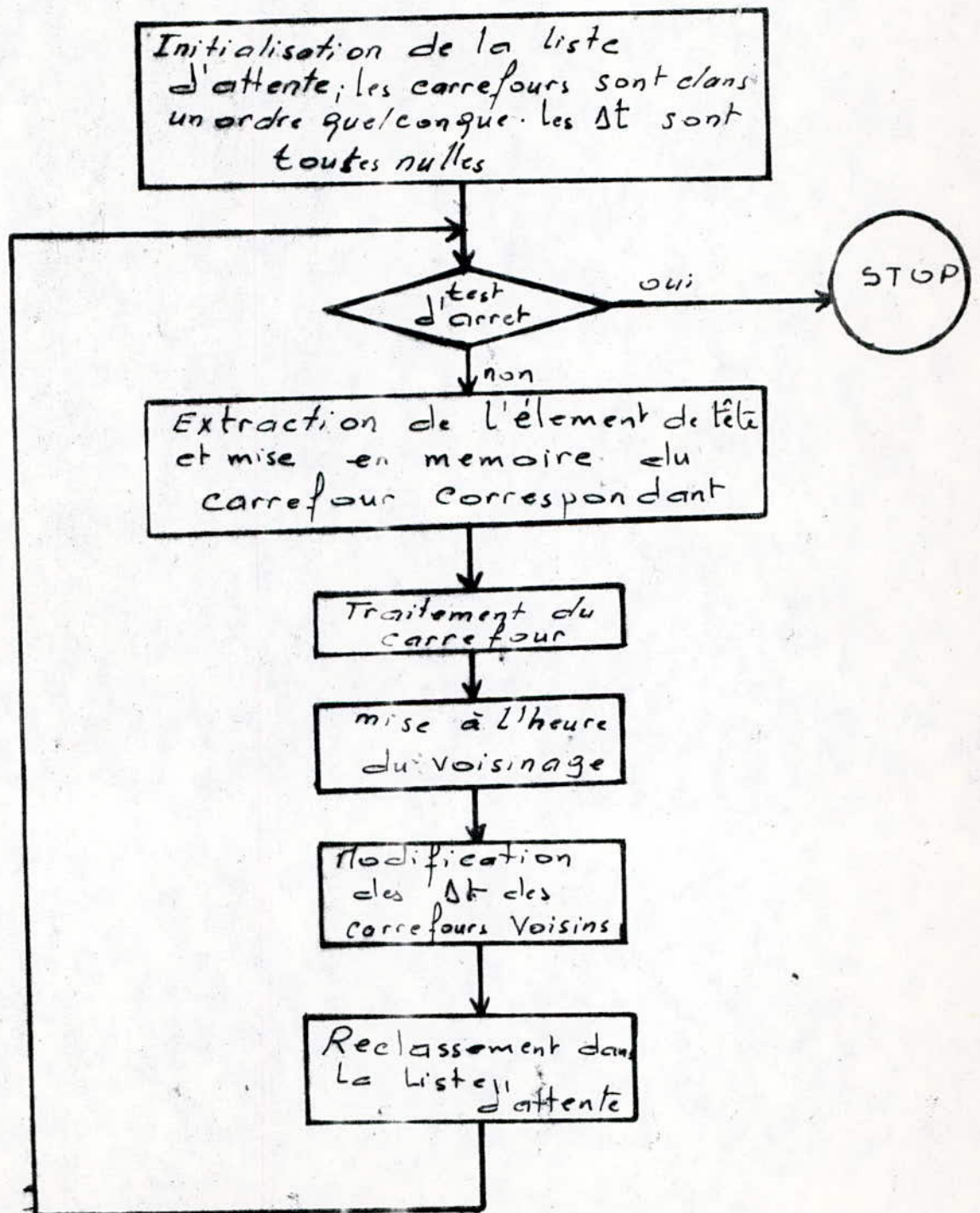
##### a) Voisinage immédiat

Il est constitué par l'ensemble de demi-tronçons partant du carrefour principal et des noeuds auxquels ils aboutissent. Si ces noeuds sont périphériques, nous faisons appel aux procédures de simulation des tronçons et de sorties de véhicules. Si ce sont des noeuds internes (carrefours) nous nous contentons de les mettre en réserve dans la liste d'attente auxiliaire.

##### b) Voisinage secondaire

Ce sont les ensembles de demi-tronçons aboutissant aux carrefours (voisins du carrefour principal) mis en réserve.





Schema Logique du moniteur

ad le des du moniteur

Nous traitons, les uns après les autres, tous ces ensembles; Cela consiste à émettre des véhicules s'il y a lieu et à simuler tous les demi-tronçons.

C'est après cette opération que l'on recalcule la date limite de traitement du carré principal, ce qui entraîne éventuellement la modification de sa période de traitement DT et par suite, sa priorité dans la liste d'attente principale.

EN RESUME nous pouvons dire que le principe de modélisation que nous venons d'étudier suit, par ordre, les opérations suivantes:

-Description des carrefours qui comprend pour chacun la lecture des données descriptives (géométrie et signalisation), la génération des entrées et sorties et la mémorisation sur périphériques.

-Description des tronçons et des itinéraires: lecture des données descriptive, lecture sur périphériques du modèle itinéraires préalablement déterminé et enregistré, génération des entrées et sorties.

-Lecture de la matrice origine-destination et des débits aux entrées du réseau.

-Initialisation des carrefours: commutation au temps zero, génération d'une demande de commutation, initialisation d'une liste (mise à l'heure) pour permettre une actualisation correcte de la liste

---

des demandes au premier appel.

-Initialisation de la liste d'attente principale du moniteur: nous recopions dans la liste toutes les adresses du repertoire des carrefours en affectant à chacun une priorité quelconque les DT sont prises au début toutes nulles.



ANNEXE

## NOTIONS SUR LE MICRO PROCESSEUR 6800 :

### 1 - ORGANISATIONS INTERNE ET EXTERNE :

L'organisation interne est donnée sur la fig 1 .

Le microprocesseur comprend :

- Une unité arithmétique et logique.
  - Une unité de contrôle et de décodification des instructions associée au registre d'instruction.
  - Une mémoire locale représentée par 6 registres disponibles à l'utilisateur.
- Les différents registres et les lignes d'E/S accessibles sont représentés par la fig 2.

- 1 bus de données 8 bits bidirectionnels
- 1 bus d'adresse 16 bits

$R/W$  : 1 le microprocesseur est en état de lecture

0 le microprocesseur est en état d'écriture

$VMA$  : Adresse mémoire valide (validation de l'adresse mémoire ; si

$VMA=1$  une adresse est représentée sur le bus d'adresse);

$IRQ$  : demande d'interruption masquable (si celui-ci est = 0)

$NMI$  : Demande d'interruption non masquable,  $\Phi_1$  et  $\Phi_2$  : signaux d'Horloge  
(2 phases sans recouvrement)

$\overline{RESET}$  : Permet l'initialisation du microprocesseur.

$DBE$  : validation du Bus de données

$TSC$  : Signal d'entrée de contrôle qui permet de faire passer à l'état haute impédance les lignes d'adresse  $R/W$ .

$BA$  : Bus disponible  $BA = 1$  indique que le microprocesseur est à l'arrêt et que le bus d'adresse est disponible.

$\overline{HALT}$  : Demande d'accès direct à la mémoire.

Permet l'arrêt de l'unité centrale.

Les 6 registres accessibles sont :

- Deux accumulateurs :  $AccA$  et  $AccB$  de 8 bits
- Un registre d'index 16 bits ( $Rx$ )
- Un pointeur de pile ( $Sp$ ) de 16 bits. Il contient l'adresse de la position mémoire disponible dans une pile externe du type LIFO (dernier rentré - premier sorti)

- Un compteur ordinal ( $PC$ ) de 16 bits

- Un registre d'état de 8 bits qui fournit les 16 informations suivantes :

Bit 0 : C retenue (carry)

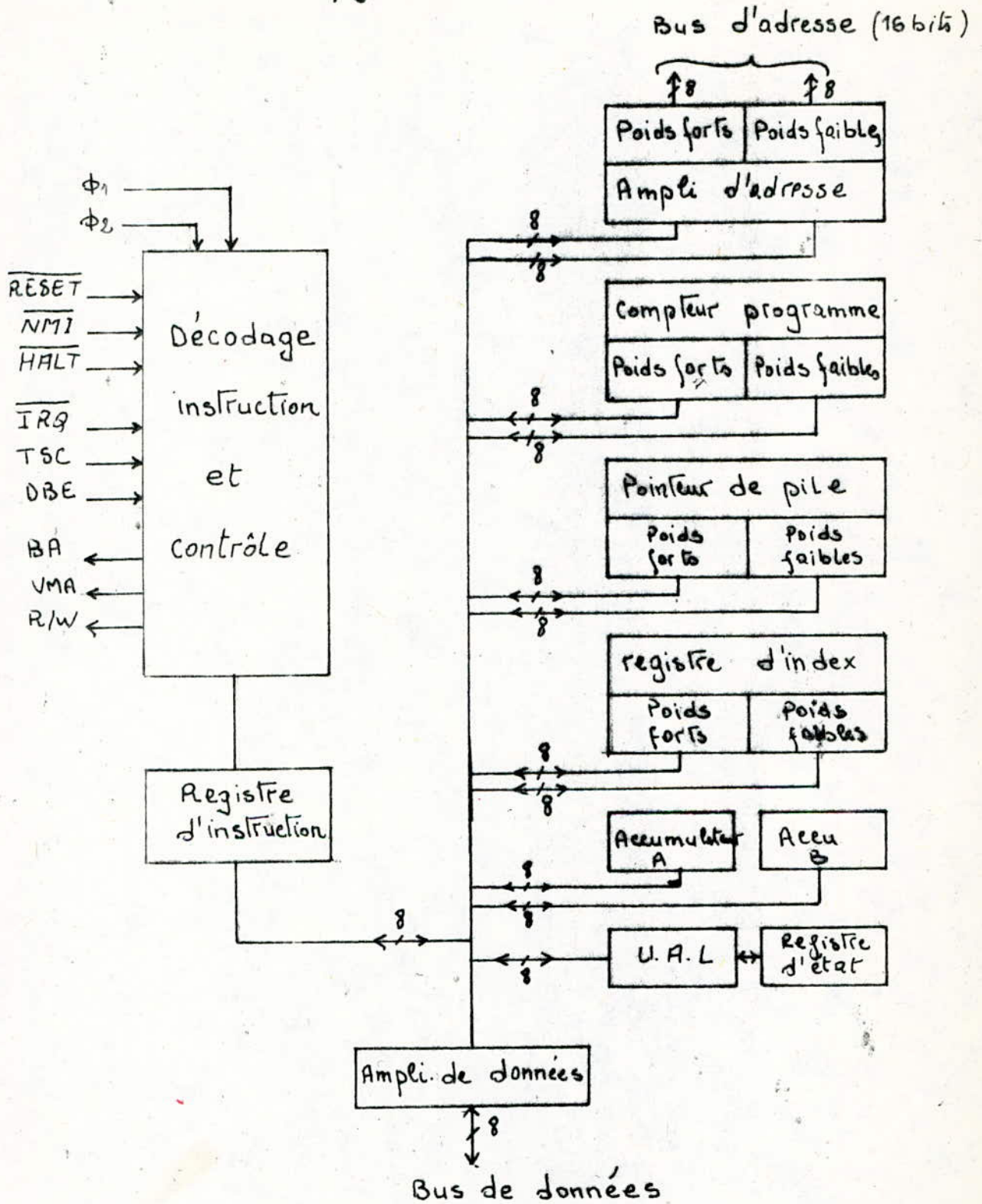
Bit 1 : V over flow

Bit 2 : Z résultat nul

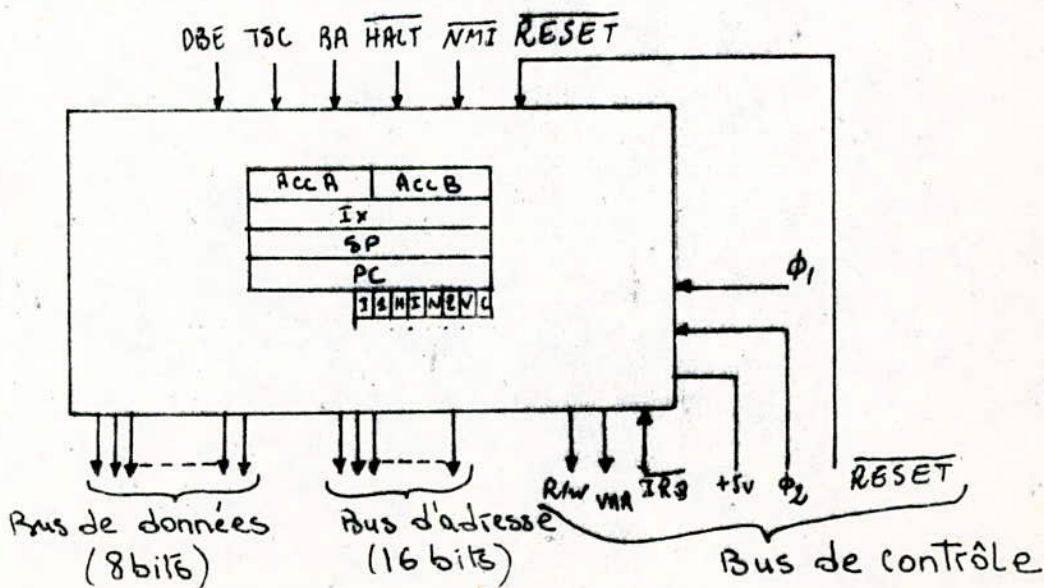
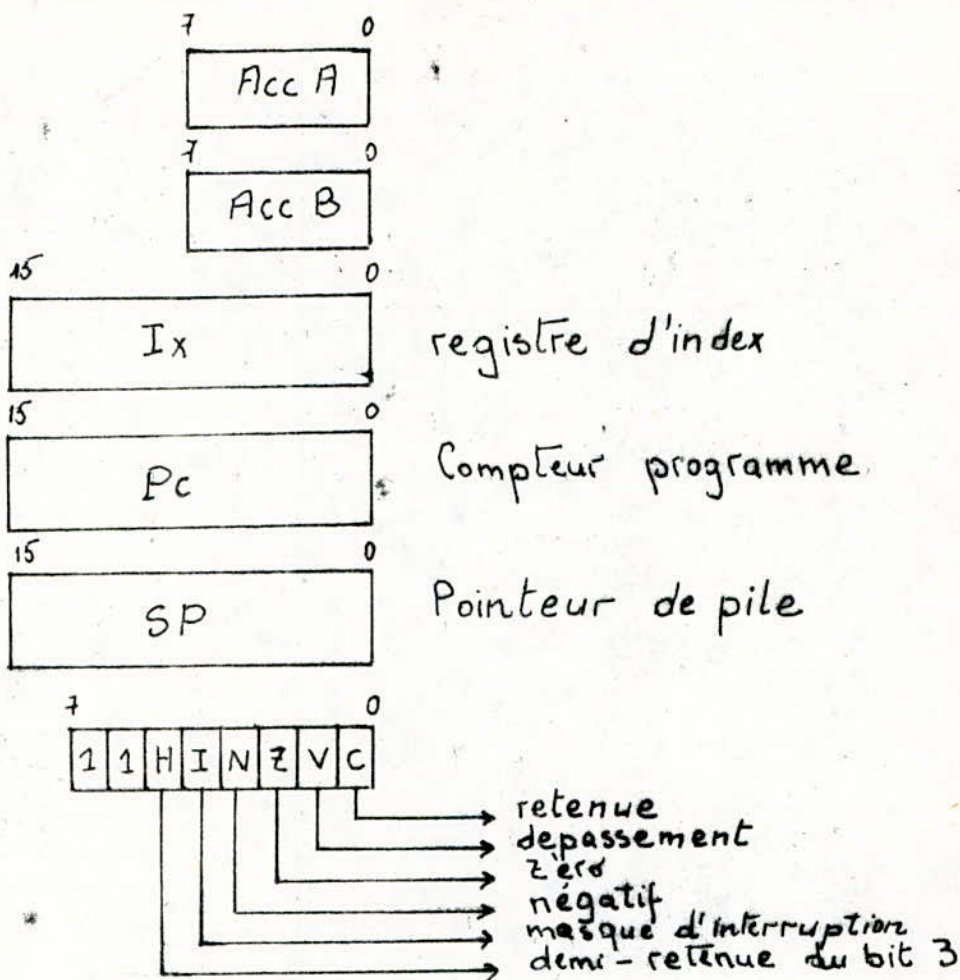
Bit 3 : N résultat  $< 0$

Bit 4 :  $\Phi$  masquage d'interruption





(Fig 1)



(fig 2)



Bit 5 : H demi - retenue  
Bit 6 et 7 : toujours au niveau 1.

### COUPLEURS ET MODES D'ADRESSAGE :

La capacité d'adressage du 6800 est de 65 536 octets, les registres d'entrées / sortie inclus dans les coupleurs sont traités comme des positions de mémoire. Le 6800 dispose de 7 modes d'adressages : registre, page zéro, étendu, indexé, relatif, immédiat et implicite.

Ses 6 registres sont programmables par l'utilisateur. La configuration minimale requiert 5 boîtiers : Horloge, microprocesseur, RAM, ROM et coupleur. Le 6800 a un jeu de 72 types d'instructions.

Compte tenu des modes d'adressage, il possède 197 instructions. Ces instructions peuvent être classées en 5 groupes : instructions logiques, instructions de saut systématique ou conditionnel et appel de sous-programmes instructions de contrôle du microprocesseur. La longueur des instructions est de 1, 2 ou 3 octets.

Le 6800 dispose de plusieurs types de coupleurs.

Le coupleur parallèle utilisé est le 6820.

### PIA 6820

Le PIA 6820 est un circuit d'adaptation entre le microprocesseur et les périphériques (dans notre cas : capteurs, feux, ...). Il permet au moyen des 8 bits de données (bus bidirectionnel) d'accéder à la plupart des périphériques.

### SIGNAUX ECHANGES AVEC LE SYSTEME :

L'interface avec le microprocesseur se fait à l'aide de 3 bus : bus de contrôle, bus de données et le bus d'adresse.

Le bus de contrôle est composé des lignes suivantes :

E : Signal d'activation des échanges. Cette ligne correspond à l'horloge du microprocesseur.

R/W : 1 lecture  
0 écriture

RESET : Remet tous les registres du PIA à zéro.

$\overline{IRQA}$  et  $\overline{IRQB}$  : demandes d'interruption pour le microprocesseur issues resp du port A et du Port B.

CS : Selection du boîtier PIA. En effet il dispose de 3 lignes de selection (CS0 - CS1 -  $\overline{CS2}$ ) facilitant le décodage d'adresse pour la selection de ce boîtier.

Le bus de données est un bus de 8 bits bidirectionnels, directement relié au bus de données du microprocesseur, mais le bus adresse est ici réduit aux lignes A0 et A1 pour la sélection des registres internes.

INTERFACE AVEC LA PERIPHERIE :

Les fonctions de l'inter-face avec la périphérie sont identiques pour le port A et le port B et regroupent au total 20 lignes :

- PA0 - PA7 } 16 lignes de données programmables individuellement en
- PB0 - PB7 } entrée et en sortie.
- CA1 - CB1 2 lignes d'entrée d'IT dans le PIA
- CA2 - CB2 2 lignes programmables en entrée d'IT ou en sortie de commande.

La consommation est d'environ 110 nA et la puissance dissipée est de 500 mW

ORGANISATION INTERNE :

Le PIA est constitué de 2 ports, chacun renferme les registres nécessaires à l'interface du 6800 avec un périphérique à savoir :

- Un registre de données DRA ou DRB
- Un registre de direction des données DDRA ou DDRB.
- Un registre de contrôle du PORT CRA ou CRB

ADRESSAGE DES REGISTRES DU PIA :

RS1	RS0	CRA2	CRB2	Reg. adressé
0	1	x	x	CRA
0	0	0	x	DDRA
0	0	1	x	DRA
1	1	x	x	CRB
1	0	x	0	DDR B
1	0	x	1	DRB



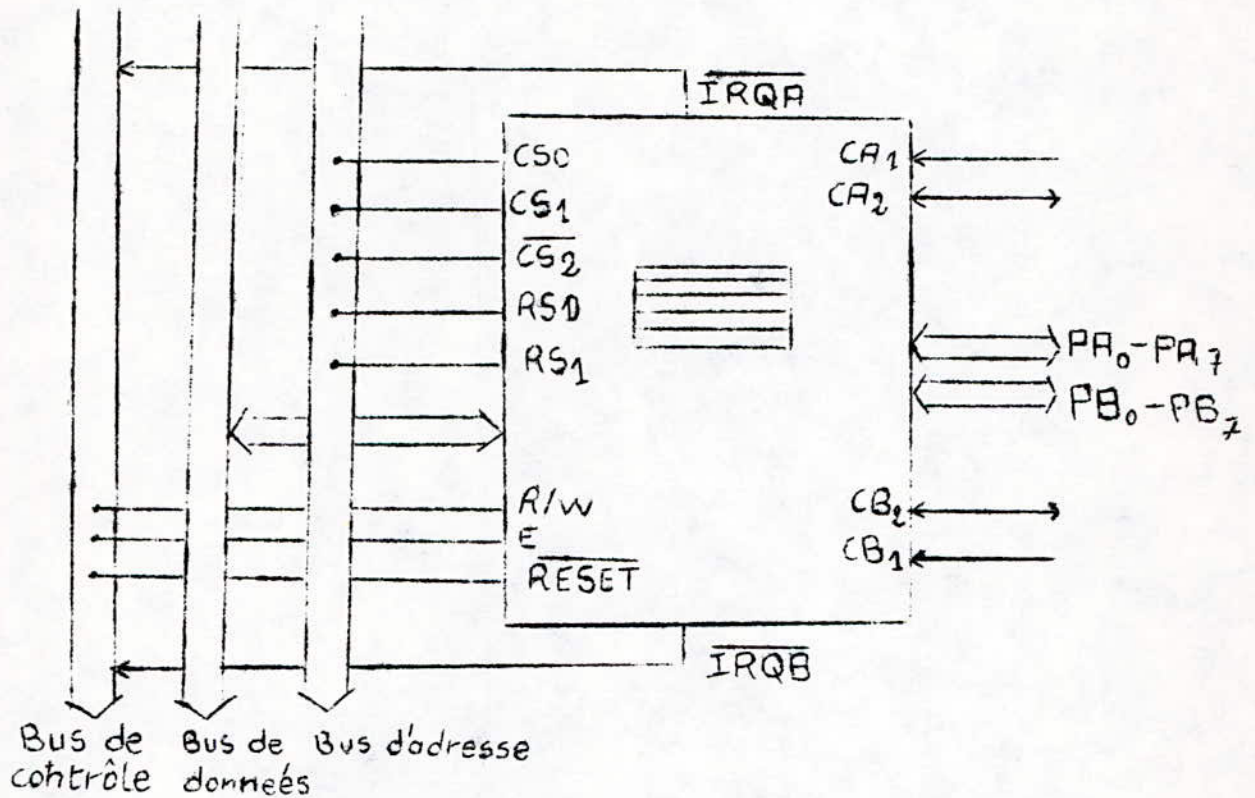
- DRA ou DRB servent à recevoir la donnée du 6800 vers la périphérie ou inversement.

DDRA ou DDRB permettent de définir le bus de transfert des données. A chaque bit de ces registres est associée une ligne (PA0 - PA7), (PB0 - PB7)

1 : définit une ligne en sortie du PIA

0 : définit une ligne en entrée du PIA

Enfin le registre CRA (contrôle registre) est à lecture - écriture sauf pour les bits 6 et 7 qui sont à lecture uniquement.



7	6	5 4 3	2	1 0
$\overline{IRQA_1}$	$\overline{IRQA_2}$	contrôle de	Accès	contrôle de
$(\overline{IRQB_1})$	$(\overline{IRQB_2})$	CA2 (CB2)	DDRA	CA1 (CB1)
			(DDRB)	

La signification des bits est la suivante :

- CRA7 - CRA6 : ces deux bits indiquent les interruptions associés respectivement à CA1 (CB1) et CA2 (CB2) : Ils sont mis à 1 suivant la programmation du mode de fonctionnement des lignes du registre de données correspondant.

- CRA5 - CRA3 (CRB5 - CRB3) servent à programmer le fonctionnement de CA2 (CB2).

. Contrôle de CA1 (CB1) : la valeur de CRA4 (CRB4) fixe la transition active de CA1 (CB1) ; la valeur de CRA0 (CRB0) valide ou non les interruptions vers le microprocesseur.

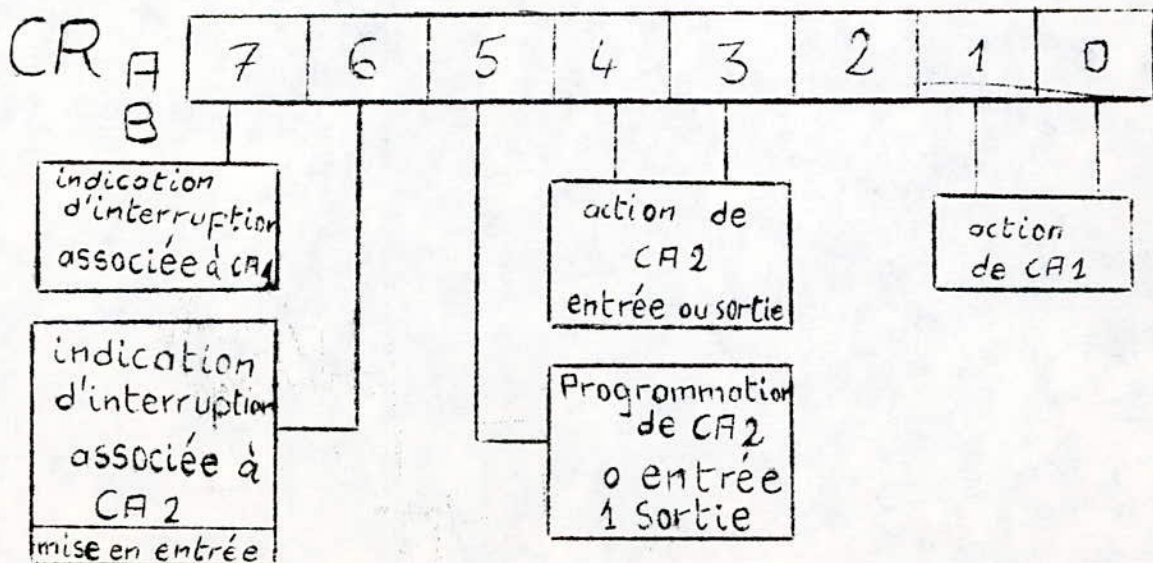
. Contrôle de CA2 (CB2) en sortie : CRA5 (CRB5) fixe CA2 (CB2) en sortie. CRA4 (CRB4) fixe le mode de fonctionnement de CA2 (CB2) mode impulsif ou contrôlé par programme.

SELECTION DES REGISTRES INTERNES :

Tous les registres du PIA étant à lecture - écriture, la selection interne par le microprocesseur ne se fait que par les lignes A0 - A1. Il est alors necessaire de disposer d'un autre bit pour selectionner 1 dans 6 registres du PIA. Ce bit est le bit 2 du CR du port considéré.

Lorsque CRA2 (CRB2) sera égal à 0 le microprocesseur accedera au registre DDRA (DDRB) et lorsque le bit est égal à 1 le microprocesseur accedera au registre DRA (DRB)

FORMAT DES REGISTRES DE COMMANDE CRA et CRB :





D'où le mode de fonctionnement (unique) du PLA peut être représenté par le tableau suivant :

$\overline{CS}_2$	$CS_1$	$CS_0$	$RS_1$	$RS_0$	$b_2$	R/w	type d'opération
0	1	1	0	0	0	0	sortie: bus de donnée → DDRA
0	1	1	0	0	1	0	sortie: Bus de donnée → DRA
0	1	1	0	0	1	1	entrée: DRA → Bus de donnée
0	1	1	0	1	x	0	sortie: Bus de donnée → CRA
0	1	1	0	1	x	1	entrée: CRA → Bus de donnée
0	1	1	1	0	0	0	Sortie: Bus de donnée → DDRB
0	1	1	1	0	1	0	Sortie: Bus de donnée → DRB
0	1	1	1	0	1	1	entrée: DRB → Bus de donnée
0	1	1	1	1	x	0	Sortie: Bus de donnée → CRB
0	1	1	1	1	x	1	entrée: CRB → Bus de donnée

as avons dit au début de cette brochure (conception mathématique) que  $f(x)$  dt croit quand la variance croit pour le cas où  $f$  est une fonction localement convexe et  $\int_C f(x) dt$  croit quand la variance décroît dans le cas où  $f$  est une fonction localement concave .

### Démonstration

F localement convexe C'est à dire sa dérivée seconde est positive  
 $x(t)$  est de la forme:  $x(t) = x_0 + w(t)$   
 $w(t)$  est une fonction aléatoire, de plus elle est supposée être stationnaire et ergodique . La probabilité pour que cette fonction ait lieu est alors:

$$P(y, t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$$

où  $\sigma$  est l'écart type et  $\mu$  est la moyenne

si  $\mu=0$  nous pouvons alors écrire  $w(t) = \gamma(0, \sigma)$

alors  $x(t) = x_0 + \gamma(0, \sigma)$

$$\int_0^T x(t) dt = \int_0^T [x_0 + \gamma(0, \sigma)] dt = \int_0^T x_0 dt + \int_0^T \gamma(0, \sigma) dt$$

$$= x_0 T + 0 = x_0 T$$

$$\int_0^T f(x(t)) dt = \int_0^T f(x_0 + \gamma(0, \sigma)) dt$$

décomposons  $f$  en série de Taylor

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x_0)}{2}(x-x_0)^2 + \dots$$

$$= f(x_0) + f'(x_0)\gamma(0, \sigma) + \frac{f''(x_0)}{2}\left(\frac{x-x_0}{2}\right)^2 + \dots$$

$$\text{Donc } \int_0^T f(x(t)) dt = \int_0^T f(x_0) dt + \int_0^T f'(x_0) \gamma(0, \sigma) dt +$$

$$+ \frac{f''(x_0)}{2} \int_0^T [\gamma(0, \sigma)]^2 dt + \dots$$

or  $\int_0^T \gamma(0, \sigma) dt = 0$ ,  $\int_0^T (\gamma(0, \sigma))^2 dt$  est proportionnelle à la variance et les autres termes tendent vers zéro

$$\text{implique } \int_0^T f(x(t)) dt = \int_0^T f(x_0) dt + \frac{f''(x_0)}{2} K \sigma^2$$

comme  $f''$  est positive et  $\int_0^T f(x_0) dt$  est une constante

alors on voit bien que  $\int_0^T f(x(t)) dt$  croit quand la variance croit



f localement concave

( la dérivée seconde négative )

de la même façon nous trouvons que :

$$\int_0^T f(x(t)) dt = \int_0^T f(x_0) dt + \frac{f''(x_0)}{2} K \sigma^2$$

comme  $f''$  est négative ceci implique que  
croît quand la variance décroît .

$$\int_0^T f(x(t)) dt$$

f localement convexe

( la dérivée seconde positive )

de la même façon nous trouvons que :

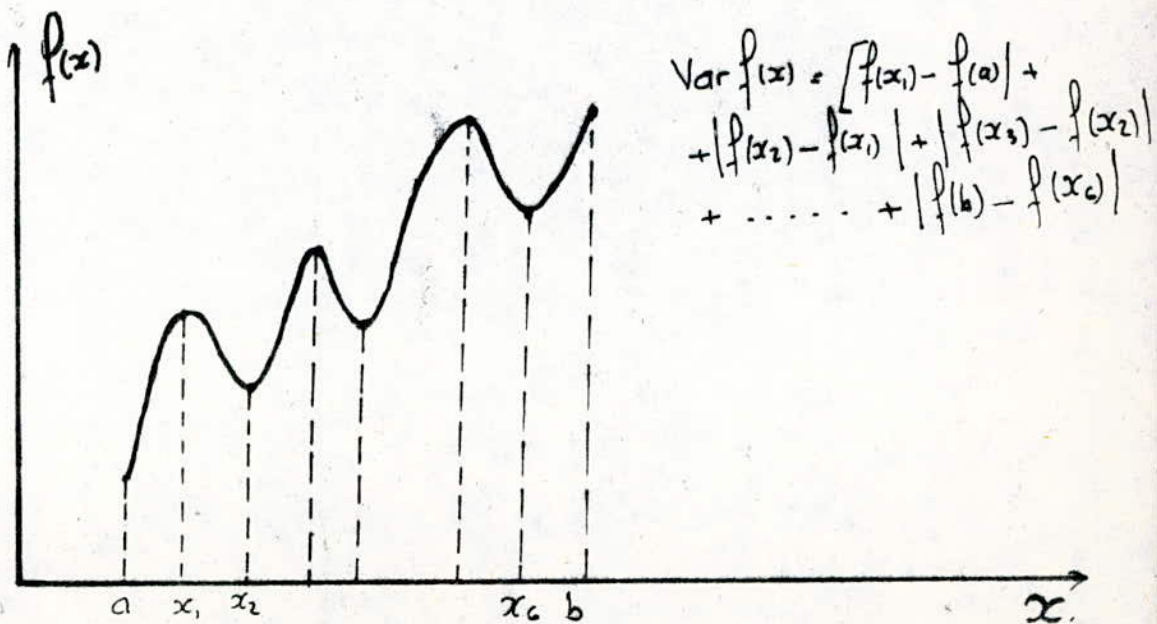
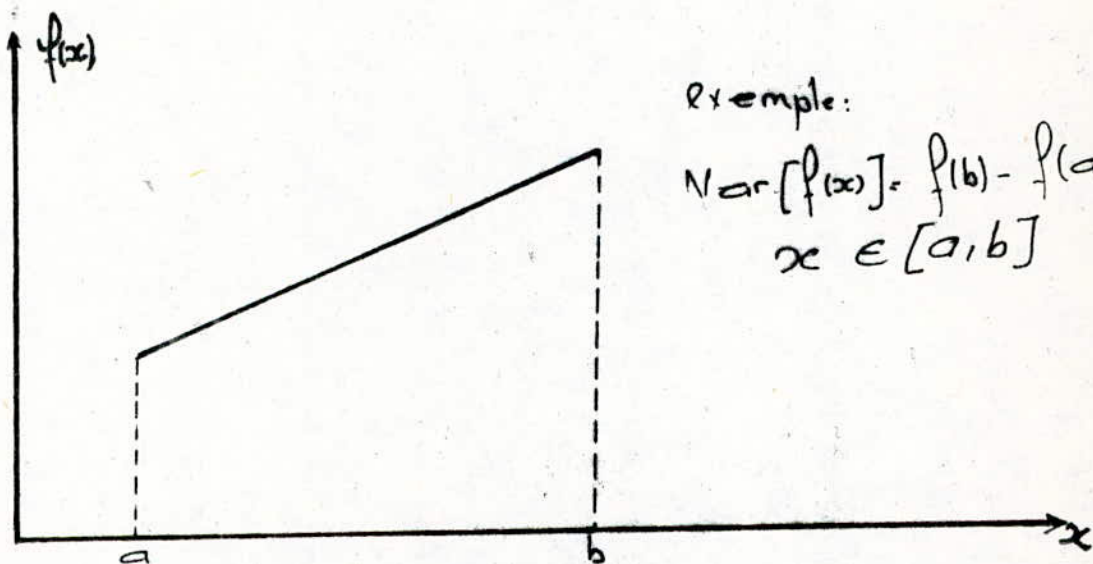
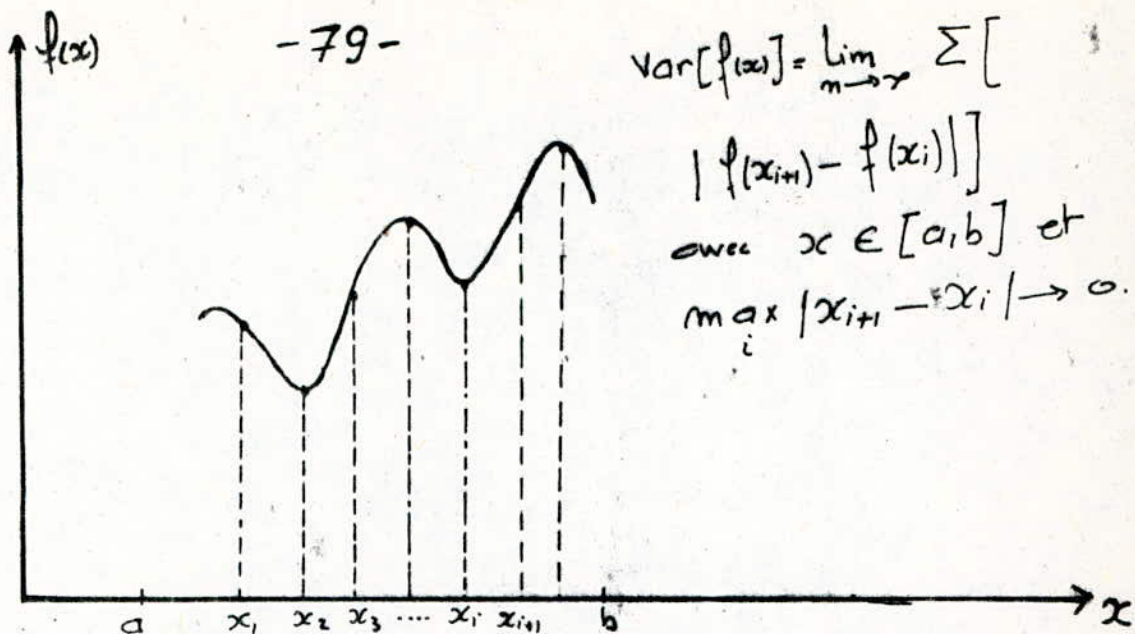
$$\int_0^T f(x(t)) dt = \int_0^T f(x_0) dt + \frac{f''(x_0)}{2} K \sigma^2$$

comme  $f''$  est positif, ceci implique que  
croît quand la variance croît .

$$\int_0^T f(x(t)) dt$$

f localement convexe

( la dérivée seconde positive )





## CONCLUSION

Dans notre modeste projet, nous avons essayé de contourner le problème de régulation et d'étudier ses différentes facettes, mais, vu le manque de documentation et la courte durée de ce semestre, il nous semble que malgré les efforts que nous avons déployés, il reste beaucoup à faire pour mener à fond cette étude. Ceci étant, car, comme nous l'avons remarqué, pour que la régulation soit vraiment efficace, elle nécessite des organes de concentration et de traitement de données particulièrement élaborés et facilement adaptables.

Nous avons essayé pour régler le phénomène "trafic" en entier de traiter simultanément deux composantes; car la nature même de ce phénomène nous impose la mise en place d'une régulation caractérisée par une structure hiérarchisée où cohabitent deux niveaux de décision:

-L'un au niveau élémentaire et individuel, donc au niveau micrographique, c'est la microrégulation

-L'autre au niveau global du groupe, de la zone ou de la ville c'est la macrorégulation

Les objectifs que nous avons essayé d'atteindre par cette régulation sont deux types contradictoires:

-L'un consiste à donner du feu vert sur un ou plusieurs itinéraires

-L'autre à minimiser le total des attentes aux feux rouges. Selon l'objectif choisi, les moyens et les stratégies à mettre en œuvre sont différentes. En réalité, on peut atteindre tous ces objectifs et on doit les atteindre, si l'on veut régler correctement la circulation d'une ville, mais on peut les atteindre successivement et non simultanément, car chaque objectif est plus approprié à un type de circulation qu'à un autre. Ainsi en cas de circulation dense (pres de la saturation) nous devons minimiser les temps d'attente globaux de façon à ne pas créer d'embouteillages; par contre dans les autres cas (densité faible ou moyenne) nous pouvons réaliser le premier objectif sur les principaux axes de la ville.

Signalons que la stratégie que nous mettons en place pour atteindre ces objectifs ne peut être réellement efficace que si elle est mise en oeuvre dans un laps de temps relativement court après que nous ayons décelé la tendance d'évolution de la commande.

Terminons cette conclusion en rappelant que, vu les avantages conçus par ordinateur (en coordination) nous avons tenté dans cette étude de donner des initialisations de coordination avec de simples algorithmes. Et afin d'achever cette étude de coordination qui est en soit même très vaste et très complexe mais dont l'efficacité de l'exploitation résoud le problème de circulation d'une façon vraiment satisfaisante, nous proposons aux étudiants qui nous suivront et qui seraient intéressés par ce sujet de continuer l'étude de cette coordination en améliorant le plus possible ces initialisations et les traduire en programmes de gestion général.



## BIBLIOGRAPHIE

- (1) WISMER D.A: Optimization methods for large scale, systems with applications ; department of engineering systems school of engineering and applied science univ of califarnia Los angeles 1971
- (2) Microprocesseur à l'usage des electroniciens: structure et fonctionnement ; JEAN PIERRE COCQUIEREZ et JEAN DEVARIS Eyrolles 1980
- (3) Ordinateurs en temps reel: Applications industrielles par J.P.NANCET
- (4) IIO études pratiques à thyristors et à triacs par R.M.MARSTON
- (4) Commande hiérarchisée et optimisation des processus complexes ; par ANDRE TITLI Dunod automatique
- ( 6) REvues : Electronique pratique et application  
Automatisme  
L'onde electrique  
Le haut parleur