50376
1985
43

N° d'ordre : 6

THÈSE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

pour obtenir le titre de

DOCTEUR

par

EL KOURSI EL MILOUDI

Maître es sciences

BIBLIOTHEQUE INTERUNIVERSITAIRE LILLE

UNE METHODE SURE DE DEVELOPPEMENT DES LOGICIELS DESTINES A LA COMMANDE DE PROCESSUS EN SECURITE :

APPLICATION A LA COMMANDE DE PORTES VEHICULES DU METRO VAL

Soutenue le 31 mai 1985 devant la Commission d'Examen

Membres du Jury :	MM.	GABILLARD,	Président
		STAROSWIECKI,	Rapporteur
		ANGUE,	Rapporteur
		DHALLUIN,	Examinateur
		BOURASSEAU,	Examinateur

D 030 143727 0

A mes Parents

Catherine

Meriem.

AVANT - PROPOS

Le présent travail a été effectué dans le laboratoire de Radiopropagation et Electronique dirigé par M. le Professeur GABILLARD. Je le remercie vivement pour l'aide et le soutien moral indispensable qu'il m'a toujours accordés malgré ses nombreuses préoccupations. Je lui témoigne ici ma plus profonde gratitude.

C'est une occasion pour moi, d'exprimer ma plus grande reconnaissance à M. JF. DHALLUIN (Ingénieur à l'Institut de Recherche des Transports). Ses idées et l'enthousiasme qu'il exprime furent indispensables à l'aboutissement de ce travail, fruit de nombreuses et fructueuses discussions entre nous.

Mes plus vifs remerciements vont également à M. le Professeur STAROSWIECKI à l'Université de LILLE I qui a bien voulu juger ce mémoire. Certains travaux effectués par l'équipe qu'il dirige m'ont été d'un précieux recours.

Monsieur le Professeur ANGUE de l'Université de Valenciennes a accepté de juger ce mémoire, qu'il trouve ici l'assurance de mes plus sincères remerciements.

Notre axe de recherche nous a conduit à quelques échanges avec la Société MATRA Transport à Lille. Je remercie son directeur Monsieur BOURASSEAU de l'attention qu'il porte à mes travaux. Sa participation au jury m'honore.

Je remercie tous les techniciens, ingénieurs et chercheurs, qui font que le laboratoire de radiopropagation et électronique soit un lieu d'entraide et d'amitié.

Enfin, je n'oublierai pas de citer Madame MEESMAECKER, qui a assuré la frappe de ce mémoire avec beaucoup de soins.

S O M M A I R E

INTRODUCTION

CHAPITRE I — Modélisation par réseau de Pétri interprété d'un automatisme logique..... P. 5

CHAPITRE II — Application des réseaux de Pétri à la description fonctionnelle de la commande-contrôle de processus discontinus..... P. 28

CHAPITRE III — Méthode d'implémentation des réseaux de Pétri interprétés sur microprocesseur..... P. 50

CHAPITRE IV — Conception d'un outil informatique d'aide au développement des programmes décrits fonctionnellement par un réseau de Pétri interprété..... P. 81

CONCLUSION ..... P. 116

ANNEXES ..... P. 119

BIBLIOGRAPHIE

INTRODUCTION

GENERALE

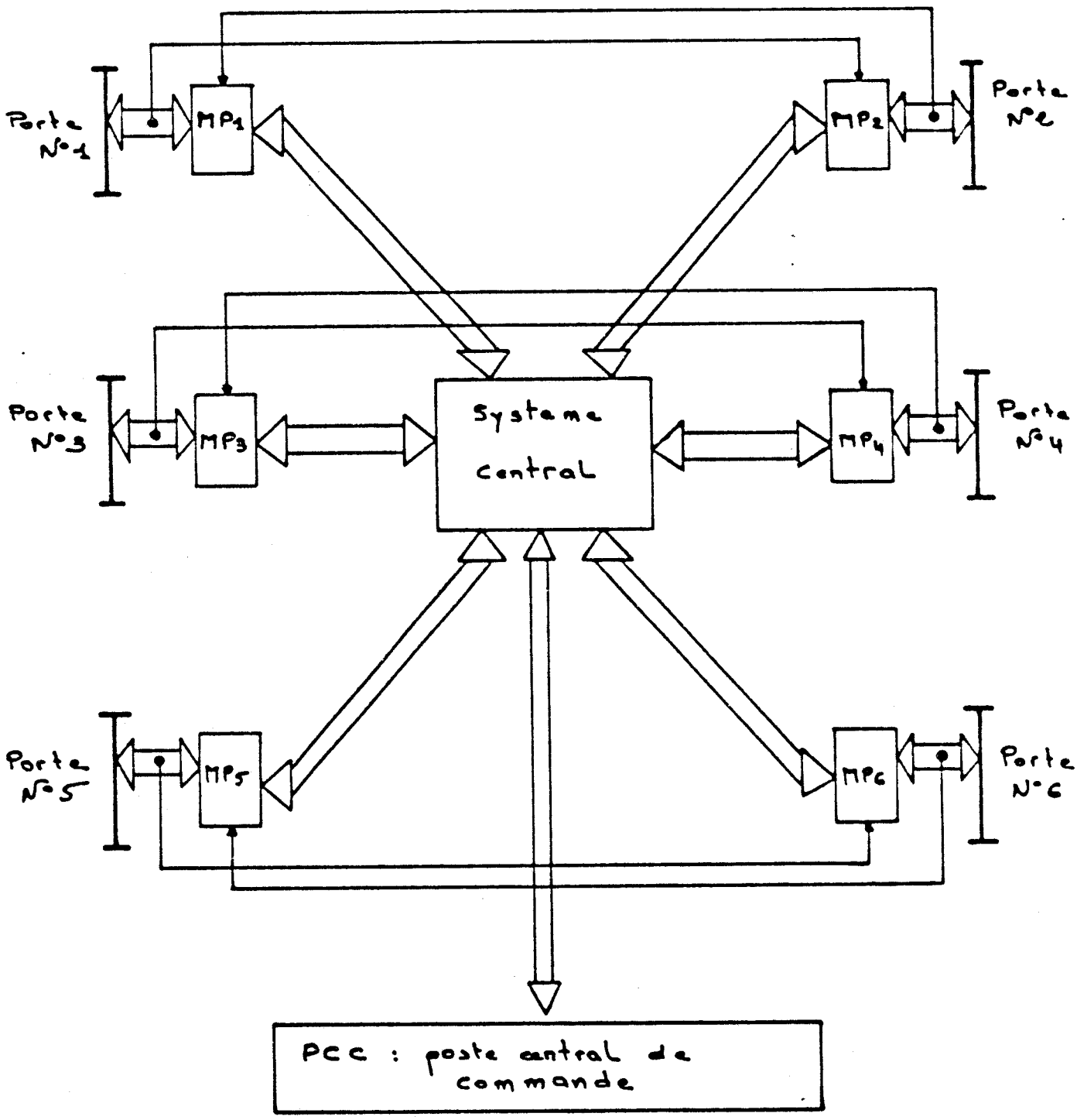
Le travail que nous avons entrepris au laboratoire de radio-propagation et électronique dirigé par le professeur GABILLARD, est la suite logique des recherches précédemment effectuées dans le domaine du transport automatisé. Le laboratoire a collaboré activement à la conception et à la réalisation du système de transport automatisé Lillois baptisé le "VAL".

C'est en s'attachant à résoudre les problèmes liés à la sécurité et à l'exploitation du métro "VAL", récemment mis en service, que le professeur GABILLARD s'est intéressé plus particulièrement à la commande-contrôle des mécanismes de portes véhicules dans les rames. En effet, l'absence du conducteur dans ce système de transport rattache en majeure partie la sécurité des voyageurs au bon fonctionnement des mécanismes portes. Il est donc nécessaire de pouvoir détecter, dès leurs apparitions, les anomalies et les pannes dangereuses ou non mettant le système dans un état d'insécurité latente. Pour permettre ceci, il faut assurer un suivi permanent de l'état de chaque mécanisme.

On s'est proposé de remplacer chaque platine à relais électro-mécanique par un dispositif à microproceseurs en sécurité. Ce dispositif est facilement modifiable et assure la même fonction que la platine. De plus, il effectue les tâches suivantes :

- Détection et localisation des pannes sur les portes afin d'apporter une aide appréciable à la maintenance.
- Génération des modes de fonctionnement dégradés issus de la détection des pannes allant vers une meilleure disponibilité.
- Etablissement d'un dialogue permanent avec un dispositif hiérarchiquement plus élevé, afin de recevoir les ordres à exécuter et rendre compte de l'état de la porte commandée.
- Surveillance de l'état sécuritaire (porte fermée, verrouillée et la demande d'évacuation d'urgence) de la porte opposée dans le véhicule.

Le choix d'un microprocesseur et l'électronique bâtie autour pour effectuer les tâches citées ci-dessus s'est inscrit dans la démarche qui consiste à établir un réseau microinformatique en étoile au sein d'un véhicule comportant six portes. Nous schématisons le réseau microinformatique par la figure 1.



Le travail qui nous a été confié dans ce projet consiste à réaliser la fonction de commande-contrôle en sécurité de chaque mécanisme de porte. En terme plus général, il s'agit de définir une méthode sûre de conception d'automatisme logique et numérique et de l'appliquer à notre procédé industriel. La recherche d'une méthode efficace se formule en deux points :

1°) Le choix d'un modèle de spécification et de description du cahier des charges. Ce choix dépend essentiellement de l'application. Cependant, il faut tenir compte des critères suivants : il faut avoir un modèle de description formel, d'une construction aisée facilement modifiable et capable de prendre en compte le parallélisme qui devient fréquent dans les automatismes. De plus, il faut que le modèle soit analysable afin d'éviter toute ambiguité et imprécision dans la conception.

Après une étude bibliographique des modèles existants, les réseaux de Pétri remplissent bien ces conditions, c'est ce que nous avons utilisé comme support à notre étude.

2°) Arrivé à l'implémentation du modèle sur un matériel donné, il faut pouvoir reproduire exactement le modèle construit et validé en gardant toute la rigueur du modèle. Pour notre application utilisant des microprocesseurs, nous avons défini une méthode microprogrammée permettant l'implantation du réseau de Pétri sur microprocesseur.

La thèse que nous présentons ici, s'articule autour des deux points formulés ci-dessus, elle se compose de quatre chapîtres :

Dans le premier chapître et après avoir rappelé la définition mathématique des réseaux de Pétri et leurs propriétés d'analyse, nous limitons notre étude aux réseaux de Pétri interprétés qui sont utilisés pour décrire les automatismes industriels et nous montrons leurs utilisations pour décrire les spécifications fontionnelles.

Nous entreprenons ensuite au deuxième chapître, une application des réseaux de Pétri interprétés à une fonction de commande-contrôle de processus (mécanisme de porte). Nous présentons d'abord comment le problème a été abordé en précisant les spécifications fonctionnelles et opérationnelles de l'automatisme. Ensuite, nous exposons la méthodologie de suivi de l'évolution du processus commandé et de détection des pannes. Nous décrivons à la fin de chapître les contraintes et les problèmes posés et comment nous avons pu les résoudre.

C'est dans le troisième chapître, que nous avons détaillé une méthode d'implémentation du réseau de Pétri interprété sauf sur un microprocesseur. Cette méthode est basée sur le traitement autour des places. Nous dresserons le bilan des avantages apportés par la méthode en faisant référence à notre automatisme.

Le quatrième chapître découle du troisième. En effet, la méthode exposée dans le troisième, qui permet la transformation graphe de Pétri/programme, est assez simple, rigoureuse et modulaire.
Il nous a apparu judicieux de créer un progiciel (produit logiciel) capable de générer à partir d'une description simple du réseau de Pétri, le langage assembleur spécifique à un microprocesseur. Cet outil d'aide à la conception est utilisé actuellement pour un microprocesseur 8751 d'INTEL. Nous présentons le matériel et logiciel mis en jeu et leurs utilisations en mettant l'accent sur les interactivités.

CHAPITRE I

CHAPITRE I : MODELISATION PAR RESEAU DE PETRI D'UN AUTOMATISME LOGIQUE

I - RESEAU DE PÉTRI, DÉFINITION ET NOTIONS FONDAMENTALES

I. 1 - DEFINITION ET REPRESENTATION GRAPHIQUE

I.1.1 - Définition

I.1.2 - Représentation graphique

I.1.21 - Succession dans un RDP

I.1.22 - Graphe d'état

I. 2 - CARACTERISATION DE LA DYNAMIQUE DES RDP

I.2.1 - Evolution du marquage

I.2.11 - Règle de franchissement des transitions

I.2.12 - Notion de conflit

I.2.13 - Marquages accessibles

I.2.2 - Propriétés associées aux marquages

I.2.21 - RDP borné

I.2.22 - RDP sain ou sauf

I.2.23 - Vivacité d'un RDP

I.2.24 - RDP conforme

I. 3 - ANALYSE

.../

/...

II - REPRÉSENTATION DES SPECIFICATIONS FONCTIONNELLES D'UN AUTOMATISME LOGIQUE PAR UN RÉSEAU DE PÉTRI INTERPRÉTÉ

II. 1 - RESEAU DE PETRI INTERPRETE

II.1.1 - Définition

II.1.2 - Représentation graphique

II.1.21 - Actions

II.1.22 - Réceptivitées

II.1.23 - Temps

II.1.3 - Règle d'évolution du marquage

II. 2 - ANALOGIE GRAFCET ET R D P I

CONCLUSION

INTRODUCTION

La demande croissante pour une automatisation complexe, prenant en compte l'aspect du parallélisme s'accompagne d'un besoin accru d'avoir des outils de spécification puissants, qui permettent de décrire et d'analyser de tels automatismes. Le réseau de Pétri et ses extensions sont les plus répandus et répondent finement à ce besoin.

Le réseau de Pétri - RDP - est utilisé pour spécifier, valider et implémenter tout système discret comprenant des évolutions parallèles. C'est un modèle qui offre une description du cahier des charges la plus proche de la physique. Les nombreuses méthodes d'analyse qui lui sont associées permettent de valider les spécifications. C'est un outil puissant et facilement modifiable.

Nous définissons d'abord le modèle de base et ses propriétés et nous nous intéressons ensuite à l'une des extensions utilisée dans le domaine des automatismes, le réseau de Pétri interprété - RDPI -. Le RDPI permet d'associer au modèle une interprétation relative aux variables de l'automatisme à concevoir.

I - DEFINITION ET NOTIONS FONDAMENTALES

I. 1 - Définition et représentation graphique

I. 1. 1 - Définition

Un réseau de Pétri (en abréviation un RDP) est un quintuplet RDP <P, T, Mo, Av, Ar> où :

- P est un ensemble fini de noeuds appelés places

$$P = \{P_1, P_2, \ldots\ldots\ldots\ldots, P_n\}$$

- T est un ensemble fini de noeuds disjoints de P appelés transitions

$$T = \{t_1, t_2, \ldots\ldots\ldots\ldots, t_n\}$$

- Mo est le marquage initial du RDP. C'est une fonction de P dans N (ensemble des entiers naturels) :

$$Mo : P \rightarrow N$$

$$p_i \rightarrow n_i$$

A chaque place p_i est attribué un nombre entier ou nul de jetons, appelés marques.

- Av est une application de P x T dans N, appelée incidence avant

$$Av : P \times T \rightarrow N$$

$$(p_i, t_j) \rightarrow n_i$$

- Ar est une application de P x T dans N, appelée incidence arrière

$$Ar : P \times T \rightarrow N$$

$$(p_i, t_j) \rightarrow n_i$$

A chaque couple (p_i, t_j) est associé un nombre entier ou nul.

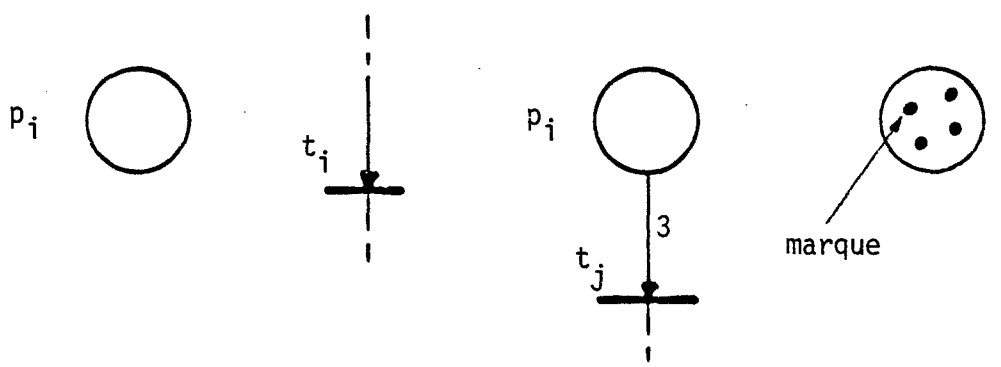
Les deux applications Av et Ar définissent le flot de marques (jetons) qui s'écoule entre les noeuds p et t |1|, |2|.

Définir un RDP sans marquage initial est analogue à définir une équation différentielle sans spécifier les variables initiales. Il est donc nécessaire d'associer à un RDP son marquage initial.

I. 1. 2 - Représentation graphique

Un réseau de Pétri est représenté par un graphe biparti orienté. Les noeuds du graphe sont les places et les transitions. L'incidence avant et l'incidence arrière sont représentées par des arcs orientés valués. A chaque arc (p_i-t_j) reliant la place p_i à la transition t_j et à (p_j,t_i) reliant la transition t_i à la place p_j, sont associés respectivement les nombres entiers Av (p_i,t_j) et Ar (p_j,t_i) appelés poids des arcs (Figure I-1c)

Une place est représentée par un cercle (Figure I-1a) et une transition par une barre (Figure I-1b). Le marquage est représenté par des points à l'intérieur des places (Figure I-1d).



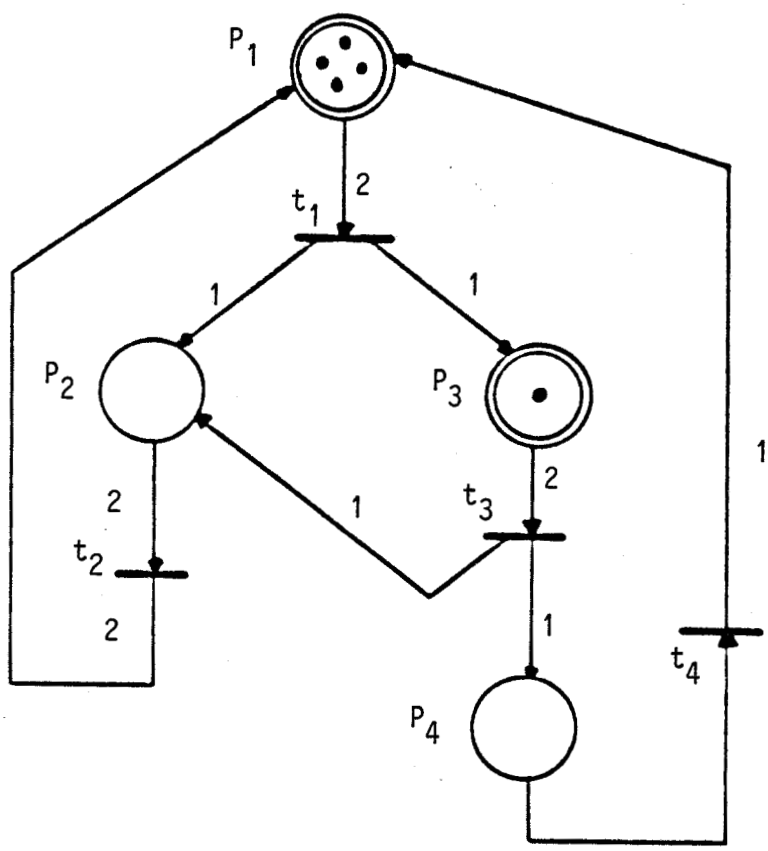
a) place p_i b) transition t_i c) arc valué à 3 d) place marquée

Figure I-1

Une place marquée initialement dans un RDP est généralement représentée par un double cercle. Le marquage initial M_o d'un RDP est représenté par un vecteur de dimension de P (ensemble des places). Il conditionne l'évolution du marquage dans les RDP (Fig. I-2).

$$Mo \left| \begin{array}{c} M(P_1) \\ M(P_2) \\ \vdots \\ M(P_n) \end{array} \right|$$

où $M(P_i)$ = Nombre de marques dans la place P_i



$$Mo \left| \begin{array}{l} 4 = M(P_1) \\ 0 = M(P_2) \\ 1 = M(P_3) \\ 0 = M(P_4) \end{array} \right.$$

Figure I-2

I. 1. 21 - Succession dans un RDP

Dans un réseau de Pétri, les places succèdent aux transitions et les transitions aux places. En respectant la même notation que dans la théorie des graphes, on note Γ l'application qui associe à un noeud ses successeurs (Γ^{-1} associe à un noeud ses prédécesseurs). Considérons le graphe de Pétri présenté Figure I-2, la succession des noeuds (places et transitions) dans ce graphe est présentée dans le tableau suivant :

Noeuds x	P_1	P_2	P_3	P_4	t_1	t_2	t_3	t_4
Successeurs $\Gamma(x)$	t_1	t_2	t_3	t_4	P_2P_3	P_1	P_2P_4	P_1
Prédécesseurs $\Gamma^{-1}(x)$	t_2t_4	t_1t_3	t_1	t_3	P_1	P_2	P_3	P_4

Dans un réseau de Pétri et comme nous le remarquons dans le tableau ci-dessus

$$\forall p \in P \quad : \quad \Gamma_{(p)} \subset T \quad , \quad \Gamma^{-1}{}_{(p)} \subset T$$

$$\forall t \in T \quad : \quad \Gamma_{(t)} \subset P \quad , \quad \Gamma^{-1}{}_{(t)} \subset P$$

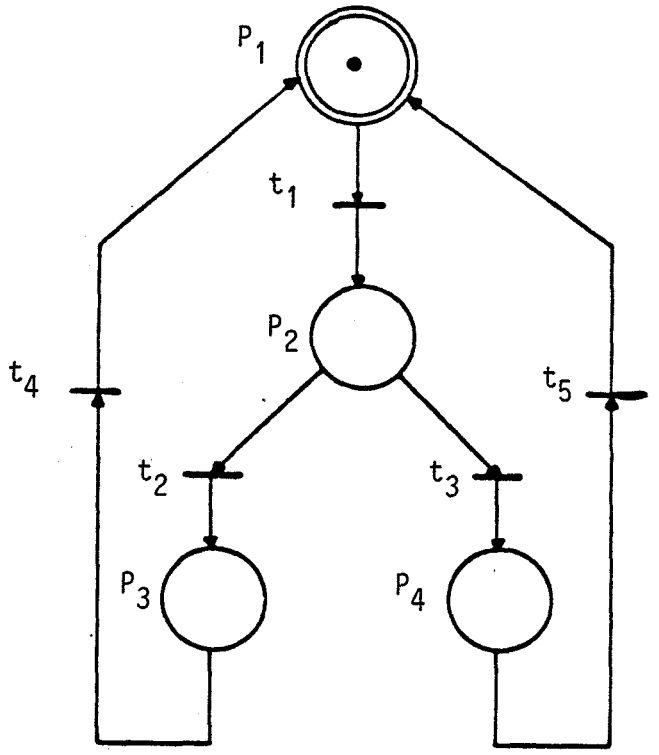
Une place ne peut succéder qu'à des transitions et vis-versa. Cette propriété doit être respectée lors de la construction d'un graphe de Pétri. Dans la suite on dira que :

$\forall p \in \Gamma_{(t)}$, p est <u>place de sortie</u> de la transition t et sera notée p_s.

$\forall p \in \Gamma^{-1}{}_{(t)}$, p est <u>place d'entrée</u> de la transition t et sera notée p_e.

I. 1. 22 - Graphe d'état

Un graphe d'état est un graphe de Pétri où chaque transition a au plus une place d'entrée P_e et une place de sortie P_s. Le graphe présenté Figure I-2 n'est pas un graphe d'état. Ci-dessous nous présentons un exemple de graphe d'état où le poids des arcs est à 1



$$Mo \begin{vmatrix} 1 \\ 0 \\ 0 \\ 0 \end{vmatrix}$$

Figure I-3

I. 2 - Caractérisation de la dynamique des réseaux de Pétri

I. 2. 1 - Evolution du marquage

Le marquage du réseau de Pétri correspond à son état dynamique. Il doit être susceptible d'évoluer. Ceci peut se faire par le franchissement des transitions.

I. 2. 11 - Règle de franchissement d'une transition

Une transition t est dite validée pour un marquage M si et seulement si pour toute place d'entrée P_e de Γ^{-1} (t) :

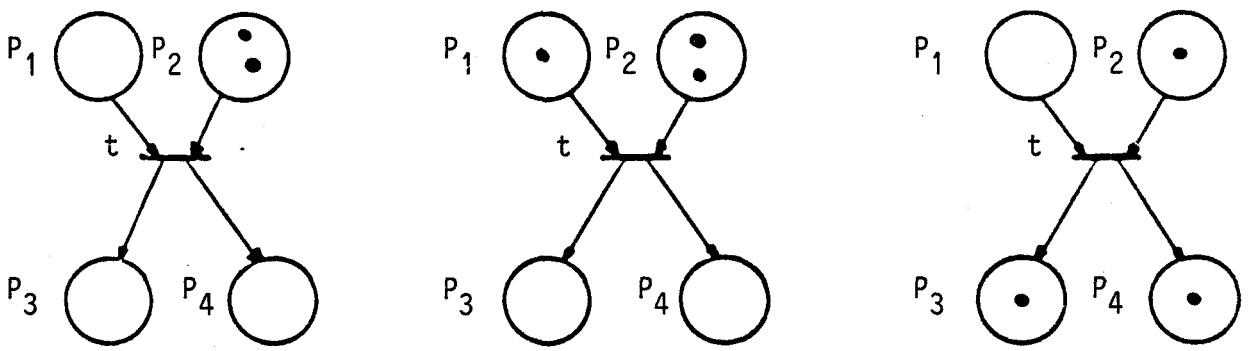
$M(P_e) \geqq Av\ (P_e, t)$ où $M(P_e)$ = Nbre de marques dans la place d'entrée P_e

Autrement dit, une transition t est validée (Fig. 4-b) si chaque place d'entrée P_e de la transition t possède un nombre de marqueurs n au moins égal au poids de l'arc ai reliant la place P_e à la transition t

poids de l'arc ai = $Av\ (P_e,t)$

Une transition ne peut être franchissable que si elle est validée. L'opération de franchissement d'une transition revient à enlever un nombre de marqueurs n = poids (ai) de chaque place d'entrée P_e de la transition et à ajouter un nombre de marqueurs q = poids (aj) dans chaque place de sortie P_s (Figure I-4)

poids de l'arc aj = $Ar\ (t, P_s)$



poids des arcs est égal à 1

a) t est non validée b) t est validée c) t est franchie

Figure I-4

Le franchissement d'une ou d'un ensemble de transitions validées simultanément peut augmenter ou diminuer le nombre de marqueurs dans un réseau de Pétri.

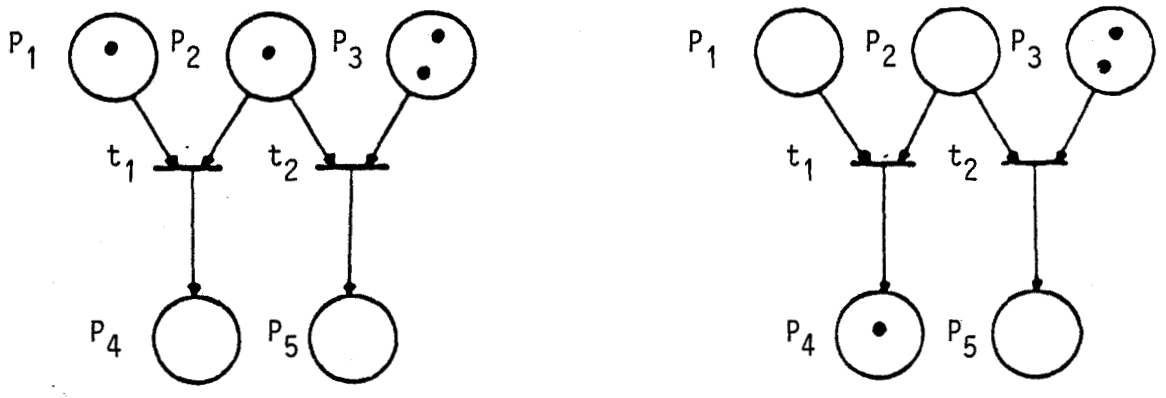
Remarque :

La définition de la règle de franchissement d'une transition est le point clé de l'évolution du marquage dans un réseau de Pétri. Nous verrons par la suite comment cette règle a été modifiée légèrement pour enrichir le modèle de base défini ci-dessus.

I. 2. 12 - Notion de conflit

Lorsqu'il existe une place d'entrée P_e, commune à deux ou plusieurs transitions validées simultanément (Fig.I-5), la règle énoncée ci-dessus ne peut pas s'appliquer. Ces transitions sont dites alors en conflits pour le marquage de P_e.

L'apparition d'un conflit pour un marquage M entraîne le blocage de l'évolution du marquage. Il est donc nécessaire de rendre une transition prioritaire par rapport aux autres. Le franchissement de cette transition modifie le marquage M et supprime la validation des autres transitions en conflits (Fig. I-5)



a) t_1 et t_2 sont en conflit

b) t_1 est prioritaire

Figure I-5

t_1 et t_2 sont en conflit, car elles ont la place P_2 en commun. Si t_1 est rendue prioritaire par rapport à t_2, le conflit est levé et c'est t_1 qui sera franchie.

I. 2. 13 - Marquage accessible

Un marquage M' est dit accessible à partir d'un marquage initial Mo, s'il existe une séquence finie de franchissement de transitions qui conduit le réseau de Pétri à l'état dynamique M'.

On note l'ensemble des marquages M' atteints à partir du marquage initial Mo par Emo. Cet ensemble dépend essentiellement de Mo.

I. 2. 2 - Propriétés associées aux marquages

Les propriétés que nous définissons ci-dessous, permettent d'analyser et de valider les réseaux de Pétri.

I. 2. 21 - RDP borné

Un réseau de Pétri est dit borné, si et seulement si aucune place ne contient un nombre de marqueurs (jetons) supérieur à une limite n max (où n max est un nombre entier positif) pour tous les marquages M' $\in$ Emo.

Un réseau de Pétri borné pour un marquage initial Mo donné peut ne pas l'être pour un autre marquage initial.

I. 2. 22 - RDP sain ou sauf

Un RDP est dit sain ou sauf pour un marquage initial Mo, si quelque soit le marquage obtenu à partir de Mo, aucune place ne possède plus d'un marqueur.

Autrement dit, c'est un RDP borné où n max = 1.

I. 2. 23 - Vivacité d'un RDP

RDP vivants :

Un RDP est dit vivant pour un marquage initial Mo, si et seulement si, toute transition t C T est validée et franchissable pour une séquence finie de franchissement des transitions.

Cette propriété caractérise le non blocage dans les réseaux de Pétri et montre que toute partie du RDP sera accessible à partir du marquage initial Mo.

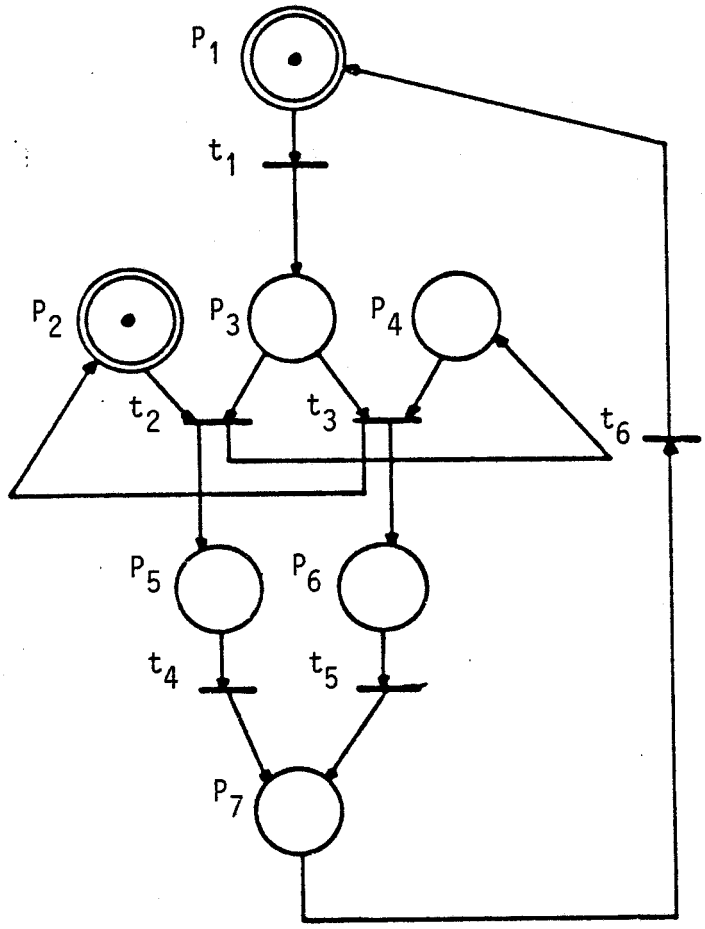
RDP propre :

Un RDP est dit propre pour un marquage initial Mo donné, si et seulement si, il existe une séquence de franchissement, telle que pour tout marquage Mi € Emo, cette séquence amène de Mi au marquage initial Mo.

Autrement dit, le RDP est réinitialisable.

I. 2. 24 - RDP conforme

Un RDP est dit conforme pour un marquage initial Mo, si et seulement si, il est borné et vivant. Contrairement à la Figure I-3, le RDP présenté ci-dessous est conforme, il est sain et vivant.



$$Mo \begin{vmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}$$

Figure I-6

I.3 - Analyse

L'analyse est l'étape qui suit la construction du modèle. Parmi les méthodes d'analyse étudiées jusqu'à nos jours, nous présentons les trois principales |1|, |18|.

a) Méthode par énumération des marquages

C'est une méthode classique basée sur la construction des marquages accessibles à partir du marquage initial et l'énumération de ses éléments. Chaque marquage atteint par le franchissement d'une ou plusieurs transitions est comparé au précédent. Selon le résultat de cette comparaison quelques propriétés peuvent être vérifiées (borné, vivant, propre...etc).

La difficulté de cet algorithme d'analyse réside dans le fait que lorsque les réseaux deviennent complexes : un nombre de places ou de transitions important ou/et un degré de parallélisme élevé. La construction des marquages accessibles devient impossible par l'explosion des combinaisons possibles. Toutefois, cette difficulté peut être atténuée par la deuxième méthode d'analyse.

b) Analyse après réduction

La réduction d'un réseau de Pétri, est obtenue en suivant des règles particulières telles que : suppression des places et transitions redondantes,... etc. Ces règles doivent conserver dans le réseau réduit, les propriétés du réseau d'origine.

c) Analyse structurelle

Cette méthode a pour principe de chercher les sous-ensembles de noeuds ou des sous-réseaux dont l'existence est significative pour la vérification des propriétés du réseau global.

Cette phase de la conception des automatismes logiques est très importante, car elle permet de prévoir la plupart des fautes de conception et, ainsi de les éviter.

Ces méthodes d'analyse s'appliquent aux réseaux de Pétri dits autonomes : l'évolution du marquage dans de tels réseaux, ne dépend d'aucune condition externe et n'obéit qu'à la règle énoncée ci-dessus. Elles prennent en compte le poids des arcs défini par les incidences Av et Ar. De ce fait, elles ont été accomplies par des systèmes informatiques assez puissants, basés au moins sur un microordinateur.

Toutefois, dans le domaine des automatismes logique et numérique, on se limite à l'emploi des RDP dans lesquels les incidences Avant (Av) et Arrière (Ar) ont leurs valeurs dans {0, 1} au lieu de N. Par conséquent, les modèles sont simples et peuvent être analysés à la main en combinant les trois méthodes résumées ci-dessus. Dans ce cas, le concepteur modifie son réseau jusqu'à l'obtention d'un réseau conforme, c'est-à-dire vivant, borné et même propre (réinitialisable).

Cependant, l'analyse n'est pas suffisante pour garantir que la spécification du système conçu soit correcte, le concepteur peut se trouver obligé de modifier la spécification au cours de la mise au point du système. Ceci conduit à des méthodes de conception descendante. De toute façon, un bon point de départ est qu'un réseau spécifiant la partie de commande soit bien formé.

Dans la suite, on ne parlera plus des poids des arcs. La présence d'un arc indiquera que l'incidence est de valeur 1, elle sera de 0 si l'arc n'existe pas.

II - REPRESENTATION DES SPECIFICATIONS FONCTIONNELLES D'UN AUTOMATISME LOGIQUE PAR UN RESEAU DE PETRI INTERPRETE

Après avoir rappelé la définition et les propriétés relatives au modèle de base le réseau de Pétri autonome (ordinaire), nous présentons ici l'utilisation de l'une des extensions du modèle dans le domaine de l'automatisation industrielle.

En effet, lorsqu'on étudie la conception d'un système automatisé, on représente généralement l'ensemble par deux parties qui coopèrent |3|, |4| : le procédé industriel à commander et la partie de commande qu'on est amené à réaliser Fig. I-7

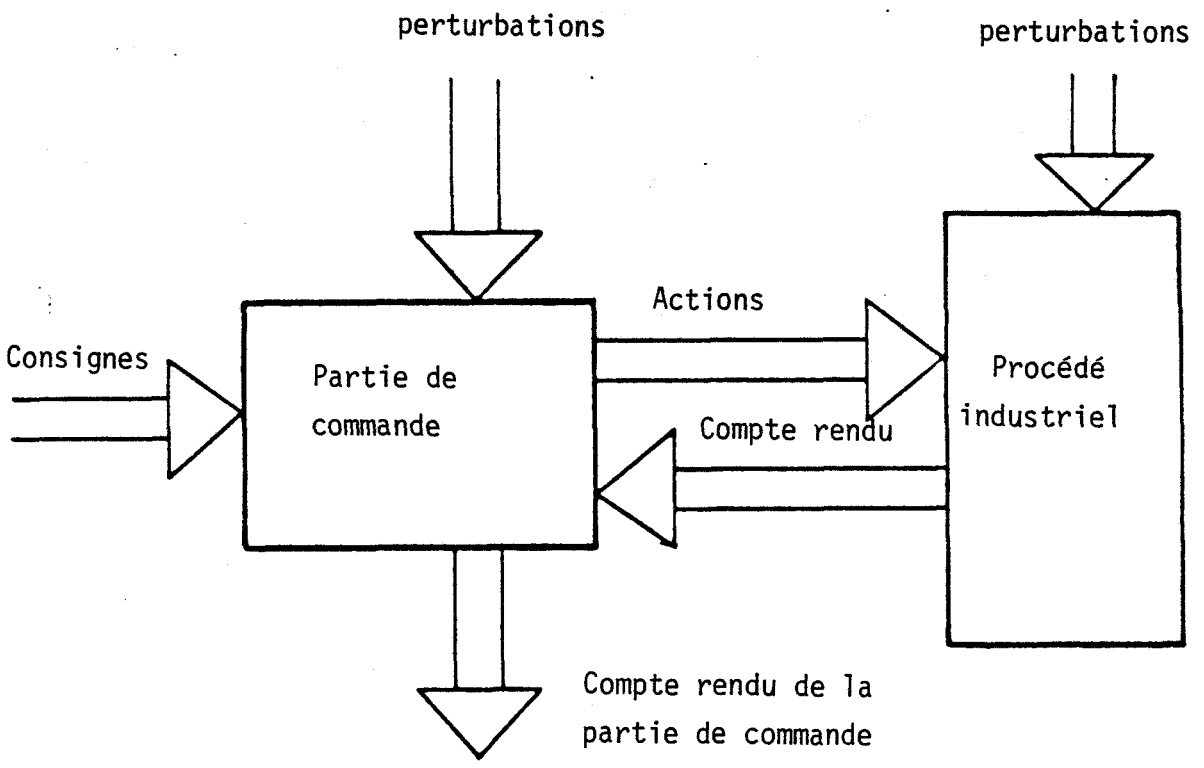


Figure I-7

Il est donc nécessaire d'avoir un modèle de description capable de traduire le lien entre la partie de commande et le procédé commandé, et l'influence de l'environnement. Le réseau de Pétri que nous allons définir et dont nous présenterons les caractéristiques permet bien celà. C'est ce modèle que nous avons utilisé pour notre travail, bien qu'existent d'autres modèles tel le GRAFCET (graphe de commande Etape - Transition) adapté à la description des automatismes, mais dont l'application reste limitée au système simple. Nous présentons donc brièvement à la fin de ce chapître, les analogies entre ce dernier et le réseau de Pétri interprété.

II. 1 - Réseau de Pétri interprété

II. 1. 1 - Définition

Un réseau de Pétri interprété (en abréviation un RDPI) est un RDP ordinaire (défini ci-dessus), auquel on a associé une interprétation. Il est défini par :

- un RDP ordinaire < P, T, Mo, Av, Ar >

- et une interprétation < Actions, réceptivités >

L'interprétation traduit le lien entre la partie de commande, modélisée par un RDPI, et son environnement (processus, consignes,... etc). Cette liaison a deux sens différents :

- De la partie de commande vers l'environnement par la commande des actions (commande de verrin pneumatique, allumage des voyants de signalisation, message... etc).

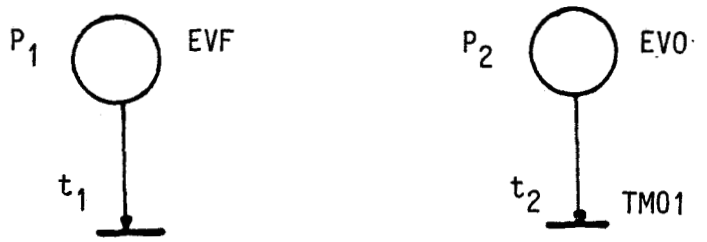
- De l'environnement vers la partie de commande par des conditions logiques appelées réceptivités. Celles-ci sont fonction des informations reçues par les parties de commande |2|, |4|.

II. 1. 2 - Représentation graphique

Un réseau de Pétri interprété est représenté par un graphe sur lequel on a porté les interprétations.

II. 1. 21 - Représentation des actions

Les actions correspondent à des fonctions à réaliser par la partie de commande. Elles peuvent être associées aux places et aux transitions. Lorsqu'elles sont décrites autour des places, on les représente par leurs symboliques à droite de celles-ci (Fig. I.8.a). Elles sont activées, tant que les places auxquelles sont associées restent marquées. Autour des transitions, elles sont représentées à droite de celles-ci et elles sont activées lorsque les transitions sont franchies (Fig. I.8.b).



a) à P_1 est associée l'action symbolisée EVF

b) à t_2 est associée l'action symbolisée TM01

Figure I-8

On associe généralement les actions de types permanent (tout ou rien) aux places, et celles de types fugitif aux transitions.
Ces actions peuvent être conditionnées.

II. 2. 22 - Représentation des réceptivitées

Les réceptivitées permettent de synchroniser l'évolution du marquage sur la réalisation des conditions logiques, numériques ou temporelles. Elles ne sont associées qu'aux transitions, on les représente à gauche de celles-ci. Fig. I-9

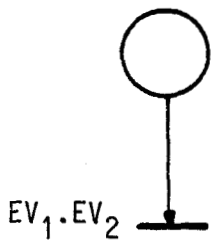
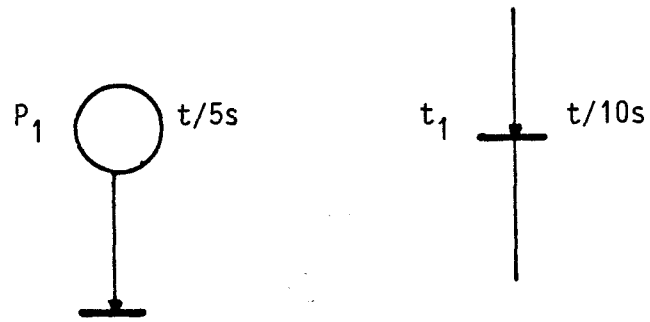


Figure I-9

Le franchissement de la transition t_1 ne se fait que si les évènements EV_1 et EV_2 sont réalisés simultanément.

II. 1. 23 - Représentation du temps dans un RDPI

Lorsque dans un système de commande, certaines actions ont des durées déterminées, on représente ces durées autour des places ou des transitions de la même façon que les actions (Fig. I-10). On parlera alors de places et transitions temporisées.



a) La place P_1 est temporisée à 5s

b) La transition t_1 est conditionnée pour une durée de 10s

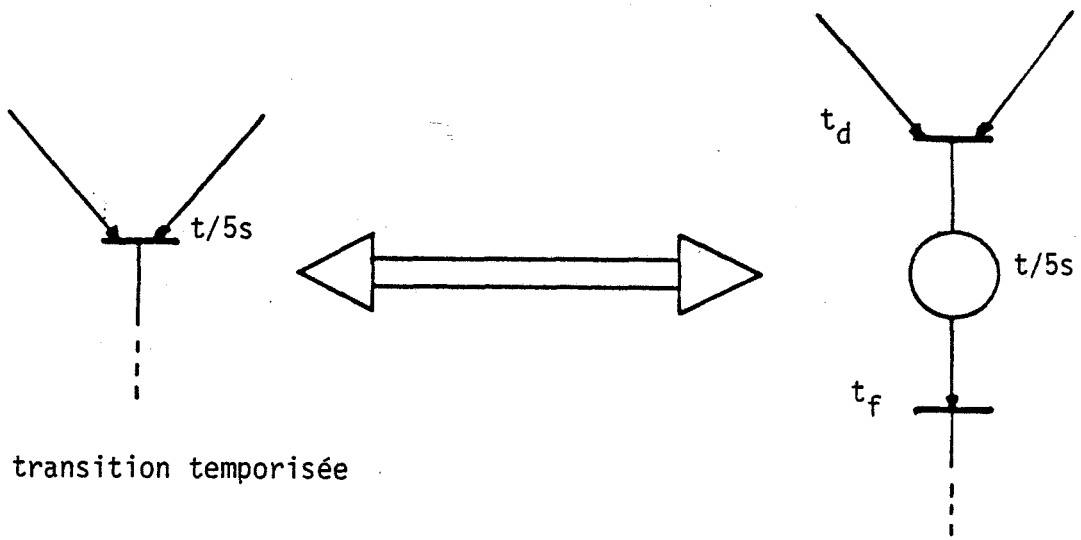
Figure I-10

Lorsqu'une marque arrive dans une place temporisée de durée T, cette marque est indisponible pendant une durée T. Dans de telles places, les marques ont deux états : disponible et non disponible.

Le franchissement d'une transition descendante à une place temporisée ne se fait que lorsque la marque dans la place est à l'état disponible.

Lorsqu'une transition temporisée à une durée T devient validée à l'instant t, elle ne peut être franchie qu'à l'instant t + T.

Dans un RDPI, on peut transformer une transition temporisée en une place temporisée et vice versa, comme l'indique la Figure 11.



t_d = transition de début
t_f = transition de fin
t_d et t_f, ne sont pas conditonnés

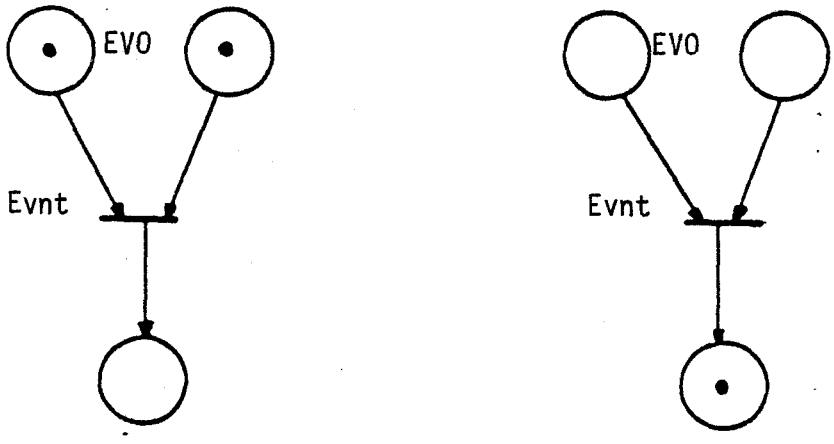
Figure I-11

Nous verrons par la suite, l'avantage des transitions temporisées par rapport aux places temporisées dans les systèmes de commande microprogrammés.

II. 1. 3 - Règle d'évolution du marquage dans un RDPI

L'évolution du marquage dans un RDPI, obéit à la règle suivante :

Lorsque toutes les places Pe $\in \Gamma^{-1}(t-)$ appelées places d'entrées de la transition t sont marquées, on dit que la transition t est réceptive à l'évènement, et si cet évènement est réalisé, on dit que la transition t est franchie. On enlève un marqueur (poids des arcs = 1) de chaque place d'entrée et on met un marqueur dans chaque place Pe $\in \Gamma$ (t) appelées place de sortie de la transition t.



a) t est validée
elle est réceptive à Evnt

b) t est franchie

Figure I-12

Les propriétés associées au marquage dans un RDP autonome définies ci-dessus, restent applicables aux RDPI.

II. 2 - Analogie GRAFCET et RDPI

Le GRAFCET (graphe de commande étape-transition) est un des outils de représentation, d'analyse et de synthèse de processus. C'est un outil qui a été présenté par la commission du groupe "Système logique" de l'AFCET , en vue de la normalisation de la représentation du cahier des charges des automatismes logiques |3|, |20|, |23|.

C'est un graphe orienté auquel est associée une interprétation comme les RDPI. Il est formé d'un ensemble fini de noeuds de deux types :

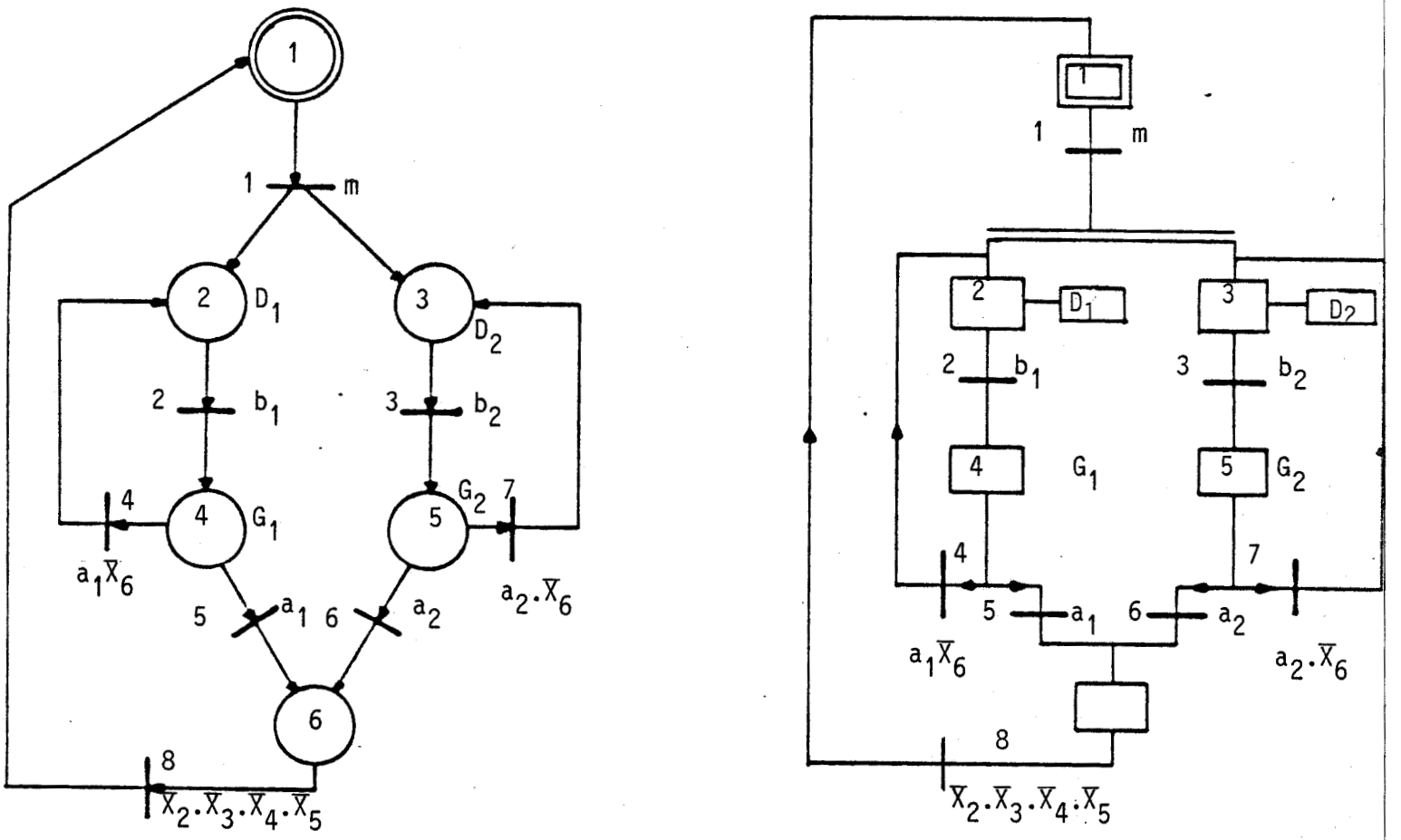
- Etapes (équivalent aux places dans le RDPI)

- Transitions.

Le GRAFCET peut être représenté de deux façons :

- Une représentation originale (Fig. I-13-a). Cette représentation est similaire à celle des RDPI.

- Une représentation normalisée. C'est une représentation spécifique aux GRAFCET (Fig. I-13-b). Les deux graphes ci-dessous décrivent le même processus.



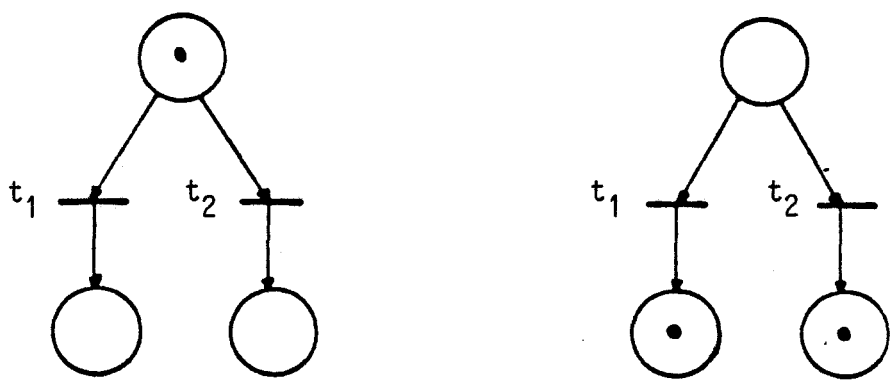
a) graphe original

b) graphe normalisé

Figure I-13

Ce modèle de description est considéré comme dérivé des RDP. Il est bien adapté à la description des automatismes logiques dont le parallèlisme est peu complexe. Il emprunte certaines propriétés de représentation et de gestion du marquage des RDPI. C'est un modèle qui se rapproche de la classe des RDPI sauf. Cependant, on peut dégager les différences essentielles suivantes |7|, |8|, |29| :

- La notion de conflit (I.2.12) dans un RDP ne se pose pas dans le GRAFCET. En effet, une étape active peut participer au franchissement de toutes les transitions descendantes validées simultanément. C'est la notion de la divisibilité d'un marqueur.



a) t_1 et t_2 sont validées simultanément

b) la marque est divisée en deux

Figure I-14

- Dans un RDP, les marques s'accumulent dans les places, alors que dans le GRAFCET, les marques qui arrivent dans la même étape se confondent.

- Dans un GRAFCET, une transition peut être conditionnée par la non activité d'une étape (Fig. I-15). Cette facilité dans la description permet une simplification d'écriture, mais alourdit toute analyse et vérification du modèle.

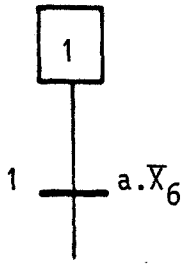


Figure I-15

La transition 1 est conditionnée par $\overline{X}_6$: la non activité de l'étape 6.

Il existe d'autres différences liées à la représentation qui ne sont pas fondamentales. Nous ne les présentons pas ici.

Le réseau de Pétri par rapport au GRAFCET, est bien connu au point de vue mathématique, et possède de nombreuses méthodes qui permettent de garantir sa conformité.

CONCLUSION

Le RDPI est un modèle puissant de description des automatismes logiques et numériques. Il permet :

- Par la représentation du parallélisme et la synchronisation du marquage par rapport à des réceptivités de visualiser l'évolution du processus, et d'avoir une représentation du cahier des charges simple, sans ambiguité et la plus proche de la physique.

- Une facilité de décomposition en sous réseau : une macro-action associée à une place peut être détaillée en décrivant la place par un sous réseau où à chaque place du sous réseau est associée une action élémentaire de la macro-action. Cette décomposition reste locale et ne nécessite aucune retouche au reste du réseau.

- Une possibilité d'analyse et de vérificiation des propriétés relatives au marquage qui permettent d'éviter certaines erreurs de conceptions.

CHAPITRE II

CHAPITRE II : APPLICATION DU R D P I A LA DESCRIPTION FONCTIONNELLE DE COMMANDE - CONTROLE DE PROCESSUS DISCONTINUS

I - COMMANDE ET CONTROLE D'UN PROCESSUS ELECTROMECANIQUE LENT : PORTE VEHICULE DU METRO VAL

I. 1 - DEFINITION DES OBJECTIFS

I. 2 - DESCRIPTION GLOBALE DU DISPOSITIF DE COMMANDE - SOLUTIONS TECHNIQUES UTILISEES

I.2.1 - Description et analyse des mécanismes de portes véhicules

a) description du mécanisme

b) analyse et reconfiguration du vecteur d'état et de commande

I.2.2 - Architecture du système de traitement décentralisé

a) Carte de commande microprogrammée

b) Réseau de connexion

c) Dispositif centralisateur des données "DCD"

.../

I I - DESCRIPTION DES TACHES DE COMMANDE ET DE CONTROLE DES PORTES PAR UN R D P I

II.1 - REPRESENTATION GRAPHIQUE DES TACHES DE COMMANDE, CONTROLE ET DE DIAGNOSTIC

II.1.1 - Détection de pannes par scrutation et analyse périodique du vecteur d'état de la porte dans son état statique

II.1.2 - Suivi séquentiel et temporel des changements d'état des capteurs pendant les manoeuvres de porte

II.1.3 - Problèmes soulevés par la commande de porte

a) Rebondissements électriques et mécaniques des capteurs

b) Gestion du dialogue avec le dispositif centralisateur de données

c) Réinitialisation en respectant le niveau de sécurité défini pour la porte.

I - COMMANDE ET CONTROLE D'UN PROCESSUS ELECTROMECANIQUE LENT : EXEMPLE D'APPLICATION : COMMANDE DES PORTES VEHICULES DU METRO VAL

I. 1 - DEFINITION DES OBJECTIFS

Nous abordons dans ce chapître une application des RDPI à la description et l'analyse de la fonction de commande-contrôle d'un ensemble de processus électromécaniques, en vue d'améliorer leur maintenabilité et leur disponibilité, tout en gardant un niveau de sécurité conforme aux objectifs. Il s'agit des portes véhicules du métro lillois "VAL".
Ce moyen de transport entièrement automatisé, utilise un système de commande de portes véhicule formé d'un matériel électromécanique classique à relais encombrant et d'une souplesse limitée. C'est un ensemble de platine à relais commandant chacune un mécanisme de porte. Photo n° 1.

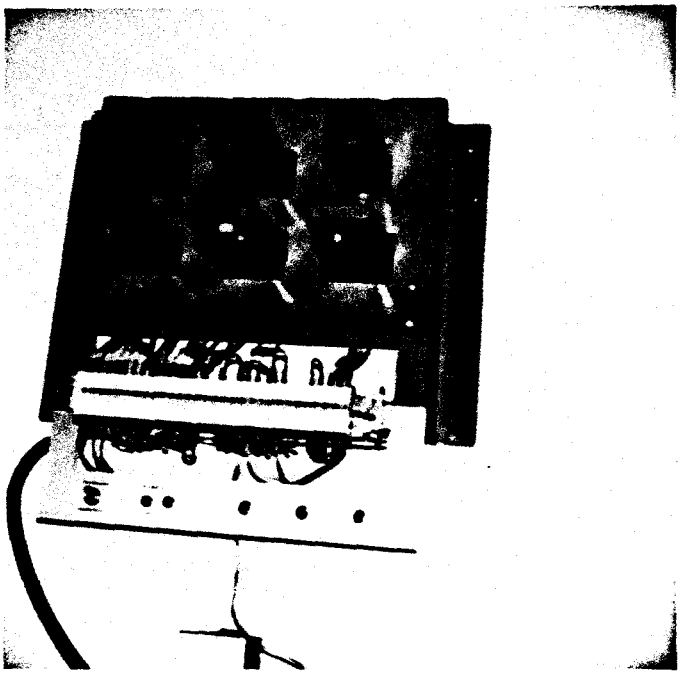


Photo n° 1 : platine de commande

Chaque platine placée à proximité d'une porte réalise les fonctions simples suivantes :

- Commande d'ouverture d'une porte
- Commande de fermeture
- Réouverture sur obstacle pendant la fermeture

Elle n'assure aucun contrôle sur les organes de commande |électro-valves| et de mesure |contacts électriques de fin de course|. Cette lacune dans chaque élément de commande est très pénalisante pour l'exploitation de l'ensemble du système de transport. En effet, une panne simple et non dangereuse, tel un collage d'un contact en position ouverte ou une détection permanente d'obstacles fictifs à la fermeture, immobilise la rame en station et par conséquent l'arrêt d'un tronçon de la ligne d'exploitation. L'objectif fixé par le laboratoire de radiopropagation et électronique sous la responsabilité du professeur GABILLARD vise à se protéger contre ces types d'incidents en associant à la fonction de commande de chaque mécanisme de porte une tâche de contrôle permanente des différents organes de porte. La solution proposée consiste à effectuer le contrôle des différents organes par l'analyse du vecteur d'état |l'ensemble des états des capteurs|. A l'issu de cette analyse, les différentes pannes sur les capteurs actionneurs et même les durs mécaniques dûs à l'usure sont détectées, localisées et signalées au dispositif de conduite automatique "DCA" : ce dispositif électronique est chargé de recueillir les informations de sécurité et d'exploitation relatives à l'ensemble de la rame et de les transmettre au poste central de commande non embarqué appelé "PCC".

On se propose de remplacer chaque platine de commande, par une structure de commande microprogrammée en sécurité, bâtie autour d'un micro-contrôleur INTEL 8751, moins encombrante, facilement modifiable et capable d'effectuer des séquences de commande sophistiquées et modifiables à volonté par retouche du logiciel. Les avantages sont multiples, dans le mesure où la conception d'une telle structure est bien faite que ce soit au niveau matériel |isolement galvanique, séparation des étages d'entrée, alimentations, etc...| On peut tirer au niveau de la commande-contrôle des portes véhicules par cette structure les trois améliorations essentielles suivantes :

- Aide à la maintenance

La structure, en effectuant une tâche de contrôle permanente, permet :

- La suppression des révisions périodiques des organes associés aux portes et déterminants pour la sécurité des voyageurs.
- La réduction du temps de remise en état des éléments défectueux et la simplicité de l'intervention de maintenance |la panne étant localisée, précisée et identifiée|.

- Amélioration de la disponibilité par tolérance aux pannes

En cataloguant toutes les pannes ayant des incidences sur l'exploitation, on peut envisager en présence de tel type de panne, de continuer l'exploitation momentanément par génération de mode de fonctionnement dégradé tout en respectant le niveau de sécurité défini pour les portes. Le dispositif de conduite automatique est régulièrement alerté de la présence du défaut.

- Traitement des informations de sécurité redondancé

On peut redondancer le traitement des informations de sécurité : chaque structure effectue en plus le contrôle de l'état sécuritaire de la porte opposée dans le véhicule |porte fermée verrouillée et demande d'évacuation d'urgence|. Elle peut ainsi signaler au "DCA" toute information relative à cet état de sécurité.

Ces améliorations ne peuvent être envisagées que si la structure à réaliser est en sécurité : la structure matérielle et logicielle doit être capable de détecter ses propres défauts de fonctionnement et, en cas d'avaries de se reconfigurer dans un état de manière à ne pas engendrer vers son environnement des ordres et des messages contraires à la sécurité.

La mise en sécurité de chaque structure de commande microprogrammée peut être décomposable en deux parties distinctes et complémentaires :

a) Etude de la sécurité fonctionnelle de la commande et du contrôle du processus traité à partir d'une structure matérielle en bon état de marche

Cette étude se situe à deux niveaux :

Au niveau descriptif. Il s'agit de définir et décrire des séquences de commande, de contrôle et de diagnostic permettant de détecter l'ensemble des pannes sur le processus, les voies de commande et sur les voies de mesure.

Au niveau du logiciel. Le logiciel traitant les tâches décrites au premier niveau doit être assez crédible. Il doit reproduire le plus fidèlement possible les descriptions initiales et être structuré et modulaire, facilitant ainsi les retouches et modifications éventuelles.

b) Etude de sécurité du fonctionnement du microcontrôleur en tant qu'unité de traitement de la fonction définie ci-dessus

La mise en sécurité de la structure matérielle et plus particulièrement l'unité de traitement s'appuie sur le principe de la sécurité positive | 9, 30, 41|. Nous rappelons que ce principe est basé sur le fait qu'une défaillance d'un composant de la structure ne doit pas baisser le niveau de sécurité défini pour l'application envisagée. Nous avons montré que les défaillances des voies de mesure et de commande sont couvertes par la première étude. Il reste donc à définir une démarche logique pour la mise en sécurité du microcontrôleur. La démarche suivie se résume en trois points :

- Choix d'une méthode d'observation du bon fonctionnement du microcontrôleur
- Réalisation d'une structure de contrôle du microcontrôleur
- Définition d'un état de sécurité que doit imposer la structure de contrôle en cas de détection d'anomalie dans le fonctionnement du microcontrôleur.

Pour notre application, un procédé |9, 31| de mise en sécurité basé sur l'observation temporelle du déroulement du programme d'application a été élaboré et adopté |32,36|.

I. 2 - DESCRIPTION GLOBALE DU DISPOSITIF DE COMMANDE - SOLUTIONS TECHNIQUES UTILISEES

I.2.1 - Description et analyse des mécanismes de portes

a) Description du mécanisme

Nous avons pu disposer au laboratoire d'un mécanisme de porte véhicule type VAL et de son système de commande à relais. Le mécanisme est constitué principalement de :

- Deux vantaux |menée et menante| et le système de conjugaison de leurs mouvements
- Un moteur pneumatique de commande d'ouverture/fermeture des vantaux
- Deux électrovalves, une qui assure le déverrouillage et l'ouverture, et une pour la fermeture notées respectivement EVO et EVF
- Sept capteurs installés, répartis sur les rails de guidage des vantaux comme le montre la figure II.1. Ces capteurs délivrent des informations relatives sur l'état du mécanisme.

b) Reconfiguration nécessaire et suffisante du vecteur d'état et du vecteur de commande

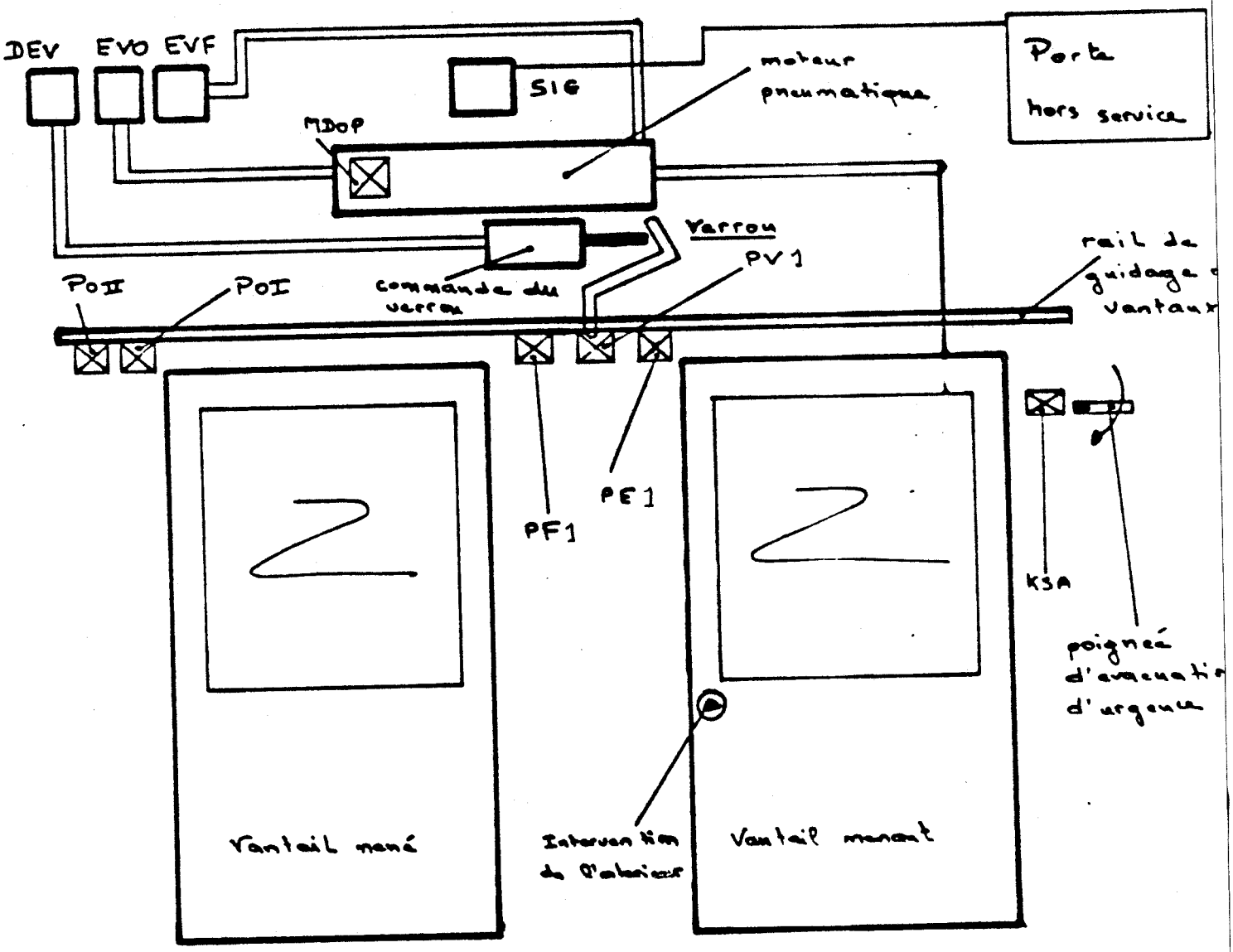
L'utilisation d'un microprocesseur pour une fonction de commande-contrôle d'un mécanisme de porte véhicule apporte une grande souplesse dans la commande qui ne doit pas être diminuée par une mauvaise configuration du vecteur d'état et de commande.

* Vecteur d'état

Le vecteur d'état a été reconfiguré | 9 | de façon à permettre une meilleure détection et localisation de pannes, et par conséquent un apport appréciable au niveau de la disponibilité.

La taille du vecteur d'état est restée la même mais deux modifications ont été apportées :

- L'information porte ouverte est redondancée, ce qui permet une meilleure descrimination en cas de défaut, et le maintien de l'exploitation en présence d'une panne sur l'un des deux capteurs |fig. II.1|.
- Le contact d'annulation de détection d'obstacle en fin de fermeture est supprimé. Cette fonction sera assurée par le contact verrou. PV1 |fig. II.1|.



VECTEUR DE COMMANDE

-EVO electrovalve d'ouverture

-EVF - - - - - de fermeture

-DEV - - - - - de deverrouillage

-SIG voyant de signalisation

VECTEUR D'ETAT

-PV1 contact porte verrouilleé

-PF1 - - - - - - - fermeé

-POI - - - - - - - ouverte n° 1

-POII - - - - - - - ouverte n° 2

-KSA contact securité alarme

-MDOP manostat de detection d'obstacle à la fermeture

-PE1 contact de deverrouillage exterieur

Fig. II.1

Configuration nécessaire et suffisante d'un mécanisme de porte

* Vecteur de commande

Le vecteur de commande est configuré de telle façon qu'une panne simple sur une voie de commande ou sur une électrovalve n'engendre pas une commande contraire à la sécurité |ouverture des portes en ligne|.

L'ouverture se fait maintenant à l'aide de deux voies indépendantes |deverrouillage, puis ouverture|.

La figure II.1 décrit la configuration du mécanisme de porte, qui a servi pour la suite de notre travail. Nous utilisons les symboliques associés aux différents capteurs et actionneurs citées dans la figure II.1.

I.2.2 - Architecture de traitement décentralisée

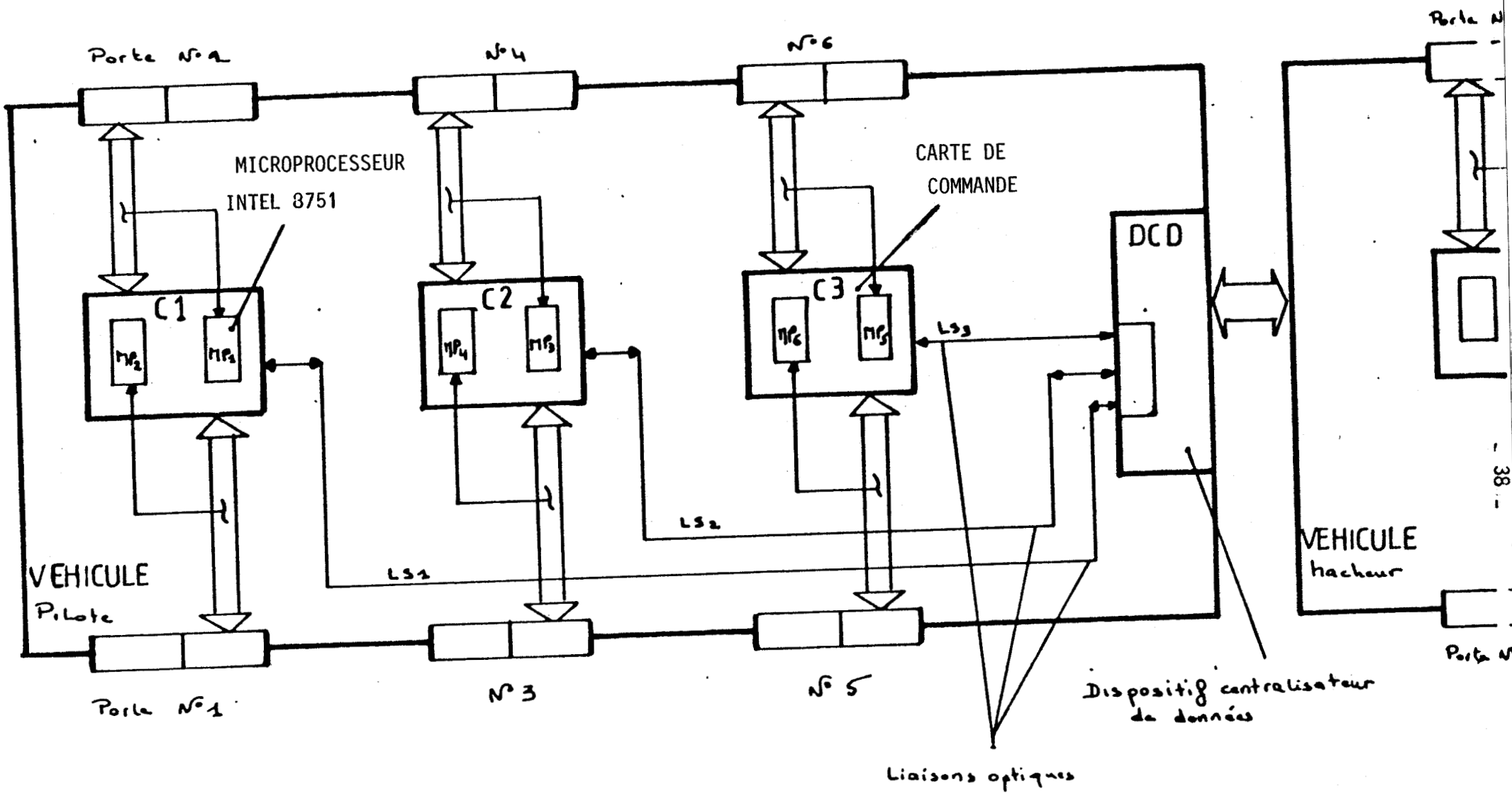
Avant d'aborder l'étude de la commande-contrôle en sécurité d'un mécanisme de porte véhicule, voyons d'abord la réalisation de la fonction dans sa globalité au sein d'un véhicule. Chaque véhicule d'une rame de métro comporte 6 mécanismes de porte |3 par face|. Vue la répartition spaciale des processus à commander, il est apparu judicieux de faire appel à un système de commande microprogrammée géographiquement réparti et modulaire |fig. II.2| où l'on a regroupé deux structures élémentaires de commande sur une même carte électronique permettant de réduire le cablage et l'encombrement.
Le système de commande dans un véhicule se compose alors de trois cartes de commandes, d'un système centralisateur de données "DCD" et du réseau d'interconnexion extensible. Chaque carte commande deux mécanismes de porte opposées dans le véhicule |fig. II.2|.

a) Carte de commande microprogrammée

Sur une carte de commande, photo n° 2, on retrouve les deux structures. Chaque structure est bâtie autour d'un microcontrôleur d'INTEL 8751. Ce microprocesseur possède sa mémoire EPROM de 4 K octet intégrée. Il assure une tâche de commande-contrôle d'une porte et la surveillance des informations de sécurité relatives à la porte opposée dans le véhicule afin d'assurer une bonne disponibilité |fig. II.2|.

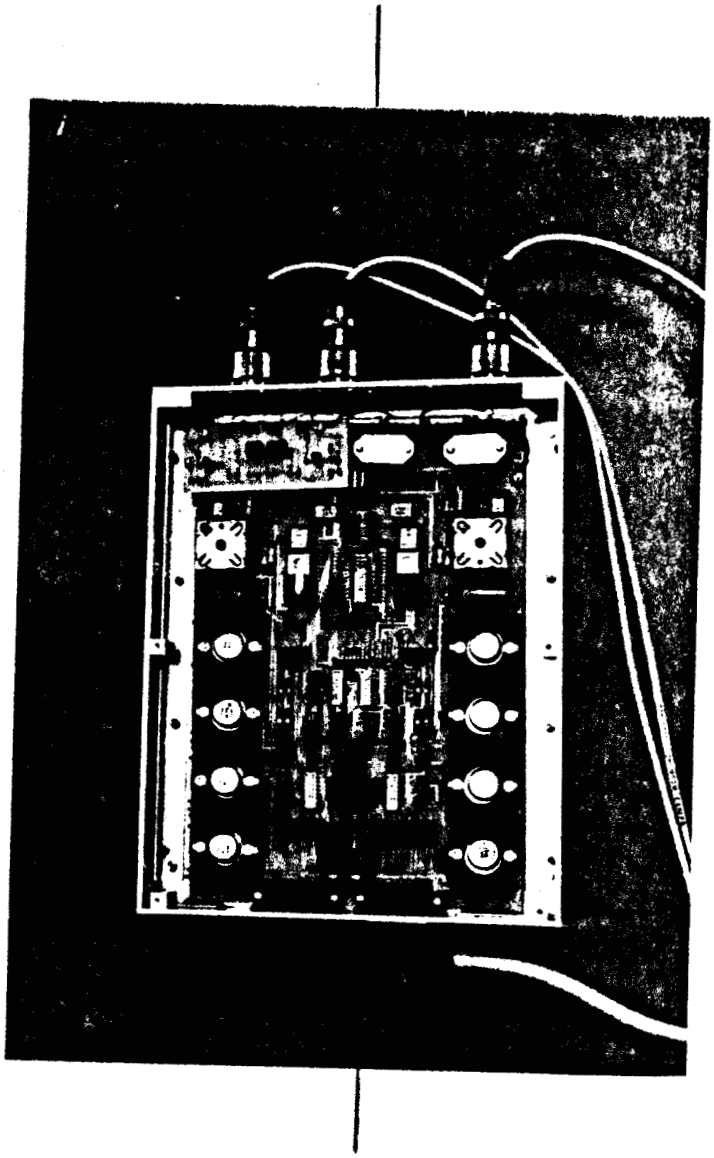
Caractéristiques du microcontrôleur utilisé

Le INTEL 8751 est classé dans la famille des microprocesseurs à 8 bit, sa principale particularité est le fait qu'il possède un processeur booléen lui



Architecture globale du dispositif de commande de porte d'un vehicule d'une rame de metro

Fig. II.2



Carte de commande

photo n° 2

permettant la manipulation de bit. Il assure des opérations arithmétiques, logiques, de transferts de données et les branchements conditionnel et inconditionnel. Il présente dans un seul boitier les caractéristiques suivantes :

- 4 K d'EPROM de mémoire
- 128 octet de RAM (adressable par bit)
- 2 timers 16 bits
- 1 port série programmable
- 5 interruptions (dont 2 externes)

L'ensemble des instructions dure 1 ou 2 µs pour un quartz de 12 MHz, et occupe 1, 2 ou 3 octet de mémoire. Ces deux données -occupation mémoire et durée d'exécution- nous ont permis l'évaluation de notre logiciel d'application. Nous présentons ceci au IIIème chapitre.

b) Réseau de connexion

L'ensemble des cartes est connecté à un dispositif centralisateur de données "DCD" par un réseau en étoile à voies point par point où chaque carte est en liaison directe avec le dispositif central. L'échange d'informations se fait par des liaisons de type asynchrone en technologie optique afin d'apporter une grande immunité aux bruits.

c) Dispositif centralisateur des données "DCD"

Ce dispositif peut être considéré comme faisant parti du "DCA". Il réalise une fonction importante d'orchestration du fonctionnement de l'ensemble des microprocesseurs. Ses trois principales fonctions sont les suivantes :

* Génération des ordres d'ouverture/fermeture des portes

Il joue le rôle d'interprête du DCA auprès des microprocesseurs. Lorsqu'il reçoit du DCA une télécommande synthétique telle que TCFPG "Télécommande de fermeture des portes gauches", celui-ci l'intrerprête après l'avoir reconnu et émet vers chaque microprocesseur commandant une porte la télécommande spécifique |dans notre cas, les microprocesseurs du côté gauche|.

* Scrutation périodique des microprocesseurs

Il réalise une fonction de scrutation périodique de l'ensemble des microprocesseurs afin de recueillir auprès de chacun, les télémesures concernant l'état de sa propre porte et l'état sécuritaire de la porte opposée dans le véhicule. Après le recueil de ces informations, il fait une synthèse des messages reçus et ne communique au DCA que les informations importantes. Cette scrutation périodique lui permet de détecter les coupures de liaison.

* Tracé historique des évènements

Il mémorise les télémesures relatives aux défauts fugitifs et permanents, parus sur chaque mécanisme de porte en vue d'une analyse plus fine par les agents de maintenance. Ces télémesures peuvent être appelées à tout moment par le DCA afin d'effectuer un traitement statistique des pannes qui permet le filtrage des pannes fugitives.

I I - DESCRIPTION DES TACHES DE COMMANDE ET DE CONTROLE DES PORTES PAR UN R D P I

II. 1 - REPRESENTATION GRAPHIQUE DES TACHES DE COMMANDE, CONTROLE ET DE DIAGNOSTIC

Nous abordons dans ce paragraphe deux méthodes de contrôle du fonctionnement du mécanisme de porte. Les séquences de commande de porte sont simples et peuvent être résumées en 3 phases |voir Annexe I|

- L'ouverture des vantaux s'effectue en deux temps : déverrouillage puis ouverture.
- Phase de fermeture avec réouverture sur obstacle et la mise en place d'un dispositif de non entraînement en fin de fermeture |Annexe I|.
- Etat porte fermée verrouillée et véhicule en ligne.

Nous remarquons que le mécanisme de porte placé dans son contexte se trouve dans deux états :

- Etat statique |vantaux immobile|
- Etat dynamique |vantaux mobile|.

Il est donc nécessaire de pouvoir déceler et localiser toutes les pannes cataloguées durant les deux états de fonctionnement du mécanisme.

II.1.1 - Détection de pannes par scrutation et analyse périodique du vecteur d'état de la porte dans son état statique

Le vecteur d'état de la porte et les informations de sécurité de la porte opposée dans le véhicule (fig. II.2) sont relevés périodiquement par le microprocesseur. Celui-ci scrute (fig. II.3) et contrôle le niveau logique de chaque capteur. Lorsqu'un capteur se trouve à un niveau logique différent à celui qu'il doit être après un traitement d'antirebond |ch.II.II.1.3|, le microprocesseur effectuera des tests d'hypothèses afin de déceler l'origine de la panne.

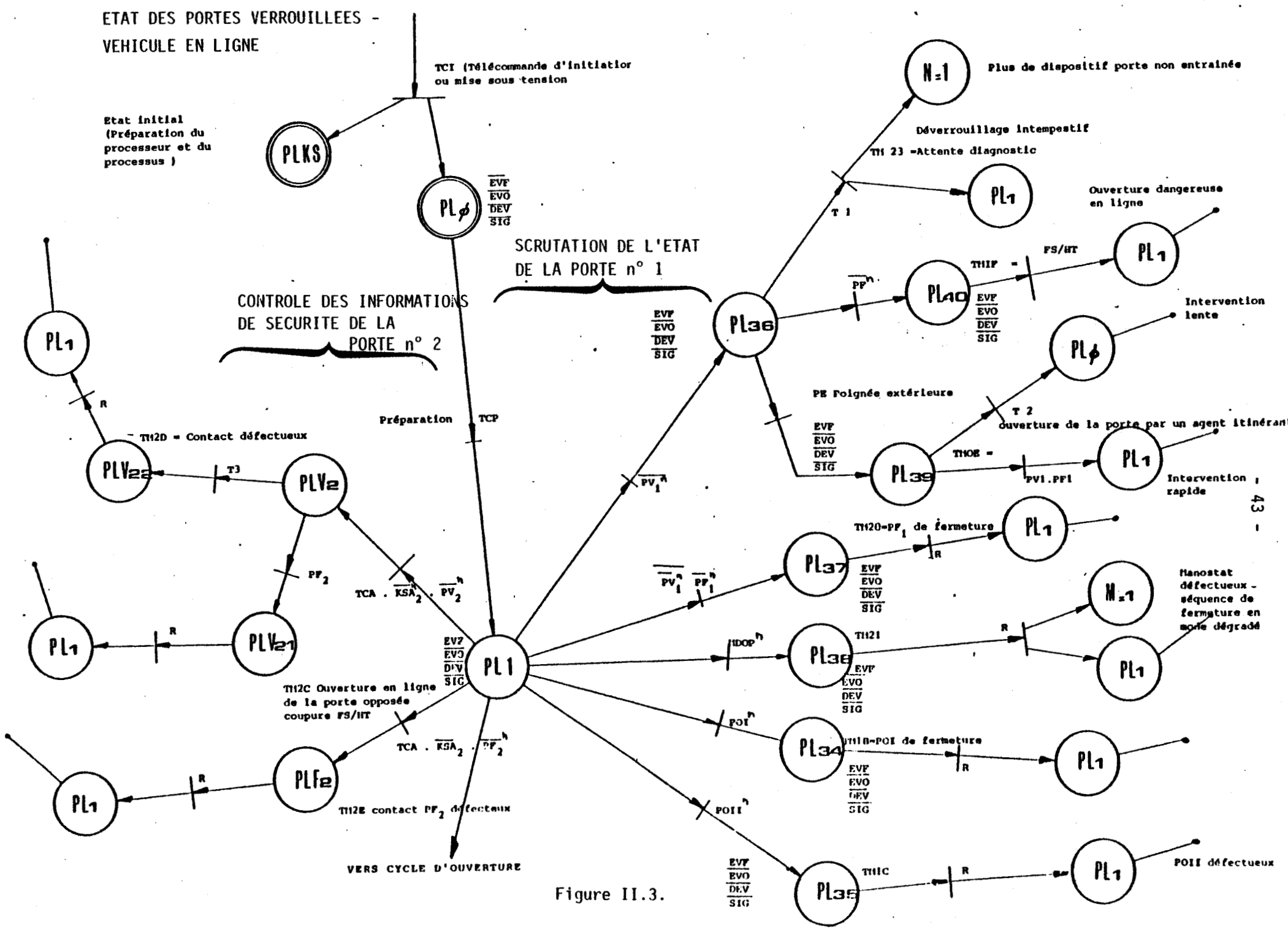

ETAT DES PORTES VERROUILLEES - VEHICULE EN LIGNE
TCI (Télécommande d'initiation ou mise sous tension
Etat initial (Préparation du processeur et du processus)
PLKS
PLφ
EVF EVO DEV SIG
SCRUTATION DE L'ETAT DE LA PORTE n° 1
CONTROLE DES INFORMATIONS DE SECURITE DE LA PORTE n° 2
Préparation
TCP
PL1
PLV22
PLV2
PLV21
PLF2
TM2D = Contact défectueux
T3
R
PF2
TCA . KSA2 . PV2
TM2C Ouverture en ligne de la porte opposée coupure FS/HT
TCA . KSA2 . PF2
TM2E contact PF2 défectueux
VERS CYCLE D'OUVERTURE
PV1
PL36
PL40
PL39
PL37
PL38
PL34
PL35
N=1
Plus de dispositif porte non entrainée
Déverrouillage intempestif
TM 23 =Attente diagnostic
T 1
Ouverture dangereuse en ligne
TM1F =
FS/HT
Intervention lente
PE Poignée extérieure
T 2
ouverture de la porte par un agent itinérant
TM0E =
PV1.PF1
Intervention rapide
TM20=PF1 de fermeture
Manostat défectueux - séquence de fermeture en mode dégradé
TM21
TM1B=POI de fermeture
TM1C
POII défectueux

Figure II.3.

Afin que la localisation de la panne se fasse sans ambiguité, ces hypothèses ont été prises en compte :

- Les pannes en mode commun sont écartées par une implantation judicieuse des composants
- Une seule panne apparaît à la fois.

Chaque défaut localisé est mémorisé pour être émis vers le dispositif extérieur sur sa demande .

II.1.2 - Suivi séquentiel et temporel des changements d'état des capteurs pendant les manoeuvres de porte

Par un suivi de la chronologie des changements d'état logique des capteurs pendant les phases d'ouverture et de fermeture de porte, le microprocesseur peut déceler les pannes sur le mécanisme de porte (capteur et effecteurs) et sur les étages d'entrée et de sortie de la carte de commande |fig. II.5|.

Pour que la localisation de pannes soit sans équivoque, on s'est limité aux pannes simples, c'est-à-dire qu'un élément soit défectueux à la fois car le diagnostic est basé sur le fonctionnement correct des autres capteurs. Ci-dessous nous présentons une partie du graphe, traitant le cas |fig. II.5| où la porte reste fermée mais déverrouillée après une commande d'ouverture en station. Lorsque la porte reste fermée, le capteur PF1 est fermé à l'état logique 5V. Trois causes à celà :

- Capteur PF1 défectueux et collé fermé
- Commande de l'ouverture défectueuse (EVO)
- Dur mécanique dû à l'usure ou à une pression faible dans la chambre d'ouverture.

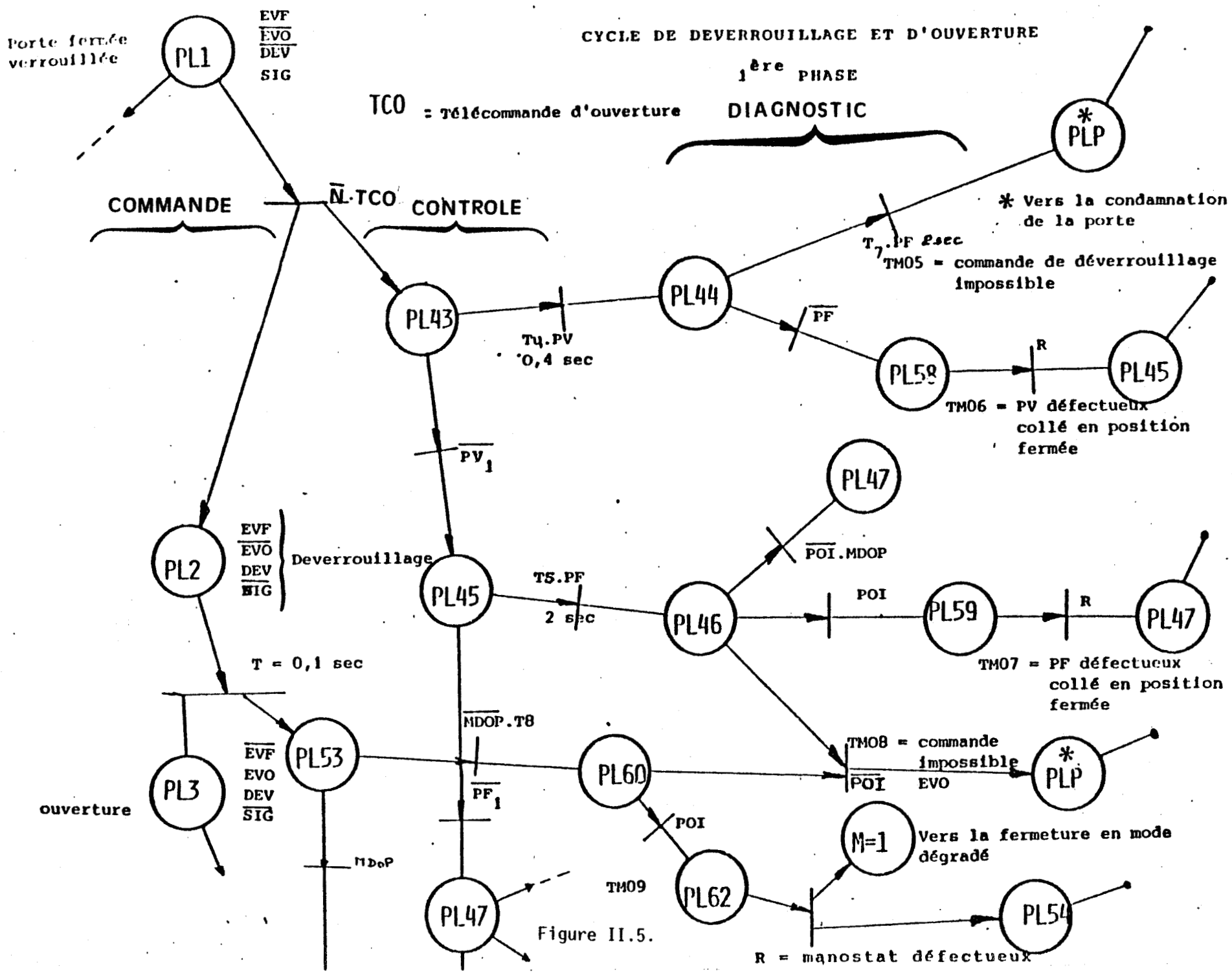

CYCLE DE DEVERROUILLAGE ET D'OUVERTURE
1ère PHASE
TCO : Télécommande d'ouverture
DIAGNOSTIC
COMMANDE
CONTROLE
Porte fermée verrouillée
PL1
EVF
EVO
DEV
SIG
N.TCO
PL43
Tq.PV
0,4 sec
PL44
T7.PF 2sec
TM05 = commande de déverrouillage impossible
PLP
* Vers la condamnation de la porte
PF
PL58
R
PL45
TM06 = PV défectueux collé en position fermée
PV1
PL47
POI.MDOP
PL2
Deverrouillage
PL45
T5.PF
2 sec
PL46
POI
PL59
R
PL47
TM07 = PF défectueux collé en position fermée
T = 0,1 sec
PL3
ouverture
PL53
MDOP.T8
PL60
TM08 = commande impossible
POI
EVO
PLP
PF1
POI
M=1
Vers la fermeture en mode dégradé
MDOP
PL47
TM09
PL62
PL54
R = manostat défectueux

Figure II.5.

Voyons la figure ci-dessus, lorsque le déverrouillage est effectué après réception de l'ordre d'ouverture. Les places PL2 et PL45 se trouvent marquées, au bout de 0,1 s la commande d'ouverture s'exécute (PL3) |marquage PL3, PL53, PL45|. Si le capteur PF1 reste à l'état logique (5V), la transition $\overline{PF1}$ n'est pas tirée, par contre la transition PF1 t/2 s est tirée au bout de 2 s. Trois cas se présentent :

1°) Le capteur POI à changé d'état logique alors la porte s'est bien ouverte, on déduit alors que le capteur est défectueux (PL59), le microprocesseur émet la télémesure "capteur PF1 collé en position fermée". On continue l'exploitation par retour à la place (PL47) en terminant le contrôle des autres capteurs.

2°) Si au même moment, le monostat MDOP ne change pas d'état au bout de 0,4 s |le MDOP est relié à la chambre d'ouverture et de ce fait le manque de pression dans la chambre d'ouverture se répercute sur le monostat| par manque ou insuffisance de pression.

Les marques se trouvent dans les places PL3, PL60, PL46 et si en plus le capteur POI ne change pas d'état on peut déduire que la commande d'ouverture ne s'est pas effectuée et la télémesure "commande impossible de l'électrovalve d'ouverture" est émise par le microprocesseur vers le "DCD". Dans ce cas on génère un fonctionnement dégradé par condamnation de la porte fermée verrouillée |AI.III|.

3°) Lorsque les capteurs PF1, POI ne changent pas d'état, et que le monostat MDOP se trouve en haute pression, alors on peut déduire que la commande d'ouverture s'effectue mais qu'il y a un dur mécanique à l'ouverture. L'exploitation continue par le marquage PL47 et la nature du dur mécanique est précisée dans la branche PL47 - PL49 |fig. A5-1|.

Ces deux techniques d'observation permettent de recouvrir toutes les pannes cataloguées sur le mécanisme, les capteurs, et sur les actionneurs |voies de mesure et de commande incluses|.

II.1.3 - Problèmes particuliers posés par la commande de porte

a) Rebondissements électriques et mécaniques des capteurs

Les capteurs installés sur les mécanismes de porte sont sujets à des rebondissements fréquents dus au déplacement de la rame et à l'usure, qui provoquent des changements intempestifs d'état logique à l'entrée des microprocesseurs. Pour remédier à ces rebondissements et aux parasites, l'état logique de chaque capteur est échantilloné un nombre n fois et n'est pris en compte que si il est stable et qu'il présente la même valeur pendant l'échantillonnage.

La prise d'une valeur d'échantillonnage est faite à intervalle de temps régulier et espacé de la durée de traitement d'un marquage |voir chap. III|. La description du traitement d'antirebonds se traduit dans le réseau de Pétri par une transition pondérée d'un nombre n (ex : fig. II.6).

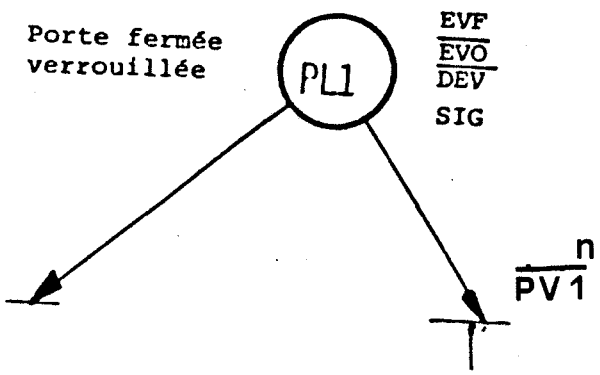


Figure II.6.

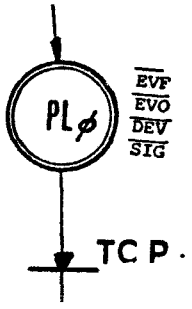
La transition ci-dessus ne peut être tirée que si l'état logique symbolisé par PV1 est maintenu à zéro durant les n valeurs d'échantillonnage.

b) Gestion du dialogue avec le dispositif centralisateur de données

Pour des raisons de sécurité liées à la gestion du réseau par le dispositif extérieur |10,13|, chaque microprocesseur reçoit les télécommandes spécifiques codées sur deux mots (16 bits), le premier est un mot de passe différent d'un microcontrôleur à l'autre, le deuxième est le mot utile de la télécommande.

Chaque télécommande reçue et reconnue est stockée afin d'être utilisée par la suite. Toutes les télécommandes sont sauvegardées de telle manière à ne garder en mémoire que la dernière arrivée.

Les télécommandes sont représentées autour des transitions.

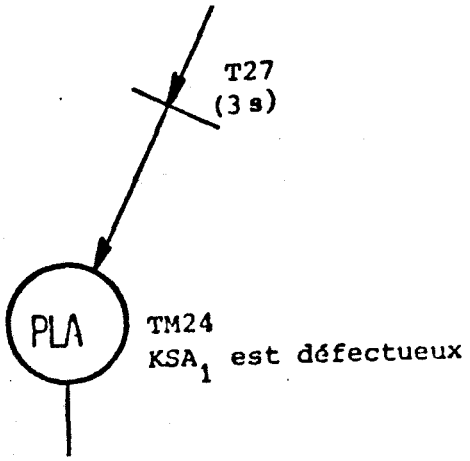


TCP = télécommande de préparation

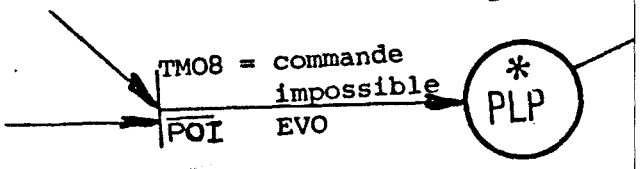
Figure II.7.

L'émission des télémesures se fait sur autorisation du dispositif. Lorsque le microprocesseur a une télémesure à émettre, il la stocke dans une pile dite de télémesures. Cette pile est vidée par la suite sur la demande du dispositif par une télécommande spécifique d'appel de télémesure "TCT".

Chaque télémesure représentée autour d'une place est automatiquement conditionnée, et ne sera émise qu'une seule fois quand la place à laquelle elle est associée reste marquée. Toutefois, nous avons adopté une représentation autour des transitions afin de réduire la taille du réseau, et n'émettre la télémesure que si la transition à laquelle elle est associée est franchie.



a) Description autour des places



b) Description autour des transitions

Figure II.8.

c) Réinitialisation de l'automatisme en respectant le niveau de sécurité défini pour la porte

Pour garantir la sécurité des voyageurs, il est indispensable de pouvoir arrêter l'évolution du processus instantanément et le figer dans un état dit de repos. Pour se faire, une télécommande dite d'initialisation a été réservée à cet effet ; lorsque le microcontrôleur reçoit cette télécommande spécifique, celui-ci force le marquage du RDP à l'état initial toutes commandes annulées. Nous décrivons cette procédure dans le troisième chapître.

CHAPITRE III

CHAPITRE III : METHODE D'IMPLEMENTATION DES RESEAUX DE PETRI INTERPRETES SUR MICROPROCESSEUR

I - DESCRIPTION ET GESTION DU PROGRAMME REPRESENTATIF DU R D P I

I. 1 - DESCRIPTION MEMOIRE DU R D P I

I.1.1 - Notion de place pivot et de synchronisme

I.1.2 - Recherche des places pivots

I.1.3 - Représentation et traitement d'un PAS

I.1.4 - Représentation du marquage

I.1.5 - Structure générale de représentation

I. 2 - PROGRAMME DE GESTION

I.2.1 - Evolution synchrone du marquage

I.2.2 - Gestion du marquage

I.2.3 - Cadencement des entrées et sorties

I.2.4 - Initialisation

I.2.5 - Programme moniteur

.../

/...

II - SOLUTIONS PARTICULIÈRES APPORTÉES AUX PROBLÈMES SPÉCIFIQUES DE COMMANDE-CONTRÔLE DE PROCESSUS ÉLECTROMÉCANIQUE

II. 1 - GESTION DES TELEMESURES ET TELECOMMANDES

II. 2 - METHODE DE TRAITEMENT DES TEMPORISATIONS

II. 3 - TRAITEMENT DES REBONDISSEMENTS ELECTRIQUES ET MECANIQUES DES CONTACTS

II. 4 - REINITIALISATION

III - PERFORMANCES

III. 1 - OCCUPATION MEMOIRE

III. 2 - EVALUATION DE LA DUREE DES CYCLES DE TRAITEMENT

III. 3 - APTITUDE A GERER LA SECURITE DE FONCTIONNEMENT DU MICROPROCESSEUR

III. 4 - POSSIBILITE DE MISE EN OEUVRE AUTOMATIQUE DU LOGICIEL

METHODE D'IMPLEMENTATION DE RESEAUX DE PETRI INTERPRETES SUR MICROPROCESSEUR

I - DESCRIPTION ET GESTION DU PROGRAMME REPRESENTATIF DU R D P I

La méthode d'implantation des réseaux de Pétri interprétés, que nous présentons dans ce chapître, est basée sur le traitement autour des places. Elle s'applique au RDPI sauf et vivant.
Le RDPI est intièrement décrit autour du seul type de noeuds, appelé Places. Autour de chaque place, on regroupe les sorties, les transitions descendantes à la place, les conditions de franchissement et les places de sortie des transitions descendantes. Chaque place est décrite par un module de programme appelé "PAS", qui effectue les différentes tâches associées à la place. Cette méthode est dite "orientée programmée" par le fait que chaque place est décrite par un module de programme écrit en assembleur |20|, |22|, |24|.

Le but recherché par l'emploi de telle méthode est d'avoir un programme portable et modulaire, afin de faciliter les modifications par adjonction ou suppression de module |PAS| et une durée de cycle de traitement |Traitement d'un marquage| la plus courte possible permettant de suivre de plus près l'évolution du processus commandé.

Le programme d'implantation comporte deux parties :

- Programme de description mémoire du RDPI : il est spécifique à l'application et décrit complètement le RDPI place par place. Il est formé de l'ensemble des modules (PAS).

- Programme de gestion : il est indépendant de l'application. Il assure la gestion du marquage des places, le cadencement des entrées et sorties et l'initialisation de l'automatisme|processeur et procédé|.

Nous présentons d'abord la modélisation statique du RDPI où l'on décrit la description mémoire des places et des marques ; ensuite, la modélisation dynamique où l'on décrit les différents blocs fonctionnels du programme moniteur, et enfin, nous évaluons les performances de la méthode par référence à notre application.

I. 1- DESCRIPTION MÉMOIRE DU R D P I

L'ensemble du RDPI est décrit par un ensemble de petits modules de programme correspondant aux places. A chaque place sont associées les informations concernant les actions et les transitions descendantes : dans un "PAS" de programme, on trouve la représentation physique de la place |adresse du début de "PAS"|, un bloc d'élaboration des sorties lorsque la place est marquée, un bloc du traitement de chacune des transitions descendantes |calcul des conditions de franchissement et les adresses des "PAS" associés aux places de sortie|.

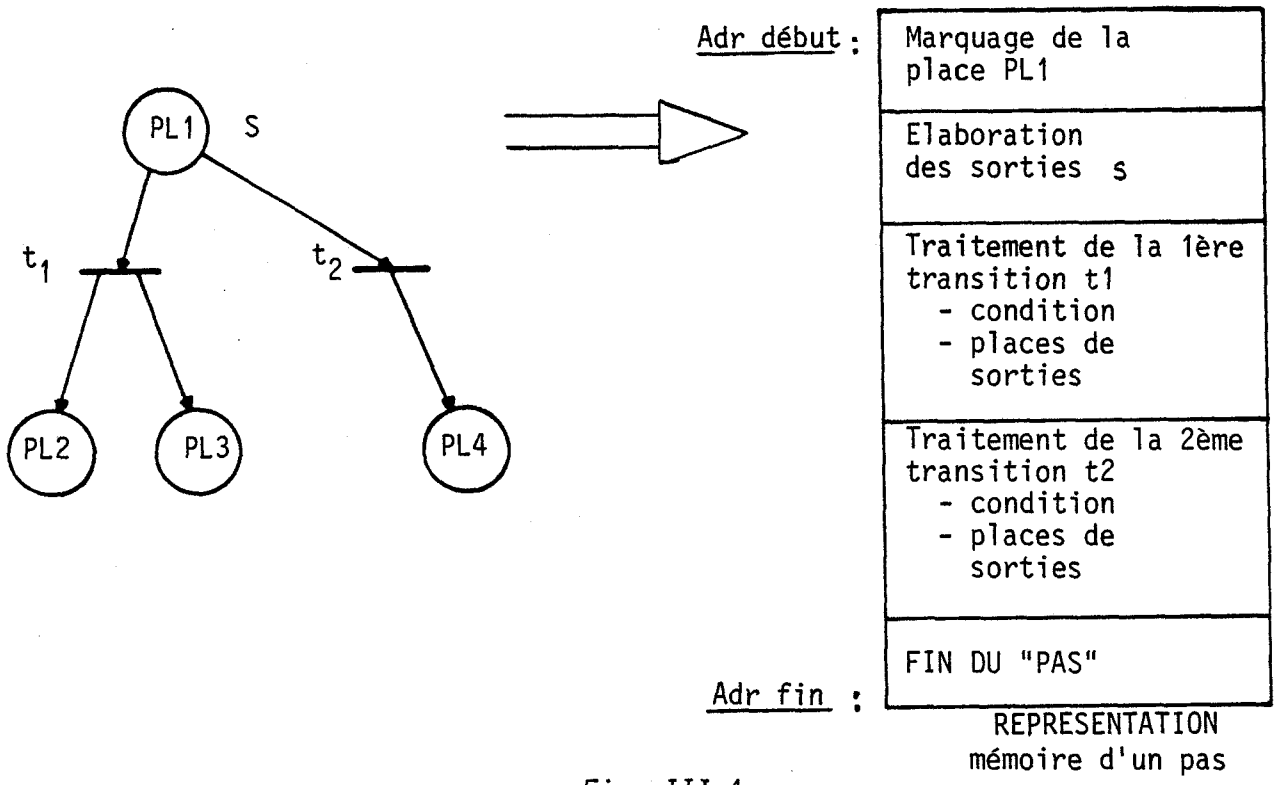


Fig. III.1

La difficulté de la représentation par des PAS réside dans le fait que, lorsqu'une transition admet plusieurs places d'entrée, celle-ci est décrite autant de fois que celle de places d'entrées. Cette redondance dans la description augmenterait la taille du programme. De plus, la synchronisation du franchissement de telle transition, et la gestion des marqueurs est assez délicate si toutes les places d'entrée sont représentées par des "PAS" de programme.

Pour remédier à cette difficulté, la notion de place Pivot et de synchronisme a été introduite |21|, |22|.

I. 1.1 - NOTION DE PLACE PIVOT ET DE SYNCHRONISME

Une place pivot pour une transition t est la seule place qui permet de valider la transition descendante t. Si cette transition admet d'autres places d'entrée, ces places seront notées places de synchronisme pour la transition t. Elles ne participent qu'au calcul de la condition du franchissement de la transition t, validée par le marquage de la place pivot. Ce type de place ne sera pas représenté par un "PAS" mais par une variable interne X_i telle que :

$X_i = 1$ si la place i est marquée

$X_i = 0$ si la place i n'est pas parquée

Ci-dessous, nous présentons une transition ayant trois places d'entrée.

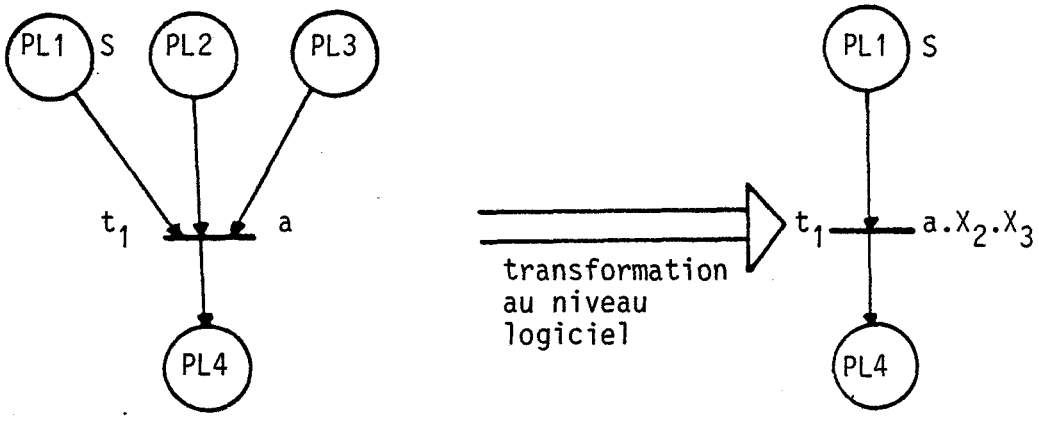


Fig. III.2

- PL_1 est place pivot pour la transition t_1. Les places PL_2 et PL_3 sont des places de synchronisme pour t_1. Elles sont représentées respectivement par les variables X_2 et X_3 qui n'interviennent que dans le calcul du franchissement de la transition t_1.

Lorsqu'une place de synchronisme X_i admet des sorties, celles-ci sont décrites autour des transitions qui permettent le marquage et le démarquage de la place de synchronisme. Autrement dit, les sorties sont activées lorsqu X_i devient égale à 1 et désactivées si X_i devient égale à 0.

Une place pivot pour une transition peut être de synchronisme pour une autre. Ces places ont une double représentation : par un PAS et une variable interne X_i. Ci-dessous la place PL_2 est notée place pivot pour t_2 et de synchronisme pour la transition t_1.

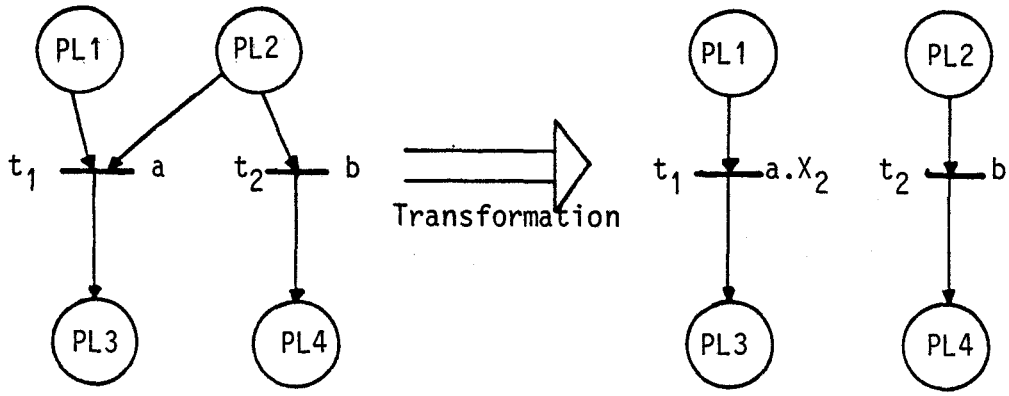


Fig. III.3

I. 1.2 - RECHERCHE DES PLACES PIVOTS

La procédure de recherche des places pivots dans un réseau de Pétri est la suivante :

1) Une place qui n'a qu'une transition descendante est dite place pivot essentielle.

2) Une place qui ne possède pas de transitions communes à d'autres places est notée place pivot.

3) Toute place admettant des sorties conditionnelles est nécessairement place pivot.

4) Pour les autres places, lorsque n places admettent au moins une transition commune, c'est la place qui admet d'abord des sorties ou/et le plus de transitions descendantes qui est considérée comme place pivot, les autres sont notées places de synchronisme pour la transition commune. Cette opération est répétée autant de fois qu'il reste de transitions à associer aux places.

L'applications de la procédure énoncée ci-dessus à l'exemple de la figure A4-A5 donne la répartition suivante :

PL2 (1ère règle), PL1, PL42, PL43, PL45, PL47, PL52, PL53 (2ème règle) et PL3 (4ème règle) sont des places pivots.

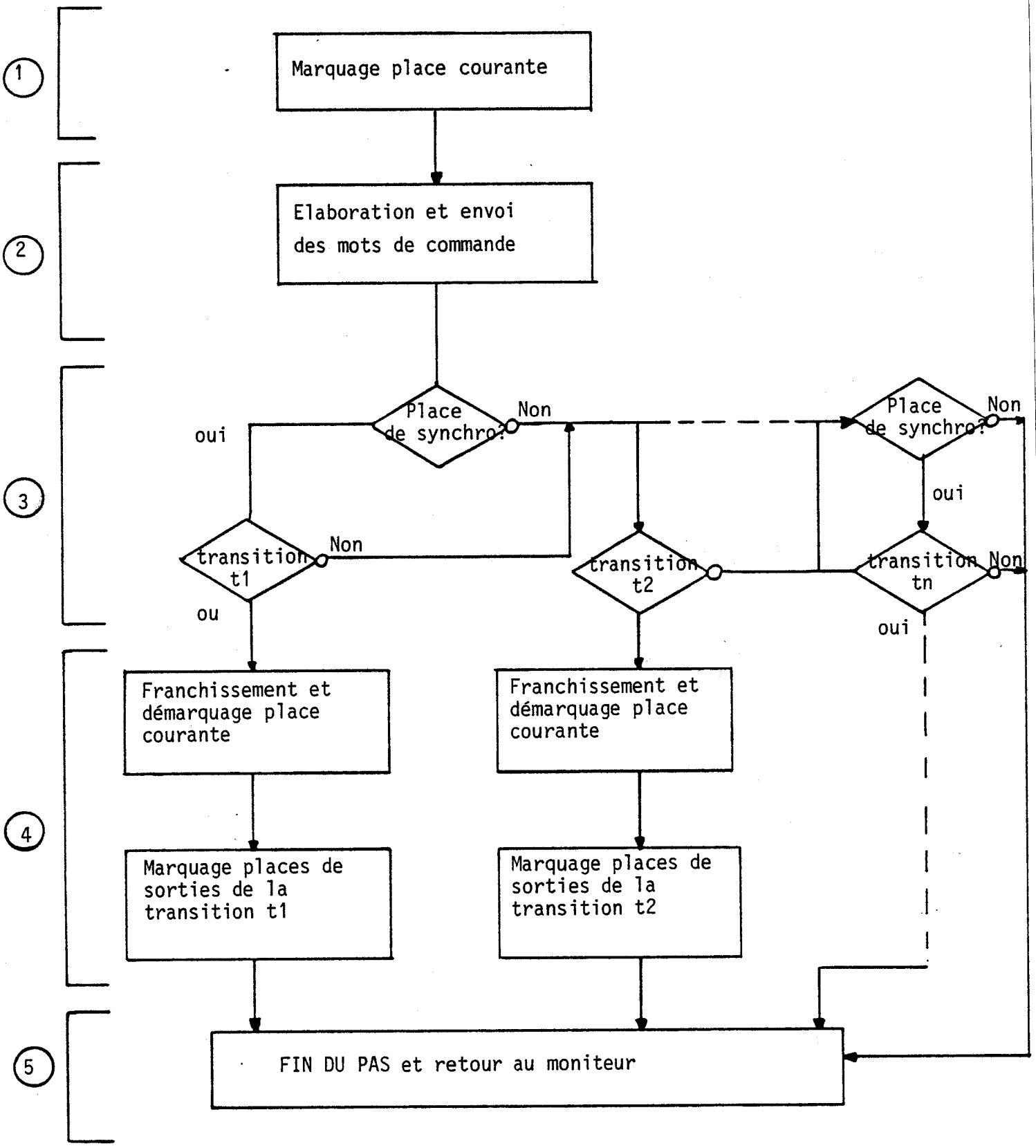
PL54 et PL55 sont des places de synchronisme.

Le RDPI à implanter sera entièrement décrit par l'ensemble des places pivots (PAS) et l'ensemble des variables X_i décrivant les places de synchronisme.

I. 1.3 - REPRESENTATION ET TRAITEMENT D'UN PAS

Le PAS représentant une place est appelé par le programme moniteur |CH. III. II.5| chaque fois que la place qu'il représente est marquée. Il permet le traitement de la place.

L'organigramme de la Figure III.4 montre le traitement d'un PAS.



Traitement d'un PAS

Fig. III.4

Le traitement d'un PAS comprend 5 blocs :

- Le bloc de marquage de la place courante permet de maintenir le marquage de la place jusqu'au franchissement d'une transition descendante.

- Le bloc d'élaboration et d'envoi des sorties vers le processus commandé.

Le bloc de test, qui comporte le test associé à chaque transition descendante |test du marquage des places de synchronisme (s'il y en a) et des évènements associés à la transition|.

- Le bloc du franchissement de chaque transition descencante qui permet le démarquage de la place courante et le marquage des places de sortie de la transition franchie.

- Le bloc de fin du "PAS" permet le retour au moniteur, qui lance les PAS suivants.

I. 1.4 - REPRESENTATION DES MARQUAGES

Une place marquée dans un réseau de Pétri interprété représente un état actif correspondant à l'exécution d'un nombre de tâches décrites autour de la place. L'exécution de ces tâches se fait par le lancement du PAS de programme associé à la place. Pour se faire, le marquage d'une place sera représenté par l'adresse du début du PAS correspondant |Fig. III.5|.

Dans un RDPI, plusieurs places peuvent être marquées simultanément, il est donc indispensable de pouvoir représenter l'ensemble des marques par les adresses des débuts des PAS associés. On utilise pour celà une structure de données où l'on écrit empilées les adresses des places marquées simultanément. Le lancement des PAS correspondants se fait par lecture des adresses empilées une à une. La construction des places marquées pendant le traitement des PAS se fait par écriture des adresses des places dans une autre structure de données qui sera lue à son tour.

Pour mieux organiser les deux fonctions d'écriture - marquage et de lecture - traitement, on utilise deux structures (piles) juxtaposées dans la zone mémoire RAM du microprocesseur. L'une sera accessible seulement en écriture où l'on construit une pile des adresses des places marquées, l'autre sera seulement accessible en lecture d'où les PAS seront lancés |Fig. III.5|.

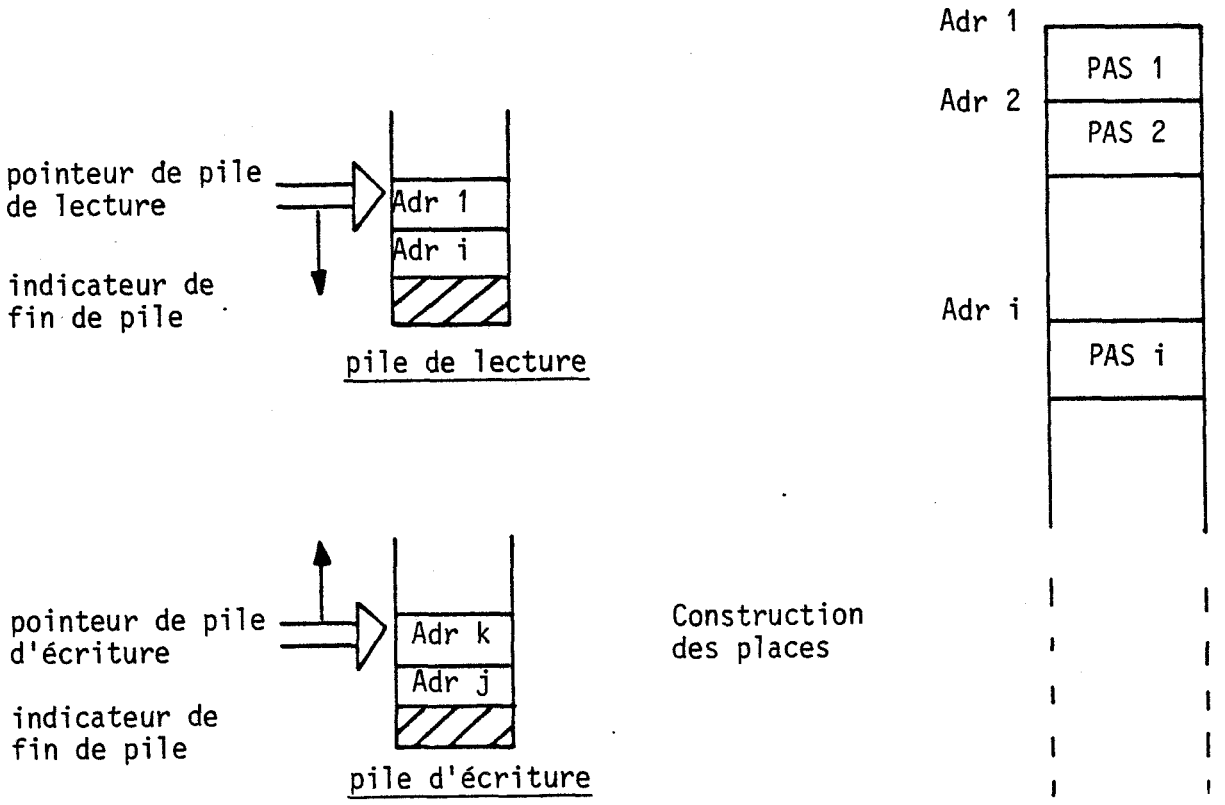


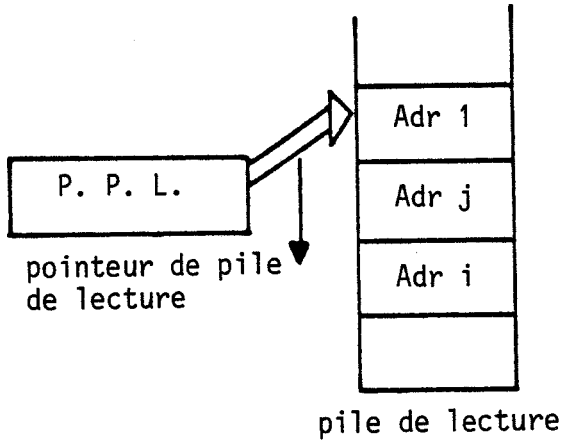
Fig. III.5

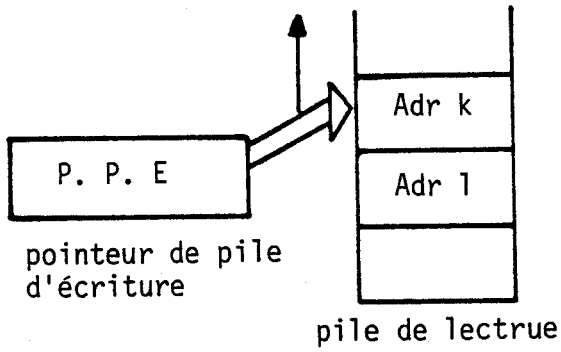
On attribue à chaque pile un pointeur permettant de suivre de plus près l'évolution de la taille de la pile et le nombre de places marquées pendant un cycle de traitement |traitement d'un marquage du RDPI|. La taille des deux piles est variable suivant le nombre de places marquées simultanément, on peut les juxtaposer en ajoutant un indicateur de fin de pile.

Pendant le cycle de traitement, la pile de lecture se rétrécit par décrémentation du pointeur de pile de lecture P.P.L. jusqu'à l'indicateur de fin de pile. Par contre la pile d'écriture s'agrandit de l'indicateur de fin de pile par incrémentation du pointeur P.P.E suivant le nombre de places à marquer.

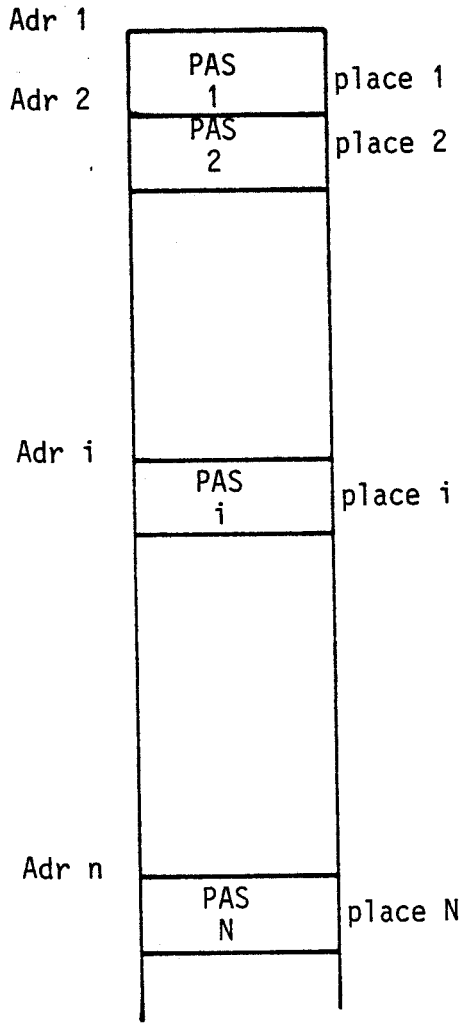
Ci-dessous, nous présentons la structure générale de description mémoire du RDPI et de son marquage.

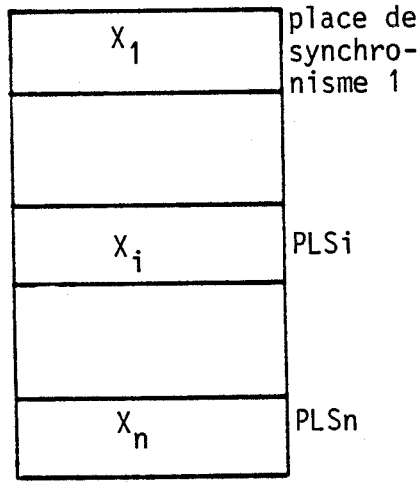
I.5 - STRUCTURE GENERALE DE REPRESENTATION





Représentation du marquage





Représentation des places

Fig. III.6

I. 2 - PROGRAMME DE GESTION

Le choix d'une méthode de représentation mémoire des RDPI est intimement lié à la méthode de gestion. En effet, on dira que la gestion du marquage est synchrone si pour un marquage M, toutes les transitions franchissables sont franchies simultanément, dans le cas contraire, la gestion est dite asynchrone. Nous avons choisi un programme de gestion synchrone.

I. 2. 1 - EVOLUTION SYNCHRONE DU MARQUAGE

Pour avoir une évolution synchrone, on gèle les conditions d'évolution du marquage pendant le cycle de traitement : on fige l'état du système par la sauvegarde en mémoire du vecteur d'entrée |états des capteurs| et du marquage du réseau au début du lancement du cycle de traitement. On ne traite que les piles des marqueurs figés en mémoire et on ne regarde que l'image du vecteur d'entrée en mémoire pour calculer les conditions d'évolution du marquage. Le nouveau marquage atteint ne sera traité qu'au cycle de traitement suivant.

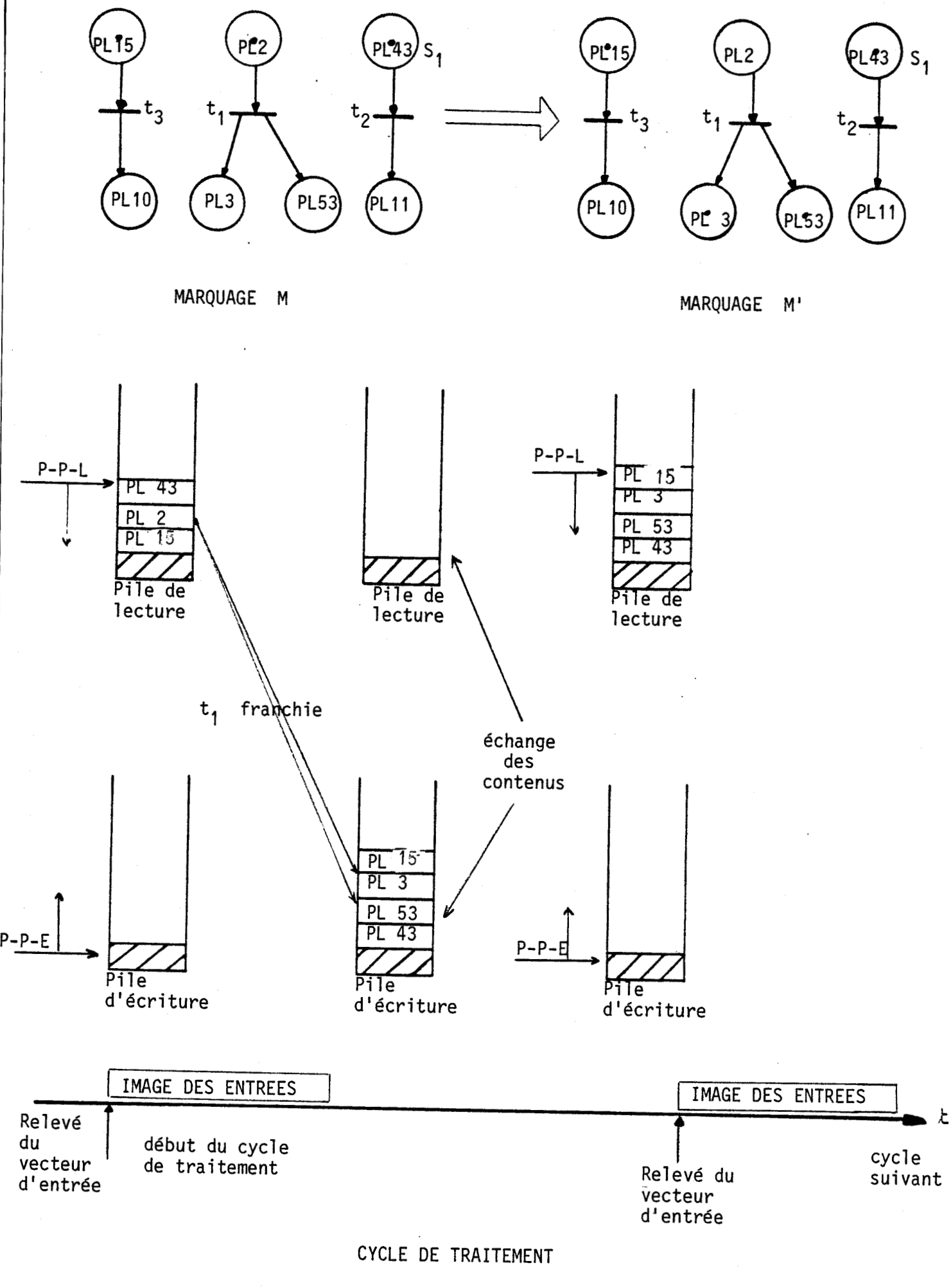
I. 2. 2 - GESTION DU MARQUAGE

Un cycle de traitement démarre à chaque lecture du vecteur d'entrée du système qui est figé par la suite dans une zone mémoire. Il s'arrête à la détection de l'indicateur de fin de pile de lecture - traitement.

Lorsqu'un cycle est lancé, on va lire l'adresse du haut dans la pile de lecture qui représente le marquage d'une place (PL_i). Celle-ci est effacée automatiquement par décrémentation du pointeur P.P.L. Elle est ensuite chargée dans le pile en formation |pile d'écriture| au début du traitement du PAS correspondant |Fig. III.7|. Si aucune transition descendante à la place (PL_i) n'est franchie, l'adresse (marquée PL_i) reste maintenue dans la pile d'écriture. Sinon elle est effacée par décrémentation du pointeur P.P.E et remplacée par les adresses des places de sortie de la transition franchie |Fig. III.7|.

Le traitement des autres places de la pile de lecture est poursuivi de la même manière jusqu'à l'indicateur de fin de pile de lecture. A la rencontre de ce dernier le contenu de la pile en formation (écriture) est échangé avec celui de la pile de lecture. A l'issu de cet échange la pile de lecture devient pleine, alors que celle d'écriture devient vide.

Nous présentons ci-dessous, la gestion d'un marquage tiré de l'exemple du RDPI donné en annexe. On suppose qu'une seule transition à été franchie (t_1).



CYCLE DE TRAITEMENT

Fig. III.7

I. 2. 3 - CADENCEMENT DES ENTREES/SORTIES

a) Vecteur d'entrée

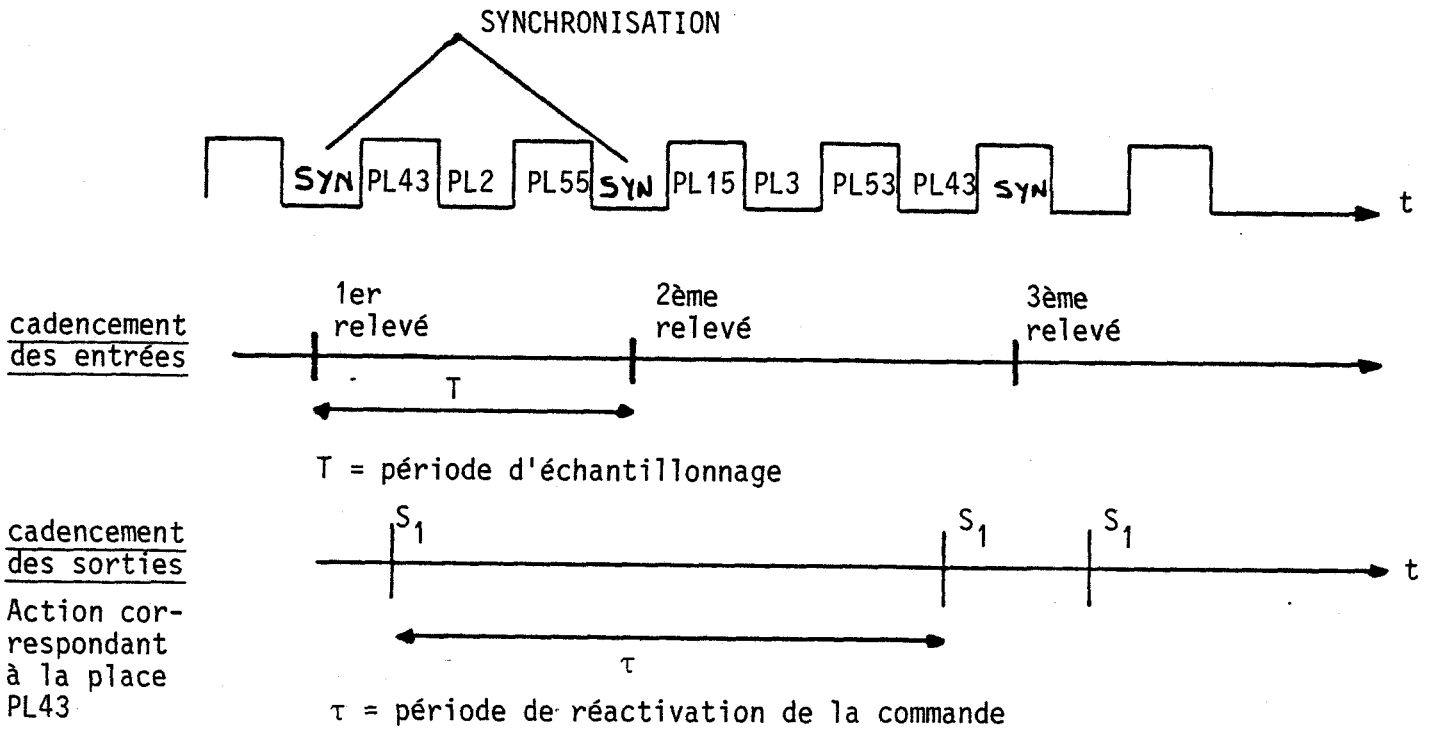
Le relevé du vecteur d'entrée est fait à chaque début du cycle de traitement. On prélève les grandeurs d'entrée que l'on stocke dans une mémoire. Ces grandeurs stockées seront utilisées pour le calcul des réceptivités associées aux transitions traitées dans le cycle de traitement. La cadence des relevés dépend du nombre de places marquées simultanément comme le montre la figure III.8. Nous verrons par la suite la normalisation des durées des cycles et par conséquent la période d'échantillonnage pour des fins tels l'ajout des tests fonctionnels du µP.

b) Vecteur de sortie

L'envoi des sorties logiques pour la commande de processus est fait directement : il consiste à valider les sorties vers le milieu extérieur au fur et à mesure de leurs élaborations à l'intérieur de chaque "PAS". Cette méthode a l'avantage par rapport à un envoi des commandes en bloc en fin de chaque cycle de traitement, de réduire l'occupation mémoire par le fait qu'on n'est pas amené à mémorisé chaque commande.

c) Représentation temporelle

En se référant à la gestion du marquage de la figure II.7, nous présentons ci-dessous la synchronisation de l'évolution du marquage avec les changements d'état du processus, et l'émission des commandes.



SYNCHRONISATION du marquage par rapport à l'évolution du processus

Fig. III.8

Remarques :

- La période d'échantillonnage dépend essentiellement du degré du parallélisme du RDPI, et des performances du microprocesseur utilisé.

Elle doit satisfaire la condition suivante :

$$T \leq \frac{Tch}{2}$$

Tch = temps entre 2 changements des grandeurs du processeur

T = temps de réponse du processus

- Le relevé cyclique du vecteur d'entrée permet de s'affranchir des défauts fugitifs intervenant entre 2 relevés (immunité contre les parasites sur les lignes d'entrée).

- La commande associée à une place marquée pendant un nombre N de cycles de traitement est réaffirmée à chaque cycle. La commande ne reste donc pas statique, mais dynamisée.

I. 2.4 - INITIALISATION

L'initialisation du système est opérée par le moniteur dès la mise sous tension de l'automatisme, ou à l'arrêt des commandes par des procédures particulières. Elle comporte l'initialisation du processeur et du réseau de Pétri, la préparation des temporisations utilisées et la mise à jour des variables internes :

- l'initialisation du processeur : elle permet l'initialisation des blocs fonctionnels du microprocesseur, en particulier les ports E/S, les timers, les registres de contrôle, le port série... etc.

- l'initialisation du RDPI. Elle se fait par la définition des deux piles (écriture et lecture), leurs tailles et pointeurs respectifs, et par l'écriture des adresses des places marquées initialement.

I. 2.5 - MONITEUR

Le programme de gestion dont l'organigramme est donné ci-dessous, appelé moniteur, comporte les blocs fonctionnels suivants :

- bloc de marquage : il est affecté à la recopie des adresses des places marquées pendant les cycles de traitement dans la pile d'écriture

- bloc de démarquage : il permet l'effacement des adresses des places démarquées par décrémentation du pointeur P.P.E.

- bloc d'échange des contenus des deux piles : ce bloc de programme est appelé par la prise de son adresse du début dans la pile de lecture : l'adresse du bloc représente en effet l'indicateur de fin de pile. Le moniteur n'effectue pas de test, mais prend les adresses une à une jusqu'à la dernière qui représente le bloc d'échange. Ce bloc est lancé comme une place

- bloc des sorties : il permet l'envoi des mots de commande vers le milieu extérieur.

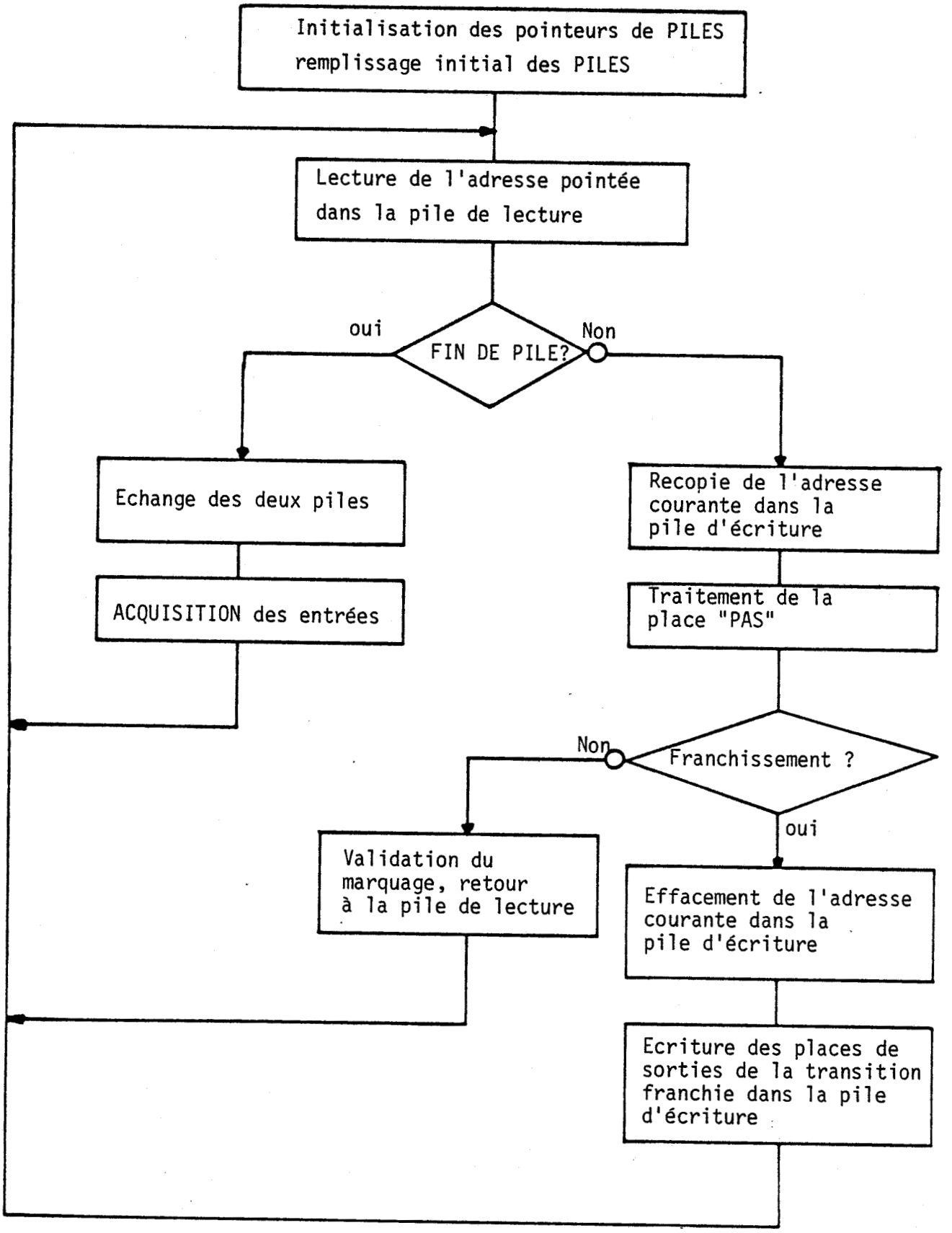


Fig. III.9

ORGANIGRAMME DU MONITEUR

II - SOLUTIONS PARTICULIÈRES APPORTÉES AUX PROBLÈMES SPÉCIFIQUES DE COMMANDE-CONTRÔLE DE PROCESSUS ÉLECTROMAGNÉTIQUES

II. 1 - GESTION DES TELEMESURES ET TELECOMMANDES

Le système de commande localisé est sous contrôle d'un dispositif hiérarchiquement plus élevé. Il est constamment amené à répondre aux ordres de ce dispositif par des télécommandes spécifiques.

Chaque télécommande est formée de deux octets, l'un représentant le mot de passe |mot code|, l'autre étant la commande |mot utile|.

Lorsque le microprocesseur reçoit une télécommande, celle-ci est décodée |reconnaissance du mot code|.

Dans le cas où le mot code est reconnu, le mot utile de la télécommande est stocké afin d'être utilisé par la suite. Cependant, il existe d'autres télécommandes, qui ont pour fonction d'arrêter ou demander des comptes rendus au microcontrôleur, celles-ci sont traitées instantanément |exemple : la réinitialisation| |ch. II. II.4|.

Chaque microcontrôleur n'émet ses télémesures que sur ordre du dispositif extérieur. Il se contente de les stocker dans une pile dite de télémesures. Ces télémesures sont émises par la suite par paquet. A chaque fin d'émission de paquet, le microcontrôleur émet un indicateur de fin de pile afin de permettre au dispositif de scruter l'ensemble du réseau de commande.

II. 2 - METHODE DE TRAITEMENT DES TEMPORISATIONS

Les temporisations associées aux places dans un RDPI sont souvent exécutées à l'aide de boucles d'attente où le microprocesseur ne fait que décompter le temps. Cette technique de temporisation est écartée. En effet, notre choix s'est porté sur la temporisation des transitions. Une transition temporisée signifie que la transition ne sera tirée qu'au bout d'un nombre n de cycle de traitement. Par cette procédure, le microprocesseur reste en contact avec son procédé à commander (relevé du vecteur d'état à chaque cycle).

La base de temps utilisée pour le calcul des temporisations est la durée des cycles de traitement. A chaque cycle, le compteur de temps est décompté. Le pas du décomptage est égal au nombre de places marquées simultanément, chaque place dure environ 100 µs. Cette valeur aujourd'hui approximative ira vers une meilleure précision lors de la normalisation de la durée de traitement des pas (Réf.31).

Ci-dessous, nous présentons une place possèdant une transition descendante temporisée. Le PAS de programme que nous présentons est généré par le traducteur |voir Ch. IV|.

Point d'entrée du module

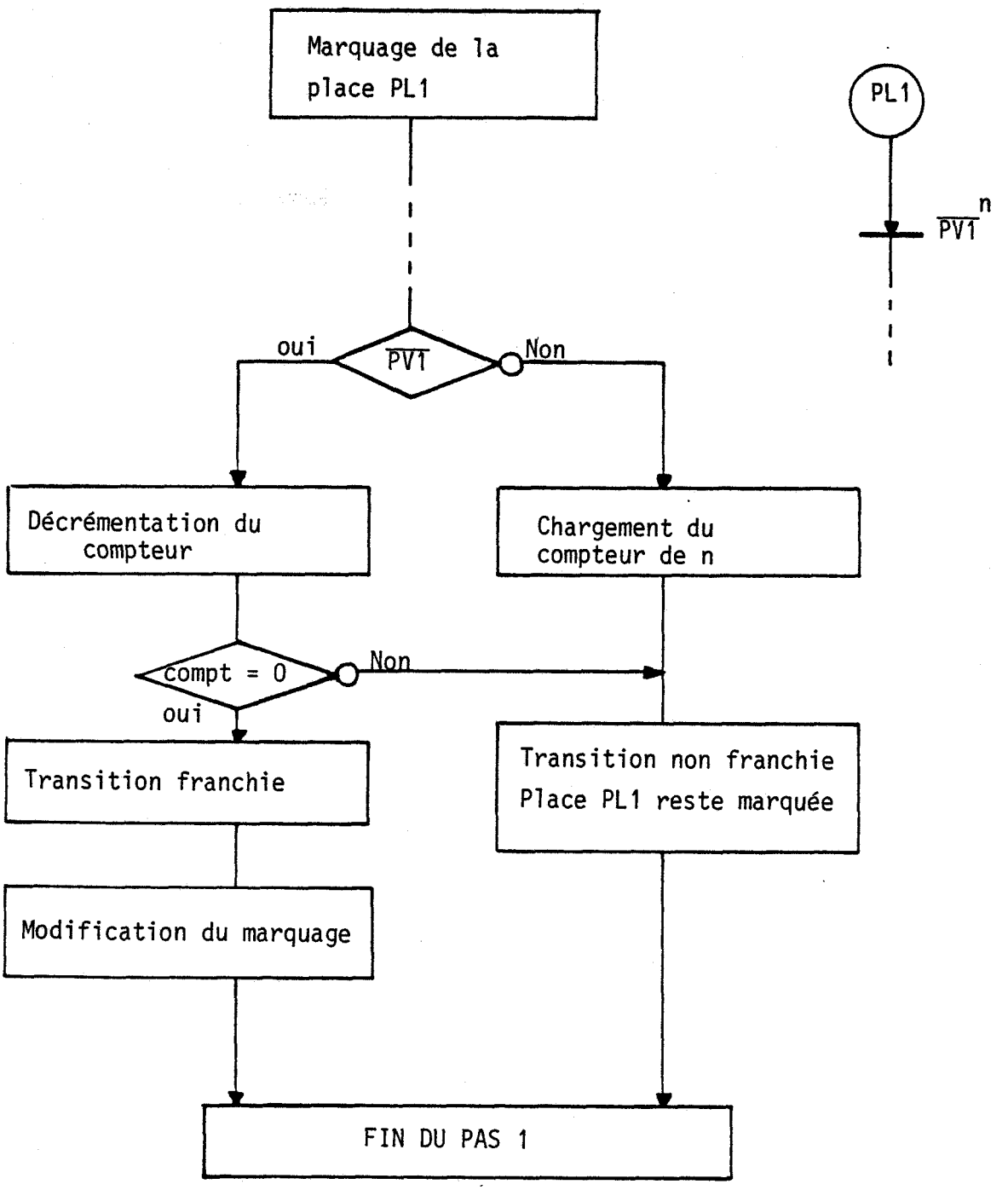
```
                                    Durées d'execution
                                     en microseconde
PL02:     CALL     MP                  ; 10;   Marquage de la place PLØ2
          MOV       SO1,#000H          ; 2;
          CALL     SORT                ; 6;   Envoi des commandes
TR021:    NOP
TT0002:   MOV       A,TT00a            ; 1;   Temporisation TToo
          SUBB      A,R2               ; 1;
          JB        ACC.7,TT0a02       ; 2;   R2, contient le nombre de places
          MOV       TT00a,A            ;1;       marquées simultanément à
          MOV       R3,#01BH           ; 1;       chaque cycle
          JMP       CG                 ;2;
TT0a02:   MOV        TT00a,#64H
          DJNZ     TT00,FI021         ; 2;
          MOV      TT00,#065H         ; 2;
FR02:     CALL      FR                 ; 6;
          MOV       DPTR,#PL04         ; 2;
          CALL      MP                 ; 10;   R3, indicateur de la durée de
          MOV       R3,#02FH                      traitement du PAS (PLØ2)
          JMP       CG
FI021:    MOV       R3,#01CH
          JMP       CG                          "CG", routine du moniteur
```

Point de sortie du module

Chaque temporisation dure un nombre n fois de 100 µs.

II. 3 - TRAITEMENT DES REBONDISSEMENTS ELECTRIQUES ET MECANIQUES DES CONTACTS

Le relevé cyclique du vecteur d'état permet de rafraichir régulièrement l'image des entrées et favorise le traitement d'antirebonds. Il s'agit de ne prendre en compte l'état logique d'un capteur que s'il garde la même valeur pour N cycles de traitement |N relevés du vecteur d'entrée|. Pour se faire, on associe à chaque capteur un compteur chargé initialement de N. Chaque fois que le capteur est testé, et qu'il a la même valeur, le compteur est décrémenté jusqu'à zéro. Dans ce cas, l'état logique est pris comme bon et la transition à laquelle il est associé peut être franchie comme l'indique l'organigramme ci-dessous. Le traitement d'antirebond est effectué autour de la transition et par conséquent à l'intérieur du PAS où la transition est décrite. Ce procédé permet de s'affranchir des rebondissements mécaniques et électriques des capteurs |fréquents dans les automatismes industriels|, rend l'unité de commande peu sensible aux parasites, et tolère les changements d'état intempestif de courte durée.



Traitement d'antirebonds

Fig. III.10

La place PL1 ne peut être démarquée que si le capteur PV1 (verrou) reste à 0 pour les n cycles de traitement |le PAS étant appelé à chaque cycle|.

Le traitement d'antirebonds comme l'indique l'exemple ci-dessus permet de s'affranchir des parasites et des changements d'état intempestifs de courte durée.

II. 4 - REINITIALISATION

La réinitialisation ou l'arrêt de l'évolution du procédé automatisé à tout instant est très importante, elle permet de se protéger contre l'apparition de situation intempestive et dangereuse, pouvant porter atteinte à la sécurité des matériels et des personnes. Il est donc nécessaire de prévoir des procédures particulières dites généralement procédure d'arrêt d'urgence. Ces procédures permettent la réinitialisation de l'automatisme en le reconfigurant dans une situation connue. La méthode décrite ci-dessus s'adapte facilement à ce genre de situation. En effet, le fait d'avoir une pile de lecture, de laquelle les modules |PAS| sont lancés et une pile d'écriture dans laquelle sont stockées les adresses des PAS à lancer dans le cycle suivant, il est envisageable lors d'une demande de réinitialisation de forcer le marquage du RDPI à un marquage permettant d'initialiser l'unité de commande de l'automatisme.

Le forçage du marquage comme l'indique la figure |III.11| se fait en réinitialisant les deux piles et en chargeant la pile de lecture par les adresses des places à marquer pour une réinitialisation. Le moniteur continuera la gestion du marquage en lançant les PAS dont les adresses étaient déposées volontairement. Les adresses déposées ne sont pas nécessairement celles des places initiales.

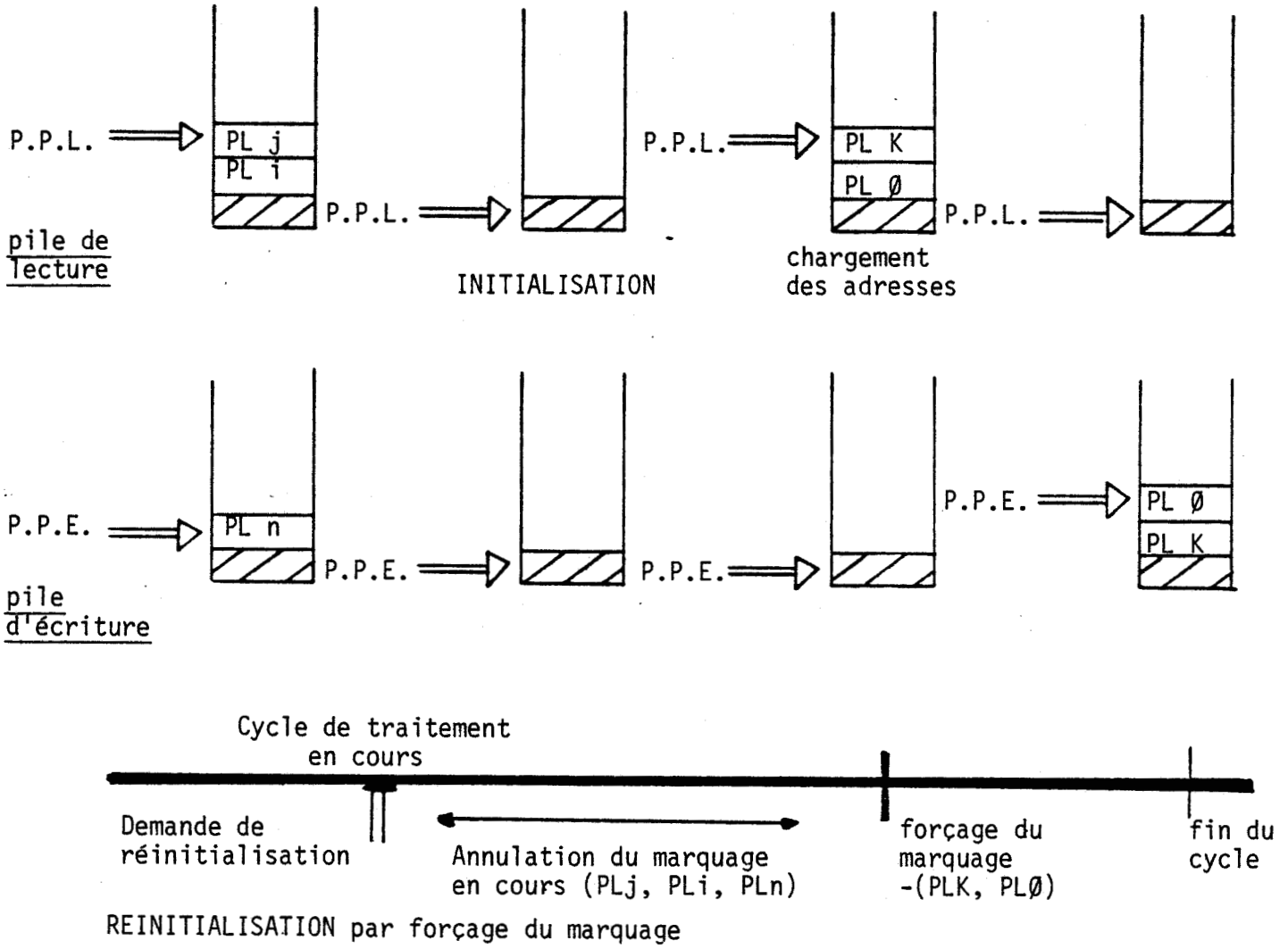


Fig. III.11

Cette méthode dite de réinitialisation a été introduite pour résoudre le problème lié à la demande d'évacuation d'urgence et à la demande d'arrêt ou d'initialisation faite par le dispositif hiérarchiquement plus élevé, auprès du microcontrôleur.

III - PERFORMANCES

L'évaluation des performances de la méthode d'implantation du réseau de Pétri décrite ci-dessus consiste à évaluer l'occupation mémoire et la durée du cycle de traitement |permet de suivre de plus près le procédé automatisé|.

- Nous allons examiner successivement ces deux points en faisant référence à notre application.

III. 1 - OCCUPATION MEMOIRE

L'occupation mémoire dépend essentiellement de la taille du réseau de Pétri et par conséquent de l'application. Pour notre application commande-contrôle d'un mécanisme de porte, nous avons implanté un réseau de Pétri de 80 places. Le programme écrit en assembleur 8051, occupe 3 K octets de mémoire EPROM et 100 octets de RAM répartis de la façon suivante :

- 3 K octets de programme :
  - le moniteur occupe une faible partie : 60 octets
  - les places occupent de 20 octets |pour une place simple ne possédant qu'une transition descendante| à 180 octets |pour une place possédant 8 transitions descendantes avec des traitements d'antirebonds |ex : PL1| |.

- 100 octets de mémoire vive de données
  - les piles de lecture et d'écriture occupent chacune 16 octets
  - la pile des télémesures occupe 40 octets |nous en reparlerons par la suite|
  - le reste est occupé par les places de synchronisme x_i, les images des entrées, les registres des temporisations et des antirebonds et les variables internes.

III. 2 - EVALUATION DE LA DUREE DU CYCLE DE TRAITEMENT

L'évaluation de la durée d'un cycle de traitement reste approximative et dépend essentiellement de l'application |complexité des places à traiter| et du degré du parallélisme dans le réseau de Pétri |nombre de place marquée simultanément|. Pour un processus électromécanique tel le mécanisme de porte, la durée du cycle de traitement reste extrèmement faible devant la durée d'évolution du processus. En effet, la durée de traitement d'une place du réseau donné en annexe ne dépasse pas 70 µs |pour un quartz de 12 MHz et écrite en assembleur 8751|. Vu la rapidité du microprocesseur, on est amené à le ralentir en introduisant des temporisations dans le programme.

Les deux paramètres |occupation mémoire et durée du cycle de traitement| sont des critères d'évaluation généralisée de la méthode. Toutefois, d'autres critères aussi importants permettent de mieux évaluer les avantages de la méthode pour une application donnée, tels les performances du logiciel produit, la souplesse de la méthode et l'aptitude à gérer la sécurité du fonctionnement du microprocesseur par observation temporelle de son fonctionnement. Ce sont des critères que nous mettons en évidence par la suite, en faisant référence à notre application.

La modularité du programme |chaque place est représentée par un PAS| et la simplicité du logiciel produit |peu d'instructions utilisées puisque les PAS s'écrivent de la même façon|, nous a permis d'envisager et développer par la suite les deux points ci-dessous.

III. 3 - APTITUDE A GERER LA SECURITE DE FONCTIONNEMENT DU MICROPROCESSEUR

Dans le chapître précédent, nous avons montré les différents niveaux d'études nécessaires à la mise en sécurité de l'unité de traitement. Il est donc logique de présenter le procédé utilisé pour notre application et les contraintes qui s'y rattachent. On se propose alors, et ceci grâce à la modularité du programme, de structurer chaque PAS |par comptage des durées de chaque instruction élémentaire| de telle sorte que la durée d'exécution quelque soit le point de sortie du PAS, reste la même. A la sortie du chaque PAS, le microprocesseur génère le signal de test. Il suffit alors d'implanter autour du microprocesseur

une structure de contrôle en logique câblée, capable d'analyser le signal équitemps et de vérifier tous les fronts qui apparaissent à intervalle régulier. Tout front décalé ou absent est détecté par la structure de contrôle, qui agit sur le mot de commande en le faisant tomber à zéro |absence d'énergie|, en n'alimentant plus en fréquence un ampli sélectif en sécurité intrinsèque*. Ci-dessous, nous avons schématisé l'ensemble microprocesseur et structure de contrôle.

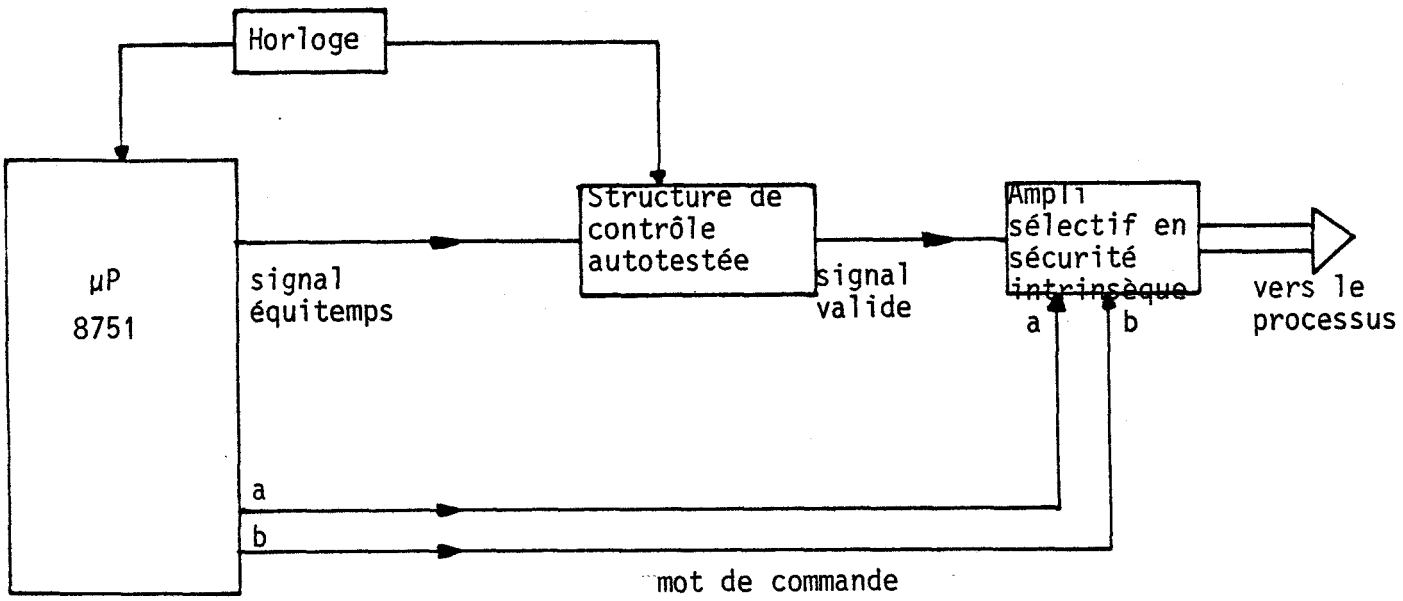


Fig. III.12

Il est bien entendu que les interruptions ne sont plus autorisées. Le dialogue doit se faire à intervalle de temps régulier afin de permettre la récupération des télécommandes sans perte d'informations. Il a fallu tester à chaque fin de PAS |un PAS dure pour notre application 100 µS| la réception des TC. Le programme de réception dure un nombre n fois de la durée d'un PAS de façon à maintenir la génération du signal de test. |fig. III. 1.4|.

L'élaboration des tests de la RAM, des registres de fonctions |ports, registres de contrôle, accumulateur... etc| se font structurés à la fin de chaque cycle de traitement |Réf. 30, 31|. Ils sont fait de la manière suivante : chaque élément à tester est chargé de la valeur hexadécimale FF, puis décrémenté jusqu'à une valeur permettant la génération du signal de test. De ce fait, tout collage de bit ou une mauvaise exécution des instructions décalent les fronts générés de quelques microsecondes qui seront détectées par la structure.

La représentation temporelle ci-dessous, montre comment ont été résolues les contraintes liées à la gestion du dialogue sans interruption du programme et l'élaboration des tests fonctionnels. Nous avons fixé la durée du cycle de traitement à 5 ms.

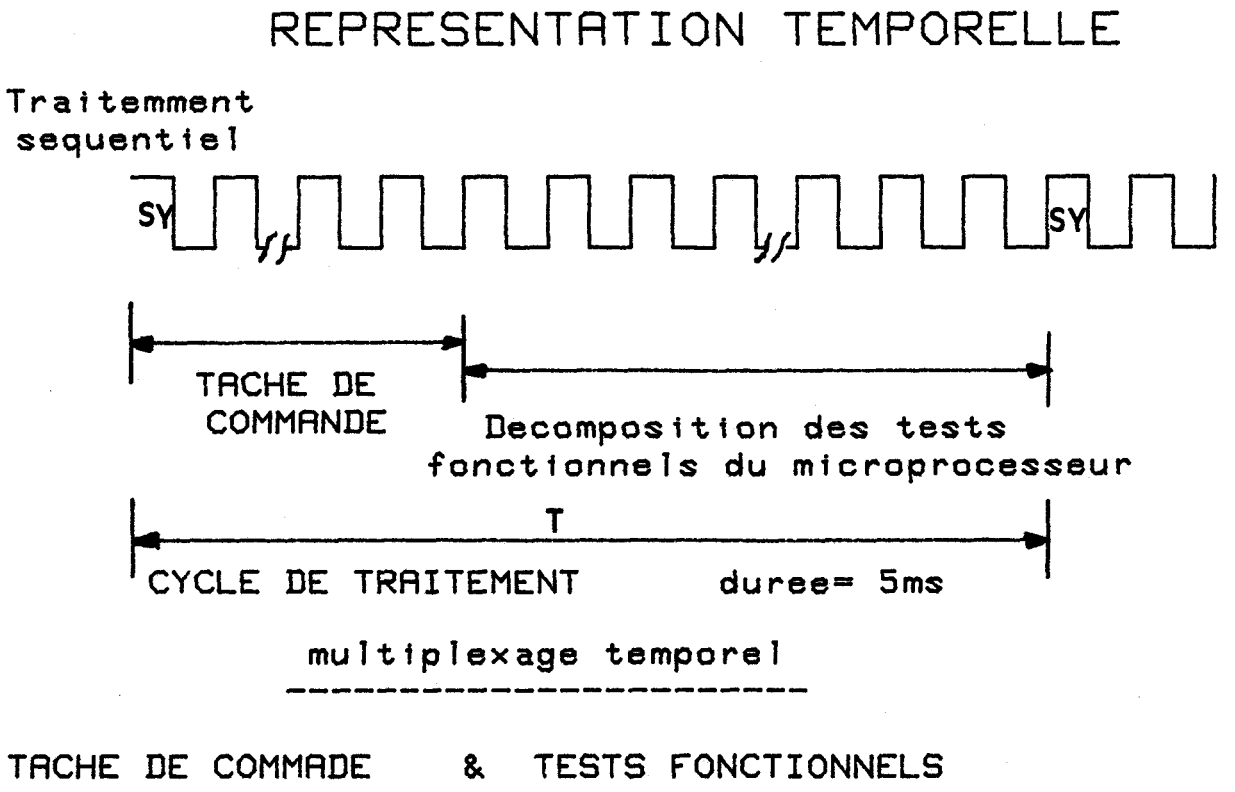


Fig. III.14

III. 4 - POSSIBILITE DE MISE EN OEUVRE AUTOMATIQUE DU LOGICIEL

La simplicité du logiciel produit par la méthode, permet d'envisager la mise en oeuvre automatique de la transformation RDPI/programme. Cela revient à créer un langage spécifique de description du RDPI et un interpréteur correspondant, capable de générer un fichier objet écrit en assembleur du microprocesseur que l'on désire programmer |ici, un INTEL 8751| en y introduisant les tests fonctionnels et la génération du signal équitemps. L'utilisateur pourra alors programmer le microprocesseur qu'il désire utiliser pour piloter son processus à partir de la simple description du RDPI et ceci, de façon simple et rapide.

C'est ce dernier point que nous présentons au quatrième chapître.

CHAPITRE IV

CHAPITRE IV : CONCEPTION D'UN OUTIL INFORMATIQUE D'AIDE AU DEVELOPPEMENT DES PROGRAMMES DECRITS FONCTIONNELLEMENT PAR UN R D P I

I - INTRODUCTION

II - LES LOGICIELS INTERACTIFS DE SAISIE DE DONNEES CARACTERISTIQUES DU R D P I

II. 1 - PREPARATION

II.1.1 - Définition des lignes E/S
II.1.2 - Préparation du processeur
II.1.3 - Initialisation du RDPI
II.1.4 - Temporisations
II.1.5 - Dialogue microprocesseur/dispositif extérieur

II. 2 - EDITION DU RDPI SAUF ET VIVANT

II.2.1 - Edition autour des places

II.2.11 - Bloc d'introduction des places
- Places
- Actions

II.2.12 - Description des transitions descendantes
- Transition
- Tests
  * Etat logique
  * Télécommandes
  * Temporisations
  * Place de synchronisme
  * Combinaison de tests

.../

/...

II.2.2 - Stockage et traitement syntaxique des données

II.2.3 - Aide à la mise en oeuvre
- Trace
- Modidication

III - TRANSFORMATION R D P I EDITE EN LANGAGE ASSEMBLEUR, OU FONCTION DE COMPILATION

III. 1 - PRE-TRADUCTION
- Affectation
- Génération du moniteur

III. 2 - TRADUCTION DES PLACES
- Places et actions
- Transitions et tests
- Places de sortie

IV - SURETE DE FONCTIONNEMENT DE L'OUTIL INFORMATIQUE D'AIDE AU DEVELOPPEMENT DU LOGICIEL

GENERALITE

IV. 1 - SURETE DE FONCTIONNEMENT DES LOGICIELS INTERACTIFS

IV. 2 - SURETE DE FONCTIONNEMENT DU TRADUCTEUR OU COMPILATEUR

I - INTRODUCTION

L'objectif recherché, lors de la conception et de la réalisation de cet outil, s'inscrit dans la démarche sécuriste de notre problème de commande-contrôle de processus industriel. Nous avons montré dans le précédent chapître, les différents intérêts présentés par la méthode d'implémentation du RDPI sur microprocesseur. Ces intérêts se résument sur le plan de la sécurité en deux points :

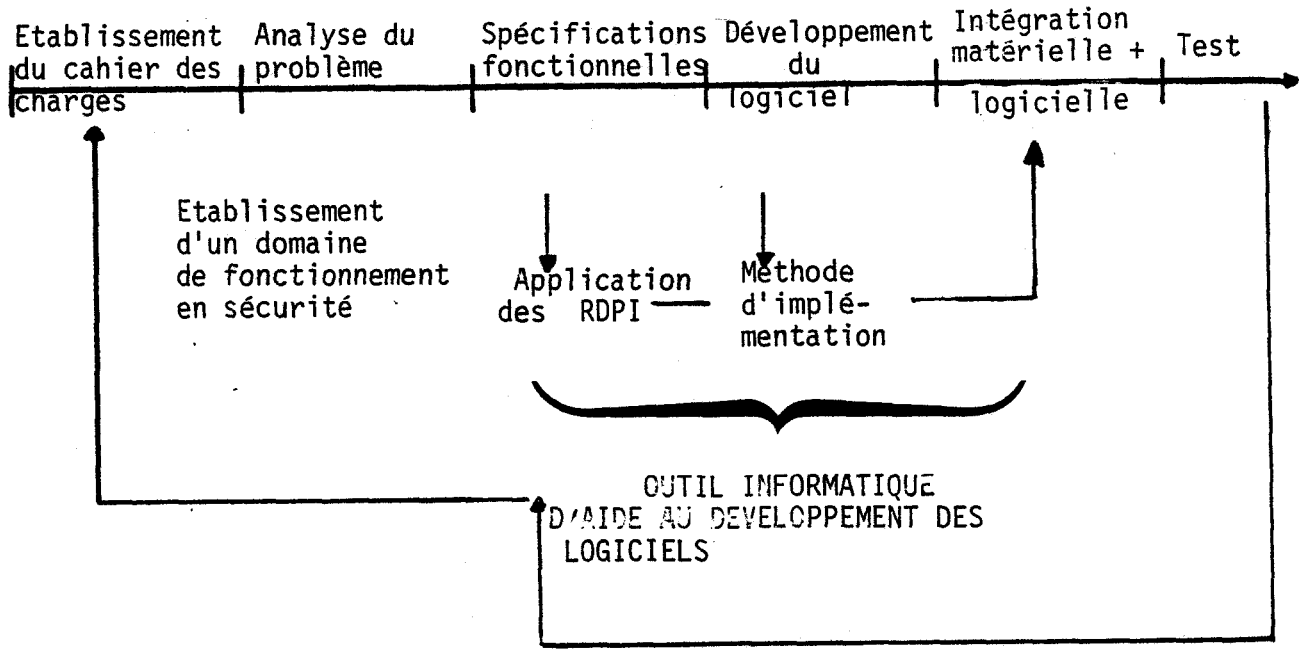
- une haute sûreté de fonctionnement du logiciel produit ;

- la possibilité de mise en sécurité du microprocesseur, selon le procédé, basé sur l'observation temporelle de son fonctionnement |31|

Le maintien de ces avantages lors de la génération automatique du logiciel en assembleur, impose certaines contraintes à caractère systèmatique, notamment la normalisation de la durée d'exécution de chaque PAS de programme et l'inhibition de toute forme d'interruption allant dans le sens contraire de la sécurité. Ces contraintes ont été résolues en respectant le mode d'exécution du programme de traitement du marquage du RDPI tel que nous l'avons défini au chapître précédent.

La mise en oeuvre d'un tel outil permet donc sur le plan de la sécurité, de couvrir une étape importante située au coeur des différentes phases consécutives du développement d'un projet |41,40 |. Cette étape se situe, comme le montre la représentation ci-après, en amont de la phase consistant à développer le logiciel.

PROGICIEL



Phases de développement d'un projet

L'outil d'aide au développement de programmes décrits par un RDPI, permet d'établir une liaison directe entre la description graphique correspondant aux spécifications fonctionnelles et l'integration materielle et logicielle . Pour le concepteur, ce lien se traduit au niveau du concept par une continuité dans son travail dans la mesure où l'outil permet de soulager la phase de transcription allant du RDPI jusqu'au code binaire exécutable par le microprocesseur.

La réalisation de l'outil informatique ou PROGICIEL d'aide à la transcription des RDPI en langage binaire doit satisfaire aux exigences de sécurité relatives au problème de commande de processus discontinus.
Il doit respecter les contraintes suivantes :

- commodité d'utilisation. Ceci afin de minimiser les erreurs de transcription et réduire le temps de mise en oeuvre.

- Vérifications intermédiaires. Celles-ci doivent être possible dans chaque phase de transcription du RDPI.

- Sûreté de fonctionnement de l'outil. Il est primordial d'écarter toute erreur de transcription et de génération de faute dans le code exécutable par le microprocesseur. Ceci doit se faire au niveau de la transcription et de traduction par des analyses et vérifications automatiques très poussées et répétitives.

Le PROGICIEL a donc été développé dans cet état d'esprit. Il a été réalisé à partir d'un programme BASIC hautement interactif |environ 3000 lignes| résidant sur un calculateur de type HEWLETT PACKARD 9836. Ce PROGICIEL génère un programme écrit en assembleur 8051 . Ce programme est récupéré par un système de développement INTEL série II-221 à travers une liaison série standard RS 232 C. L'ensemble de la configuration matérielle utilisée pour notre travail est donnée détaillée en annexe III.

Nous développons dans les paragraphes suivants les différents aspects de notre travail. Tout d'abord, dans le paragraphe II, nous décrivons la partie de saisie de données caractéristiques du RDPI en mettant l'accent sur l'interactivité et les moyens mis en oeuvre pour limiter les risques d'erreurs de transcription. Dans le paragraphe III, nous exposons la fonction de compilation ou traduction proprement dite avec les problèmes de sûreté de fonctionnement qui s'y rattachent.

II - LES LOGICIELS INTERACTIFS DE SAISIE DES DONNEES

CARACTERISTIQUES DU RDPI

Les différentes tâches propres à remplir notre fonction sont écrites en basic interprété sur le calculateur HP 9836. Elles sont au nombre de quatre et peuvent se céder la place mutuellement à l'exception du programme de transfert comme l'indique l'organigramme ci-dessous.
On y trouve les logiciels suivants :

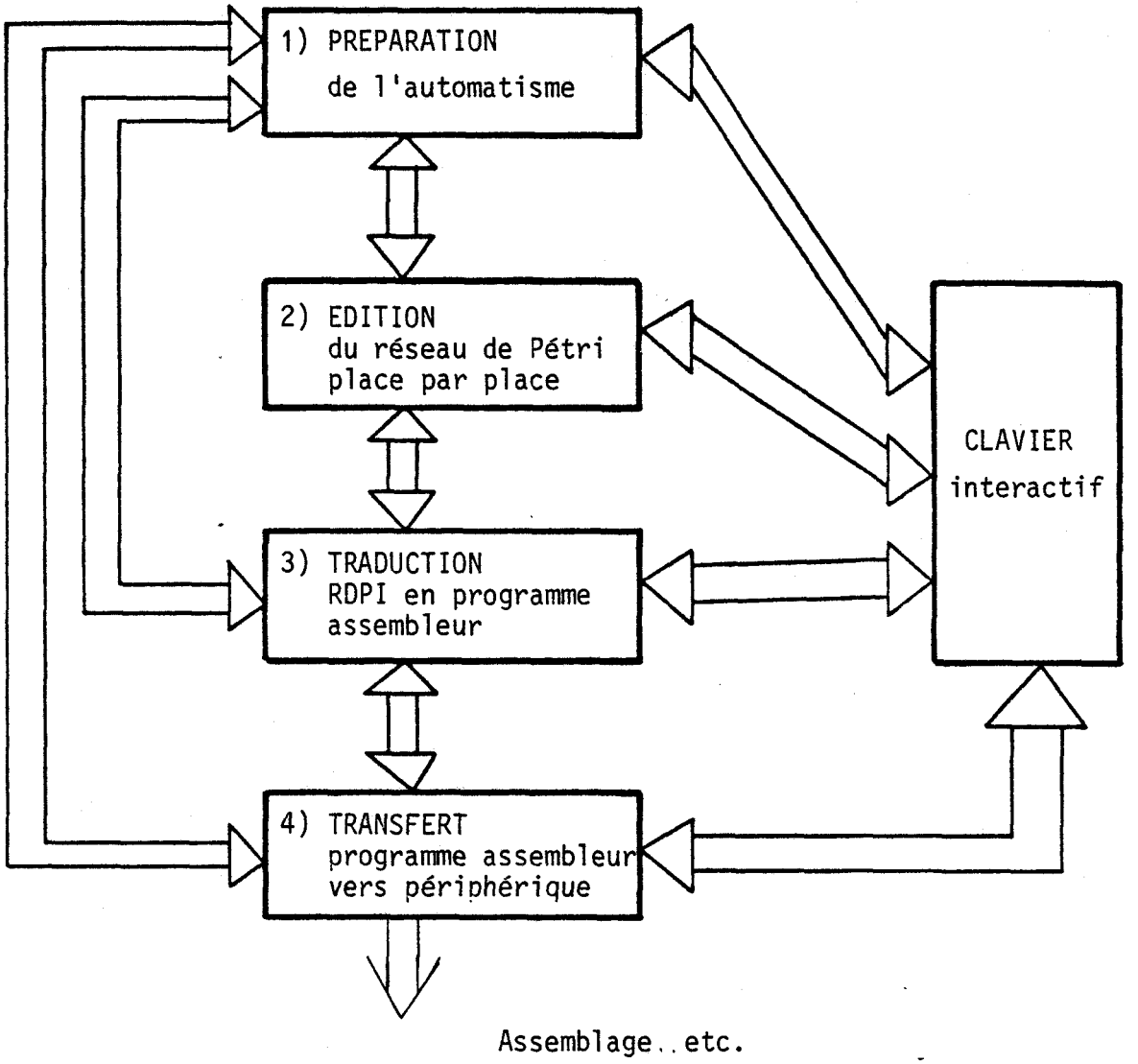
- La préparation du microprocesseur et de l'ensemble du système de commande : définition physique et assignation symbolique des vecteurs d'entrée et de sortie.

- L'édition ou la transcription du RDPI, elle est indépendante du microprocesseur utilisé et permet l'introduction du RDPI place par place.

- La traduction ou compilation du RDPI en programme assembleur du microprocesseur utilisé. Il permet l'envoi du programme assembleur.

- Le transfert vers le système de développement.

Ci-après, nous présentons schématisé l'organigramme décrivant l'accès aux logiciels utilitaires écrits en BASIC.



Afin de limiter les risques d'erreurs de transcription, nous avons dû définir autour des réseaux de Pétri un certain nombre de conventions graphiques couvrant toutes les particularités que nous avons développées au chapître II et solutionnées au chapître III. Ces particularités se résument ainsi :

- Le RDPI doit être réceptif et non sensible |38| : toutes les actions doivent être inconditionnelles.

- Les actions impulsionnelles telles que les télémesures doivent être exclusivement décrites autour des places. Ce type de places est automatiquement franchie afin de n'émettre qu'une seule fois la télémesure associée.

- Les places de synchronismes définies au chapître précédent ne doivent pas posséder des actions.

II. 1 - LOGICIEL DE PREPARATION

Ce logiciel permet au programmeur de définir avant toutes autres choses les conditions du travail du microprocesseur : la définition des lignes d'entrée/sortie, la préparation des registres de fonctions spéciales |Timers, registres de contrôle...|, la définition du dialogue avec un dispositif hiérarchiquement plus élevé, l'introduction des places initiales du RDPI qui vont être marquées dès la mise sous tension, et enfin la préparation des temporisations utilisées dans le réseau. Nous allons examiner dans le détail les différentes procédures de préparation énoncées ci-dessus. Ces procédures correspondent à notre microcontroleur utilisé.

II. 1. 1 - Définition des lignes entrée/sortie

Pour un microprocesseur INTEL 8751, les quatres ports E/s parallèles peuvent être définis. Pour celà, le concepteur est amené à symboliser chaque ligne de chaque port utilisé pour le pilotage de son processus. En effet, chaque ligne doit être symbolisée par trois caractères |ex : EVF pour notre application| et définie en entrée ou en sortie comme nous le présentons par l'organigramme ci-dessous |Fig. IV.3|.

II. 1. 2 - Préparation du processeur

Il s'agit pour l'utilisateur de définir les modes de travail du microprocesseur : préparation des registres de fonctions speciales du microcontroleur.L'utilisateur introduit l'octect d'initialisation de chaque registre en suivant les explications donnees par le logiciel de PREPARATION

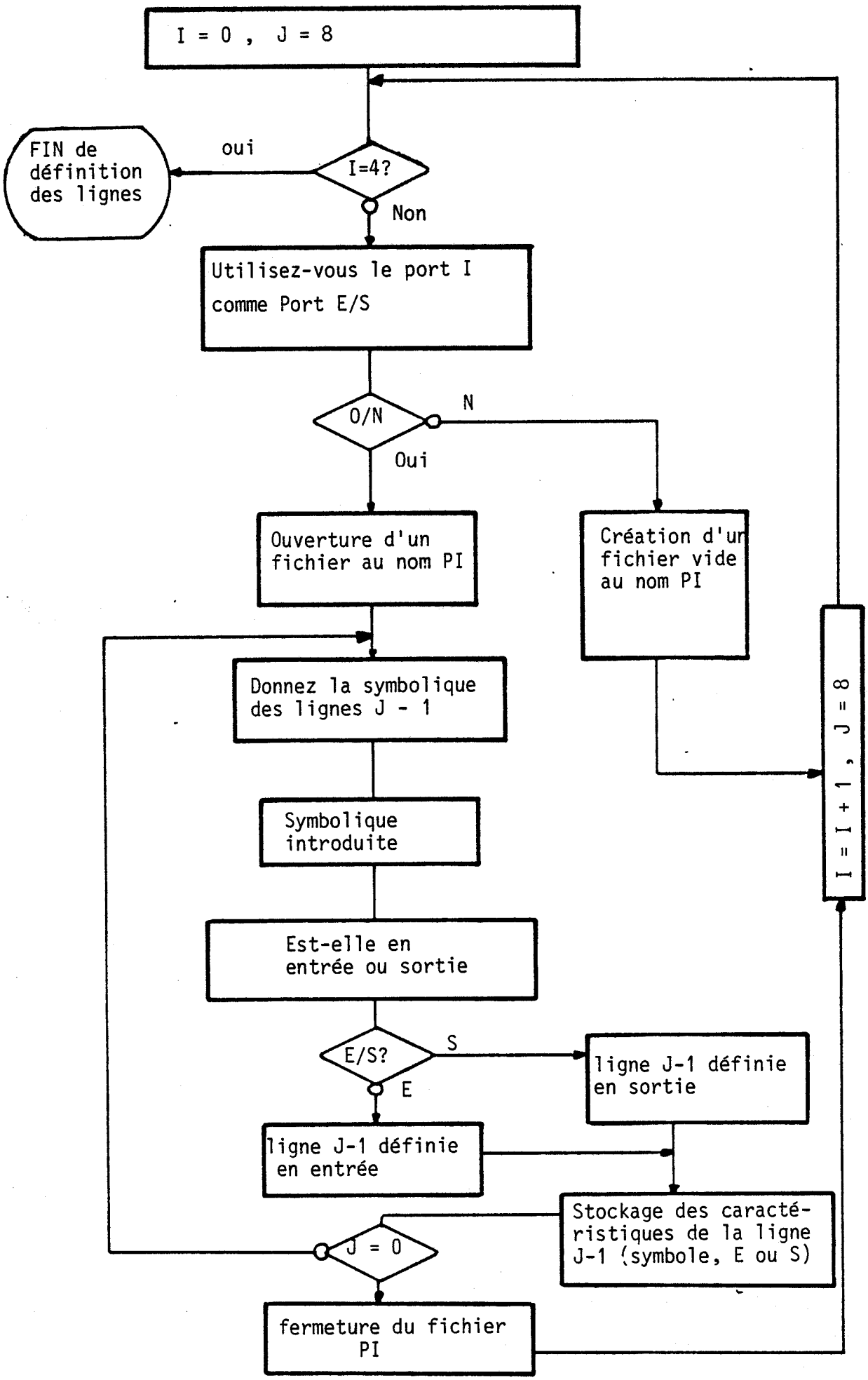


Fig. IV.3 : Affectation des lignes d'entrée et de sortie

II. 1. 3 - Initialisation du réseau de Pétri

Cette fonction permet d'introduire les places initialement marquées dans le RDPI. Le concepteur ne peut ici introduire que des places pivot |ex : PL 11|. Le traducteur retrouvera les PAS associées aux places initiales.

Les adresses origine et les tailles des deux piles |lecture et écriture| |CH. III. II.2| sont fixées automatiquement par le traducteur dans la zone de mémoire RAM intégrée dans le microcontrôleur.

II. 1. 4 - Préparation des temporisations

Le concepteur a la possibilité de choisir dix valeurs de temporisation différentes comprises entre 1/100 et 2,55 s. Pour celà, il est amené à numéroter les temporisations différentes dans son RDPI de 00 à 09. Ainsi, lors de la préparation des temporisations, il introduira les valeurs utilisées dans l'ordre (de 00 à 09) en indiquant les valeurs en un multiple de 1/100 s.

II. 1. 5 - Le dialogue microprocesseur/dispositif extérieur

Cette partie de préparation est optionnelle. Le concepteur peut envisager d'instaurer un dialogue entre son microprocesseur et un dispositif extérieur hiérarchiquement plus élevé et selon un protocole d'accés défini par ailleurs.Réf. |13|. Il doit pour cela définir les télécommandes particulières telles que : le mot de passe qui permet de valider la réception des télécommandes, la télécommande d'initialisation nécessaire pour l'arrêt instantané du système de commande par le dispositif extérieur et enfin, la télécommande d'appel des télémesures, celles-ci étant émises sous forme de paquet.

Les télécommandes et les télémesures sont ici gérées sous interruption et sont formées chacune par un octet. Dans le cadre du développement du procédé de mise en sécurité du microcontrôleur (31,9) il est prévu en ce qui concerne les télétransmissions, de garder le même protocole d'accés, de supprimer les interruptions et de coder les messages | 34, | en vue de s'affranchir des erreurs de transmission. Ceci déborde du cadre de notre travail, toutefois ces dispositions ne constituent qu'une modification mineure du progiciel. Néanmoins, il est important de signaler que la liste des messages d'exploitation -télémesures et télécommandes- sont introduites par l'utilisateur d'une manière symbolique. La taille de chaque symbole n'excède pas dix caractères.

Pendant la procédure de préparation, le concepteur peut à tout moment modifier les données introduites auparavant localement. A chaque fin de préparation les données sont visualisées à l'écran pour permettre à l'utilisateur de les valider et les stocker par la suite.

Ci-dessous nous présentons les différentes données caractérisant la préparation pour une application donnée.

TEMPORISATIONS

SYMBOLS	DUREE EN SECONDE
TT 0 0	0800ms
TT 0 1	0100ms
TT 0 2	1000ms
TT 0 3	0300ms
TT 0 4	1200ms

PROCESSEUR

TIMER1 (TH1)	=00H	TIMER0(TH0)	=00H
TMOD	=40H	TCON	=50H
SCON	=50H	IE	=90H
IP	=80H		=90H

PLACES INITIALES : PL00

DIALOGUE :

MOT DE PASSE	=66H
MOT D'INITIALISATION	=55H
MOT D'APPEL DES TELEMESURES	=33H

II. 2 - EDITION DU RDPI SAUF ET VIVANT

L'édition du RDPI se fait place par place où à chaque place sont associées les sorties |mots de commande|, les transitions descendantes et leurs tests et les places de sortie. L'éditeur créé, permet l'introduction d'un RDPI possédant jusqsu'à 100 places et autant de transitions. Cependant, il nécessite les quelques préparatifs préliminaires suivants :

- validation du RDPI |sauf et vivant|
- numérotation des places et transitions |00 à 99|
- présentation des commandes en symbolique
- présentation des entrées en symbolique
- séparation de places pivot, de synchronisme double et de réinitialisation.

Sous édition, le concepteur peut éditer un nouveau RDPI, modifier un ancien, appeler la procédure de préparation |CH IV.II.1| ou lancer la procédure de traduction d'un RDPI précédemment introduit sous édition en programme assembleur du microprocesseur utilisé. Le logiciel d'édition est indépendant du microprocesseur utilisé mais fait appel aux fichiers créés sous préparation. Il occupe 45 K octet. Ci-dessous nous schématisons l'accés à partir des clés fonctions du calculateur HP 9836 aux différentes procédures.

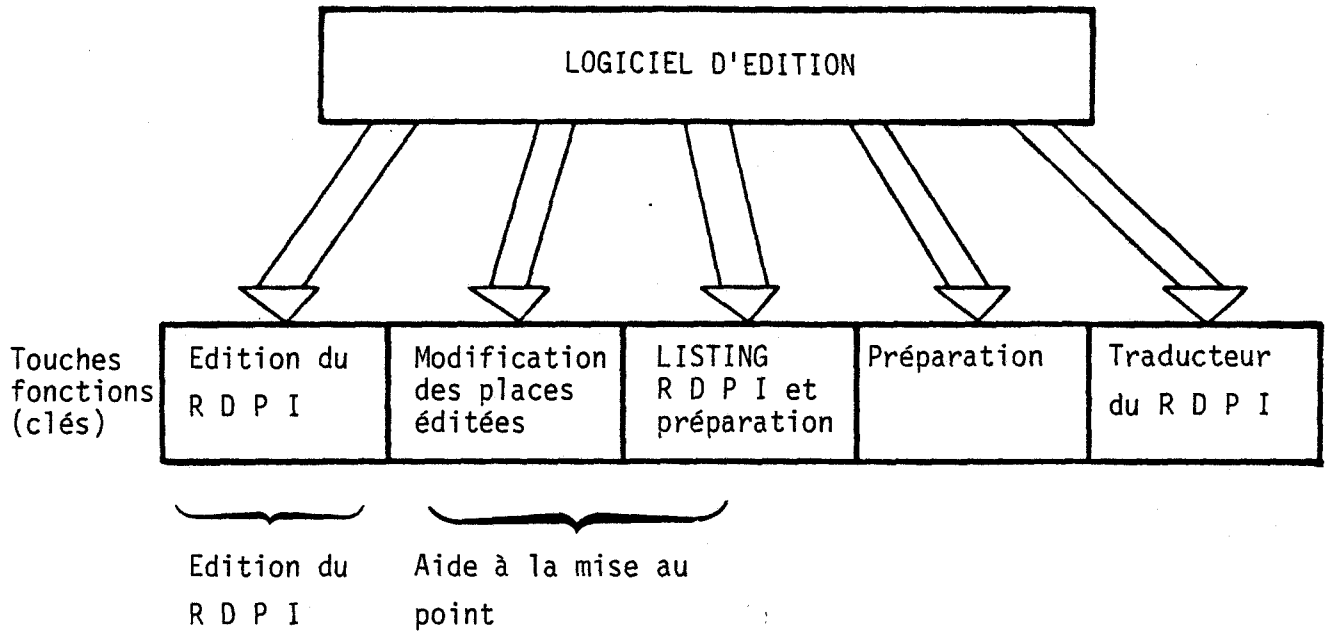


Fig. IV.4

II. 2. 1 - Edition du RDPI autour des places

Chaque place éditée, est écrite sous forme de phrase qui sera le noyau à partir duquel le traducteur générera le programme assembleur. L'écriture de la phrase se fait d'une manière transparente pour l'utilisateur. Ce dernier n'introduira sous éditeur que les numéros des places et des transitions et les symboliques des commandes et des capteurs. Chaque phrase éditée est stockée séparément sous le même nom du fichier associé au RDPI de telle façon qu'elles peuvent être appelées individuellement pour des fins de modifications.

L'édition d'une place se compose de trois blocs : l'introduction de la place |représentation physique et les actions décrites autour|, la description des transitions descendantes |la représentation de la transition, la description des tests associés et les places de sortie| et le stockage après le traitement syntaxique. Ci-dessous, nous avons schématisé simplifiée l'édition complète d'une place.

L'organigramme montre bien que l'utilisateur peut contrôler et intervenir aux différents niveaux de la construction de la phase.

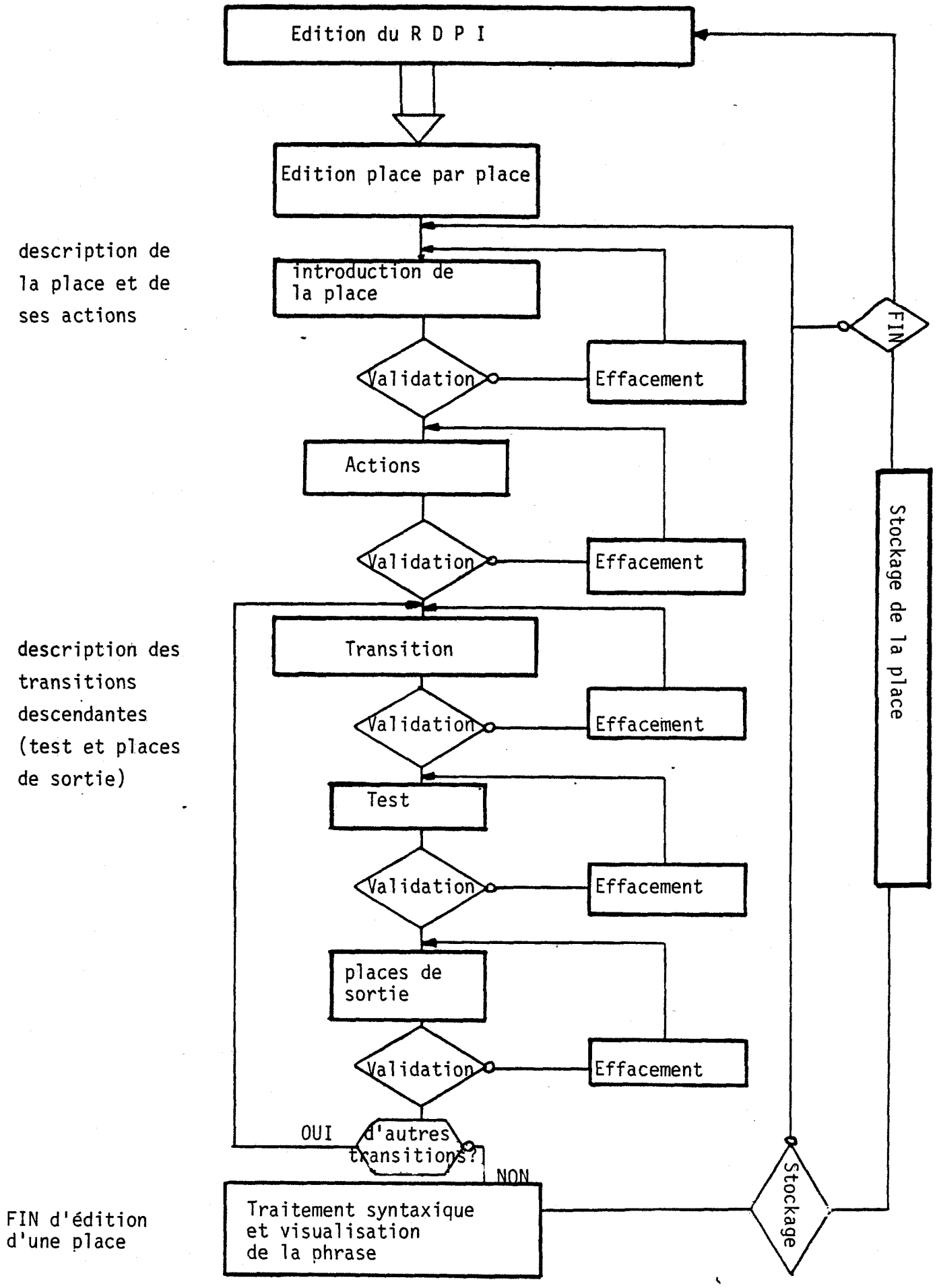

Edition du R D P I
Edition place par place
description de la place et de ses actions
introduction de la place
Validation
Effacement
Actions
Validation
Effacement
description des transitions descendantes (test et places de sortie)
Transition
Validation
Effacement
Test
Validation
Effacement
places de sortie
Validation
Effacement
OUI
d'autres transitions?
NON
FIN d'édition d'une place
Traitement syntaxique et visualisation de la phrase
Stockage
FIN
Stockage de la place

Fig. IV.5

II. 2. 11 - Bloc d'introduction des places

a) Places

Trois types de places peuvent être décrites et représentées par un module de programme appelé "PAS".

- place pivot notée PL
- place double |pivot et de synchronisme| PD
- place de réinitialisation PI.

L'utilisateur choisira la place qu'il veut décrire par les touches fonctions |clés| comme le montre la figure IV-6. Il introduira par la suite le numéro de la place et choisira par les touches fonctions les actions à définir autour de la place.

b) Actions

Deux types d'actions peuvent être décrites autour de la place :

. Commande logique

Pour les commandes logiques, l'utilisateur est amené à préciser le nombre de lignes de sortie à mettre ou à maintenir actives |5V| et à introduire par la suite la symbolique associée à chacune d'eux. |ex : EVF pour l'application précédemment décrite|.

A partir de la symbole, le programme éditeur retrouvera le port et la ligne à laquelle la commande symbolisée est liée. Lorsque le nombre de lignes précisé est retrouvé, l'éditeur formera les mots de commande à émettre sur les ports E/S quand la place à laquelle ces mots sont associés est marquée. Les lignes d'entrée des ports sont, bien entendu, masqués lors d'émission des mots de commande.

Si l'éditeur ne retrouve pas une symbolique dans les fichiers ports créés sous la préparation |CH. IV-II.1|, il générera le message "symbolique non trouvé" et fera visualiser à l'intention de l'utilisateur les différentes lignes des ports avec leurs symboliques. L'utilisateur pourra alors corriger la dernière symbolique.

Le travail en symbolique permet à l'utilisateur pendant l'édition du RDPI de ne pas s'encombrer des différents schémas de cablages commandes/ ports du microprocesseur. Ceci est aussi valable pour les lignes d'entrée.

. Commandes impulsionnelles |Télémesures|

Chaque télémesure est représentée par un octet. L'utilisateur introduira l'octet représentatif de la télémesure, qui sera émise à travers le port série du 8751. La télémesure est stockée une fois dans une pile des télémesures dès que la place à laquelle cette télémesure est associée est marquée. Pour cela, une place possédant une télémesure doit avoir une seule transition descendante franchissable sans condition. En effet, si la place reste marquée pendant un nombre N de cycle, la télémesure associée sera stockée n fois de suite. Ceci est dû au fait que notre éditeur n'autorise pas les actions conditionnelles.

Après validation des actions, le début de la phrase en cours de construction est visualisé. A ce niveau d'édition le noyau |phrase représentative d'une place| se retrouve écrit de la manière ci-dessous.

Places	Actions logique, impulsionnelles		
PLxx	SO Ø yy		
ou	ou/et		
PDxx	SO1yy	ou	TMyy
ou	SO2yy		
PIxx	SO3yy		

xx = numéro introduit par l'utilisateur |ex : 10|

yy : mot de commande formé par l'éditeur à partir des symboles de ligne de commande

0 à 3 : sont les numéros du port

ex : PL 10 SO Ø FØ SO140...

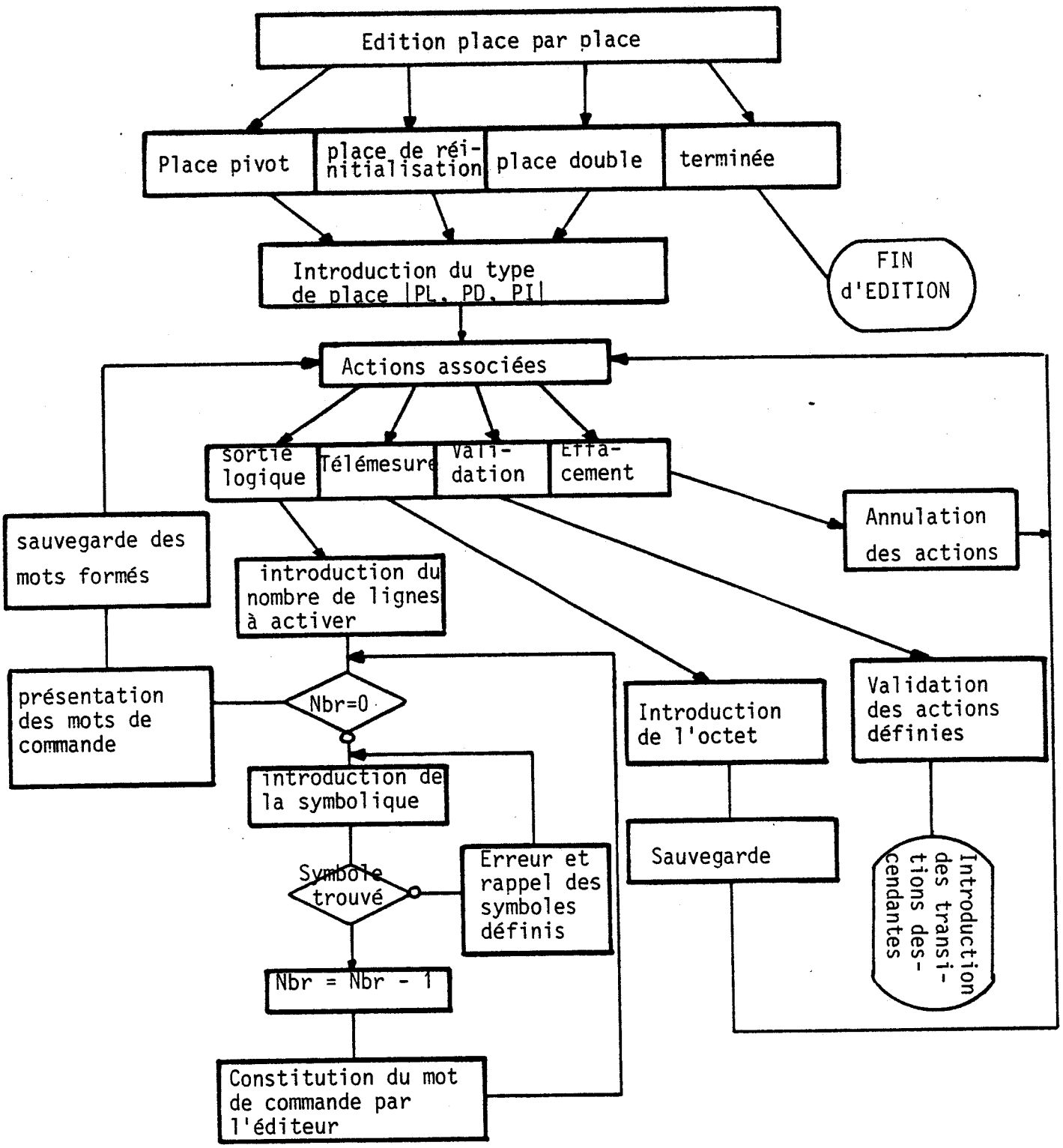

Edition place par place
Place pivot
place de réi-nitialisation
place double
terminée
Introduction du type de place |PL, PD, PI|
FIN d'EDITION
Actions associées
sortie logique
Télémesure
Vali-dation
Effa-cement
Annulation des actions
sauvegarde des mots formés
introduction du nombre de lignes à activer
présentation des mots de commande
Nbr=0
Introduction de l'octet
Validation des actions définies
introduction de la symbolique
Sauvegarde
Symbole trouvé
Erreur et rappel des symboles définis
Introduction des transi-tions des-cendantes
Nbr = Nbr - 1
Constitution du mot de commande par l'éditeur

Fig. IV.6

II. 2. 12 - Bloc de description des transitions descendantes

* Transitions

Dix transitions descendantes à une place peuvent être transcrites. Chaque transition est numérotée |00 à 99| dans le RDPI. Ils lui sont associés, les tests |conditions logiques, numériques et/ou temporelles|, et les places de sortie. Nous présentons ci-dessous l'organigramme permettant l'écriture des transitions.

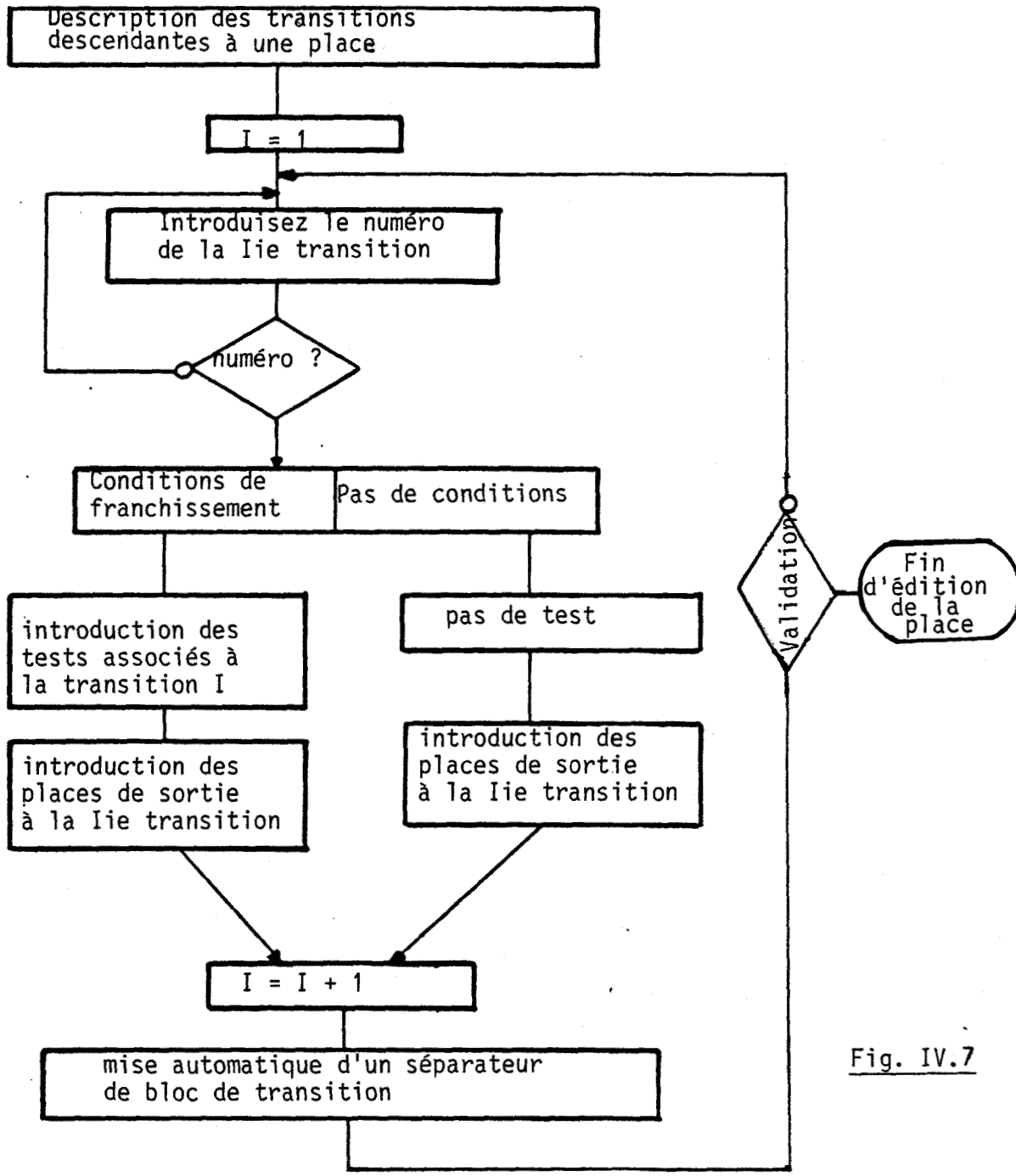


Fig. IV.7

Dans la phase présentée par l'organigramme ci-dessus, l'utilisateur introduira transition descendante par transition à la place en cours d'édition. L'introduction d'une transition regroupe les tests associés et les places de sortie.

BU LILLE

Dans le programme final généré par le traducteur, le traitement des transitions descendantes à une place se fait dans le même ordre que leurs éditions.

Lorsqu'une transition ne possède pas de test, l'utilisateur introduira directement les places de sortie. La transition est franchie dès que la place d'entrée est marquée. Fig. IV.7.

* Tests

Le test associé à une transition peut être une combinaison de tests élémentaires définis ci-dessous :

- test sur un état logique des lignes d'entrée
- une temporisation autour de la transition
- test sur les places d'entrée dites de synchronisme
- test sur les télécommandes reçues à travers le port série du 8751.

La figure IV.8 présente l'accès à l'écriture des différents tests élémentaires.

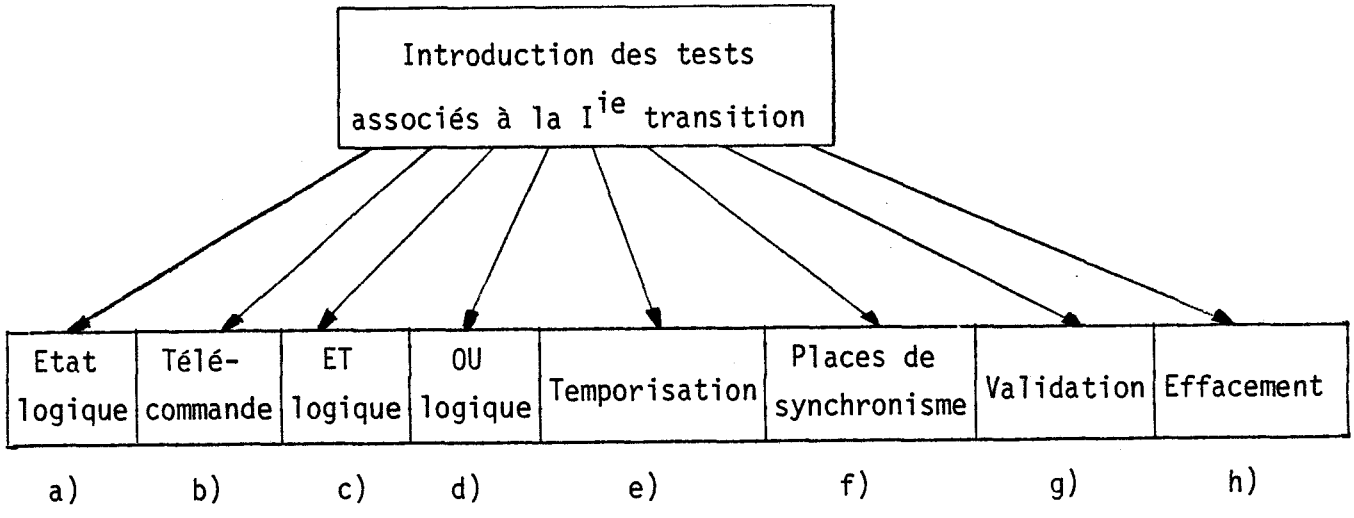


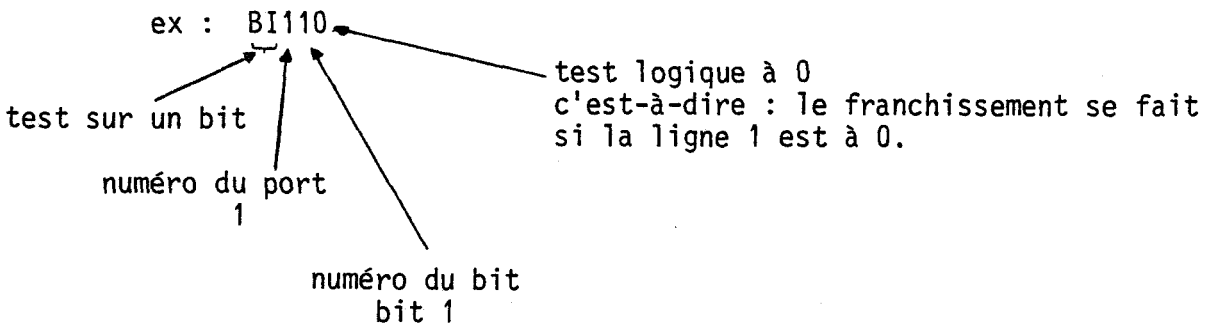
Fig. IV.8

a) Etat logique

Chaque ligne d'entrée a été symbolisée |3 caractères| |ex : PV1 pour notre application|. Il suffit à l'utilisateur d'introduire la symbolique et l'état logique |0 ou 1(5 V)| de la ligne à tester. A partir de la symbolique,

l'éditeur retrouvera la ligne |le numéro, et le port auquel elle est affectée|. S'il ne trouve pas la ligne, un récapitulatif des différents symboles et lignes |numéro et ports| auquelles elles ont été affectées, est visualisé à l'écran, afin de permettre à l'utilisateur de corriger le symbole introduit.

Le test logique s'écrit d'une manière transparente pour l'utilisateur de la façon suivante :



b) Télécommande |test numérique|

Chaque télécommande est représentée par un octet (8 bit). Il suffit d'introduire 0 octet de la télécommande à tester. En effet, chaque télécommande reçue est stockée dans une zone mémoire appelée "BY". Le test de la télécommande se fait par la comparaison de l'octet introduit par l'utilisateur et le contenu de la mémoire "BY". Si les deux octets sont égaux le test est vrai.

Le test s'écrit : BY octet en hexadécimal

Il existe une deuxième version du progiciel en cours de finition où les télécommandes sont introduites d'une manière symbolique, codées sur deux octets à l'aide d'un code linéaire arithmétique.

e) Temporisation

Les temporisations ne sont décrites qu'autour des transitions. Elles sont introduites par leurs numéros (00 à 09) car elles ont été préparées (CH IV.II.1).

Le test s'écrit : TT 02

transition — temporisée — numéro

f) Places de synchronisme

Elles sont décrites comme test pour le franchissement de la transition à laquelle elles sont associées. Pour cela, l'utilisateur aura deux informations à donner :

- le numéro du bit associé à la place de synchronisme
- l'état logique du bit permettant le franchissement de la transition |place de synchronisme marquée (1), démarquée (0)|

Le test s'écrit : PSS ␣ ␣ ex : PSS 11

place de synchronisme

numéro du bit

état logique

c et d) Tests combinés

L'utilisateur peut associer les tests en un OU logique |touches fonctions OU|. Par défaut ou par la touche |ET logique|, il affecte les tests élémentaires en un ET logique.

Lest tests combinés s'écrivent :

+	OU logique	
*	ET logique	BI110*BY40 + TT02...
	par défaut	BI110BY40 + TT02...

h) Effacement

Au fur et à mesure de l'introduction des tests élémentaires, le sous noyau "test" s'écrit. L'utilisateur peut à tout moment annuler le test par la touche effacement. Cette opération reste locale et autour de la dernière transition introduite.

g) Validation

Après validation |clé|, le test écrit est sauvegardé. L'utilisateur peut introduire les places de sortie de la transition en cours d'édition.

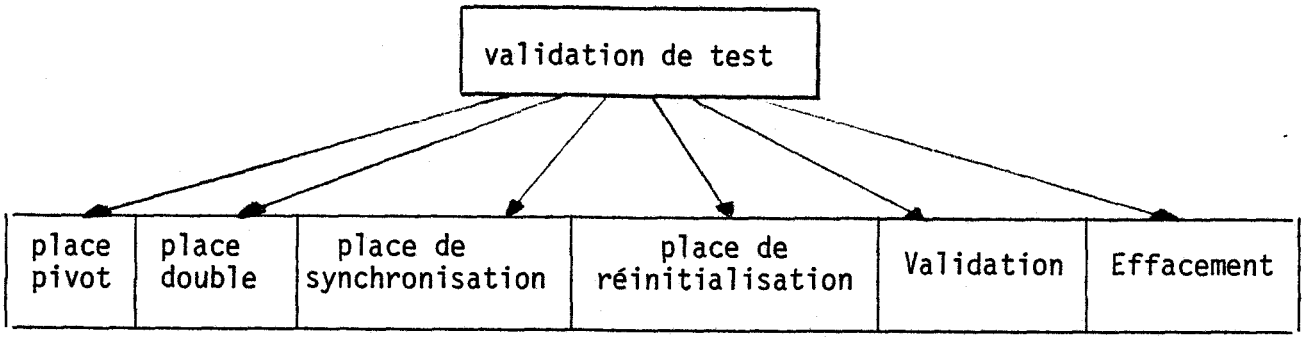


Fig. IV.9

* Introduction des places de sortie

Chaque type de place de sortie est introduit par la touche fonction correspondante. L'utilisateur introduira par la suite le numéro de la place |place pivot, place double et place de réinitialisation| et dans le cas des places de synchronisme, le bit associé et l'état logique auquel le bit est mis (0 démarquage, 1 marquage).

Les places de sortie s'écrivent de la façon suivante :

- place pivot — PL ␣ ↖ numéro
- place de réinitialisation — PI ␣ ↗ numéro
- place de synchronisme — PLS ␣ ␣ ↖ état marqué ou démarqué (1 ou 0) ; numéro
- place double — PL ␣ PLS ␣ ␣ ↖ Etat logique ; numéro ; numéro du bit

Après validation de l'écriture des places de sortie, l'utilisation peut introduire les transitions descendantes restantes ou lancer la procédure de stockage.

II. 2. 2 - Stockage temporaire et traitement syntaxique

La procédure de stockage se compose de deux parties : le traitement syntaxique et le stockage dans le fichier du RDPI.

Le traitement syntaxique de la phrase (place) éditée est fait par décomposition de la phrase en groupement de caractères. Chaque groupe de caractère est analysé afin de déceler les erreurs d'écriture. Cependant, l'utilisateur n'a accès à la phrase que par l'écriture des numéros ou des valeurs |la phrase étant écrite automatiquement par l'éditeur| de ce fait l'apparition des erreurs est minimisée.

L'analyse syntaxique est importante lorsque l'utilisateur désire modifier une phrase|place| éditée sans la réécrire sous édition. La phrase à modifier est appelée et visualisée sur l'écran, l'utilisateur peut la modifier, et lancer la précédure de stockage. Lorsqu'une erreur est rencontrée, l'éditeur visualise le noyau |phrase| en indiquant l'endroit de l'erreur |voir II.2.3|.

Le stockage de chaque phrase se fait dans le fichier portant le nom du RDPI. La place éditée est stockée sous un numéro dans le fichier afin de pouvoir l'appeler individuellement.

Nous avons présenté ci-dessous, le tableau des différents caractères utilisés pour la constitution de chaque noyau.

	Symbols	Numéro	Valeur	N° BIT	N° PORT	ETAT Logique	
Places Pivot	PL	00 à 99	-	-	-	-	
	PD	00 à 99	-	-	-	-	
	PI	00 à 99	-	-	-	-	
Actions	SO	-	00 à FF	-	-	-	
	TM	-	00 à FF	-	-	-	
Transition	TR	00 à 99					
TEST	BI	-	-	0 à 7	0 à 3	0 ou 1	
	BY	-	00 à FF	-	-	-	
	TT	00 à 09	-	-	-	-	
	PSS	-	-	0 à 7	-	0 ou 1	
	*	-	-	-	-	-	
	+	-	-	-	-	-	
Place de sortie	PL	00 à 99	-	-	-	-	
	PD	00 à 99	-	-	-	-	
	PI	00 à 99	-	-	-	-	
	PLS	-	-	0 à 7	-	0 ou 1	
Fin d'une Transition	.	-	-	-	-	-	-
Fin d'une Place	s	-	-	-	-	-	-

Exemple d'une place "PL02" éditée

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 2 PL02TR07BI370PL03.TR08BI160PL04.TR09TT03PL05.&

PLACE  :  PL02                    NUMERO DE STOCKAGE  :          2

      SORTIES A EXCITER      :
      TRANSITIONS                 TESTS et PLACES DE SORTIE
          ---->TR07               BI370PL03.
          ---->TR08               BI160PL04.
          ---->TR09               TT03PL05.&
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

II. 2. 3 - Aide à la mise au point

Il est primordial de pouvoir avoir une trace de l'ensemble des tâches de préparations |CH IV.II| et du RDPI introduit sous édition. L'utilisateur a la possibilité d'avoir un listing sur imprimante de l'ensemble des places accompagné des préparations |définition des lignes E/S, microprocesseur, ... etc|.

Sous édition, deux types de modifications peuvent être utilisés par le concepteur qui a édité son RDPI :

- Modification rapide. Elle consiste à faire appeler la phrase (noyau) et de la modifier directement par retouche au noyau. Après modification, la phrase est stockée sous le même numéro après le traitement syntaxique. Cette opération nécessite une bonne connaissance des composants du noyau (phrase).

- Modification lente. Elle consiste à rééditer la phrase et la stocker sous le même numéro. L'ancienne place est écrasée par la nouvelle. Cette opération est lente mais facilement utilisable.

III - TRANSFORMATION R D P I EDITE / LANGAGE ASSEMBLEUR DU MICROPROCESSEUR OU FONCTION DE COMPILATION

Le traducteur écrit en basic permet l'obtention d'une façon indépendante des deux parties : le module de gestion (moniteur) et le programme de description ou de traitement des pas. Le traducteur se compose de deux parties :

- Le pré-traducteur, qui génère le programme de gestion et organise l'occupation mémoire RAM du 8751.

- Le traducteur des places.

III. 1 - PRE-TRADUCTEUR

C'est un sous programme qui peut être appelé de n'importe quel moment d'introduction du RDPI. Il permet à partir du fichier créé sous préparation (IV-II), l'organisation de la RAM du microprocesseur |affectation de symbols, réservation de mémoire et initialisation du microprocesseur et du RDPI|.
Il génère d'une façon automatique et indépendamment de l'application envisagée par le concepteur, le programme de gestion.

- Phase d'affectation

Elle consiste pour le pré-traducteur à effectuer aux ports E/S aux temporisations, aux places de synchronisme et aux télécommandes des étiquettes et des zones mémoires (registres) (voir annexe).

SOø	EQU	40H	Image du port de sortie Pø
BI1	EQU	21H	Image du port d'entrée P1

De même, chaque registre affecté à une temporisation est changé par la valeur calculée auparavant sous préparation

MOV	TT01, ≠ 14H	chargement de la temporisation N° ø1 par la valeur 14 hexadécimale.

- Dans le cas où l'utilisateur avait désiré au cours de la préparation, l'établissement d'un dialogue avec un dispositif hiérarchiquement plus élevé, le pré-traducteur génère le programme de réception/émission et de stockage des télécommandes reçues et des télémesures à émettre.

- Génération du programme moniteur

Le programme moniteur généré comporte :

- L'initialisation du microprocesseur |timers, ports E/S...|
- L'initialisation du RDPI |définition des piles lecture et écriture, leurs pointeurs et tailles et les places initialement marquées|
- Les routines utilitaires suivantes :

. Marquage des places	"MP"
. Démarquage |franchissement|	"FR"
. Echange des deux piles (synchronisation)	"SYN"
. Envoi des commandes (sortie)	"SORT"
. Lecture des adresses dans la pile de lecture	"CG"
. Stockage des télémesures	"BTME"

La pré-traduction peut être lancée sans la traduction des places. Le programme généré peut être récupéré sur imprimante, écran ou sur le système de développement des microprocesseurs afin d'être consulté.

La figure IV.10 nous montre les différentes possibilités d'avoir des traductions partielles.

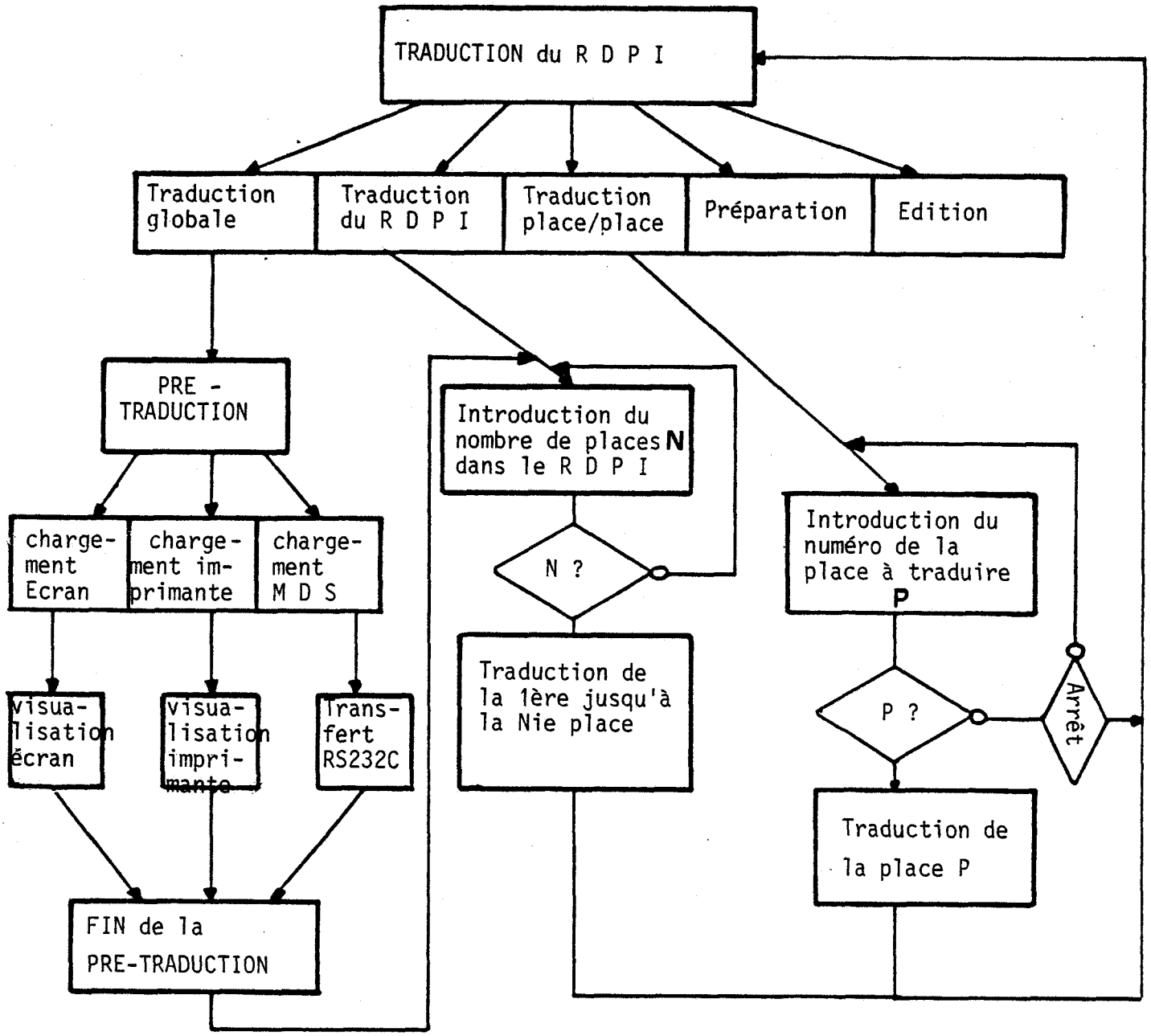


Fig. IV.10

MONITEUR | N PLACES | PLACE P

Description du traducteur

III. 2 - TRADUCTION DES PLACES

La traduction du RDPI édité se fait place par place en utilisant le fichier créé sous édition du nom du RDPI. Chaque place (phrase) est traduite en attribuant à chaque sous ensemble de caractères du noyau (phrase) un ensemble d'instructions en assembleur 8051.

* Traduction du bloc place et actions

Le sous noyau représentatif de la place et des actions s'écrit :

PL numéro SO Valeur TM Valeur ..... TR ...

(1) (2) (3)

La transformation se fait en associant à chaque groupe de caractère un ensemble d'instructions en assembleur comme le montre l'exemple ci-dessous

(1) représente l'adresse de la place

exemple :

PL10 SOø øC SO1FF TM20

(1) (2) (2) (3)

(2) commandes logiques

(3) télémesures à émettre

PL10 :	CALL	MP
	MOV	SOø, ≠ øCH
	CALL	SORT
	MOV	SO1, ≠ øFFH
	CALL	SORT
	MOV	A, ≠ 20H
	CALL	BTME

(MP, SORT, BTME) sont des blocs fonctionnels du moniteur (voir annexe II).

* Traduction des transitions et des tests

Parmi les difficultés rencontrées lors de l'élaboration du traducteur, nous avons été confronté au problème de l'existence de plusieurs transitions descendantes possédant des tests combinés. Cela a pour effet d'augmenter le nombre de sauts dans un même PAS (place traduite) et d'accroître la difficulté de repérage des endroits des sauts.

Pour remédier à cette difficulté, nous avons adopté la technique suivante :

- Le traducteur lit d'abord la transition jusqu'à l'indicateur de fin de transition "." . Il repère par la suite la transition suivante "TR numéro" ou la fin de la phrase "ß".

- Les tests élémentaires combinés sont regroupés selon les blocs en ET logique et ensuite en OU logique. Chaque test élémentaire traduit est repéré par une étiquette appropriée.

ex : TR11BI ø 11 TTo2

```
TR 11      : NOP
BI01 11    : JNB     BIø.1, FIN11
TT02 11    : DJN2    TTø2,  FIN11
               '        '
               '        '
               '        '
               '        '
FIN 11       JMP       CG
```

11 représente le numéro de la transition

* <u>Traduction des places de sortie</u>

La traduction des places de sortie s'opère de la même façon que ci-dessus : a chaque groupe de caractères est associé un ensemble d'instructions assembleur 8051.

ex : ÷ - - PL10 PD20 PLS31 PI40

```
MOV     DPTR, ≠ PL10
CALL     MP
MOV     DPTR, ≠ PD20
CALL     MP
SETB    PSS.3
MOV     DPTR, ≠ PI40
CALL     MP
```

A l'indicateur de fin de phrase "ß" correspond l'instruction : JMP CG (*) qui permet d'aller lire les autres adresses.

* Lecture des adresses dans la pile mémorisant les places marquées.

IV - SURETE DE FONCTIONNEMENT DE L'OUTIL INFORMATIQUE D'AIDE AU DEVELOPPEMENT

GENERALITES

La confiance que l'on peut accorder à notre outil informatique ne se justifie pas par rapport à sa vie opérationnele, mais par une analyse des différentes composantes matérielles et logicielles mises en oeuvre.

Cette analyse se situe à deux niveaux :

- Support matériel et logiciel de l'outil

L'emploi de machines performantes types HEWLETT PACKARD et INTEL ayant des logiciels largement diffusés et donc ayant fait leur preuve (BASIC HP et ASM 51).

- Ecriture du PROGICIEL

La rédaction du progiciel sous forme de programmation structurée et modulaire apporte souplesse et crédibilité à notre outil informatique. Nous allons montrer par la suite les phases d'analyses de la sûreté de fonctionnement relatives à notre PROGICIEL.

IV. 1 - SURETE DE FONCTIONNEMENT DES LOGICIELS INTERACTIFS

La sûreté de fonctionnement de ces logiciels peut être évaluée par leur capacité à remédier aux problèmes suivants :

* Transcription non ambigüe d'une place en une chaîne de caractères (à une chaîne ne doit correspondre qu'une et une seule place).

* Chaque chaîne de caractères comprend un nombre suffisant de symboles |informations| pour définir entièrement la place associée.

Cette capacité de minimiser les erreurs de transcription et de correspondance place/chaîne se situe à trois niveaux :

a) Interactivité

L'introduction de données caractéristiques d'un RDPI se fait d'une façon automatique rendant les risques d'erreur ou d'oubli extrèmement minimes. En effet, l'utilisateur est fortement guidé pendant les différentes phases d'édition : introduction des places, des transitions, des tests et des places de sortie. Pendant l'édition et à différents niveaux, l'utilisateur n'a pour tâches que le suivi des explications et des menus affichés, et l'appui sur la touche fonction correspondante à la fonction élémentaire qu'il désire éditer |ex : test logique|.

Autrement dit, la formation du noyau (phrase) correspondant à une place, se fait par une suite d'étapes successives où la construction s'effectue par rajout de caractères associés spécifiquement à une fonction élémentaire. Chaque fonction élémentaire est éditée par une touche fonction du calculateur.

b) Vérification syntaxique

La semantique utilisée contient un nombre relativement faible de caractères. Ceci rend l'analyse syntaxique plus aisée et permet de diversifier les niveaux d'analyse. Cette analyse syntaxique se fait sur trois niveaux :

- Pendant la construction du noyau

A chaque introduction d'une fonction élémentaire, celle-ci est affichée afin d'être analysée par l'opérateur en vue de sa validation.

- Avant le stockage

Avant le stockage de chaque phase, celle-ci subit une analyse syntaxique. Lorsqu'une erreur est détectée, la phase est visualisée en indiquant l'endroit de l'erreur. L'utilisateur peut ainsi la rectifier ou rééditer sa phrase.

- Avant la traduction en langage assembleur

Chaque phrase à traduire est à nouveau décomposée et analysée automatiquement avant de générer le langage assembleur correspondant. Ce niveau d'analyse est différent des autres car il fait appel à un logiciel indépendant des logiciels interactifs.

c) Modularité

Le programme permettant la transcription RDPI/chaîne de caractères est décomposé en plusieurs sous-programmes courts réalisant une fonction élémentaire. Chacun ne peut être appelé que par une touche fonction spécifique du calculateur. Cette décomposition en sous-programme facilite la maintenabilité du logiciel et augmente sa clarté.

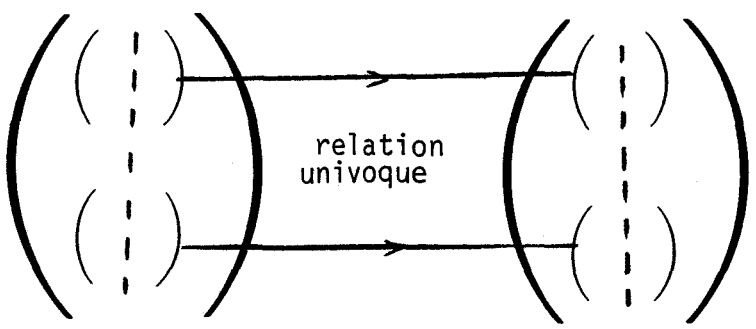
Chaque sous-programme utilise ses propres ressources et indicateurs.

IV. 2 - SURETE DE FONCTIONNEMENT DU TRADUCTEUR OU COMPILATEUR

La traduction est une phase importante où l'utilisateur n'intervient plus. Toutefois celui-ci est prévenu dans le cas où une erreur est détectée pendant la transformation RDPI/langage assembleur. Elle permet en tenant compte de la fonction d'assemblage (ASM51 INTEL), d'obtenir directement le code objet exécutable par le microprocesseur. Nous allons examiner les différents points qui contribuent à susciter la confiance dans le traducteur.

- Transformation phrase/langage assembleur

Cette transformation est opérée après décomposition de la chaîne de caractères (noyau) en des sous-chaînes auxquelles la traduction fait correspondre un ensemble d'instructions en langage assembleur. On établit ainsi une relation univoque en faisant correspondre à un élément de la chaîne un et un seul jeu d'instructions en assembleur.



Noyau ou chaîne de caractères

Ensemble de jeu d'instruction en assembleur 8051

- Etiquettage du langage assembleur

La phase d'assemblage du langage généré par le traducteur, ne doit pas être interrompue. Nous avons présenté l'intérêt de ne générer que des jeux d'instructions limités et qui ont été stockés au préalable. Ceci écarte toute mauvaise interprétation des instructions en assembleur, mais n'empêche pas les problèmes relatifs aux repérages des adresses de sauts dans le programme généré. Pour ce faire, les adresses de sauts sont étiquettées de telle manière que le langage d'assemblage affecte de façon univoque à chaque étiquette une et une seule adresse mémoire dans le microprocesseur. Chaque "PAS" possède ses propres adresses de sauts étiquettés de façon différente par rapport aux autres "PAS" de programme. Le PAS ci-dessous montre comment ont été étiquettées les adresses de saut.

```
Point d'entrée du module
                                    Durées d'execution
                                     en microseconde
PL12:    CALL    MP                  ; 10;
         MOV      SO1,#040H          ; 2;
         MOV      SO3,#008H          ; 2;
         CALL    SORT                ; 6;
TR121:   NOP
         JB      BI1.2,TR221         ; 2;
         JNB     BI1.1,TR222         ; 2;
         JNB     BI3.6,TR223         ; 2;
FR12:    CALL     FR                 ; 6;
         MOV      DPTR,#PL13         ; 2;
         CALL     MP                 ; 10;
         MOV      R3,#030H
         JMP      CG
TR221:   NOP
         NOP
TR222:   NOP
         NOP
TR223:   NOP
         JB      BI3.6,FI121         ; 2;
FR22:    CALL     FR                 ; 6;
         MOV      DPTR,#PL15         ; 2;
         CALL     MP                 ; 10;
         MOV      R3,#033H
         JMP      CG
FI121:   MOV      R3,#021H
         JMP      CG

Point de sortie du module
```

Module de programme généré par le traducteur

CONCLUSION

CONCLUSION

L'objectif de l'étude, proposée dans cette thèse, est d'apporter une solution au problème de la sûreté du logiciel à implémenter dans les systèmes de commande microprogrammés amenés à fonctionner en sécurité.

Dans cette perspective, nous nous sommes attachés à présenter une démarche simple et rigoureuse allant du choix du modèle de description et d'analyse des automatismes au développement du logiciel spécifique, en respectant les contraintes de sécurité fixées par le cahier des charges. Cette démarche a été suivie pour la conception de notre système de commande-contrôle d'un ensemble de portes véhicules du métro type "VAL".

En résumant les principaux point qui ont guidés ce travail, on peut dire que le réseau de Pétri interprété présente un intérêt considérable par le fait qu'il permet la description aisée du parallélisme qui devient fréquent dans les automatismes. Cependant, nous avons introduit la notion de place de réinitialisation qui ne diminue pas la puissance du modèle d'origine, mais apporte une certaine souplesse. En effet, la place de réinitialisation permet la récupération de tous les jetons dans le RDPI quand elle est marquée, pour les redistribuer après à travers une transition descendante. Cette notion allège considérablement le modèle et apporte une solution au problème de la sécurité liée à l'arrêt d'urgence et à la reconfiguration de l'automatisme dans un état défini par le concepteur. Associée à celà, une méthode sûre permettant de reproduire en langage assembleur le modèle, le plus fidèlement possible, et apte à gérer la sécurité du microprocesseur portant le modèle, nous pensons que cette démarche constitue une voie pour l'amélioration de la sécurité dans les systèmes de commande microprogrammés.

.../

/...

Nous avons par la suite, automatisée cette démarche en créant un outil informatique d'aide au développement des programmes décrits fonctionnellement par un réseau de Pétri interprété. Cet outil soulage son utilisateur de la phase de traduction du modèle en langage assembleur et permet le passage direct du modèle construit à son implantation sur le microprocesseur que l'utilisateur adopte pour le pîlotage de son processus.

Nous estimons que des travaux restent à effectuer sur l'outil informatique, nous pensons plus particulièrement à son élargissement à d'autres familles de microprocesseurs autre que la famille INTEL 8051. Ceci ne constituera que quelques modificactions mineures au niveau du TRADUCTEUR. Nous pensons par exemple à d'autres ajouts dans l'outil afin d'augmenter sa crédibilité telle que la génération en parallèle du langage assembleur du graphe représentatif du modèle d'origine, ainsi l'utilisateur peut vérifier la conformité du graphe généré à son modèle initial.

ANNEXES

ANNEXE I

DESCRIPTION DE LA COMMANDE-CONTROLE DES PORTES VEHICULES DU METRO TYPE 'VAL'

A N N E X E I

I - SEQUENCES NORMALES DE COMMANDE - CONTROLE

Les séquences de commande qu'effectue chaque microprocesseur dans le cas de fonctionnement normal (absence de pannes) sont simples et peuvent être décomposées en trois phases : la phase de contrôle de l'état du mécanisme de porte, le véhicule étant en ligne |en marche|, la séquence d'ouverture de porte en station et le contrôle du déroulement de la séquence, enfin la séquence de fermeture et de verrouillage des portes en station.

Avant de décrire ces phases de fonctionnement, nous présentons l'état initial de l'automatisme. Il définit l'état que prendra le processeur commandant vis-à-vis du procédé commandé dès la mise sous tension ou, dans certains cas, à l'arrêt de la commande par des procédures particulières |arrêt d'urgence|. Généralement c'est un état de repos de l'automatisme (actionneurs non activés).

(Fig. A.I) présente l'état initial décrit par le RDP (fig. ci-dessous) modélisant le fonctionnement des portes avec les procédures de détection de pannes. Cet état prend en compte toute demande d'évacuation d'urgence sur la porte commandée (porte n° 1) et celle contrôlée (porte n° 2).

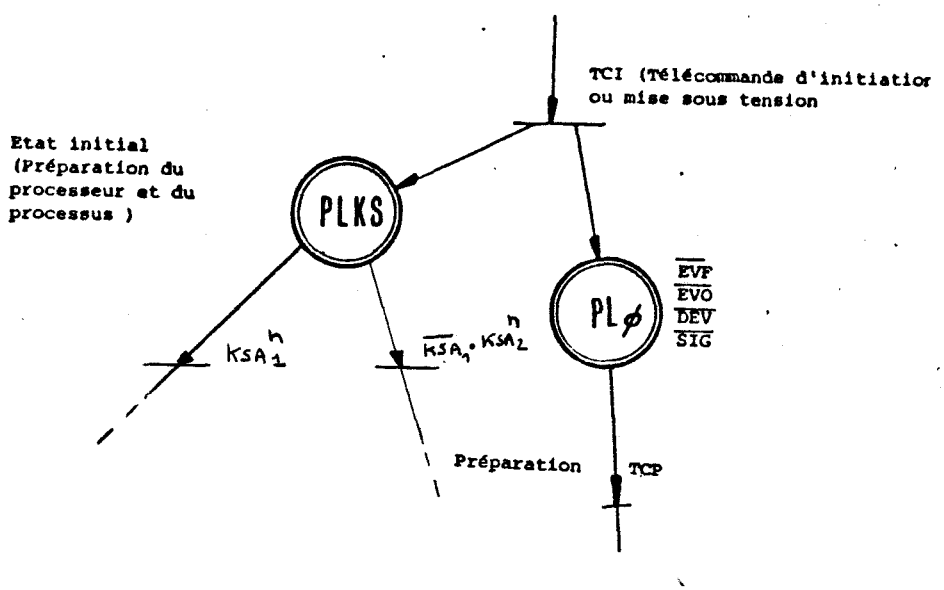


Fig. A.I

A l'état initial, le microprocesseur est réceptif aux trois évènements suivants :

a) La télécommande de préparation "TCP" qui lui permet de se configurer dans l'état porte fermée verrouillée |ch. II. AI-1.1|.Cette télécommande est envoyée par le "DCD" vers les six microprocesseurs. Chaque microprocesseur aquitte la réception par émission de la télémesure "porte N°x fermée verrouillée".

b) La demande d'évacuation d'urgence sur sa propre porte, modélisée par la transition "KSA_1^n" qui est prioritaire sur l'évènement C.

c) La demande d'évacuation d'urgence sur la porte 2 opposée dans le véhicule modélisée par la transition ($\overline{KSA}_1^n$. KSA_2). |ch. II. fig. A9|.

I. 1 - Séquence de contrôle, porte fermée verrouillée et véhicule en ligne

C'est une séquence atteinte de deux manières : de l'état initial par réception de la télécommande de préparation "TCP", et de la fin de la séquence de fermeture des portes en station après validation de verrouillage des vantaux et la séquence de non entrainement | AI - I.3-b|. Lorsque le véhicule est en ligne chaque microprocesseur effectue les tâches suivantes |fig. A3|.

- Le maintien de la fermeture des vantaux par excitation de l'électrovalve de fermeture "EVF".

- Le contrôle périodique des différents capteurs afin de détecter les pannes par analyse séquentielle du vecteur d'état et par ordre d'importance des capteurs |fig. A3 |.

- Le contrôle du verrouillage et de fermeture de la porte opposée dans le véhicule sur autorisation du "DCD". En effet, lorsque le "DCD" ordonne l'ouverture des portes du côté station par l'envoi d'une télécommande d'ouverture "TCO" aux microprocesseurs concernés |ex : côté impair|, il émet une télécommande de masquage "TCM" aux autres microprocesseurs |ex : côté pair|

de telle sorte que ces derniers ne contrôlent plus le verrouillage et la fermeture des portes opposées (ex : impair) puisque celles-ci vont s'ouvrir normalement. Pour annuler ce masquage, le "DCD" émet vers les microprocesseurs (côté pair) une télécommande d'autorisation "TCA" lorsque toutes les portes sont fermées verrouillées au démarrage de la station. Le contrôle des portes opposées ne s'effectue qu'en ligne.

I. 2 - Cycle d'ouverture en station

Lorsque la rame entre en station, le "DCD" interprète la télécommande synthétique "TCOPG" : télécommande d'ouverture porte gauche qu'il reçoit du "DCA" et envoie la télécommande spécifique "TCO" aux microprocesseurs concernés. Ces derniers effectuent la tâche en deux temps :

- 1°) Deverrouillage

Le déverrouillage s'effectue par excitation de l'électrovalve de verrouillage "DEV". Cette commande est maintenue jusqu'à l'ouverture complète des vantaux.

- 2°) Ouverture

Par excitation de l'électrovalve d'ouverture EVO et de déverrouillage "DEV". Les vantaux s'ouvrent par admission d'air comprimé dans la chambre d'ouverture.

Parallèlement à cela, le contrôle de la commande et de l'ouverture des vantaux se fait par l'analyse séquentielle de l'état des capteurs de position |fig. A4|.

En fin d'ouverture, le déverrouillage est desexcité, la commande d'ouverture est maintenue par excitation de l'"EVO" et la télémesure (message) "porte N° 1 est ouverte" est envoyée par le microprocesseur au DCD |fig. A5|.

I. 3 - Cycle de fermeture

Le cycle de fermeture normale |cas où la détection d'obstacle est opérative| se décompose en deux phases :

a) phase de fermeture

Dès réception de la télécommande de fermeture "TCF" le microprocesseur effectue la séquence suivante.

- Début de fermeture sans détection d'obstacle pendant 0,6 s |fig. A6|
- puis fermeture avec détection d'obstacle, comptage du nombre de réouverture sur obstacle et envoi de la télémesure "obstacle à la fermeture" toutes les deux réouvertures
- arrêt de détection d'obstacle apèrs le verrouillage des vantaux.

Parallèlement à cela, le microprocesseur contrôle la fermeture des vantaux, par l'analyse séquentielle de l'état des capteurs de position |fig. A6|.

b) Phase de non entraînement

Lorsque le microprocesseur valide la fermeture et le verrouillage des vantaux, il déclenche le dispositif de non entraînement dans le cas où le système de verrouillage est en bon état de marche.

La phase de non entraînement consiste à desexciter l'électrovalve de fermeture pendant dix secondes ainsi la porte peut s'ouvrir manuellement ou sous l'effet du démarrage de la rame, cette ouverture est partielle car la porte reste sous contrôle du verrou à l'aide d'un système à ressort dont la distance d'ouverture est réglable.

Le but est de permettre aux objets de faibles épaisseurs (ex : écharpe, sac...) coincés entre les vantaux de se libèrer lors du démarrage de la rame et permettre ainsi de se prémunir de ces types d'accidents mettant parfois des vies humaines en péril.

Le système central "DCD" est averti du déclenchement de ce dispositif par l'émission de la télémesure "porte N° 1 non entrainante". Le "DCD" peut alors autoriser le démarrage de la rame.

- Lorsque le système de détection d'obstacle est défectueux et ne peut plus assurer sa fonction, la phase de fermeture se fait en mode dégradé : fermeture en battement (fermeture et réouverture) jusqu'au verrouillage complet des vantaux. Le "DCD" est averti par l'émission de la télémesure "fermeture dégradée porte N° 1".

En fin de phase de non entraînement, le microprocesseur revient à la séquence de contrôle, porte fermée verrouillée et véhicule en ligne, en émettant vers le "DCD" la télémesure "porte N° 1 fermée".

II - PRISE EN COMPTE DE LA DEMANDE D'EVACUATION D'URGENCE

Dans un système de transport tel le "VAL", la prise en compte de la demande d'évacuation d'urgence doit être instantanée car de là dépend la sécurité des voyageurs. Il est donc primordial pour chaque microprocesseur d'avoir à l'oeil l'information relative à la demande d'évacuation. Pour ce faire, l'évacuation d'urgence est traitée parallèlement aux différentes séquences décrites ci-dessus.

Initialement, on commence par deux marqueurs, l'un permettant la gestion des différentes séquences (PLØ) et l'autre réservé à la demande d'évacuation d'urgence (PLKS) |fig. A3|.

Ci-dessous, nous présentons la partie du graphe décrivant le traitement de l'évacuation d'urgence par chaque microprocesseur. Prenons l'exemple du microprocesseur commandant la porte N° 1 et traitant les informations de sécurité de la porte N° 2 comme l'indique la figure A8

- D'abord l'information (KSA1) de la demande d'évacuation d'urgence sur la porte 1 est traitée en priorité par rapport à celle de la porte N ° 2. Lorsqu'un voyageur actionne la poignée d'évacuation, il déverrouille en même temps la porte ; de ce fait, le microprocesseur valide la demande en contactant les deux changements d'état du KSA1 et du verrou PV1, après le traitement d'antirebonds |ch. II. II.1.3|.

La demande d'EVAC peut intervenir à n'importe quel moment et en particulier pendant l'exécution d'une séquence d'ouverture ou de fermeture, ...etc. L'arrêt des séquences se fait par réinitialisation du marquage du réseau de Pétri et annulation des commandes |retour à l'état initial|. Le retour à l'état initial se fait par annulation de tous les marqueurs |ch.II.I.1.3|. Le microprocesseur N° 1 émet la télémesure "Demande d'Evacuation d'urgence sur la porte N° 1, vue par le microprocesseur N° 1", et le N° 2 émet à son tour vers le "DCD" la télémesure "Demande d'évacuation d'urgence sur la porte N° 1 vue par le µP N° 2" |fig. A9|.

III - GENERATION DE FONCTIONNEMENT DEGRADE

Les fonctionnements en mode dégradé telle une condamnation momentanée d'une porte ou une fermeture en battement lorsque la détection d'obstacle ne peut pas se faire |fig. II. A6| permettent une amélioration sensible de l'exploitation de la porte que nous avons envisagée dans les cinq cas suivants |voir annexe RDP| :

- Une commande de déverrouillage impossible qui entraîne l'impossibilité d'ouverture de porte.

- Une commande de l'électrovalve d'ouverture impossible |ouverture impossible|.

- Coupure de l'électrovalve d'ouverture "EVO" impossible pendant la phase de fermeture nécessitant l'appel d'un agent de maintenance |fermeture impossible|.

- Commande de l'électrovalve de fermeture impossible, nécessitant un appel d'un agent de maintenance pour fermer la porte, celle-ci est condamnée par la suite.

- Une condamnation par un agent itinérant : lorsque la porte n'arrive pas à se fermer pour une raison quelconque |ex : obstacle permanent|, le microprocesseur fait appel à un agent par émission d'une télémesure. L'agent peut alors décider la condamnation de la porte par un contact situé au-dessus de la carte de commande, ou laisser continuer l'exploitation.

Une porte condamnée doit être fermée et verrouillée, et signalée aux voyageurs à l'aide d'une plaque signalitique"SIG". Pour cela, et avant de la condamner, le microprocesseur contrôle les capteurs PV1 (verrou) et PF1 (porte fermée) et exécute la fermeture en battement (ouverture et fermeture jusqu'au verrouillage complet de la porte, en fin de cycle il allume la signalitique.

Il est indispensable de continuer le contrôle d'une porte condamnée. En effet, le microprocesseur a toujours à l'oeil les deux capteurs (verrou et fermeture). Lorsque l'un des capteurs est défectueux, il est signalé au "DCD" par l'émission d'une télémesure. De plus, il assure le contrôle de l'état sécuritaire de la porte opposée dans le véhicule.

- Le traitement des demandes d'évacuation d'urgence est maintenu pendant la condamnation d'une porte. Le dialogue reste établi entre le microprocesseur et le "DCD".

La condamnation d'une porte se traduit sur le graphe de Pétri par la forçage du marquage dans un état. En effet, lorsqu'une porte est condamnée cela signifie qu'elle est devenue dangereuse pour les voyageurs et qu'elle ne peut plus évoluer normalement, alors le microprocesseur la condamne en annulant les séquences de commande en cours (ouverture ou fermeture) par annulation du marquage en cours et forçage du marquage de condamnation de porte, cet artifice est beaucoup plus efficace qu'un cycle de dégagement vers un marquage, qui augmente la taille du RDPI et par conséquence les logiciels implantés dans les microprocesseurs. Le forçage du marquage est permis grâce à la méthodologie d'implantation du réseau de Pétri sur microprocesseur développée au troisième chapitre.

Le forçage du marquage est décrit par un astérisque à l'intérieur d'une place, |PLP fig.A10|. Nous noterons par la suite cette place, Place de réinitialisation. Nous décrivons ce type de place dans le quatrième chapitre.

La libération d'une porte condamnée se fait sur demande du "DCD" par l'envoi de la télécommande d'initialisation "TCI". Cette télécommande permet au microprocesseur de revenir à l'état initial et bien entendu par arrêt d'alimentation en énergie de la carte de commande.

Dans la suite, nous présentons le réseau de Pétri, qui a servi de modèle pour la programmation du microprocesseur INTEL 8751. Il est représenté morcelé pour avoir une description détaillée de chaque phase du fonctionnement.

Voici quelques éléments nécessaires à la compréhension du réseau de Pétri.

- Une place qui se termine par un ↗ existe déjà dans la description.

- Une transition décrite comme suit : $\overline{PV}^{n}$ signifie que la transition n'est franchie que si l'état du verrou ($\overline{PV}$) reste inchangé n fois de suite : c'est le traitement d'antirebondissement.

- Les différents contacts portant des informations sur l'état de la porte sont :

  PV 1 : contact porte 1 verrouillée
  PF 1 : contact porte 1 fermée
  MDOP : manostat de détection d'obstacles
  PO I : contact I porte 1 ouverte
  PO II : contact II porte 1 ouverte
  KSA 1 : contact de demande d'évacuation d'urgence sur la porte 1
  PE : contact d'intervention extérieur porte 1 (intervention d'agent)
  PV 2 : contact porte 2 verrouillée
  PF 2 : contact porte 2 fermée
  KSA 2 : contact de demande d'évacuation d'urgence sur la porte 2.

- Les quatre voies de commande sont :

  EVO : électrovalve d'ouverture
  EVF : électrovalve de fermeture
  DEV : électrovalve de déverrouillage
  SIG : plaque signalée

  L'excitation des quatre commandes se fait bit par bit.

- Les télémesures sont représentées de la façon suivante :

  T M octet - ex. T M 23, où l'octet est la valeur à émettre sur le port séri

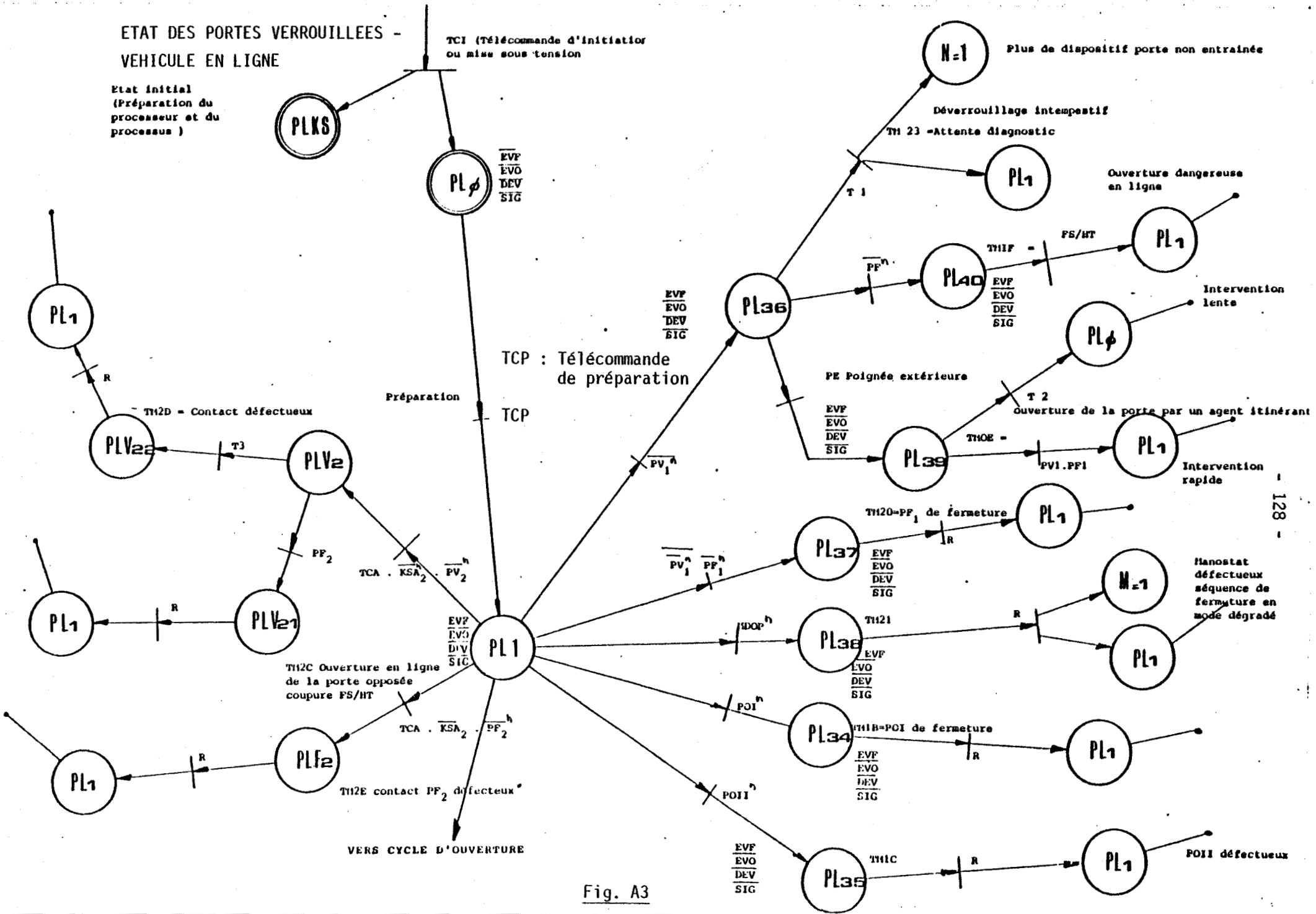

ETAT DES PORTES VERROUILLEES - VEHICULE EN LIGNE
Etat initial (Préparation du processeur et du processus)
PLKS
TCI (Télécommande d'initiation ou mise sous tension
PLφ
EVF EVO DEV SIG
TCP : Télécommande de préparation
TCP
Préparation
PL1
PL1
R
TH2D = Contact défectueux
PLV22
T3
PLV2
PF2
PL1
R
PLV21
TCA . KSA2 . PV2
TH2C Ouverture en ligne de la porte opposée coupure FS/HT
PL1
R
PLF2
TCA . KSA2 . PF2
TH2E contact PF2 défecteux
VERS CYCLE D'OUVERTURE
PV1
PL36
N=1
Plus de dispositif porte non entrainée
Déverrouillage intempestif
TH 23 =Attente diagnostic
T 1
PL1
Ouverture dangereuse en ligne
PF
PL40
TH1F =
FS/HT
PL1
Intervention lente
PLφ
PE Poignée extérieure
T 2
Ouverture de la porte par un agent itinérant
PL39
THOE =
PV1.PF1
PL1
Intervention rapide
PV1 PF1
PL37
TH20=PF1 de fermeture
R
PL1
Manostat défectueux séquence de fermeture en mode dégradé
SDOP
PL38
TH21
R
N=1
PL1
POI
PL34
TH1B=POI de fermeture
R
PL1
POII
PL35
TH1C
R
PL1
POII défectueux

Fig. A3

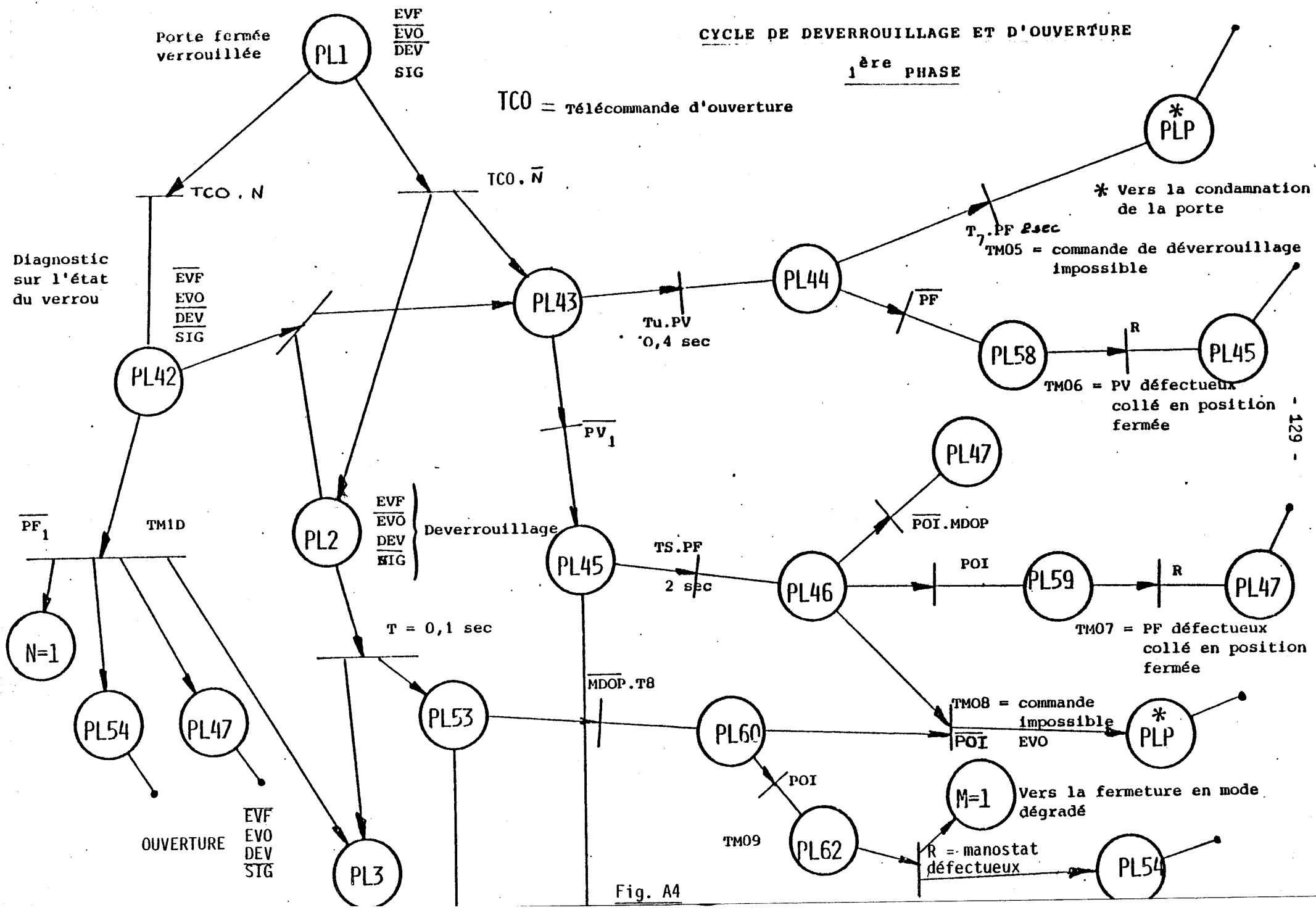

CYCLE DE DEVERROUILLAGE ET D'OUVERTURE
1ère PHASE
TCO = Télécommande d'ouverture
Porte fermée verrouillée
PL1
EVF
EVO
DEV
SIG
TCO . N
TCO . N̄
Diagnostic sur l'état du verrou
EVF
EVO
DEV
SIG
PL42
PL43
Tu.PV
0,4 sec
PL44
T7.PF 2sec
PLP
* Vers la condamnation de la porte
TM05 = commande de déverrouillage impossible
PF
PL58
R
PL45
TM06 = PV défectueux collé en position fermée
PV1
PF1
TM1D
N=1
PL54
PL47
OUVERTURE
EVF
EVO
DEV
SIG
PL2
EVF
EVO
DEV
SIG
Deverrouillage
T = 0,1 sec
PL3
PL53
PL45
TS.PF
2 sec
PL46
POI.MDOP
PL47
POI
PL59
R
PL47
TM07 = PF défectueux collé en position fermée
MDOP.T8
PL60
TM08 = commande impossible
POI
EVO
PLP
POI
TM09
PL62
M=1
Vers la fermeture en mode dégradé
R = manostat défectueux
PL54

Fig. A4

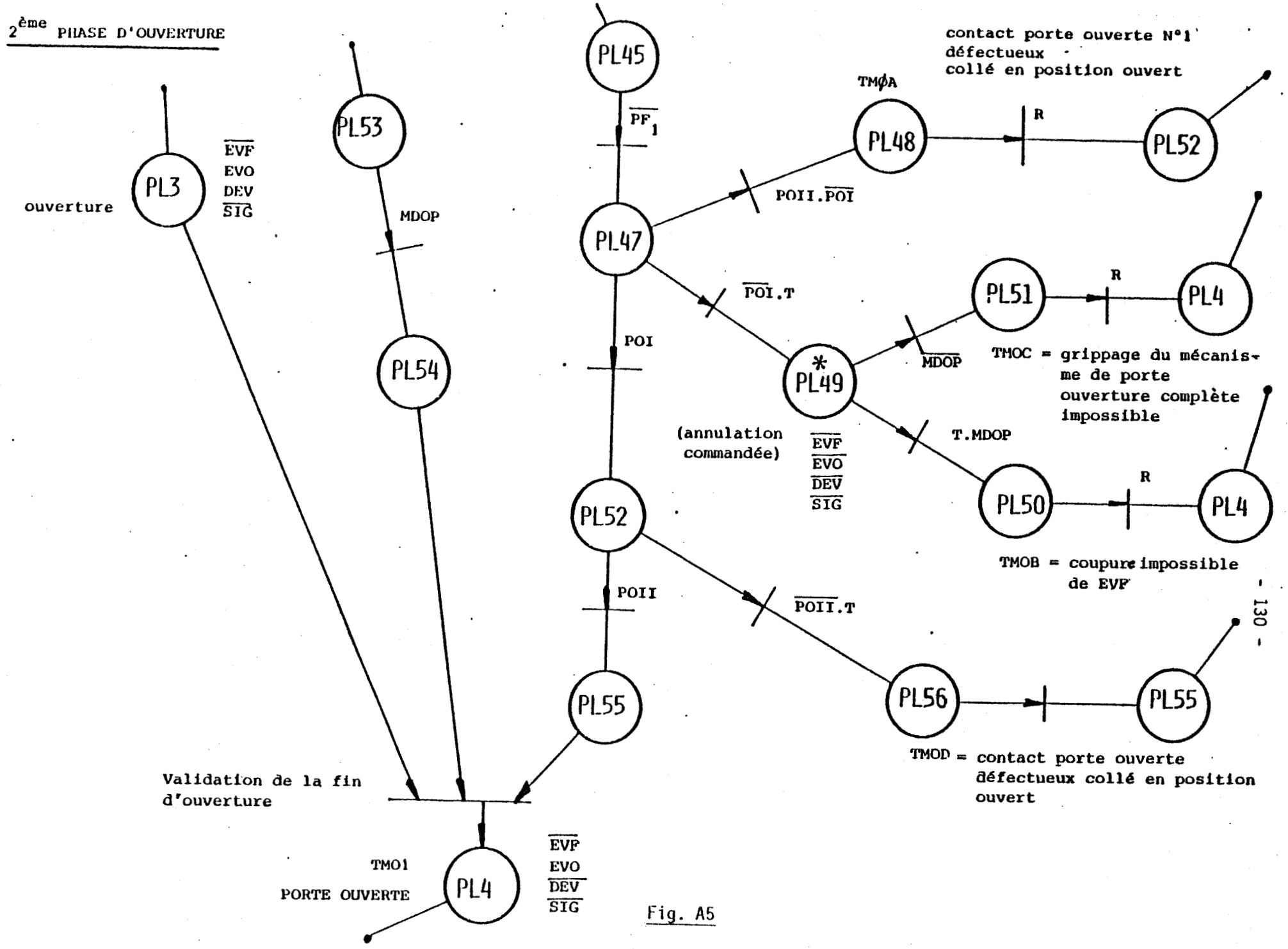

2ème PHASE D'OUVERTURE
ouverture
PL3
EVF
EVO
DEV
SIG
PL53
MDOP
PL54
PL45
PF1
PL47
POI
PL52
POII
PL55
Validation de la fin d'ouverture
TMO1
PORTE OUVERTE
PL4
EVF
EVO
DEV
SIG
POII.POI
TMØA
PL48
R
PL52
contact porte ouverte N°1 défectueux collé en position ouvert
POI.T
PL49
(annulation commandée)
EVF
EVO
DEV
SIG
MDOP
PL51
R
PL4
TMOC = grippage du mécanisme de porte ouverture complète impossible
T.MDOP
PL50
R
PL4
TMOB = coupure impossible de EVF
POII.T
PL56
PL55
TMOD = contact porte ouverte défectueux collé en position ouvert

Fig. A5

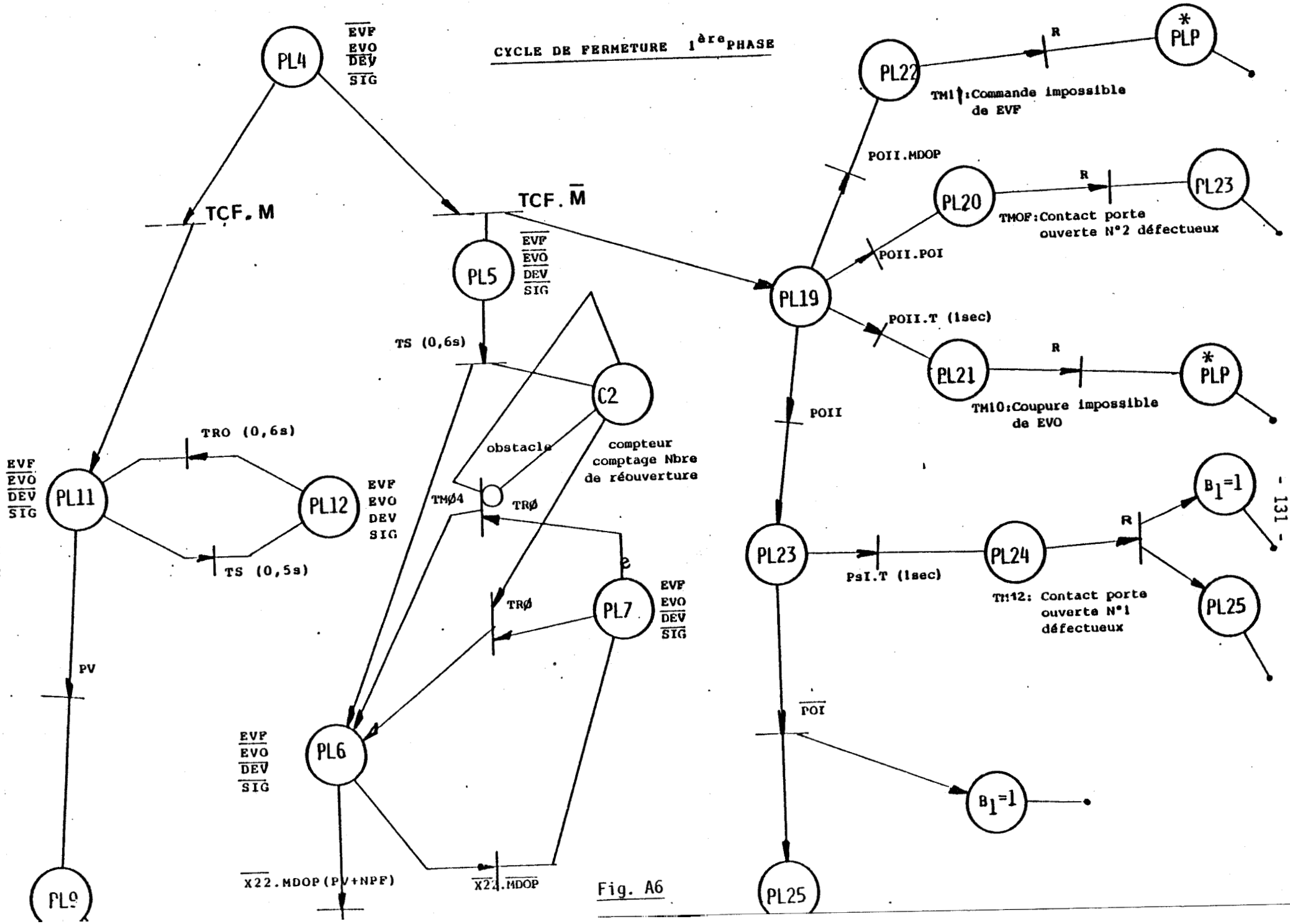

CYCLE DE FERMETURE 1ère PHASE
PL4
EVF
EVO
DEV
SIG
TCF.M
TCF.M̄
PL5
TS (0,6s)
C2
compteur
comptage Nbre
de réouverture
obstacle
TMØ4
TRØ
PL7
PL11
TRO (0,6s)
PL12
TS (0,5s)
PV
PL6
X22.MDOP(PV+NPF)
X22.MDOP
PL9
PL19
POII.MDOP
POII.POI
POII.T (1sec)
POII
PL22
TM1↑:Commande impossible
de EVF
PL20
TMOF:Contact porte
ouverte N°2 défectueux
PL23
PL21
TM10:Coupure impossible
de EVO
PLP
PsI.T (1sec)
PL24
TM12: Contact porte
ouverte N°1
défectueux
PL25
POI
B1=1
R

Fig. A6

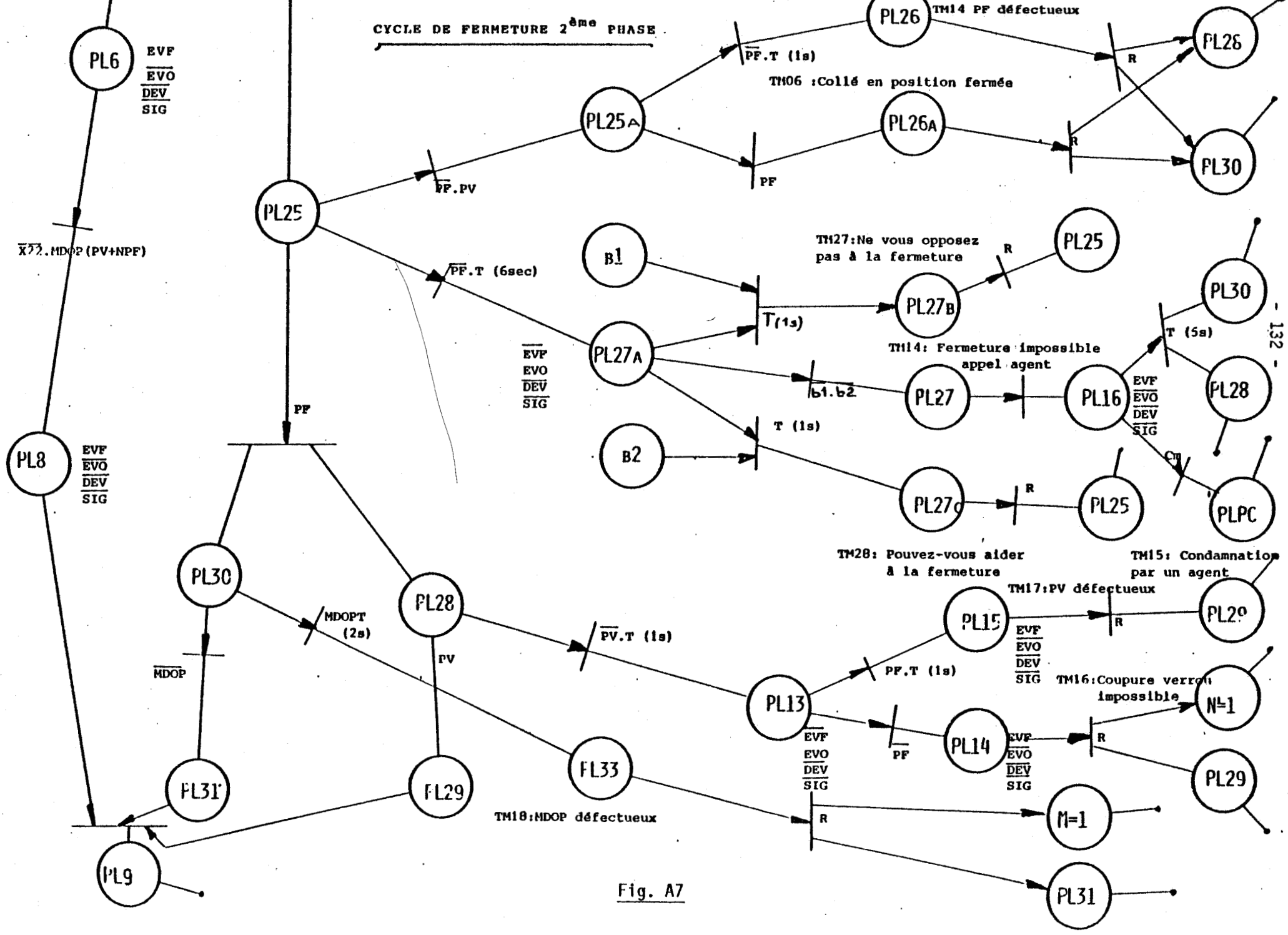

CYCLE DE FERMETURE 2ème PHASE
PL6
EVF
EVO
DEV
SIG
X22.MDOP(PV+NPF)
PL8
EVF
EVO
DEV
SIG
PL25
PF
PL25A
PF.PV
PF.T (1s)
PF
PL26
TM14 PF défectueux
PL26A
R
PL26
PL30
TM06 :Collé en position fermée
PF.T (6sec)
B1
B2
PL27A
EVF
EVO
DEV
SIG
T(1s)
T (1s)
PL27B
TM27:Ne vous opposez pas à la fermeture
PL25
PL27
b1.b2
TM14: Fermeture impossible appel agent
PL16
EVF
EVO
DEV
SIG
T (5s)
PL30
PL28
Cm
PLPC
PL27C
PL25
TM28: Pouvez-vous aider à la fermeture
TM15: Condamnation par un agent
TM17:PV défectueux
PL15
PL29
PL30
MDOPT (2s)
MDOP
PL28
PV
PV.T (1s)
PL13
PF.T (1s)
PL14
PF
TM16:Coupure verrou impossible
N=1
PL29
PL31
PL29
PL33
TM18:MDOP défectueux
M=1
PL9
PL31

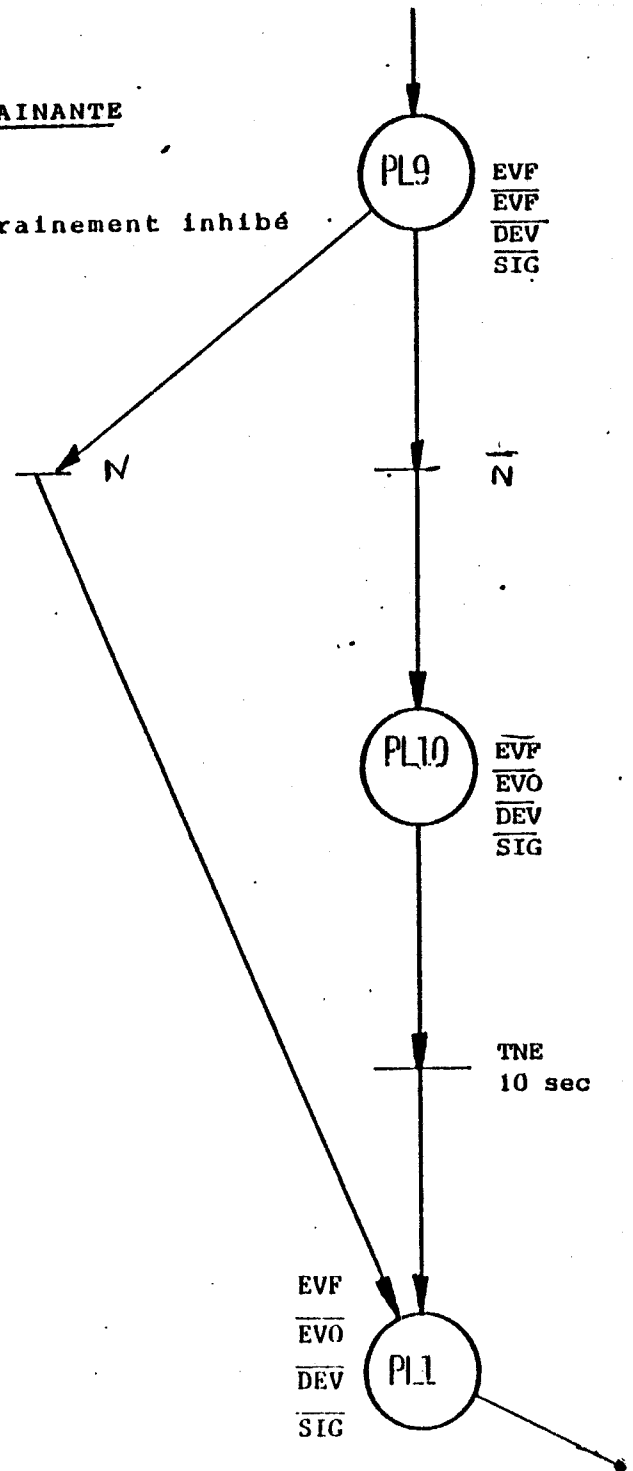
Fig. A7

DISPOSITIF PORTE NON ENTRAINANTE

3ème PHASE

N=1 dispositif de non entrainement inhibé

N =^0 dispositif de non entrainement activé



porte fermée verrouillée
TM19 = départ autorisé

porte non entrainante

porte fermée verrouillée
véhicule en ligne

Fig. A8

TRAITEMENT DE L'EVACUATION D'URGENCE

PORTE 1 ET PORTE 2 PAR LE MICROPROCESSEUR 1

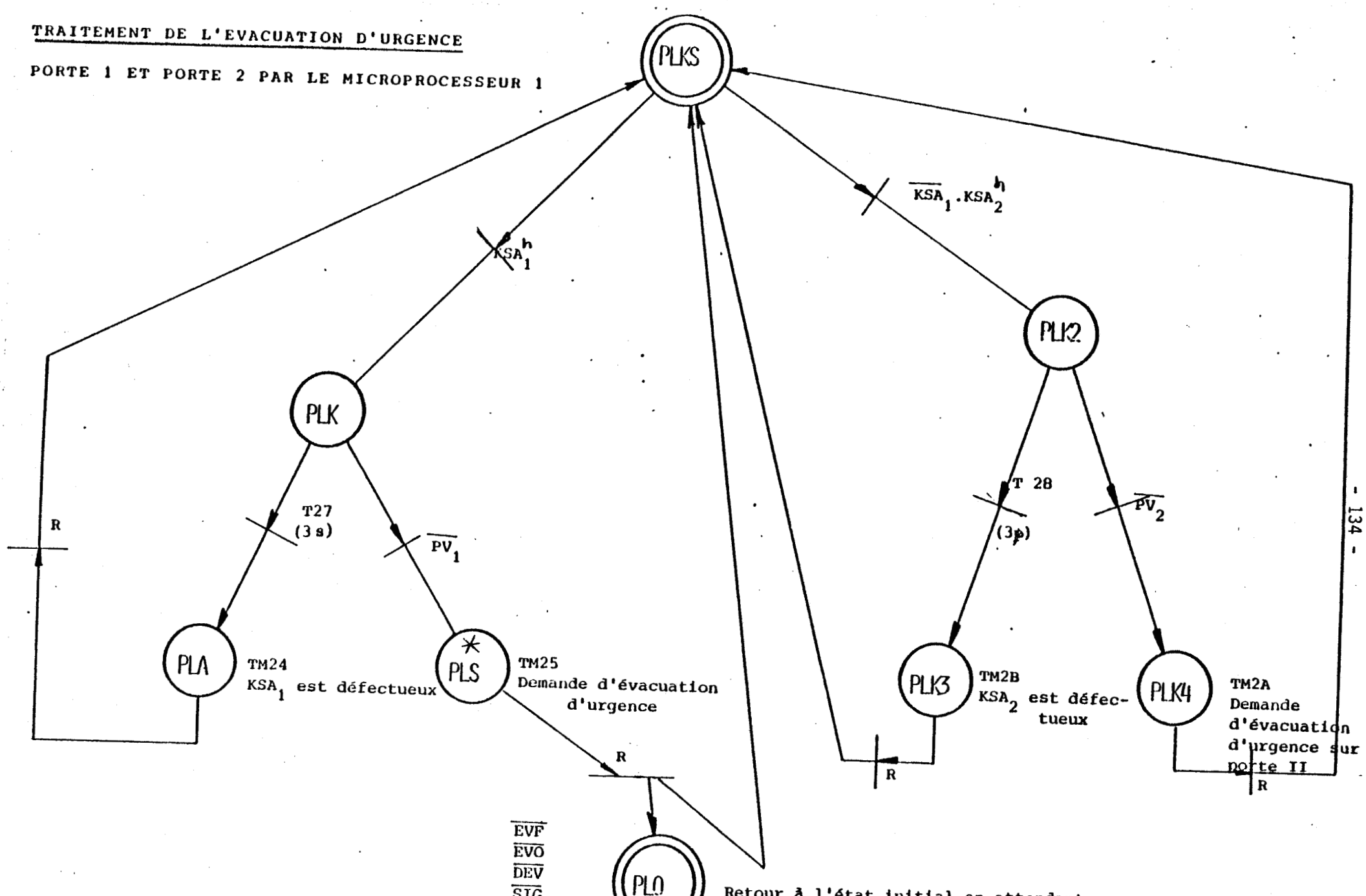


Fig. A9

TRAITEMENT PORTE CONDAMNEE

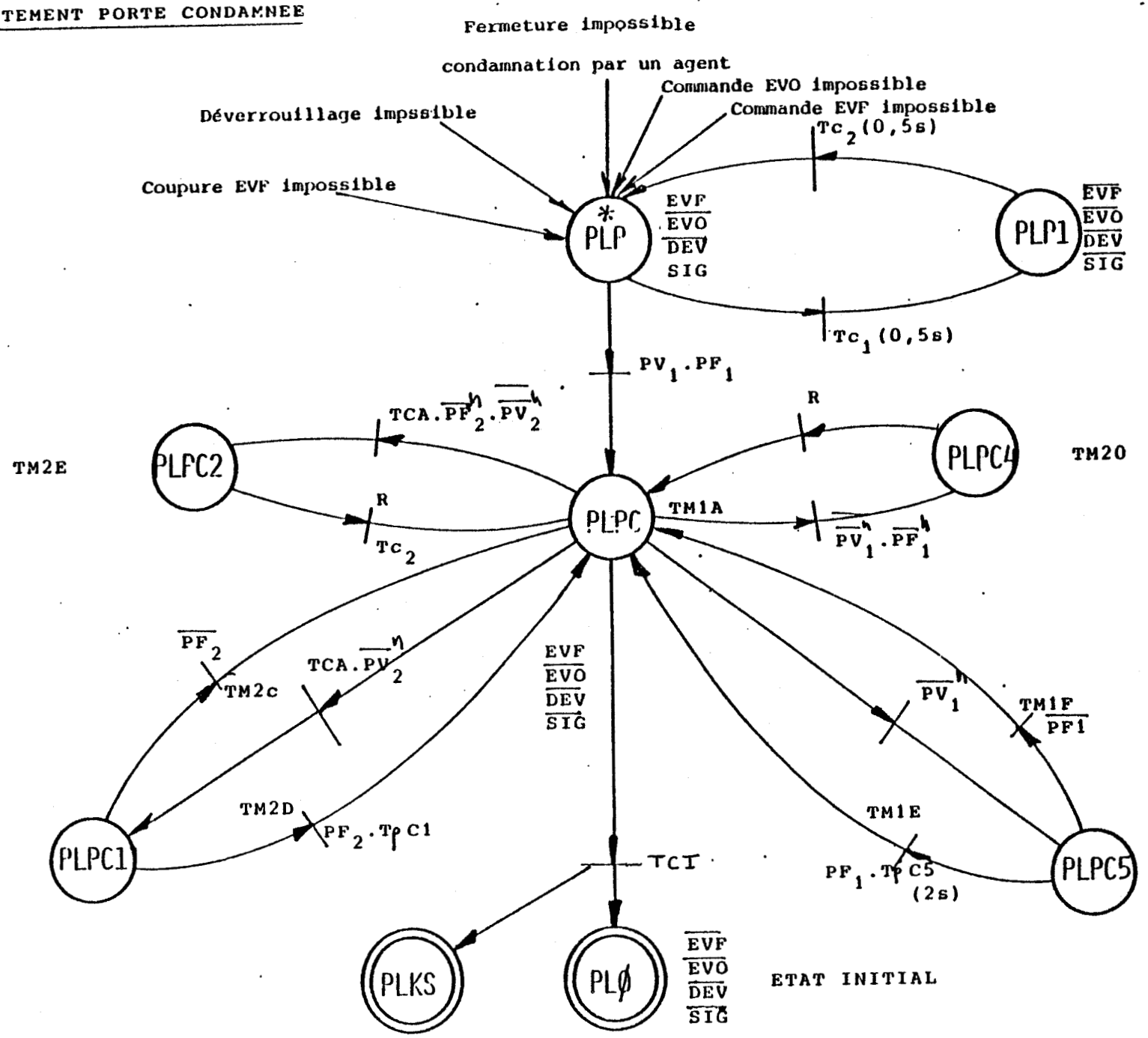


Fig. A10

ANNEXE II

LES LOGICIELS DE TRAITEMENT ET DE DESCRIPTION D'UN RESEAU DE PETRI SUR LE MICROCONTROLEUR INTEL 8751

ANNEXE II

Les différents modules de programmes presentés dans cet annexe , permettent l'ecriture du logiciel en assembleur(intel 8751) à partir d'une description graphique en RDPI d'un automatisme numerique (voir chapitre III)

L'ensemble de ces programmes ont été génerés par le TRADUCTEUR (voir chapitre IV_3). On y trouve :

I] Le programme MONITEUR

Ce logiciel comporte l'initialisation du processeur(INTEL 8751) et les routines utilitaires permettant La gestion des différents modules de programmes specifiques à l'application decrite par un reseau de PETRI interprété [RDPI].

a// Affectation des variables de l'automatisme et organisation de la RAM du microprocesseur pour une application donnée .

```
;____  IMAGE DES SORTIES ____

       S00          EQU          44H
       S01          EQU          45H
       S02          EQU          46H
       S03          EQU          47H

;____  IMAGE DES ENTREES ____

       BI0          EQU          20H
       BI1          EQU          21H
       BI2          EQU          22H
       BI3          EQU          23H

       PSS    EQU     24H     ;IMAGES DES PLACES DE SYNCHRONISME
       BY     EQU     48H   ;IMAGES DES MOTS RECUS A TRAVERS LE PORT SERIE
;    ____  COMPTEURS  DISPONIBLES  ____

       TT00         EQU          30H
       TT00a        EQU          31H
       - -          - -          --
       - -          - -          --
       TT09         EQU          42H
       TT09a        EQU          43H
```

b// Initialisation du processeur et du RDPI

#/ PROCESSEUR

```
DEPT: MOV     TH1,#0FCH
      MOV     TH0,#000H
      MOV     TMOD,#020H
      MOV     TCON,#040H
      MOV     SCON,#040H
      MOV     IE,#000H
      MOV     IP,#000H
      MOV     R0,#50H
```

INITIALISATION DES PORTS E/S UTILISES

```
      MOV     P1,#0FFH
      MOV     P2,#0FFH
      MOV     P3,#08FH
```

PREPARATION DES COMPTEURS UTILISES

(L'utilisateur a droit à 10 valeurs de temporisatios defférentes [0.01 à 2.55s])

```
      MOV     TT00,#00BH
      MOV     TT00a,#64H
      MOV     TT01,#015H
      MOV     TT01a,#64H
```

#/ RDPI (application)

INITIALISATION DES PILES

```
INIT: MOV     R1,#60H        __ pile d'ecriture
      MOV     SP,#70H        __ pile de lecture
      MOV     DPTR,#SYN      __ positinnement de l'indicateur
      CALL    MP                de fin de pile
      PUSH    DPL
      PUSH    DPH
```

ECRITURE DES PLACES INITIALES

```
      MOV     DPTR,#PL01
      PUSH    DPL
      PUSH    DPH
      MOV     DPTR,#PL11
      PUSH    DPL
      PUSH    DPH
      MOV     DPTR,#PL20
      PUSH    DPL
      PUSH    DPH
      MOV     R3,#30H
      JMP     CG
```

c// Routines utilitaires

#/ Procedure de marquage

La procedure de marquage s'effectue en inscrivant l'adresse de la place à marquer dans la pile d'ecriture pointée par le registre R1 du microprocesseur.

```
MP:    INC     R1
       MOV     @R1,82H
       INC     R1
       MOV     @R1,83H
       RET
```

#/ Routine de lecture des places marquées

Le lancement de la place (PAS) à traiter se fait de la maniere suivante :

** lecture de l'adresse pointée par le registre SP

** normalisation de la durée de traitement de la place en cours de traitement à 100 us

** lancement du PAS associe à la place lue

```
CG:    POP     DPH                 __
       POP     DPL                 __| LECTURE DES PLACES

       MOV     A,#50H             __
       CLR     C                    |
       SUBB    A,R3                 |
       JNB     ACC.0,PAIR           |
       NOP     '                    |  Calcul du temps restant
PAIR:  CLR     ACC.0                |  à écouler pour atteindre
       RR      A                    |  les 100 us
       MOV     R3,A                 |  R3 est le registre contenant
       DJNZ    R3,$                 |  la durée de chaque PAS
       CLR     A                    |
       JMP     @A+DPTR            __|  lancement du PAS suivant
```

#/ Routine d'echange des deux piles

Cette routine du moniteur est appelée à chaque fin de pile de lecture .Elle est traitée comme un PAS et dure 100 us. Elle permet :

** le passage d'un cycle de traitement au suivant
** le calcul du nombre de places marquées simultannèment afin d'ajuster les temporisations en parallèles par rapport au nombre de places traitées en même temps.
** le relevé du vecteur d'etat du processus commandé.

```
SYN:  MOV    A,SP          __
      XCH    A,R1            ! Echange des deux piles
      MOV    SP,A          __!


      ANL    A,#00FH       __
      RR     A               ! Calcul du nombre de places marquèes
      MOV    R2,A          __! simultanément au debut du cycle

      MOV    DPTR,#SYN     __
      CALL    MP           __! Remise de l'indicateur de fin de pile


      MOV    BI1,P1          !  Releve du vecteur  d'etat
      MOV    BI2,P2          !  du processus commandé
      MOV    BI3,P3          !
      MOV    R3,#20H         !
      JMP    CG            __!
```

#/ routine de demarquage

Le demarquage d'une place (adresse) se fait par decrémentation du pointeur de pile d'ecriture R1

```
FR:   DEC    R1
      DEC    R1
      RET
```

#/ routine d'envoi du mot de commande

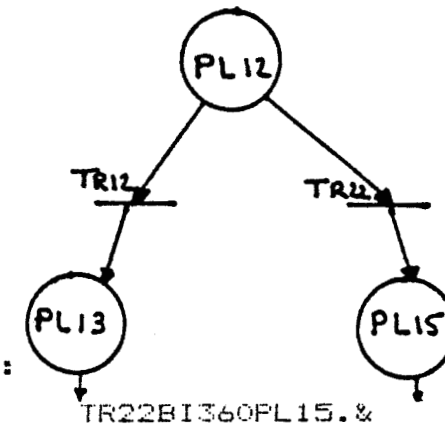
le mot de commande est emis vers les ports E/S après masquage des lignes definies en entrée

```
SORT: NOP
      ORL    S03,#08FH
      MOV     P3,S03
      RET
```

II] LES MODULES SPECIFIQUES A UNE APPLICATION

Selon la methode presentée dans le troisieme chapitre, trois types de places peuvent être representés par un module de programme appelé 'PAS'. Chaque PAS fait appel à des routines du moniteur tel que : MP (marquage),FR(franchissement),SORT(sorties) et CG (combinatoire generale)[voir le programme moniteur].

1// PLACE PIVOT



place 12 est pivot
pour les transitions TR12 et TR22.

Novau gènèré par le logiciel EDITEUR(chapitre IV) :

12 PL12S0140S0308 TR12BI120BI111BI361PL13. TR22BI360PL15.&

PLACE PIVOT : PL12 NUMERO DE STOCKAGE : 12

MOTS DE COMMANDE : S0140S0308

TRANSITIONS	TESTS	PLACES DE SORTIE
---->TR12	BI120BI111BI361	PL13.
---->TR22	BI360	PL15.&

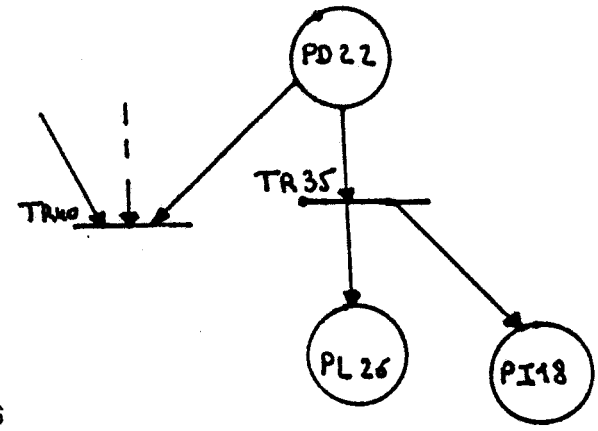
Point d'entrée du module

Durées d'execution en microseconde

```
PL12:    CALL    MP              ; 10;
         MOV      S01,#040H      ; 2;
         MOV      S03,#008H      ; 2;
         CALL    SORT            ; 6;
TR121:   NOP
         JB      BI1.2,TR221     ; 2;
         JNB     BI1.1,TR222     ; 2;
         JNB     BI3.6,TR223     ; 2;
FR12:    CALL     FR             ; 6;
         MOV      DPTR,#PL13     ; 2;
         CALL     MP             ; 10;
         MOV      R3,#030H
         JMP      CG
TR221:   NOP
         NOP
TR222:   NOP
         NOP
TR223:   NOP
         JB      BI3.6,FI121     ; 2;
FR22:    CALL     FR             ; 6;
         MOV      DPTR,#PL15     ; 2;
         CALL     MP             ; 10;
         MOV      R3,#033H
         JMP      CG
FI121:   MOV      R3,#021H
         JMP      CG
```

Point de sortie du module

2// PLACE DOUBLE



La place 22 est une place double. Elle est pivot pour la transition TR35 et de synchronisme pour TR40. Elle est representée par le PAS PD22 ci_dessous et par la variable de synchronisme de TR40 definie par le bit 5 du registre PSS

Noyau gènèré par le logiciel EDITEUR(chapitre IV) :

22 PD22S0100S0323 TR30PS05PLS05. TR35BI121BI111PL26PI18.&

PLACE DOUBLE : PD22 NUMERO DE STOCKAGE : 22

MOTS DE COMMANDE : S0100S0323

TRANSITIONS	TESTS	PLACES DE SORTIE	
----->TR30	PS05		PLS05.
----->TR35	BI121BI111	PL26	PI18.&

Point d'entrée du module

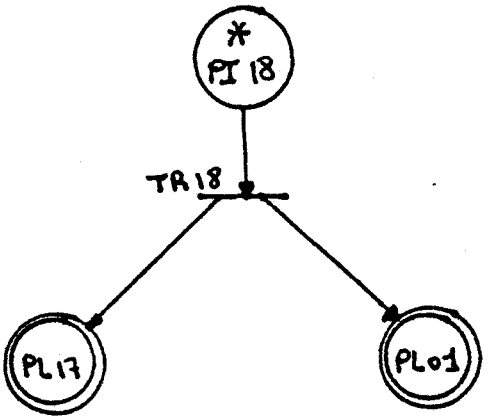
Durées d'execution en microseconde

```
PD22:    CALL    MP                ; 10;
         MOV      S01,#000H        ; 2;
         MOV      S03,#023H        ; 2;
         CALL    SORT              ; 6;
         NOP
         JB      PSS.5,TR351       ; 2;
         CALL     FR               ; 6;
         CLR     PSS.5             ; 1;
         MOV      R3,#021H
         JMP      CG
TR351:   NOP
         JNB     BI1.2,FI221       ; 2;
         JNB     BI1.1,FI222       ; 2;
FR35:    CALL     FR               ; 6;
         MOV      DPTR,#PL26       ; 2;
         CALL     MP               ; 10;
         MOV      DPTR,#PI18       ; 2;
         CALL     MP               ; 10;
         MOV      R3,#03DH
         JMP      CG
FI221:   NOP
         NOP
FI222:   MOV      R3,#01FH
         JMP      CG
```

Point de sortie du module

3// PLACE DE REINITIALISATION

La place 18 a une fonction speciale. Elle a été definie pour resoudre le probleme de la demande d'evacuation d'urgence(voir chapitre II et III). elle annule le marquage du reseau dés qu'elle est marquée. Elle permet ainsi de réinitialiser le reseau



Noyau gènèré par le logiciel EDITEUR(chapitre IV) :

18 PI18S0100S0304 TR18BI120BI110BI100PL01PL17.&

PLACE DE REINITIALISATION : PI18 NUMERO DE STOCKAGE : 1

MOTS DE COMMANDE : S0100S0304

TRANSITIONS	TESTS	PLACES DE SORTIE	
----->TR18	BI120BI110BI100	PL01	PL17.&

Point d'entrée du module

```
                                    Durées d'execution
                                     en microseconde
PI18:   MOV     R2,#62H             ; 1;
        MOV     SP,#72H             ; 2;
        CALL    MP                  ; 10;
        MOV      S01,#000H          ; 2;
        MOV      S03,#004H          ; 2;
        CALL    SORT                ; 6;
TR181:  NOP
        JB      BI1.2,FI181         ; 2;
        JB      BI1.1,FI182         ; 2;
        JB      BI1.0,FI183         ; 2;
FR18:   CALL     FR                 ; 6;
        MOV      DPTR,#PL01         ; 2;
        CALL     MP                 ; 10;
        MOV      DPTR,#PL17         ; 2;
        CALL     MP                 ; 10;
        MOV      R3,#03FH
        JMP      CG
FI181:  NOP
        NOP
FI182:  NOP
        NOP
FI183:  MOV      R3,#021H
        JMP      CG
```

Point de sortie du module

ANNEXE III

DESCRIPTION DE LA CONFIGURATION MATERIELLE PROPRE AU DEVELOPPEMENT DES LOGICIELS REPRESENTES FONCTIONNELLEMENT PAR GRAPHE DE PETRI

ANNEXE

Le but recherché est de permettre l'implantation rapide et efficace d'un RDPI sauf et vivant selon la méthode décrite au IIIe chapître sur un microprocesseur donné. Il s'agit d'apporter une aide (réduction de temps de mise en oeuvre, logiciel sûr) à tout concepteur d'un système de commande logique à microprocesseurs.

L'idée est que le concepteur n'aura à effectuer comme tâches que :

- La description du fonctionnement de son automatisme par un RDPI sauf et vivant
- L'introduction du RDPI dans un système dit d'édition de RDPI
- Le lancement de la procédure de transformation RDPI/programme
- La récupération du microprocesseur programmée en assembleur pour le pilotage de son processus selon la méthode du IIIe chapître.

La mise en oeuvre d'un tel outil nécessite un support matériel adapté aux différentes tâches |Edition du RDPI, programmation du microprocesseur| et un support logiciel permettant l'écriture du langage spécifique à la description et du langage interprêteur permettant la génération sans erreur du programme assembleur du microprocesseur utilisé à partir de la description du RDPI.

L'outil mis en oeuvre au laboratoire s'applique à l'heure actuelle à la famille des microprocesseurs INTEL 8051.

Nous présentons ci-dessous le choix du matériel utilisé et les raisons de ce choix.

STRUCTURE MATERIELLE UTILISEE

Le choix du matériel est lié essentiellement au microprocesseur à programmer. Néanmoins, il est aussi important d'avoir un seul appareil |micro-ordinateur| permettant :

1 - L'écriture d'un langage éditeur structuré interactif, facilement utilisable, où le concepteur est bien dirigé, a le moins d'informations à écrire et peut à tout instant modifier ces informations.

2 - L'écriture du langage traducteur permettant l'obtention du programme assembleur du microprocesseur d'une façon transparente pour l'utilisateur.

3 - La possibilité de programmer le plus de famille de microprocesseurs.

A l'heure actuelle, il est prématuré de pouvoir réunir les 3 caractéristiques ci-dessus dans le même microordinateur. La solution que nous avons retenu est d'utiliser deux microordinateurs, l'un permettant l'écriture des logiciels utilitaire avec aisance et rendant leurs utilisations facile. L'autre permettant la programmation d'un nombre important de famille de microprocesseurs (photo n° 3).

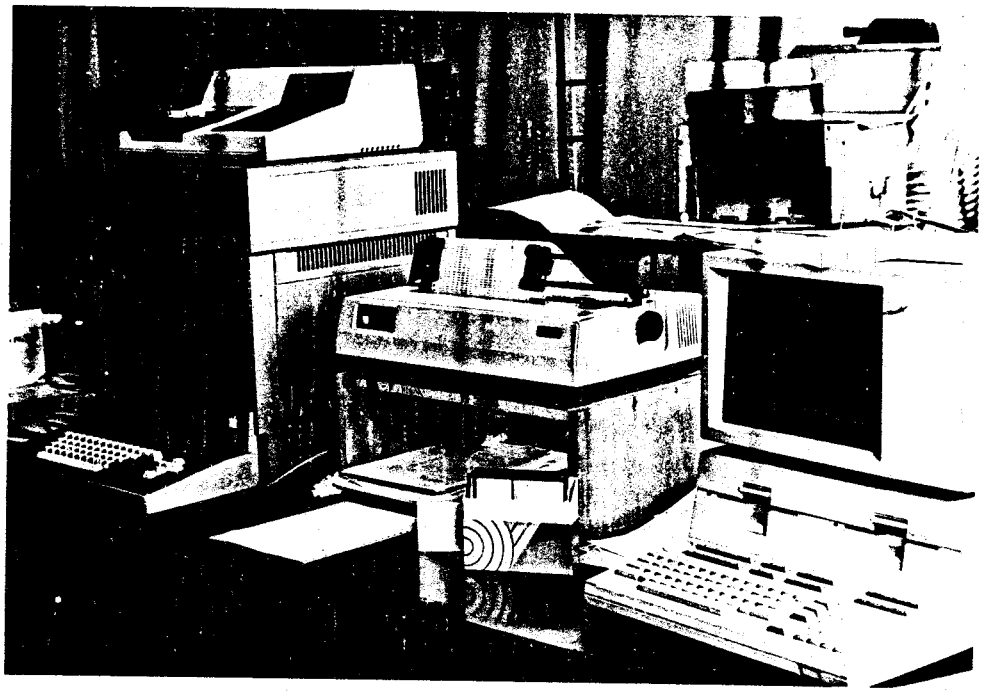
I - MATERIEL D'EDITION ET DE TRADUCTION DU RDPI

Pour des raisons de disponibilité et de souplesse nous avons adopté pour la partie d'édition et de traduction un microrodinateur de type HEWLETT PACKARD 9836 -200 . Nous ne nous attarderons pas sur ses caractéristiques, mais cet appareil possède un atout considérable par le fait qu'il possède des touches fonctions |clés| programmables et dont le contenu est visualisé en bas de l'écran. Cet atout sera extrèmement utile pour l'utilisation des différents programmes |Edition et Traduction| avec beaucoup de simplicité et d'agrément.

Le concepteur utilisant le langage de description n'aura pendant les phases d'édition du RDPI que :

- la lecture des explications affichées
- la lecture du contenu des clés
- l'appui de la clé correspondante à ses désirs.

outil de développement INTELLEC série II



Minicalculateur HP 9836

- Assemblage ASM 51
- Mémorisation (8751 EPROM 2732)

- Préparation
- Edition x modification
- Traduction

Photo n° 3 : Vue d'ensemble du matériel informatique nécessaire au développement des programmes.

II - MATERIEL DE PROGRAMMATION DES MICROPROCESSEURS

Notre choix a été fait sur un système de développement de microprocesseurs "MDS" de la série II 221, pour les raisons suivantes :

- Possibilité de programmation d'un nombre important de microprocesseurs de chez INTEL
- Notre application qui utilise un INTEL 8751
- Disponibilité.

Ce système de développement récupère le programme assembleur généré par la HP 9836 et le transforme en code objet, qui sera par la suite implanté sur le microprocesseur.

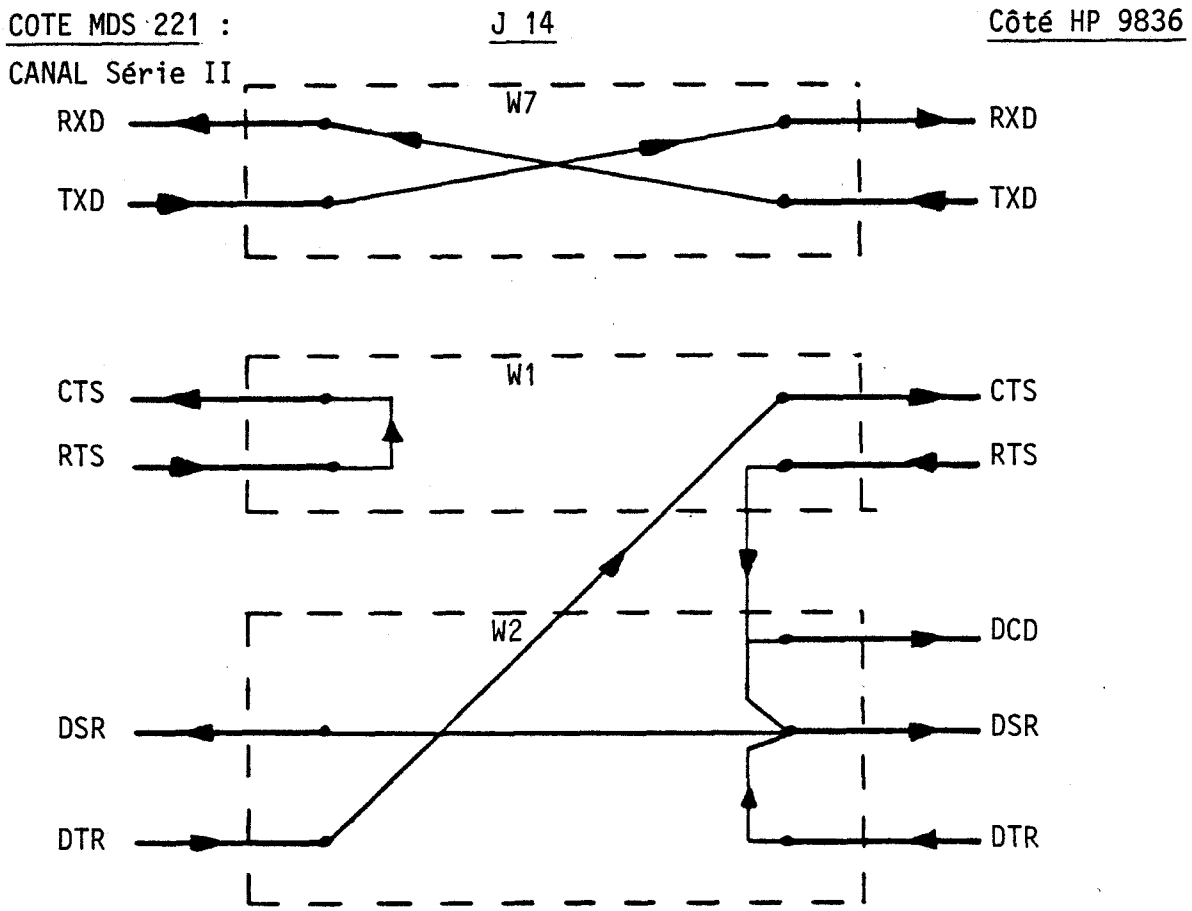
III - INTERCONNEXION ENTRE LES DEUX APPAREILS

La liaison standard RS 232 C a été utilisée pour sa simplicité afin d'assurer le dialogue entre les deux calculateurs. La liaison physique HP 9836 - MDS 221 est donnée ci-dessous ainsi que le programme de transfert écrit par nos soins. La transmission se fait à la vitesse standard 9600 bds. La taille du programme assembleur générée par la HP 9836 ne dépend que de la capacité mémoire RAM du MDS qui ne peut pas excèder les 42 Koctets |après chargement d'ISIS et programme de dialogue|. Cependant, on peut transmettre un programme assembleur de grande taille > 42 K octets morcelé.

Ci-dessous, nous présentons la liaison entre les lignes de dialogue des deux calculateurs et les logiciels d'émission et de réception permettant l'initialisation de l'interface série du calculateur HP 9836.

LIAISON HP 9836 - MDS 221 : CANAL II

Liaison RS 232 C




```
10          SUB Emission2
20 !#################################################################
30                      !#  EMISSION  #
40                      !##############
50     ! PROGRAMME DE CONNEXION HP9836  _ MDS 221
60     ! CONNEXION RS 232
70     ! ON NE PEUT PAS EMETTRE PLUS DE 40K OCTECT
80     ! RTS EST RELIE AU DSR DU MDS 221
90     ! CTS EST RELIE AU DTR DU MDS 221
100       !#####################################
110       !# PREPARATION DE LA LIAISON RE232C #
120       !#####################################
130 Pars1:     Charge=11
140              CONTROL Charge,0;1                  ! RESET
150              CONTROL Charge,3;9600               !  VITESSE    9600 BAUDS
160              CONTROL Charge,4;3                  ! 8 BITS DATA,1BIT STOP,
170              CONTROL 1;10,10
180              OUTPUT 1;" HP9836    ------------------->  MDS 221"
190              DISP " DEMANDE D'EMISSION "
200              CONTROL Charge,5;1                  !RTS a 1 DTR a 0
210 Cts:         STATUS Charge,11;Sign1              !CTS ?
220              IF Sign1<>16 THEN GOTO Cts
230              DISP " EMISSION ACCEPTEE  "
231              SUBEXIT
240 Fin1:        CONTROL Charge,5;0                   !RTS a 0  DTR a 0
250              OUTPUT Charge;CHR$(3)&CHR$(3);       !ENVOI DE FIN DE TEXTE
260 Cts1:        STATUS Charge,11;Sign2               !CTS
270              IF Sign2<>0 THEN GOTO Cts1
280              DISP "    FIN  DE L'EMISSION "
290         SUBEND
```

```
1                    !#  RECEPTION2  ##
10     ! PROGRAMME DE CONNEXION HP9836  / MDS 221
20     ! CONNEXION RS 232
30     ! ON NE PEUT PAS RECEVOIR PLUS DE 32K OCTETS DANS LE BUFFER
40     !DTR EST RELIE AU DSR, DCD ET AU DSR DU MDS
50     !##############################################################
60             COM H$[32000] BUFFER,I,K,Serie
70             H$=""
80     !##############################################################
90 Pars1:      Serie=11                          !CODE DE L'INTERFACE SERIE
100            CONTROL Serie,0;1                 ! RESET
110            CONTROL Serie,3;9600              !  VITESSE    9600 BAUDS
120            CONTROL Serie,4;7                 ! 8 BITS DATA,2 BIT STOP,
130            CALL Reception                    ! APPEL RECEPTION DU TEXTE
150            END
160    !##############################################################
170                  !#  RECEPTION  ##
180                  !#################
190            SUB Reception                        !SOUS PROGRAMME DE RECEPTION
200            INPUT "Reception d'un programme ou visualisation ?  R/V",R$
210                  SELECT R$
220                  CASE ="R","r"
230                  CASE ="V","v"
240                         GOTO N
250                  END SELECT
260            COM H$[20000] BUFFER,I,K,Serie!VARIABLES DECLAREES EN COMMUN
270                  DISP " DEMANDE DE RECEPTION"
280                  Serie=11                     ! INTERFACE SERIE
290                  ASSIGN @B TO BUFFER H$       !H$ ETANT LE TEXTE
300                  ASSIGN @S TO Serie           !@S=INTRFACE SERIE
310                  CONTROL Serie,5;3            !RTS a 1 DTR a 1
320 Dsr:             STATUS Serie,11;Sign1        !DCD et DSR ?
330                  IF Sign1<>160 THEN GOTO Dsr! DSR a 1 et DCD a 1
340                  DISP " RECEPTION ACCEPTEE  "
350                  CONTROL @B,3;1
360                  S1=1
370                  STATUS Serie,10;Etat
380                  TRANSFER @S TO @B;COUNT 32000! TRANSFERT DES DONNEES
390                  STATUS @B,4;S                !LE COMPTAGE DES CARACTERS
400                  CONTROL 1,1;1
410                  CONTROL 1,0;1
420 !-----------------------------------------------------------------------
```

BIBLIOGRAPHIE

BIBLIOGRAPHIE

|1| G.W. BRAMS

Réseaux de Pétri : Théorie et pratique
Tome I : Théorie et analyse
Tome II : Modélisation et applications
Edition Masson, 1983.

|2| Sylvain THELLIERZ

Pratique séquentielle et réseaux de Pétri
Edition Eyrolles, 1978.

|3| M. BLANCHARD

Comprendre, maitriser et appliquer le grafcet
Edition Cepadus, 1979.

|4| E. GELEMBRE

Sûreté de fonctionnement des systèmes informatiques
- Monographies d'informatique de L'AFCET
Editions Hommes et techniques, 1980.

|5| M. TIMOULALI

"Modélisation graphique et simulation des systèmes de production"
Docteur-ingénieur, Lyon I, avril 1981.

|6| P. GOSSET

"Conception des systèmes de commande"
Nouvel automatisme, p.48-53, septembre 1980.

|7| R. VALETTE

"Etude comparative de deux outils de représentation :
Grafcet et Réseau de Pétri"
Nouvel automatisme, p. 377-382, décembre 1978.

|8| M. BLANCHARD

"Automatismes logiques : Grafcet ou Réseau de Pétri"
Nouvel automatisme, p. 45-52, mai 1979.

|9| J.F. DHALLUIN

Commande-contrôle de processus en sécurité : application à la commande d'un ensemble de portes véhicule d'une rame de métro de type VAL
Thèse docteur ingénieur, Lille 1983.

|10| C. MAGNIEZ

Commande d'un ensemble de processus en sécurité :
Application à la commande d'un système de porte de métro
DEA - Lille, 1983.

|11| M. CUVELIER

Commande des portes VAL :
Carte de commande à microprocesseurs
Rapport d'activité - U.S.T.L. - mars 1983.

|12| D. POWELL

Réseaux locaux de commande-contrôle de processus sûrs de fonctionnement
Thèse d'état - Institut National Polytechnique de Toulouse - octobre 1981.

|13| EL KOURSI, CH. MAGNIEZ, J.F. DHALLUIN

"Commande-contrôle de processus"
Note interne - Lille - août 1984.

|14| Sécurité et automates programmables
Electronique Industrielle
pp. 57-62, 1er juin 1982.

|15| J.P. HAUTIER

Sur la description fonctionnelle et la simulation numérique d'un onduleur à transistors : application au contrôle de la dynamique d'une machine asynchrone
Docteur Ingénieur - Lille, 1984.

|16| H. GHESTEM

Sur la description et la programmation de manoeuvres d'ascenceur
Thèse 3ème cycle - U.S.T.L. - Lille, 1979.

|17| D. DEI-SVALDI, M. COLLIER, J.P. VAUTRIN

"Machine commandée par automate programmable :
amélioration de la sécurité"
Nouvel automatisme - mai 1983.

|18| D. POWELL

"Sûreté de fonctionnement des réseaux locaux"
Electronique industrielle - septembre 1982.

|19| M.E. TAFAZZOLI

"Réalisation d'un interpréteur matériel de réseau de Pétri"
Application à la commande d'un système multiprocesseurs
Thèse 3ème cycle - Nice, 1979.

|20| J.M. TOULOTTE et S. THELLIEZ

"GRAFCET et logique industrielle programmée"
Eyrolles, 1980.

|21| J. DEFRENNE

"Implantation des réseaux de Pétri sur automate biprocesseur à haute sûreté de fonctionnement"
Thèse 3ème cycle - Lille - juin 1979.

|22| SILVA, R. DAVID

"Synthèse programmée des automatismes logiques décrits par réseaux de Pétri. Une méthode de mise en oeuvre sur calculateur"
RAIRO - Vol. 13 - N° 4 - 1979.

|23| J.M. TOULOTTE, S. THELLIEZ

"Applications industrielles du GRAFCET"
Eyrolles - 1983.

|24| SAMI MAHMOUD

"Conception et réalisation d'un automate programmable par schémas à réseaux de Pétri"
Docteur Ingénieur - Toulouse, 1980.

|25| J.P. BANATRE

"Contribution à l'étude des méthodes et d'outils de construction de programmes parallèles et fiables"
Thèse d'état - Rennes - 1980.

|26| EL KOURSI

"Implantation de réseaux de Pétri sur microprocesseur INTEL 8085 Application à la commande d'un système de porte de métro"
D.E.A. - LILLE - Octobre 1982.

|27| J.P. PARSY

"Dispositif interactif d'aide à la conception des programmes de commande de processus logiques industriels"
Thèse 3ème cycle - U.S.T.L. - 1981.

|28| N. LAHMAR

"Système de gestion de processus multitâches - Application à la commande et à la simulation"
Thèse 3ème cycle - LILLE I - 1981.

|29| J.M. TOULOTTE

Réseaux de Pétri et automates programmables -
Automatisme Tome XXII - N° 7-8 - juillet-août 1978.

|30| A. SCHWEITZER, J.P. GERARDIN

Méthode permettant d'améliorer le niveau de sécurité des systèmes à logique programmée
Electronique, technique et industrie (Paris) - N° 12 novembre - N° 13 décembre - 1984.

|31| PH. DELANGHE

Etude de différentes structures de contrôle du bon fonctionnement d'un microprocesseur
Note interne - U.S.T.L. - septembre 1983.

|32| PH. DELANGHE

Présentation d'une structure de contrôle autotestée
Note interne - U.S.T.L. - décembre 1983.

|33| EL KOURSI

"MANUEL d'utilisation d'une méthode d'implantation des réseaux de Pétri sur microprocesseur"
Note interne - U.S.T.L. - mars 1983.

"Commande de portes VAL
Le logiciel de commande et contrôle"
Note interne - U.S.T.L. - mars 1983.

|34| A. OUADGHIRI

Transmission de message en sécurité -
établissement d'une répartition de messages codés à distance maximum et de procédures de détection d'erreur s'y rapportant
Rapport d'activité - U.S.T.L. février 1984.

|35| C. GROSSE

"Progiciels et langages de contrôle-commande"
Electronique industrielle - N° 39 - 1er octobre 1982.

|36| F. BARANOWSKI

"Utilisation des circuits câblés ssi pour la conception de fonctions logiques en sécurité intrinsèque"
D.E.A. - LILLE - juin 1984.

|37| D. PELAND, G. SAUCIER

"Conception de systèmes temps réel à très haute sécurité sur microprocesseur"
Rapport de recherche IMAG - 1978.

|38| NAIFI

CONTRIBUTION à l'étude des systèmes logiques industriels
Thèse 3ème cycle - LILLE - septembre 1983.

|39| J.C. LAPRIE

"Evaluation de la sûreté de fonctionnement des logiciels"
Séminaire AFCET "Approche qualitative en génie logiciel"
Sophia - Antipolis - 7-8 juin 1984.

|40| C. BEOUNES, F. CEREJA, J.C. LAPRIE

"Méthodologie de conception des microcalculateurs tolérant aux pannes"
ASMAR : automate sûr et modulaire adapté aux régulations anioniques
RAIRO - Vol. 13 - N° 4 - pp. 395-419 - 1979.

|41| J.C. LAPRIE

Journées "Sécurité des applications des automatismes numériques dans les transports"
Villeneuve d'Ascq, 24-25 octobre 1984.
"L'INFORMATIQUE dans le système"
de sécurité problème et approche.

BIBLIOTHEQUE INTERUNIVERSITAIRE LILLE