

50376
1979
23

N° d'ordre : 229

50376
1979
23

THÈSE

présentée à

L'UNIVERSITÉ DES SCIENCES ET TECHNIQUES DE LILLE

pour obtenir le titre de

DOCTEUR-INGÉNIEUR

par

Hubert GHESTEM

Ingénieur CNAM



SUR LA DESCRIPTION ET LA PROGRAMMATION DE MANŒUVRES D'ASCENSEUR

Soutenue le 25 juin 1979 devant le Jury d'Examen

MM. G. SEGUIER	Président
C. MAIZIERES	Rapporteur
S. THELLIEZ	Examineur
G. MANESSE	Directeur du Travail
J. EVIN	Invité

A V E R T I S S E M E N T

Si vous me le dites, j'oublierai.
Si vous me le montrez, je m'en
souviendrai peut-être. Mais si
vous m'y faites participer, je le
comprendrai.

Proverbe chinois.

Cette étude, menée dans le Laboratoire d'Electrotechnique,
revêt un caractère pratique grâce à la Société AUTINOR (59790 RONCHIN)
avec laquelle j'ai pu travailler.

J'ai d'abord un grand plaisir à remercier Monsieur MAIZIERES
de m'avoir accueilli dans son laboratoire et de m'avoir toujours facilité
la tâche. Je dois ensuite mentionner la largeur d'esprit et les sages
conseils de Monsieur MANESSE qui m'a guidé tout au long du travail.

Je dois encore beaucoup de remerciements à Monsieur DURIEZ,
Directeur d'AUTINOR et Monsieur EVIN, directeur technique, qui m'ont
accordé leur confiance en me faisant participer à la vie de leur bureau
d'études. Le chef de bureau d'études, Monsieur MARY, m'a toujours expliqué
clairement et patiemment tous les détails de l'ascenseur. Je lui témoigne
une profonde reconnaissance.

Je remercie enfin tous ceux qui m'ont aidé d'une manière ou d'une
autre, et ils sont nombreux, au cours de cette thèse.

.....

	Pages
INTRODUCTION	
CHAPITRE 1 : METHODES D'ETUDES	1
1.1. Les commandes dans l'ascenseur	1
1.2. La manoeuvre "collective montée-descente"	2
1.2.1. Caractéristiques de la manoeuvre	3
1.2.1.1. Loi de parcours	3
1.2.1.2. Détection de la position de la cabine	4
1.2.1.3. Auxiliaires - Sécurité	7
1.2.2 Difficultés de réalisation	8
1.3. Techniques utilisées	9
1.3.1. Les relais	9
1.3.2. Les diodes, transistors, circuits intégrés	9
1.3.3. Le microprocesseur	9
1.4. Moyens de description	10
1.4.1. Le langage courant	10
1.4.2. Tables logiques et chronogrammes	10
1.4.3. L'organigramme	11
1.4.4. Réseau de Pétri	12
1.5. Caractéristiques des systèmes de développement	13
1.5.1. Picolog 80	13
1.5.2. SBD MOSTEK F8	14
CHAPITRE II : CHOIX DU SENS ET DETERMINATION DES ARRETS	15
2.1. Description par réseaux de Pétri	15
2.1.1. Mémorisation des demandes	15
2.1.2. Détermination de la position	17
2.1.3. Choix du sens	17
2.1.4. Détermination des arrêts	22
2.2. Méthode de codage	25
2.2.1. Codages utilisés	25
2.2.1.1. Codages des envois-cabine	25
2.2.1.2. Codage des appels-montée et des appels- descente	26
2.2.1.3. Codage de la position de la cabine	26
2.2.1.4. Somme logique des demandes d'arrêt	27

2.2.2. Choix du sens	27
2.2.3. Détermination des arrêts	28
2.3. Ascenseurs jumelés	30

CHAPITRE III : PREMIERES APPLICATIONS

3.1. Commande cablée	33
3.2. Commande par microcalculateur	35
3.2.1. Détermination du sens	36
3.2.2. Détermination de la position de la cabine	37
3.2.3. Détermination des arrêts	39
3.2.4. Contrôles des auxiliaires	39
Annexe 3.1. Commande programmée d'ascenseur sur Microcalculateur Picolog 80, Organigramme général	40
Annexe 3.2. Programme ascenseur, double comparaison sur Picolog	45

CHAPITRE IV : CONTROLE PROGRAMMÉ

4.1. Les automates programmables industriels	52
4.2. Le microcalculateur	53
4.2.1. Solutions diverses	53
4.2.2. Solution adoptée	54
4.3. Automate programmable à partir de graphes d'état	55
4.3.1. Microcalculateur	55
4.3.2. Description de la commande	55
4.3.3. Exemple	56
4.3.4. Suivi des réseaux de Pétri	59
4.3.4.1. Table de structure	59
4.3.4.2. Table des événements	60
4.3.4.3. Table des actions	60
4.3.4.4. Table des variables	60
4.3.4.5. Sous-programme "Evolution"	61
4.3.5. Organisation générale	61
4.3.6. Mise en oeuvre	63
4.3.6.1. Les tables	63
4.3.6.2. Les programmes	64

CHAPITRE V : MISE EN OEUVRE GLOBALE

5.1. Organisation générale 71

5.2. Visualisation 74

5.3. Réseaux de Pétri 75

5.4. Temporisations 85

Annexe 5.1 : Programme d'une manoeuvre collective,
24 Niveaux, double comparaison, Pétri,
sur MOSTEK F8 87

CONCLUSION 101

BIBLIOGRAPHIE 102

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

L'évolution des technologies (relais, transistors, circuits intégrés) a permis d'améliorer la commande logique des ascenseurs. Il est facile d'imaginer que l'avènement du microprocesseur permettra d'élaborer des commandes de plus en plus sophistiquées. La mise en oeuvre de cette technique, qui aboutit à un matériel simple et peu coûteux, n'est cependant pas aisée puisqu'elle reporte les difficultés au niveau de la programmation de ces systèmes. Les méthodes de conception traditionnelles sont bouleversées car il faut employer de nouvelles démarches de pensée qui nécessitent de se forger de nouveaux outils de travail.

Après avoir expliqué succinctement le fonctionnement de l'ascenseur et réalisé un tour d'horizon des techniques habituellement employées par les constructeurs, nous scindons l'étude de la commande en deux parties. Nous donnons d'abord le principe d'une manoeuvre déterminant le sens de parcours et les arrêts par un procédé de double comparaison facile à programmer. Nous envisageons ensuite l'ensemble des systèmes de contrôle dont la caractéristique essentielle réside dans la diversité des cahiers des charges. Pour ce contrôle des auxiliaires, qui entre dans le cadre plus général de la commande des machines logiques industrielles, nous proposons un automate programmable d'une grande souplesse d'emploi, construit sur la base d'un microprocesseur MOSTEK F8.

Cet automate accepte directement la description des machines sous forme de graphes d'état dont les structures sont tabulées. Evénements, actions, et variables élémentaires sont également rangés dans des tables de façon à s'adapter facilement aux diverses spécifications sans entraîner de modification de programme.

Les autres opérations (détection de la position de la cabine, choix du sens, obtention des arrêts) sont programmées directement en langage machine sur le même microprocesseur.

CHAPITRE I

M E T H O D E S D ' E T U D E

La prospective a une conscience si aiguë de la transformation prodigieusement rapide du monde qu'elle ne conçoit plus l'homme autrement qu'en état d'éducation permanente.

Marcel DEMONQUE

Après avoir précisé le sujet de notre travail et cité les techniques utilisables, nous passons en revue les méthodes d'étude à notre disposition pour faciliter la mise en oeuvre des techniques. Pour clore ce chapitre, nous donnons quelques précisions sur les deux systèmes de développement dont nous disposons pour nos applications.

1.1. LES COMMANDES DANS L'ASCENSEUR

L'ascenseur est constitué principalement d'une cabine qui se déplace verticalement dans une gaine grâce à un treuil mû par un moteur électrique. C'est un moyen de transport sur lequel se penchent beaucoup de spécialistes. Le mécanicien construit la cabine, les guides, le treuil... L'électricien calcule le moteur adapté aux démarrages et freinages fréquents. L'automaticien étudie les commandes. De plus l'ascenseur est fait pour transporter des personnes ; il est donc soumis à des normes de sécurité (3) et contribue lui-même à la sécurité de la population d'un immeuble en cas de sinistre. On peut dire enfin qu'il ne doit pas nuire à l'esthétique des bâtiments et qu'il doit fonctionner sans bruit.

Définissons deux termes utilisés par les ascensoristes : entraînement et manoeuvre.

L'entraînement est le dispositif constitué du moteur et de sa commande, permettant de déplacer la cabine. La commande du moteur peut être plus ou moins élaborée. Il existe des systèmes de régulation qui permettent de régler l'accélération pour obtenir un déplacement confortable. Dans un hôpital, il est par exemple, important que le malade sortant de la salle d'opération ne subisse pas de secousse dans l'ascenseur. Il faut également que la cabine s'arrête bien au niveau du palier, quelle que soit la charge et quel que soit le sens de marche. Nous ne nous sommes pas intéressé dans ce travail à l'entraînement. C'est un domaine qui est aussi à explorer. Les performances peuvent certainement être améliorées en utilisant des techniques de pointe.

Par manoeuvre, on entend la commande logique de l'ascenseur. La manoeuvre surveille donc l'état des capteurs et donne des ordres à l'entraînement (marche, arrêt, montée, descente) ainsi qu'aux dispositifs annexes (fermeture des portes, éclairage ...) Tout comme on peut créer des entraînements plus ou moins élaborés, on peut également envisager des manoeuvres plus ou moins sophistiquées. Ces types de manoeuvres sont définis par des conventions d'usage. On distingue par exemple :

- la manoeuvre "blocage" : l'ascenseur ne satisfait qu'une demande à la fois.
- la manoeuvre "interception descente" : l'ascenseur peut être intercepté en descente seulement au dessus du niveau principal, et en montée seulement en dessous du niveau principal.
- la manoeuvre "collective montée - descente" : elle est aussi appelée "collective complète" ou "collective sélective" ; elle est l'objet de notre travail, aussi ses caractéristiques sont décrites dans le détail ci-après.

1.2. LA MANOEUVRE "COLLECTIVE MONTEE-DÉSCENTE"

Notre travail a donc consisté à étudier une manoeuvre et plus particulièrement la manoeuvre "collective montée - descente".

1.2.1. Caractéristiques de la manoeuvre

1.2.1.1. Loi de parcours

Sur chacun des paliers l'utilisateur dispose de deux boutons : le bouton d'appel-montée qu'il appuie s'il désire monter et le bouton d'appel-descente qu'il appuie s'il désire descendre. Au niveau inférieur, le bouton d'appel-descente n'existe pas et au niveau supérieur, le bouton d'appel-montée n'existe pas également. L'appui sur l'un ou l'autre de ces boutons est immédiatement "enregistré" c'est-à-dire mémorisé. Un voyant s'allume pour indiquer l'enregistrement. Un appel demeure enregistré aussi longtemps que la cabine ne s'est pas arrêtée au niveau demandé dans le sens convenable. L'arrivée de la cabine est indiquée par un voyant ou un signal sonore.

A chaque niveau, la porte palière est verrouillée si la cabine n'est pas présente derrière la porte. La cabine elle-même peut être munie ou non de porte. Si elle n'en possède pas, les passagers voient défiler la paroi de la gaine lorsque la cabine est en mouvement. Pour éviter tout accident, cette paroi doit posséder certaines qualités ; en particulier, elle doit être lisse. Cette disposition particulière est appelée "à paroi lisse". Elle sera interdite en Europe à partir de 1981.

Dans la cabine, les passagers disposent d'un bouton par niveau. Dès leur entrée en cabine, ils peuvent appuyer sur le bouton correspondant au niveau désiré. Ces demandes, appelées "envois-cabine" sont également enregistrées et le demeurent tant que la cabine ne s'est pas arrêtée au niveau demandé.

Comment la cabine va-t-elle se déplacer pour satisfaire toutes ces demandes des paliers et de la cabine ? Il est tout-à-fait inconcevable de respecter l'ordre chronologique des demandes : l'ascenseur serait peu rempli et ferait des parcours en tous sens. Il est nécessaire de définir une loi plus ou moins complexe qui régit son déplacement. Une loi simple est celle-ci : quand l'ascenseur a pris un sens de parcours, il garde ce sens jusqu'à la demande la plus éloignée, qu'elle provienne de la cabine ou d'un palier. Il satisfait en cours de route les demandes d'arrêt provenant de la cabine et les demandes des paliers correspondant au sens de déplacement. Cette loi est celle de la manoeuvre "collective montée - descente".

Cette manoeuvre est destinée au cas d'échanges variés et fréquents entre niveaux (immeubles à bureaux, grands magasins, par exemple).

Il peut arriver à certains moments que l'ascenseur ne soit plus sollicité. La cabine peut demeurer là où le dernier passager l'a quittée, ou aller d'elle-même à un niveau privilégié. Lorsqu'un usager trouve ainsi la cabine présente à son niveau, en attente de nouvelles sollicitations, il a priorité pendant trois secondes après la fermeture des portes pour définir le sens de marche.

Le personnel d'entretien et de dépannage peut se placer sur le toit de la cabine. Il a alors à sa disposition une commande permettant d'inhiber la fonctionnement normal de l'ascenseur et deux commandes manuelles de montée et descente de la cabine.

1.2.1.2. Détection de la position de la cabine

La position de la cabine doit être repérée avec suffisamment de précision et de sûreté. Les deux techniques suivantes sont maintenant abandonnées :

- à chaque niveau, un capteur électromécanique de type "basculeur" à trois positions indique si l'ascenseur est présent ou si l'ascenseur montait ou descendait lors de son dernier passage.
- en machinerie, un dispositif mécanique reproduit à l'échelle réduite le mouvement de la cabine. Ce répéteur agit sur des contacts qui indiquent la position de la cabine.

L'électronique offre maintenant un procédé plus précis et plus robuste. Une fourche munie d'un capteur photoélectrique est fixé sur un côté de la cage. Des écrans sont disposés dans la gaine de façon qu'ils traversent la fourche lors de son passage, coupant ainsi le faisceau optique (fig 1.1.). Ces impulsions recueillies sont acheminées dans un compteur- décompteur qui peut être prépositionné par un capteur supplémentaire situé sur une fin de parcours de la cabine. En cas de coupure du secteur, le compteur perd l'information de position. A la remise sous tension, la cabine est envoyée d'office en bout de course pour "recaler" le compteur.

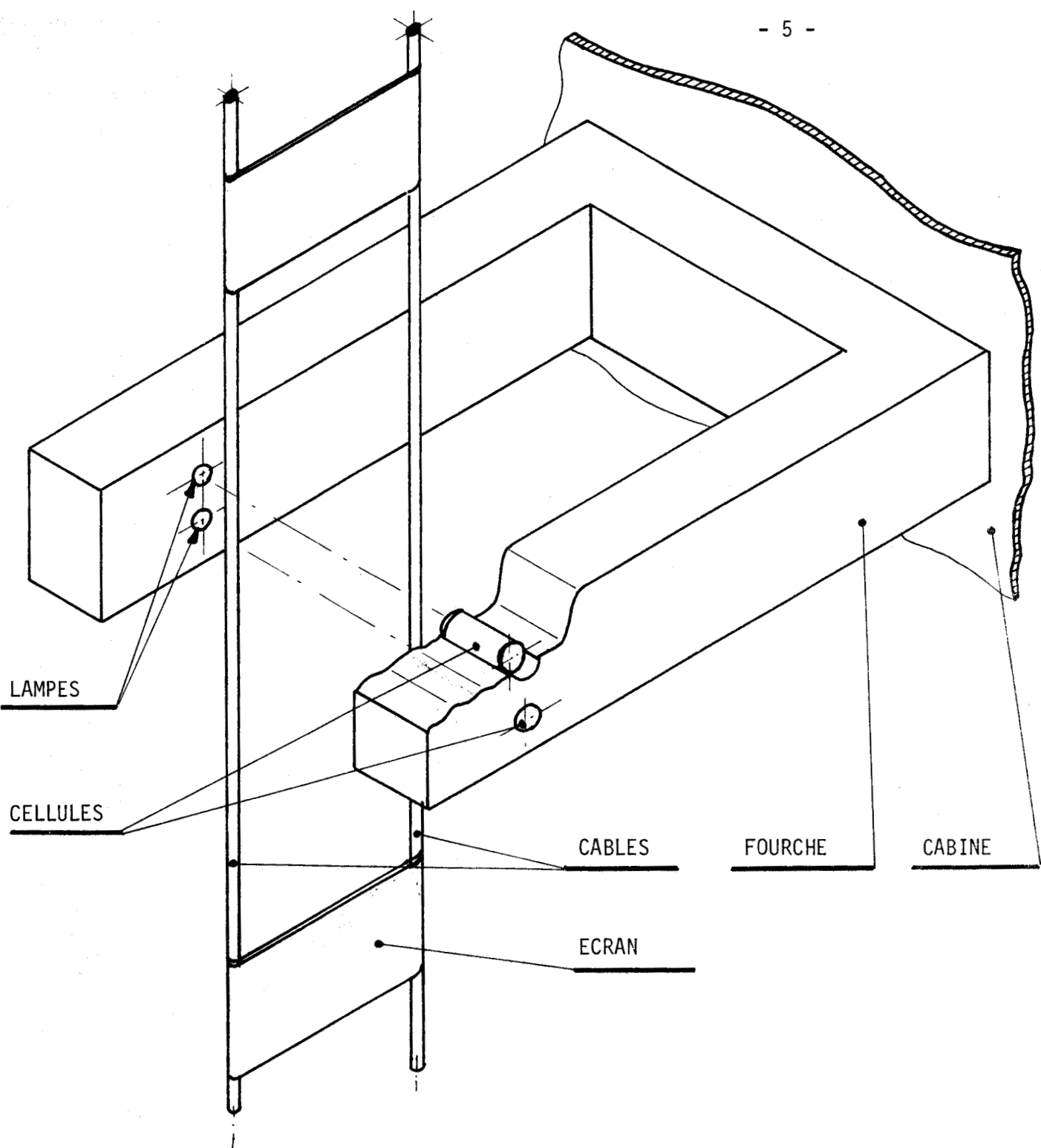


Figure 1.1.

En pratique, pour les appareils à deux vitesses, on dispose quatre écrans à chaque niveau (fig 1.2.).

En descente et pour un arrêt prévu à l'étage i, l'arrivée sur l'écran K1 provoque le passage en petite vitesse et la sortie de l'écran K2 commande l'arrêt. Avec l'inertie, la cabine s'immobilisera un peu plus loin. Les écrans K3 et K4 jouent des rôles identiques dans le cas d'un arrêt prévu à l'étage i dans le sens de la montée.

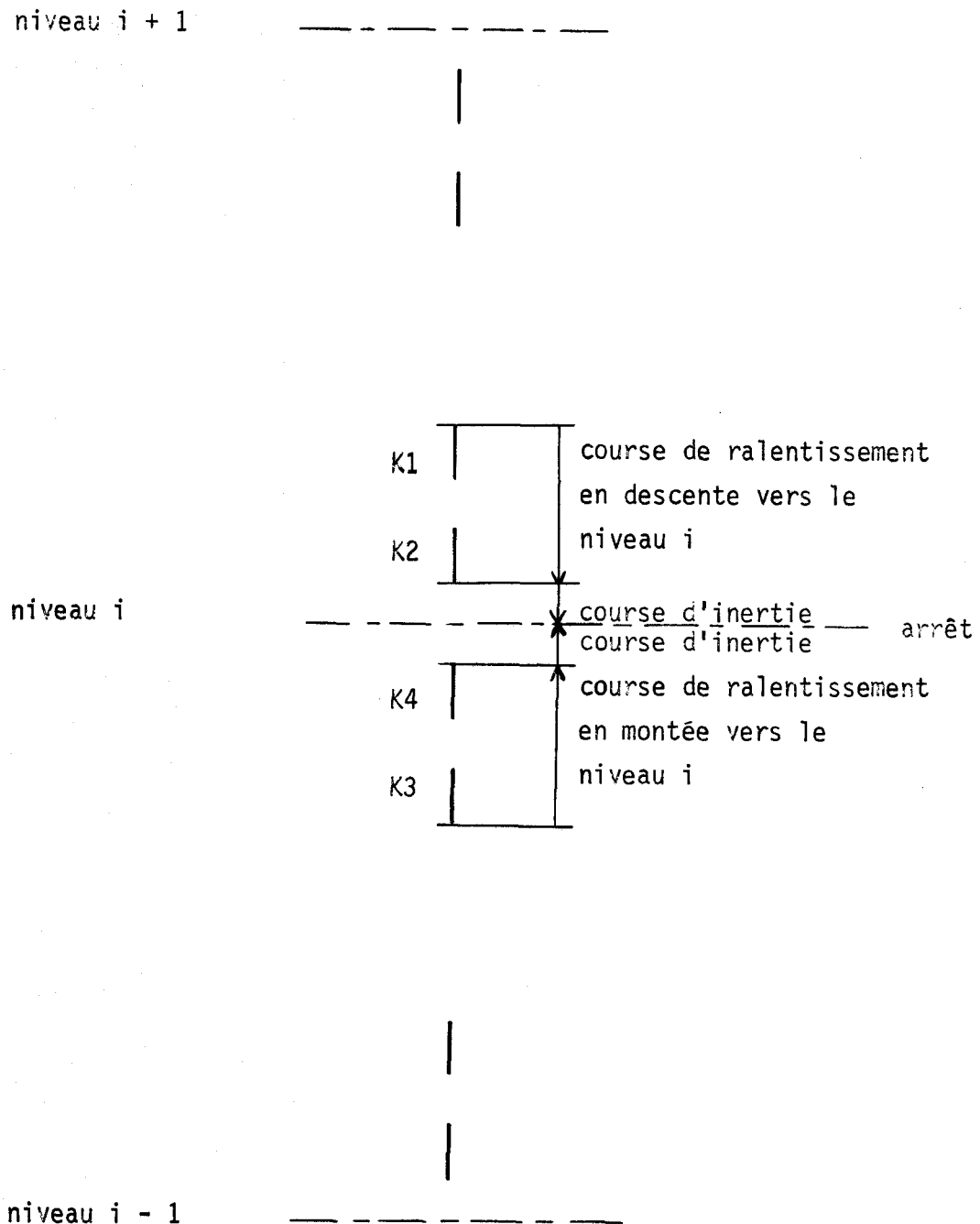


Figure 1.2. .



On peut ajouter un perfectionnement en plaçant dans la fourche deux capteurs, l'un au dessus de l'autre. Suivant l'ordre dans lequel ces capteurs sont excités, on peut connaître le sens de parcours réel de la cabine qui peut, en cas d'incident, être différent de celui qui est commandé. Lors de travaux sur la distribution électrique d'un immeuble, par exemple, l'ordre des phases peut-être changé et le moteur de traction tournera alors en sens inverse du sens normal.

1.2.1.3. Auxiliaires - Sécurité

Dans ce paragraphe, nous donnons un aperçu sur les dispositifs auxiliaires électriques qui se trouvent dans l'ascenseur ; il faut avoir à l'esprit qu'il faut veiller à la sécurité des personnes et du matériel.

Les portes palières sont diversement constituées : on trouve la porte battante, la porte coulissante, la porte pliante. Elles peuvent être ouvertes et fermées manuellement ou automatiquement. Les portes des cabines sont aussi de diverses espèces et peuvent même ne pas exister (système à paroi-lisse). Dans les constructions récentes, si les portes des cabines existent, elles sont commandées automatiquement et entraînent en même temps les portes palières. Prenons le cas d'une installation à paroi lisse, aux portes palières battantes, entraînées manuellement. La fermeture de la porte est d'abord contrôlée par un premier contact du type à arrachement : une partie du contact se trouve sur la porte ; en cas de soudure accidentelle, le contact est "arraché" par l'utilisateur qui ouvre la porte. Le verrouillage de la porte est ensuite obtenu par le retrait d'une came mobile solidaire de la cabine et commandée par un électroaimant. Un deuxième contact lié mécaniquement au pêne de la serrure vérifie le verrouillage. Ce contrôle est préalable à toute possibilité de départ de la cabine. Cette serrure est appelée "à contrôle de pêne préalable".

La cabine doit être éclairée lorsqu'un usager pénètre dans celle-ci. L'éclairage peut cesser quelques instants après le départ du dernier usager.

En plus des boutons et signalisations indiqués précédemment, on trouve encore dans la cabine :

- une signalisation de position et de sens de parcours
- un voyant de surcharge. Il s'allume lorsque la charge dans la cabine dépasse la limite permise. Dans ce cas, un ou plusieurs passagers doivent descendre.
- un bouton d'arrêt d'urgence. L'appui sur ce bouton permet aux passagers d'obtenir l'arrêt immédiat de la cabine. Le redémarrage n'est autorisé qu'après un appui sur un bouton d'envoi-cabine. L'appui sur les boutons des paliers est dans ce cas inopérant. La pose de ce bouton d'arrêt d'urgence est actuellement déconseillé, et sera même interdite, dans les appareils à portes automatiques.

- un bouton de sonnerie. Dans le cas où la cabine est coïncée entre deux étages, l'appui sur ce bouton fait retentir une sonnerie chez le concierge, par exemple, qui appelle alors le service de dépannage.

Il faut prévoir parfois un bouton d'appel prioritaire pour les pompiers appelé "appel-pompiers", et situé au palier du niveau d'accès du public.

Si le moteur d'entraînement s'échauffe anormalement, un relais thermique permet d'arrêter l'ascenseur au niveau le plus proche. De même un défaut d'isolement (perte de courant à travers la masse) fait arrêter l'installation.

Enfin deux contacts de fin de course haut et bas doivent impérativement arrêter le moteur, s'ils sont excités.

En marche, l'adhérence des câbles sur la poulie d'entraînement est contrôlée en permanence. Pratiquement le contrôle consiste à vérifier la fermeture régulière et passagère d'un contact appelé contact d'intégrateur.

1.2.2. Difficultés de réalisation

Dans une manoeuvre on peut distinguer deux parties :

- choix du sens de parcours et détermination des arrêts
- contrôle des dispositifs auxiliaires.

Le profane qui réfléchit au problème de la commande d'ascenseur pense immédiatement à la première partie. Sens et arrêt sont certes deux résultats importants de la manoeuvre. Mais il est nécessaire pour une réalisation, de descendre dans les détails. Ainsi par exemples :

- les rebondissements du contact de verrouillage de la porte doivent être tolérés par le dispositif de contrôle.
- en cas de coupure du secteur, l'ascenseur doit reprendre son service normal dans tous les cas, à la réapparition de la tension d'alimentation.

Ce deuxième aspect de la manoeuvre est à priori moins exaltant à étudier. C'est là pourtant qu'il faut faire preuve d'imagination et de savoir-faire.

1.3. TECHNIQUES UTILISEES

Suivant le prix de revient des composants, les techniques ont évolué.

1.3.1. Les relais

L'ascenseur est né avec la technique des relais électromécaniques. Les commandes à relais ne se fabriquent plus mais beaucoup sont encore en service. Il faut dire que le relais lui-même a évolué : miniaturisation, meilleure fiabilité, diminution de prix. Pour l'entretien et de dépannage, le technicien voit réellement tous les circuits. Il trouve facilement une liaison coupée.

1.3.2. Les diodes, transistors, circuits intégrés.

Les tubes à vide n'ont pas été utilisés mais par contre les diodes et transistors ont permis d'élaborer une logique sur cartes d'encombrement réduit. Les circuits intégrés T.T.L. n'ont pas eu de succès à cause de leur sensibilité aux parasites. Par contre, les circuits C.MOS, moins sensibles aux parasites, sont utilisés. Pour le dépannage, le technicien doit maintenant être spécialiste, il doit avoir des connaissances en électronique.

1.3.3. Le microprocesseur

Il y a quelques années, on a peut-être pu imaginer d'utiliser un ordinateur pour commander un ascenseur. Cet ascenseur aurait fonctionné à la perfection, grâce à des contrôles multiples. En cas d'ennuis techniques, il se serait quelquefois dépanné lui-même ou aurait fonctionné sur un programme réduit en attendant l'intervention du technicien. L'ordinateur indiquerait quel contact est en défaut, quel voyant est grillé. Il aurait même pu converser à distance avec un service technique au moyen d'une ligne téléphonique. Ce rêve de science-fiction est en train de se réaliser grâce à l'avènement des microprocesseurs. Les composants ne coûtent pas chers, mais la mise en oeuvre des techniques programmées demande du temps. Elle n'est pas évidente car il faut bien connaître et l'ascenseur et ces techniques programmées, mais elle progresse.

Dans l'état actuel des choses, le technicien d'entretien, habitué aux techniques câblées est désorienté. C'est à la fois plus simple et plus compliqué :

- plus simple, car la commande donne des indications sur la nature de la panne, et si c'est la carte de commande elle même qui est défectueuse, on ne peut guère que la remplacer.

- plus compliqué, car les mesures directes sont moins aisées. Ainsi, par exemple, le microcalculateur explore la position des boutons par multiplexage. Ce procédé économise le nombre des conducteurs mais ne permet pas des vérifications aisées de bon fonctionnement.

les autres sont plus simples

1.4. MOYENS DE DESCRIPTION

Nous avons vu que l'ascenseur est une machine complexe. Ajoutons que dans chaque type de manoeuvres, il existe beaucoup de variantes et de cas particuliers. Il faut par exemple prévoir une priorité pour le chef de service dans l'utilisation de l'ascenseur installé dans un hôpital. De quels moyens dispose-t-on pour décrire le comportement de la machine et des variantes et cas particuliers ?

1.4.1. Le Langage courant

Le langage courant est nécessaire pour exposer les problèmes, définir le vocabulaire utilisé, expliquer les solutions retenues. Cependant le langage à lui seul devient lourd si on descend dans le détail du fonctionnement et si on veut étudier toutes les configurations possibles.

1.4.2. Tables logiques et chronogrammes

Pour définir ou éclairer certaines situations, on s'aide de tables de vérité ou de tables de Karnaugh. Les chronogrammes permettent également de décrire l'évolution de signaux dans le temps. Les gens de métier arrivent ainsi au moyen du langage, de tables et de chronogrammes à construire des commandes câblées aux performances remarquables. Les notices sont naturellement complétées par les schémas de réalisation qui aident aussi à la compréhension du fonctionnement.

1.4.3. L'Organigramme

Avec l'avènement des techniques programmées il est nécessaire d'utiliser des moyens de description adaptés. Le premier qui vient à l'esprit est l'organigramme, très répandu dans le monde de l'informatique. Les opérations sont décrites par un rectangle s'il s'agit d'un traitement ou par un losange s'il s'agit d'un test. La nature des opérations est indiquée à l'intérieur du rectangle et du losange. Les rectangles et losanges sont reliés entre eux par des arcs orientés suivant l'ordre de déroulement des opérations. Le concepteur d'automatismes logiques, habitué à réfléchir au moyen de tables et d'expressions logiques doit revoir sa façon de penser s'il adopte l'organigramme.

L'organigramme est intéressant pour décrire une suite d'opérations à effectuer pour résoudre un problème de type mathématique. Il n'est pas très bien adapté pour la description du fonctionnement/de machines automatiques. En prenant pour base le jeu d'instructions d'un microprocesseur, il n'est pas possible de programmer directement des expressions logiques combinatoires. Un système séquentiel simple est par contre plus facilement programmable. La difficulté apparaît lorsque le système séquentiel se complique. Il faut avoir à l'esprit que le calculateur, aussi puissant soit-il, ne fait qu'une opération à la fois et que d'autre part, il ne peut s'arrêter. Les machines logiques, au contraire, peuvent avoir des sous-ensembles qui évoluent simultanément, et sont souvent dans des états stables en attente d'un ou plusieurs événements. Le fonctionnement du calculateur est donc en contradiction avec le fonctionnement des machines logiques qu'il doit commander. Dans le cas de la résolution d'un problème mathématique, le calculateur travaille à sa cadence, sans être lié à des contingences extérieures.

Il est certes possible d'utiliser quand même l'organigramme pour décrire la commande d'une machine. Il faut cependant être très perspicace pour établir un organigramme qui permette de surveiller toutes les entrées nécessaires dans tous les cas : des chemins dans l'organigramme doivent être tracés pour tous ces cas, et on refait des tests bien souvent inutiles. En outre, les modifications et les extensions sont très difficiles à réaliser.

Une autre manière de décrire les machines logiques industrielles est constituée par les réseaux de Pétri qui permettent l'attente dans une ou plusieurs places et l'évolution simultanée.

Il est donc nécessaire d'utiliser un calculateur de commande qui connaisse les règles d'évolution des réseaux de Pétri. Ces machines spécialisées commencent à peine à apparaître sur le marché. Il est cependant très possible d'écrire un logiciel qui transforme un microcalculateur ordinaire en une machine spécialisée qui comprenne les réseaux de Pétri. Le microcalculateur ne fera toujours qu'une opération à la fois et ne s'arrêtera jamais, mais vu de l'utilisateur, le microcalculateur muni de son logiciel peut suivre plusieurs réseaux ou plusieurs branches dans un réseau et peut s'arrêter dans les places aussi longtemps qu'on le désire.

Pour l'utilisation des réseaux de Pétri comme moyen de représentation des systèmes logiques, nous prenons les règles générales habituellement retenues dans ce cas.

Dans un réseau de type S, on trouve :

- les places, auxquelles on affecte les actions, ou ordres ou commandes ou encore sorties,
- les transitions, auxquelles on affecte les événements, fonctions logiques des entrées,
- le marquage, qui évolue à partir d'un marquage initial.

Dans un réseau de type T, les actions sont affectées non aux places, mais aux transitions et sont donc lancées au franchissement de la transition. Ces actions demeurent efficaces jusqu'à la commande des effets contraires.

Lors de l'étude des machines logiques, il nous a semblé plus simple de ne nous intéresser qu'aux variables qui changent. D'autre part, les organes de sortie des calculateurs sont munis de mémoires. Nous avons donc adopté la représentation des machines logiques par des réseaux de type T.

Cependant nous avons noté les actions au niveau de la place qui suit la transition de lancement. La mise à 1 d'une variable est indiquée par l'écriture de la variable affirmée, la mise à 0 par l'écriture de la variable complémentée. Ainsi sur la figure 1.3, l'action M est au niveau 1 aux places 1,2 et 4 et l'action N est au niveau 1 pour la place 1.

Nous avons ensuite pris une représentation qui n'est constituée que de graphes d'état, en nous référant à la définition classique des graphes d'état : chaque transition d'un graphe d'état a une place d'entrée et une place de sortie.

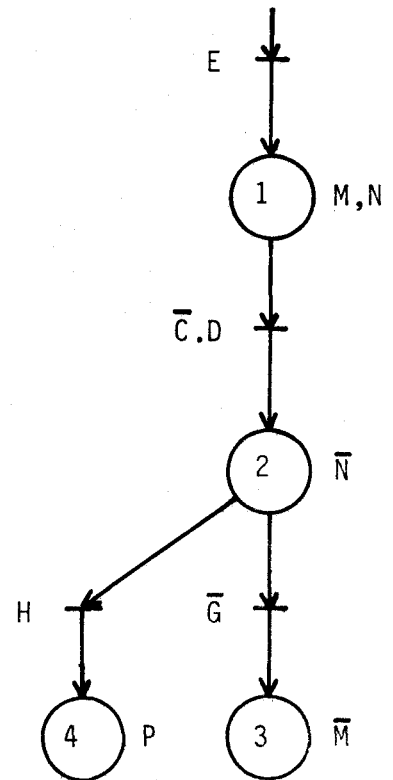


Figure 1.3

La description d'une machine peut être faite de façon globale par un seul graphe. Dans ce cas, il peut y avoir évolution simultanée de marqueurs et le nombre de ces marqueurs peut varier. La machine peut aussi être décomposée en sous-ensembles de manière que chaque sous-ensemble soit décrit par un graphe d'état. Chaque graphe d'état ne possède qu'un marqueur. Nous avons remarqué que pratiquement, la description d'une machine logique est aussi aisée, et même souvent plus aisée, à effectuer en décomposant cette machine en sous-ensembles qu'en la considérant globalement. De toutes façons, on peut toujours décomposer un graphe quelconque en plusieurs graphes d'état.

I.5. CARACTERISTIQUES DES SYSTEMES DE DEVELOPPEMENT

I.5.1. Picolog 80

Ce système de développement, paru très tôt sur le marché et basé sur le 8080 d'INTEL, comprend le microcalculateur enfermé dans un coffret, un téléimprimeur et un lecteur rapide de rubans. Le microcalculateur est muni d'une carte horloge programmable à 1, 10, 100, et 1000 Hz. On dispose d'un moniteur résident, d'un éditeur et d'un assembleur non résidents. Cet

ensemble a l'avantage de n'être pas très onéreux, mais n'est pas d'exploitation très aisée. La moindre erreur sur le ruban source demande le rechargement de l'éditeur et de l'assembleur.

1.5.2. S B D MOSTEK F8

Le système de développement est, comme le nom l'indique (Single Board Development), un système sur une seule carte basé sur le F8. D'origine, il peut être relié à un téléimprimeur ou à un écran-clavier. L'assembleur est résident et l'éditeur, non résident à l'origine a été aménagé et mis en PROM par AUTINOR. Pour faciliter son exploitation, ce système a été entouré progressivement d'un lecteur de bandes rapide, d'une imprimante à 110 caractères par seconde et d'une mémoire à disques souples.

En fait, nous avons suivi AUTINOR dans son choix. Cette société a choisi en 1976 le microprocesseur MOSTEK F8 parce que ce dernier permet d'aboutir à des réalisations ne comportant que très peu de boîtiers (le système minimal est constitué de deux circuits, dans la version traditionnelle et même d'un seul circuit avec le "monochip" MOSTEK 3870). Le CPU possède un double jeu d'entrées et de sorties, bidirectionnelles, sur 8 bits. Il renferme 64 octets de RAM, un timer et une possibilité d'interruption vectorisée.

Il est apparu aussi pour AUTINOR que le jeu d'instructions convient très bien pour les traitements logiques. Pour notre part, la présence d'un pointeur de données nous a semblé intéressante pour les traitements de tableaux que nous voulions faire. En cas d'appels à des sous-programmes imbriqués, la gestion de pile n'est pas automatique. Cet inconvénient ne nous gêne guère étant donné la modularité de nos programmes.

o o
o

Les nouvelles techniques sont merveilleuses. Le particulier fait avec précision des calculs sur sa machine de poche. L'étudiant manipule aisément des expressions matricielles récurrentes sur un calculateur de table. La commande des machines peut être réalisée également par calculateur. Dans ce domaine cependant, l'application n'est pas immédiate. Il faut d'abord connaître le calculateur, le tester dans le contexte d'utilisation, faire un choix entre plusieurs modèles et savoir quel sera le coût de la réalisation. Après ce travail préliminaire, un autre travail important d'études s'impose, où l'on doit avoir des méthodes, avant de sortir un prototype.

CHAPITRE II

CHOIX DU SENS ET DETERMINATION DES ARRETS

Il serait sage, à mon avis, que nous nous réunissions pour discuter de la situation.

AGATHA CHRISTIE

Dans ce chapitre nous examinons la première partie de la commande : choix du sens de parcours et détermination des arrêts. Cette partie est d'abord décrite par les réseaux de Pétri. Puis nous introduisons une méthode de codage. Nous terminons par le cas des ascenseurs jumelés.

2.1. DESCRIPTION PAR RESEAUX DE PETRI

Les réseaux de Pétri, avec les règles particulières définies au chapitre premier, permettent de décrire la mémorisation des demandes, de définir la position de la cabine, de choisir le sens de parcours et de déterminer les arrêts. Dans ce paragraphe, nous prenons comme exemple un ascenseur à 24 niveaux notés de 00 à 23.

2.1.1. Mémorisation des demandes

Les envois-cabine et les différents appels-paliers sont mémorisés par de simples cellules mémoires à deux places à déclenchement prioritaire. En prenant les notations suivantes :

M0 : montée (voir § 2.1.3)

$\overline{M0}$: descente

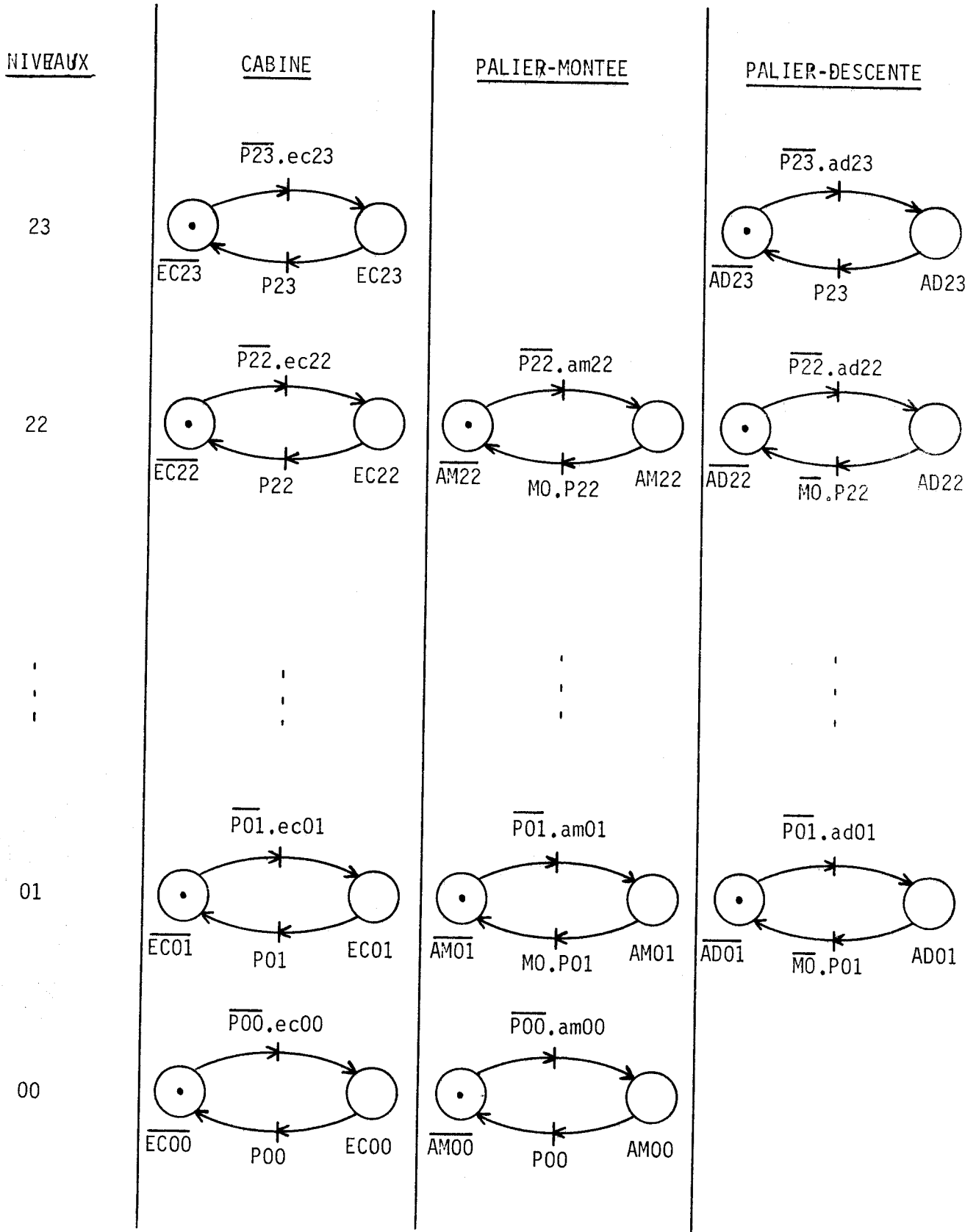


Figure 2.1 : Mémorisation des demandes



$ec\ i$: signal d'envoi-cabine au niveau i
 $ad\ i$: signal d'appel-descente du niveau i
 $am\ i$: signal d'appel-montée du niveau i
 $EC\ i$: envoi cabine au niveau i enregistré
 $AD\ i$: appel-descente du niveau i enregistré
 $AM\ i$: appel-montée du niveau i enregistré
 P : indication de position de la cabine au niveau i (fournie par un autre réseau).

on obtient, pour 24 niveaux, les graphes de la figure 2.1. Un envoi-cabine ou un appel-palier ne peuvent être enregistrés si la cabine est présente au niveau demandé.

2.1.2. Détermination de la position

Le capteur est constitué de deux cellules A et B situées dans une fourche solidaire de la cabine et par des écrans placés dans la gaine à raison de quatre écrans par niveau (voir § 1.2.1.2. et figure 2.2.). La détection du sens est étudiée au chapitre suivant.

La position de la cabine est connue très exactement par le réseau de Pétri de la figure 2.3 qui se présente comme un registre dans lequel le marqueur peut se déplacer, dans un sens ou dans l'autre, à chaque variation du mot AB.

2.1.3. Choix du sens

Le sens de parcours est défini par une seule sortie MO qui prend les deux valeurs :

MO = 1 si la cabine doit monter

MO = 0 si la cabine doit descendre.

A l'arrêt, MO peut garder indifféremment les valeurs 0 ou 1.

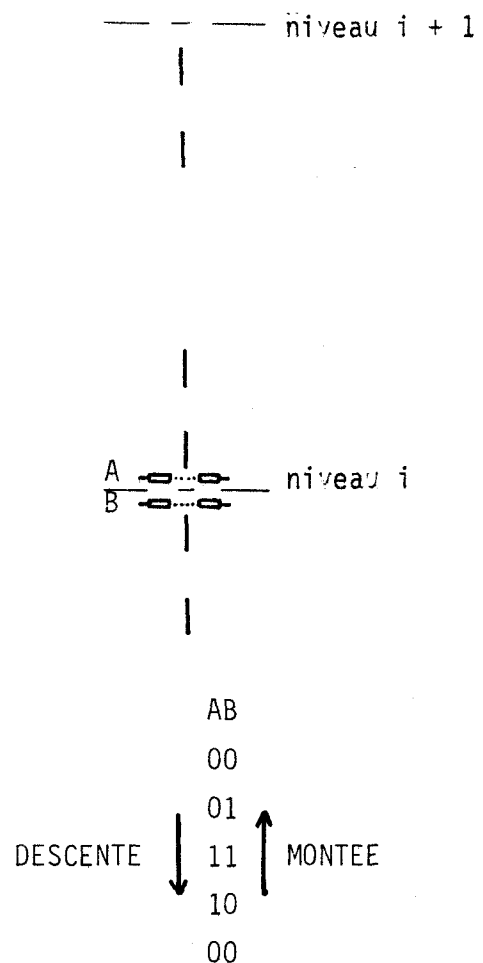


Figure 2.2.

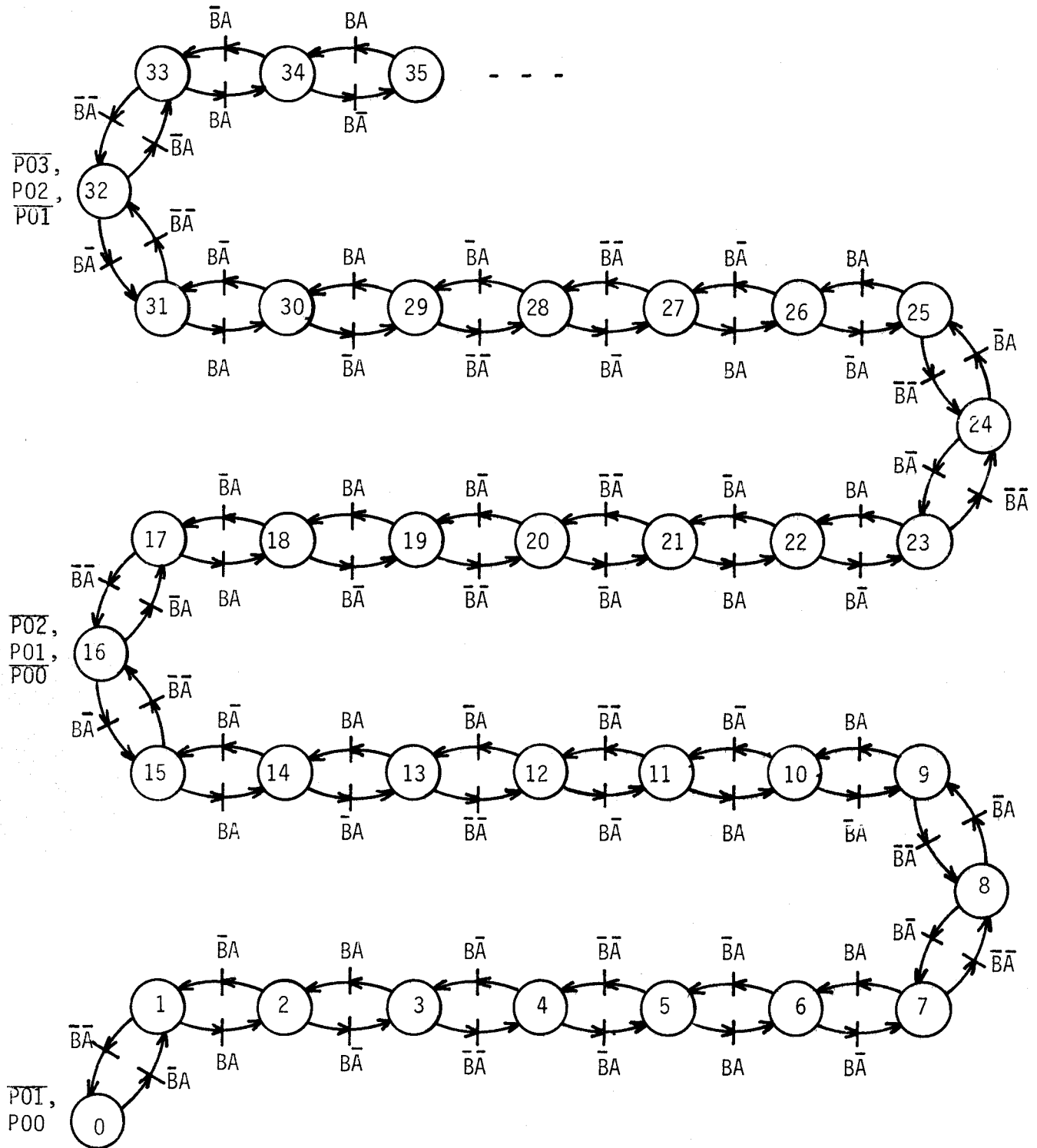


Figure 2.3



Le choix du sens de parcours dans la manoeuvre "collective" se représente, d'après la définition donnée au chapitre précédent, suivant le réseau de Pétri de la figure 2.4.

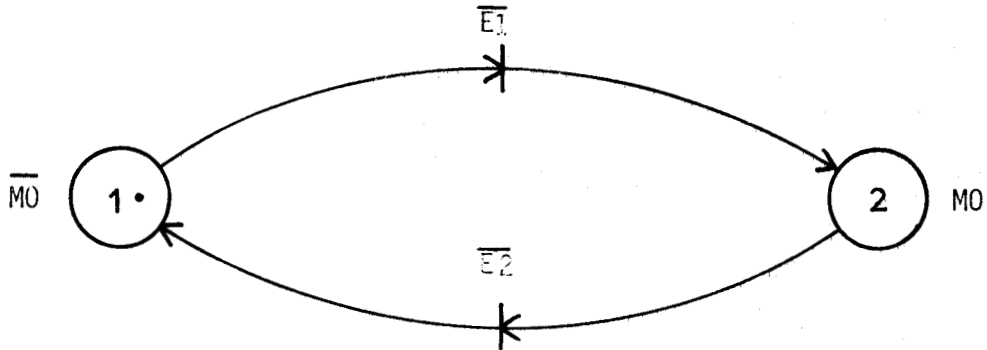


Figure 2.4.

L'évènement $\overline{E1}$ est présent si la proposition :
" il n'y a plus de demande en dessous de la position "
 est vérifiée.

De même l'évènement $\overline{E2}$ est présent si la proposition :
" il n'y a plus de demande au dessus de la position "
 est vérifiée.

Les évènements $\overline{E1}$ et $\overline{E2}$ sont les compléments des fonctions $E1$ et $E2$ explicitées ci-après :

$$\begin{aligned}
 E1 = & P23 (EC\ 22 + AM\ 22 + AD\ 22 \\
 & + EC\ 21 + AM\ 21 + AD\ 21 \\
 & + \dots \\
 & + EC\ 00 + AM\ 00) \\
 & + P22 (EC\ 21 + AM\ 21 + AD\ 21 \\
 & + EC\ 20 + AM\ 20 + AD\ 20 \\
 & + \dots \\
 & + EC\ 00 + AM\ 00) \\
 & + \dots \\
 & + P01 (EC\ 00 + AM\ 00)
 \end{aligned}$$

$$\begin{aligned}
 E2 = & P_{00} (EC_{01} + AM_{01} + AD_{01} \\
 & + EC_{02} + AM_{02} + AD_{02} \\
 & + \dots\dots\dots \\
 & + EC_{23} + AD_{23}) \\
 & + P_{01} (EC_{02} + AM_{02} + AD_{02} \\
 & + EC_{03} + AM_{03} + AD_{03} \\
 & + \dots\dots\dots \\
 & + EC_{23} + AD_{23}) \\
 & + \dots\dots\dots \\
 & + P_{22} (EC_{23} + AD_{23})
 \end{aligned}$$

Ces expressions s'écrivent également :

$$\begin{aligned}
 E1 = & (EC_{00} + AM_{00} \dots\dots\dots) (P_{23} + P_{22} + \dots\dots\dots + P_{01}) \\
 & + (EC_{01} + AM_{01} + AD_{01}) (P_{23} + P_{22} + \dots\dots\dots + P_{02}) \\
 & + \dots\dots\dots \\
 & + (EC_{22} + AM_{22} + AD_{22}) (P_{23})
 \end{aligned}$$

$$\begin{aligned}
 E2 = & (EC_{23} + AD_{23} \dots\dots\dots) (P_{22} + P_{21} + \dots\dots\dots + P_{00}) \\
 & + (EC_{22} + AM_{22} + AD_{22}) (P_{21} + P_{20} + \dots\dots\dots + P_{00}) \\
 & + \dots\dots\dots \\
 & + (EC_{01} + AM_{01} + AD_{01}) (P_{00})
 \end{aligned}$$

Le réseau obtenu constitue une mémoire SR. On peut affecter $\overline{E1}$ à S, $\overline{E2}$ à R et M0 à Q (figure 2.5)

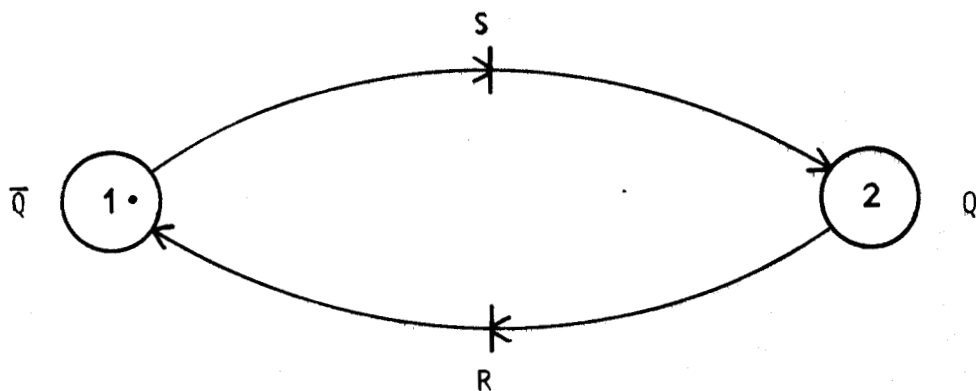


Figure 2.5.

Tant qu'il y a des commandes, cette mémoire fonctionne correctement. Dans le cas où il n'y a plus de commande, on obtient $E1 = E2 = 1$. Les deux transitions sont toujours franchies, le pointeur va d'une place à l'autre la sortie Q oscille. On peut éviter ces oscillations en définissant la sortie pour $S = R = 1$, de l'une ou l'autre des trois manières suivantes :

- la sortie garde l'état précédent : voir figure 2.6.

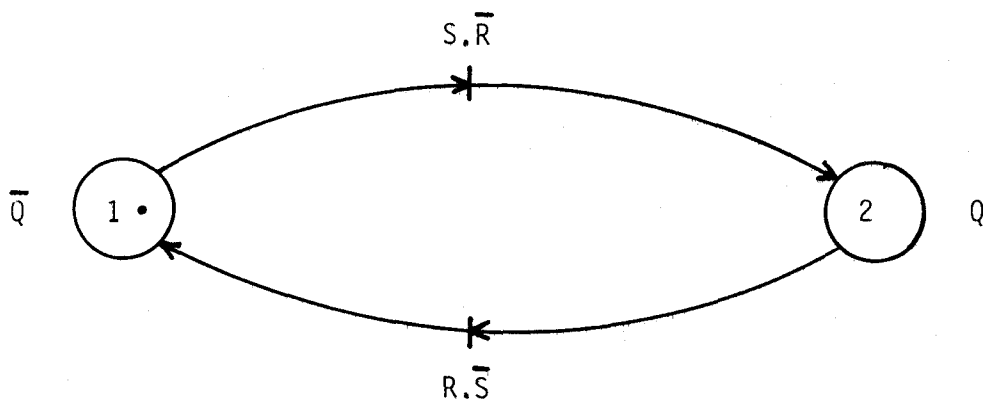


Figure 2.6.

- la sortie prend la valeur 1 : voir figure 2.7

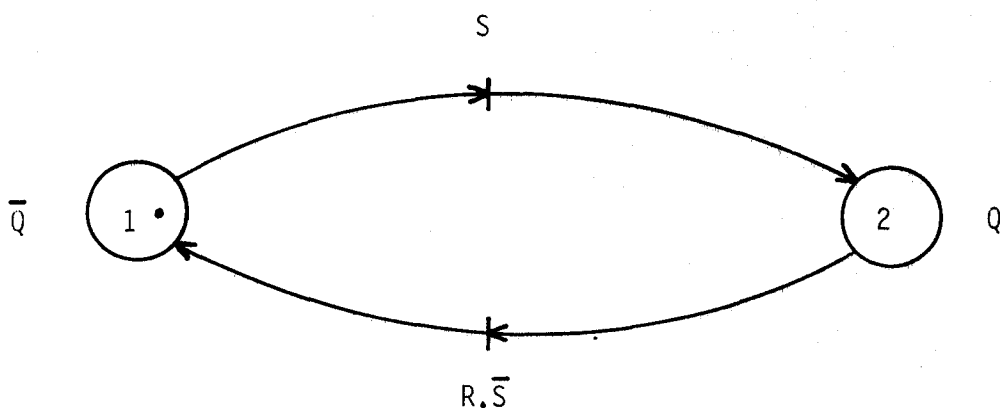


Figure 2.7.

- la sortie prend la valeur 0 : voir figure 2.8.

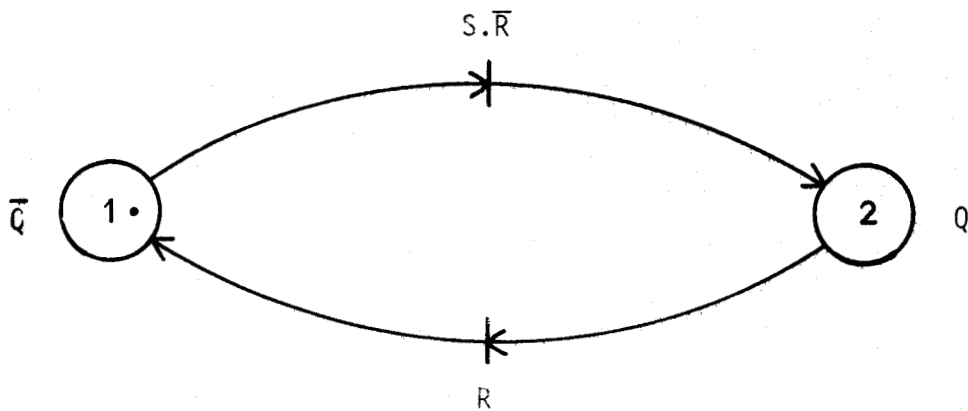


Figure 2.8.

Pour l'application qui nous concerne, la première solution est à retenir. Elle évite un changement de la sortie qui peut être inutile (figure 2.9)

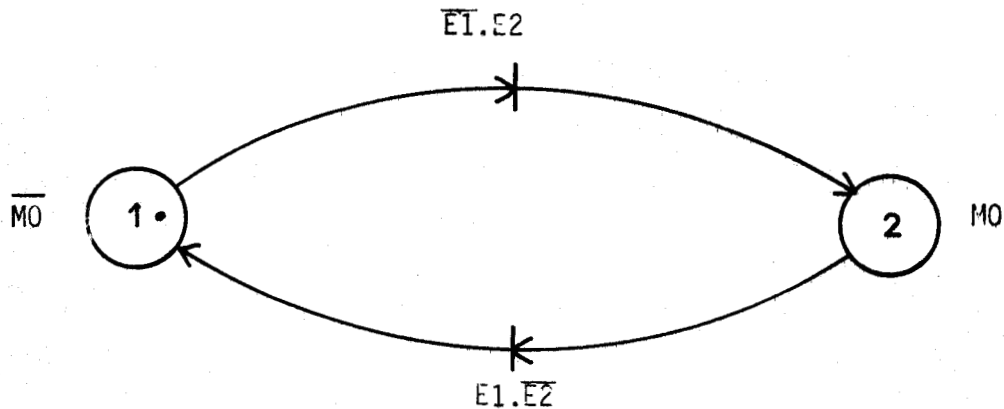


Figure 2.9.

2.1.4. Détermination des arrêts

Précisons, pour commencer ce paragraphe, que la marche et l'arrêt ne sont pas seulement fonctions des demandes et de la position. L'ordre de marche n'est donné que lorsque toutes les conditions de sécurité sont remplies. De même l'arrêt peut-être demandé par l'apparition d'un défaut, ou inhibé par



un appel-pompiers. Aussi, nous ne définirons pas un ordre de marche à proprement, mais une "demande de marche" notée DM. Le complément \overline{DM} de DM est également une demande d'arrêt qui entre en ligne de compte avec d'autres impératifs. La variable DM sera donc reprise par le sous-ensemble de contrôle qui définit exactement la marche et l'arrêt de la cabine.

La demande de marche existe dès qu'un appel-palier ou un envoi-machine est enregistré, c'est-à-dire si $EN = 1$ avec :

$$\begin{aligned} EN = & EC\ 00 + AM\ 00 \\ & + EC\ 01 + AM\ 01 + AD\ 01 \\ & + \\ & + EC\ i + AM\ i + AD\ i \\ & + \\ & + EC\ 23 + AD\ 23 \end{aligned}$$

La variable EN est aussi une fonction de E1 et E2. En effet, s'il n'existe plus de demande ni pour monter, ni pour descendre, c'est à dire si $E1 = E2 = 0$, l'ascenseur doit rester à l'arrêt. Il doit fonctionner dans les cas contraires. On peut donc écrire :

$$EN = E1 + E2$$

ou compléter le graphe de la figure 2.10 pour élaborer cette variable EN.

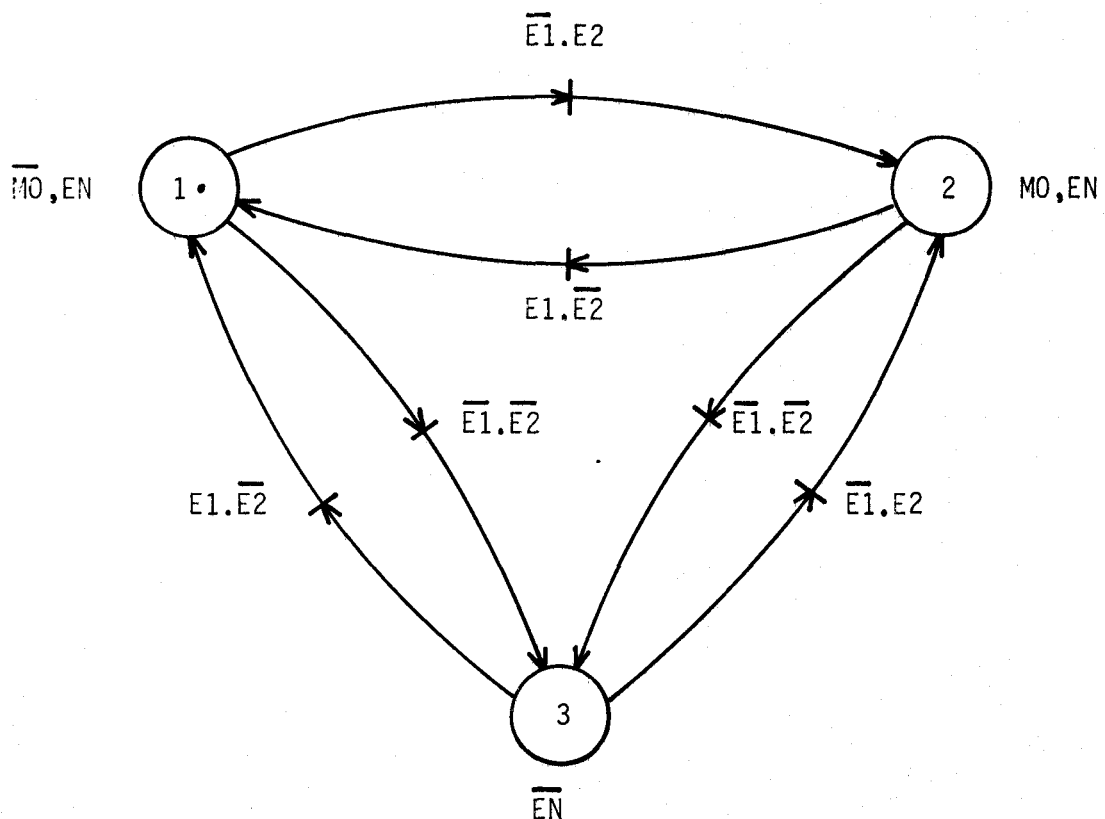


Figure 2.10

Lorsque la cabine passe au niveau i , la demande d'arrêt a lieu dans les cas suivants :

- un appel-montée est enregistré à ce niveau et la cabine monte
- un appel-descente est enregistré à ce niveau et la cabine descend
- un envoi-cabine est enregistré à ce niveau, quel que soit le sens de parcours de la cabine
- un appel-montée est enregistré à ce niveau, la cabine descend et cet appel est le dernier dans le sens de la descente
- un appel-descente est enregistré, à ce niveau, la cabine monte et cet appel est le dernier dans le sens de la montée.

Les trois premières demandes d'arrêt sont appelées : demandes d'arrêt "parcours" et les deux dernières, demandes d'arrêt "bout de course".

Toutes ces conditions d'arrêt peuvent s'écrire :

$$\begin{aligned}
 CO &= P \ 00 \ (EC \ 00 + AM \ 00) \\
 &+ P \ 01 \ (EC \ 01 + MD \ (AM \ 01 + AD \ 01 \ (\overline{EC \ 02 + AM \ 02 + AD \ 02} \\
 &\quad + \overline{EC \ 03 + AM \ 03 + AD \ 03} \\
 &\quad + \dots\dots\dots \\
 &\quad + \overline{EC \ 23 + AD \ 23} \)) \\
 &\quad + MD \ (AD \ 01 + AM \ 01 \ (\overline{EC \ 00 + AM \ 00} \))) \\
 &+ \dots\dots\dots \\
 &+ P \ i \ (EC \ i + MD \ (AM \ i + AD \ i \ (\overline{EC \ i+1 + AM \ i+1 + AD \ i+1} \\
 &\quad + \overline{EC \ i+2 + AM \ i+2 + AD \ i+2} \\
 &\quad + \dots\dots\dots \\
 &\quad + \overline{EC \ 23 + AD \ 23} \)) \\
 &\quad + MD \ (AD \ i + AM \ i \ (\overline{EC \ i-1 + AM \ i-1 + AD \ i-1} \\
 &\quad + \overline{EC \ i-2 + AM \ i-2 + AD \ i-2} \\
 &\quad + \dots\dots\dots \\
 &\quad + \overline{EC \ 00 + AM \ 00} \))) \\
 &+ \dots\dots\dots \\
 &+ P \ 22 \ (EC \ 22 + MD \ (AM \ 22 + AD \ 22 \ (\overline{EC \ 23 + AD \ 23} \)) \\
 &\quad + MD \ (AD \ 22 + AM \ 22 \ (\overline{EC \ 21 + AM \ 21 + AD \ 21} \\
 &\quad + \overline{EC \ 20 + AM \ 20 + AD \ 20} \\
 &\quad + \dots\dots\dots \\
 &\quad + \overline{EC \ 00 + AM \ 00} \))) \\
 &+ P \ 23 \ (EC \ 23 + AD \ 23)
 \end{aligned}$$

On remarque que l'écriture des conditions d'arrêt en bout de course est très lourde.

Le graphe de la figure 2.11 permet de déterminer la demande de marche DM ou d'arrêt \overline{DM} suivant l'état des variables EN et CO.

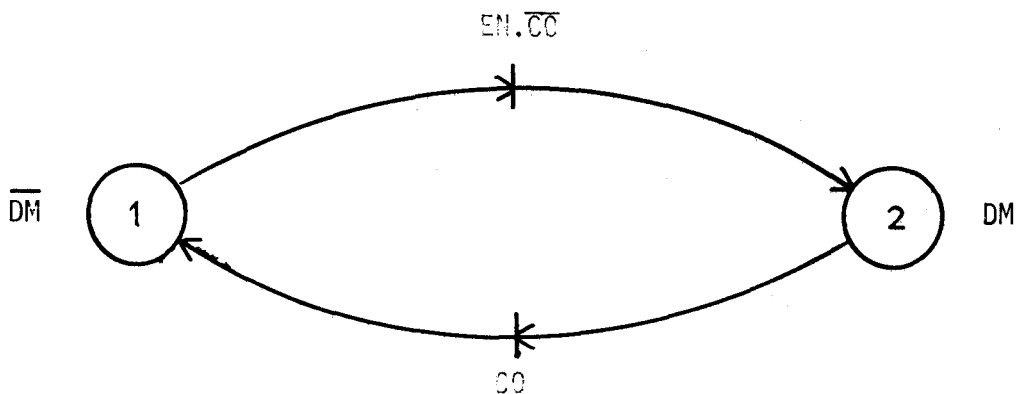


Figure 2.11

2.2. METHODE DE CODAGE

La description par réseau de Pétri a permis de préciser le comportement de l'ascenseur. Après l'écriture des graphes, il ne reste aucune ambiguïté. Nous remarquons que le graphe de détermination de la position est très long et que certains événements sont exprimés de façon compliquée. Cette lourdeur s'explique par le fait que dans les réseaux de Pétri, on prend un codage de position rudimentaire (une place par position élémentaire, donc un codage appelé "1 parmi N") et qu'on n'emploie pas d'opérateurs élaborés.

Nous allons donc reprendre cette étude du choix du sens et de la détermination des arrêts de l'ascenseur par une méthode de codage des informations sur des mots binaires et par l'utilisation d'opérateurs élaborés. (comparateur en particulier).

2.2.1. Codages utilisés

2.2.1.1. Codages des envois-cabine

Les envois-cabine peuvent apparaître de façon quelconque sur les n niveaux. On peut obtenir en principe, à un moment ou à un autre, toutes les configurations possibles d'ordres. Le nombre des ces configurations est 2^n .

En binaire, on affecte donc un mot de n bits à l'ensemble des envois-cabines. Le codage le plus indiqué est le suivant :

- le rang d'un bit correspond au niveau
- en l'absence d'envoi-cabine, les bits sont à 0 ; un envoi-cabine met à 1 le bit correspondant.

2.2.1.2. Codage des appels-montée et des appels-descente

Les appels-montée ainsi que les appels-descente des paliers sont au nombre de $n-1$. L'appel-montée n'existe pas au dernier niveau; de même l'appel-descente n'existe pas au premier niveau. On peut également obtenir toutes les configurations possibles sur les $n-1$ niveaux pour les appels-montée ou pour les appels-descente. En toute rigueur, un mot de $n-1$ bits est suffisant pour coder toutes les combinaisons des appels-montée ou toutes les combinaisons des appels-descente. Cependant, il est intéressant de prendre également un mot de n bits pour coder les appels-montée ou les appels-descente, de façon que le rang du bit corresponde également au niveau.

2.2.1.3. Codage de la position de la cabine

Pour faciliter les opérations de comparaison et de coïncidence, il est tout à fait opportun de garder également un mot de n bits pour repérer la position de la cabine. Ce mot contient un "1" parmi $n - 1$ "0" dont l'emplacement indique :

- soit la position de la cage
- soit la dernière station quittée lorsque celle ci est en mouvement et n'a pas encore atteint la station suivante.

Cette façon de coder la position est utile pour le traitement qui va suivre. Pratiquement, elle n'est pas toujours acquise immédiatement en particulier lorsque la position est repérée comme c'est le cas ici, par odométrie relative. Le contenu d'un compteur-décompteur, préaffiché au départ, permet de connaître la position de la cabine. Il faut alors convertir le contenu de ce compteur-décompteur en un code "1" parmi $n - 1$ "0".

Nous appelons m_p le mot code donnant la position de la cabine :

$$m_p = P_{n-1} P_{n-2} \dots P_j \dots P_0$$

2.2.1.4.. Somme logique des demandes d'arrêt

Appelons :

E C : le mot code des envois-cabine

A M : le mot code des appels-paliers pour monter

A D : le mot code des appels-paliers pour descendre.

Une dernière donnée utile pour notre système est un mot code m_Q constitué par la somme logique rang par rang de EC, AM et AD.

$$m_Q = Q_{n-1} Q_{n-2} \dots Q_j \dots Q_0$$

Ce mot code contient 0,1 ou plusieurs "1" parmi des "0" dont les emplacements indiquent globalement les positions d'arrêt demandés. Voir l'exemple de la figure 2.12

2.2.2. Choix du sens

Comme les comparateurs de nombres binaires existent tout faits, aussi bien en logique câblée qu'en logique programmée, nous utilisons ces opérateurs pour ne plus élaborer dans le détail des expressions telles que celles de E1 et E2.

Si l'on associe aux mots codes m_P et m_Q les nombres binaires

$$P_N = \sum_{j=0}^{n-1} P_j 2^j$$

$$\text{et } Q_N = \sum_{j=0}^{n-1} Q_j 2^j$$

la cabine doit descendre dès que

$$P_N > Q_N \quad (2.1.)$$

Si l'on associe aux mots codes m_P et m_Q les nombres binaires

$$P_R = \sum_{j=0}^{n-1} P_j 2^{n-1-j}$$

$$\text{et } Q_R = \sum_{j=0}^{n-1} Q_j 2^{n-1-j}$$

la cabine doit monter dès que

$$P_R > Q_R \quad (2.2.)$$

La détermination du sens de parcours est ainsi réalisée.

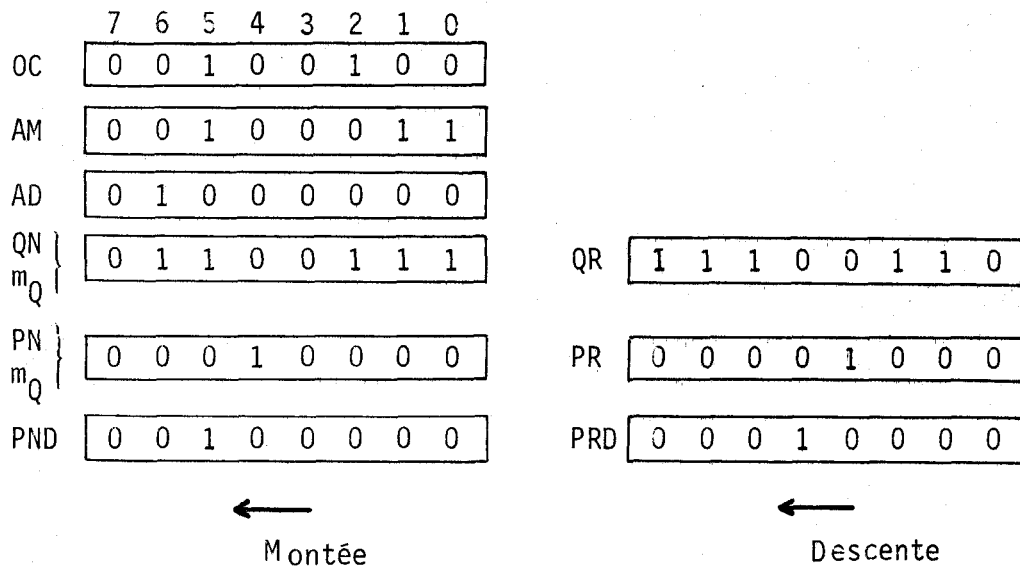


Figure 2.12 : Exemple sur 8 niveaux

Posons :

$CN = 1$ si l'égalité (2.1) est vérifiée

et $CR = 1$ si l'égalité (2.2) est vérifiée

CN et CR correspondant alors à $\overline{E2}$ et $\overline{E1}$ respectivement. Les événements du graphe de la figure 2.10 sont dans ce cas des prédicats.

Voyons la façon de résoudre le cas du "premier appel". Lorsque l'ascenseur est au repos, l'utilisateur dispose d'un laps de temps pour appuyer sur un bouton de la cage. Pour satisfaire à cette exigence particulière, il suffit de définir temporairement m_Q comme étant le mot des envois-cabine. Une fois le laps de temps écoulé, on redéfinit m_Q comme étant la somme logique des envois-cabine, des appels-montée et des appels-descente.

2.2.3. Détermination des arrêts

Les arrêts parcours sont faciles à déterminer. Il suffit de regarder s'il y a un bit, au rang correspondant à la position de la cabine,

- dans le mot des envois-cabine
- dans le mot des appels-montée, si on monte
- dans le mot des appels-descente, si on descend.

Diverses solutions sont envisageables pour déterminer les arrêts "bout de course". L'essentiel est d'examiner s'il y a encore des demandes dans le sens de la marche au delà du niveau atteint. On peut d'abord tester le mot m_Q : on regarde si ce mot ne contient que des 0 au delà du rang de la position de la cabine. Cette solution, en câblé par exemple, est facilement réalisable pour un nombre réduit de niveaux au moyen de quelques modules combinatoires qui imposent l'arrêt en cette circonstance. Elle devient lourde pour un nombre important de niveaux.

Une autre solution consiste à utiliser également l'opérateur de comparaison. Prenons le cas de la montée. Si on décale dans PN, le digit du niveau dans le sens de la montée, le nombre obtenu est plus grand que QN si on est dans le cas de l'arrêt "bout de course". Le cas de la descente se traite de la même façon.

Appelons PND le nombre obtenu à partir de PN en décalant le digit de position dans le sens de la montée. De même, soit PRD le nombre obtenu à partir de PR en décalant le digit de position dans le sens de la descente. Alors l'arrêt "bout de course" A_{BC} est donné par l'expression :

$$A_{BC} = M_0 (PND > QN) + \overline{M_0} (PRD > QR)$$

Cette deuxième solution, très simple, fait appel aux opérateurs de décalage et de comparaison. Elle est en particulier très aisément programmable.

On peut encore envisager une troisième solution valable en technique programmée. Si on est en présence d'un arrêt "bout de course", le sens doit changer au prochain départ. On peut donc, dès l'arrivée à l'étage "bout de course", changer à l'intérieur du programme, le sens de parcours, avant même de commander l'arrêt. Le changement de sens est automatique si on décale, comme on a vu précédemment, le bit de position de PN dans le sens de la marche et si on repasse dans le programme de détermination de sens. Une fois le changement de sens effectué, on a affaire à un "arrêt parcours". Cette manière de procéder paraît séduisante. Elle se réalise simplement dans la manière d'agencer le programme. On teste d'abord les "arrêts parcours" et ensuite on transforme l'arrêt "bout de course" éventuel en un "arrêt parcours". Seulement en pratique, il faut faire attention que dans la commande, l'inversion de sens et l'arrêt n'apparaissent pas en même temps.

2.3. ASCENSEURS JUMELES

D'un point de vue général, les ascenseurs en batterie ne sont pas faits à priori pour accroître la possibilité de transport. Il suffirait pour cela de prévoir une cabine plus spacieuse, mue par un moteur plus puissant. Le but principal est de minimiser les temps d'attente et de transport des usagers. Avant d'envisager les moyens d'atteindre ce but, il faut d'abord connaître la loi qui régit les déplacements des usagers. Ces lois sont très variables et dépendant non seulement de la situation de la batterie d'ascenseurs (immeuble administratif, hôtel ...) mais aussi du moment (jour, nuit....). Les mouvements d'usagers peuvent être classés typiquement de la manière suivante (figure 2.13) :

- distribution : un niveau donné reçoit les usagers qui se rendent à d'autres niveaux.
- rassemblement : les usagers proviennent de partout pour se rendre à un niveau particulier.
- déplacement : il n'y a plus de niveau préférentiel, la circulation est répartie entre des niveaux quelconques.

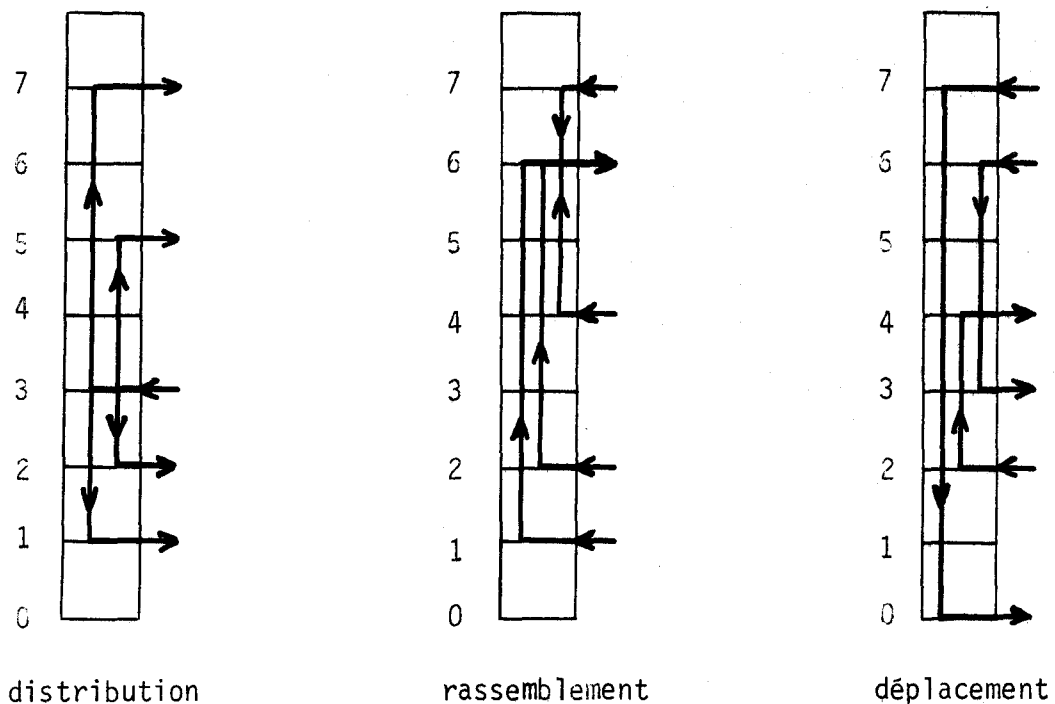


Figure 2.13

Si les lois de circulation sont connues, on peut mettre au point une solution qui satisfasse les usagers dans leur ensemble et qui obéisse également à des critères économiques (il est inutile de laisser 6 ascenseurs en service pour quelques demandes éparpillées). Des réalisations de ce genre, gérées par ordinateur, ont été conçues.

Voyons rapidement le cas plus courant de deux ascenseurs jumelés (manoeuvre "duplex"). Sur chaque palier, on dispose d'un unique bouton d'appel-montée et d'un unique bouton d'appel-descente. Supposons avoir à faire face principalement à un trafic de déplacement. Une solution consiste à adjoindre à chaque ascenseur une carte de commande "collective", munie de quelques dispositifs supplémentaires que l'on va voir. D'abord chaque ascenseur peut fonctionner seul en cas de mise au repos du voisin. Dans le fonctionnement normal simultané, on peut imaginer que leurs mouvements soient opposés : l'un monte pendant que l'autre descend. Si le trafic de déplacement était parfait, le premier ascenseur arriverait au point haut pendant que l'autre arriverait au point bas. En pratique la loi n'est pas parfaite ; on compense alors l'imperfection de la façon suivante :

Dès qu'un ascenseur fait le même mouvement que l'autre, ce deuxième ne s'arrête plus pour prendre en charge des passagers jusqu'à ce qu'il change de sens.

On ajoute donc à la carte "collective" une commande d'inhibition des arrêts.

Le fonctionnement est décrit par le réseau de Pétri de la figure 2.14.

Le mouvement des ascenseurs est indiqué par :

M1 : montée de l'ascenseur n°1
M2 : montée de l'ascenseur n°2
D1 : descente de l'ascenseur n°1
D2 : descente de l'ascenseur n°2

L'inhibition des arrêts pour prendre en charge des passagers est :

IM1 : pour l'ascenseur n°1 dans le sens de la montée
IM2 : pour l'ascenseur n°2 dans le sens de la montée
ID1 : pour l'ascenseur n°1 dans le sens de la descente
ID2 : pour l'ascenseur n°2 dans le sens de la descente.

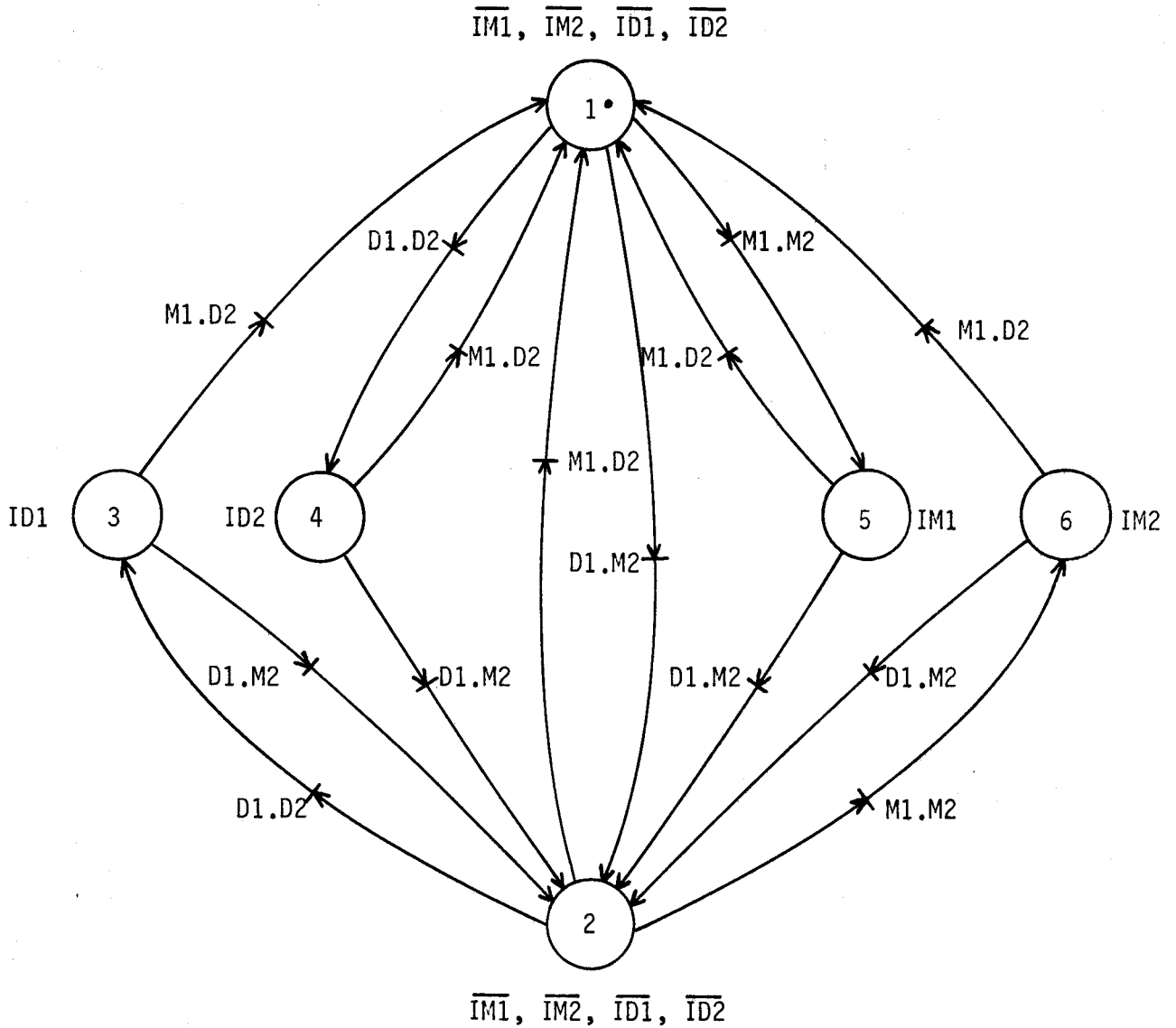


Figure 2.14

o o
o

Les réseaux de Pétri ont permis de faire l'analyse de la première partie de la commande d'un ascenseur et de mettre en forme en même temps des solutions. Les évènements d'un graphe peuvent être des prédicats. La commande des auxiliaires est étudiée également par réseau de Pétri au dernier chapitre.

CHAPITRE III

P R E M I E R E S A P P L I C A T I O N S

Nous avons vu combien il échappait de choses inexactes et fausses ; combien on en omettait de vraies.

Encyclopédie - DIDEROT

Un premier essai d'application des principes exposés dans le chapitre précédent a été effectué en logique câblée et ensuite en logique programmée. Ces deux réalisations préliminaires sont décrites dans ce chapitre. Elles permettent de tirer quelques conclusions pour mener à bien une réalisation définitive.

3.1. COMMANDE CABLEE

Les modules de base que l'on rencontre sur le marché sont de plus en plus évolués. Les circuits intégrés dont on dispose depuis une décennie offrent un choix de fonctions élaborées : mémoires, comparateurs, monostables, pour ne citer que les circuits qui nous intéressent dans notre travail. Les constructeurs de modules logiques industriels nous proposent également depuis peu des fonctions élaborées.

Nous disposons d'une maquette d'ascenseur à quatre niveaux. La détection des niveaux se fait par contacts de passage. Les boutons de commande sont ramenés sur un pupitre. Dans cette première maquette nous n'avons pas prévu la commande des auxiliaires (portes, came, lumière...).

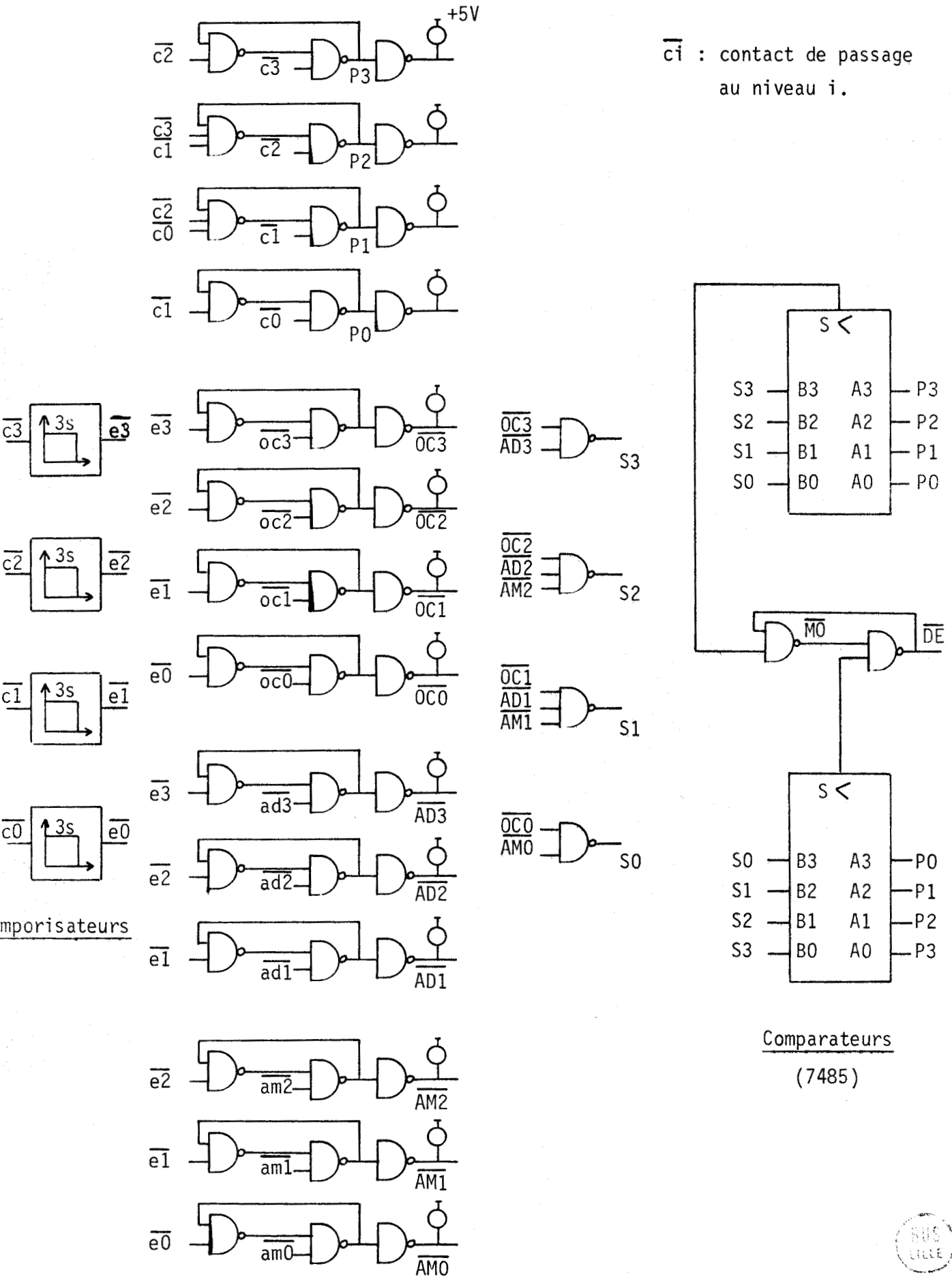
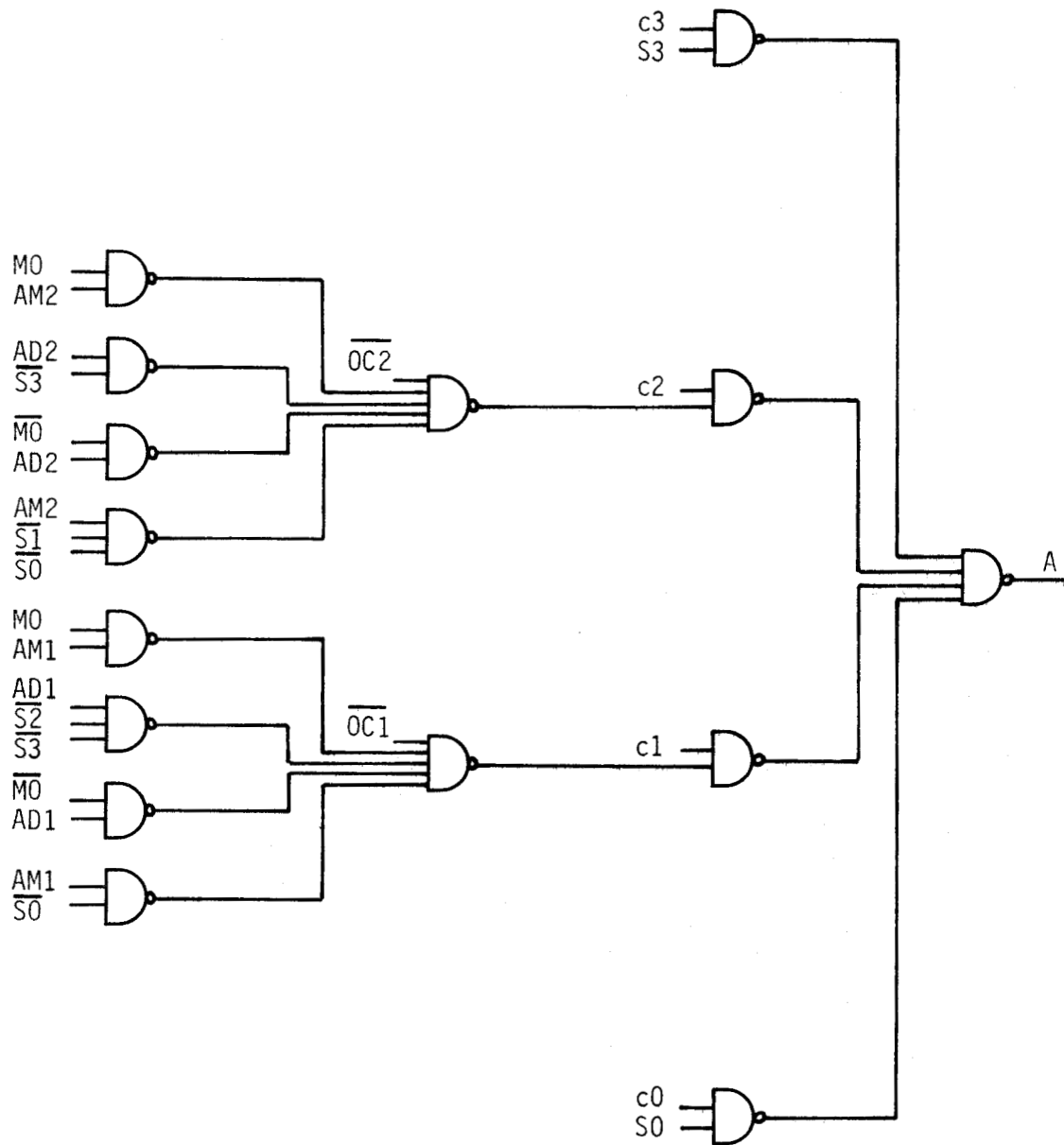


Figure 3.1.a



Circuit combinatoire d'arrêt

Figure 3.1.b



Le schéma logique est dessiné à la figure 3.1. La mise en mémoire de la position de la cabine et des commandes est réalisée par des mémoires $\overline{S} \overline{R}$. La technologie des circuits intégrés offre une sensibilité au niveau bas. Le signal actif provenant d'un bouton ou d'un capteur, doit être de niveau 0. Lorsque les contacts sont ouverts, les entrées sont maintenues au niveau haut par des résistances de $1\text{ K}\Omega$ reliées à l'alimentation 5 V. La fermeture des contacts provoque une mise à la masse des entrées. La somme logique des demandes est effectuée ici au moyen d'opérateurs \overline{ET} . Deux comparateurs effectuent les comparaisons nécessaires. Le retournement des mots PN et QN est fait par câblage. Une mémoire $\overline{S} \overline{R}$ branchée aux sorties des comparateurs donne le sens de marche. Un sous-ensemble combinatoire élabore la fonction d'arrêt.

Quatre temporisateurs ne remettent à zéro les demandes satisfaites que trois secondes après l'arrivée de la cabine. La cabine reste donc en attente trois secondes au palier avant de repartir si d'autres demandes sont enregistrées.

Ce montage d'essai fonctionne très bien sur table. La détection des niveaux aurait pu être faite également par comptage-décomptage d'impulsions. Cet ensemble est facilement extensible par tranches de 4 niveaux. Les éléments logiques utilisés ne sont pas d'un coût élevé. Seulement le montage est très sensible aux parasites. Il faudrait adjoindre une cellule d'antiparasitage à chaque entrée, et le prix de revient serait alors supérieur aux réalisations actuelles faites au moyen de composants discrets (diodes, transistors, résistances ...) beaucoup moins sensibles aux parasites.

3.2. COMMANDE PAR MICROCALCULATEUR

Pour débiter nos essais en commande programmée, nous disposons d'un microcalculateur PICOLOG, basé sur le microprocesseur INTEL 8080. Comme ce système travaille normalement sur des octets, nous avons envisagé une commande d'ascenseur pour huit niveaux.

Ce graphe a été transposé suivant l'organigramme de la figure 3.3. On constate sur ce petit exemple que le réseau de Petri est plus concis et plus facile à comprendre que l'organigramme. Le programme correspondant se trouve sous le titre "Double comparaison". PN, QN, PR, et QR sont rangés respectivement dans les registres D, B, L et C.

3.2.2. Détermination de la position de la cabine

Les signaux A et B du capteur (voir § 2.1.2.) évoluent suivant la séquence rappelée à la figure 3.4. Un compteur est incrémenté à chaque changement de A et de B suivant le sens de parcours de la séquence.

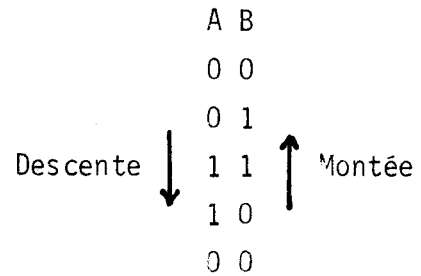


Figure 3.4.

L'organigramme de la figure 3.5. permet de déterminer la position de la cabine par rapport aux écrans.

Après l'entrée du mot AB, on examine si ce mot a changé par rapport à son ancien état noté (AB)⁻. S'il n'a pas changé le compteur reste dans le même état. Il n'est pas normal ensuite que les informations A et B changent toutes les deux : il y a défaut. Un parasite, par exemple, a troublé le fonctionnement du capteur. Disons, en passant, qu'il faut prévoir une scrutation de A et B suffisamment rapide pour éviter de perdre des états (cas des ascenseurs à grande vitesse). Sinon on pourrait aussi trouver deux changements sur A et B.

Ensuite le programme examine par une série de tests quel est l'état qui précédait l'état actuel. Suivant le résultat, on incrémente ou décrémente le compteur de position.

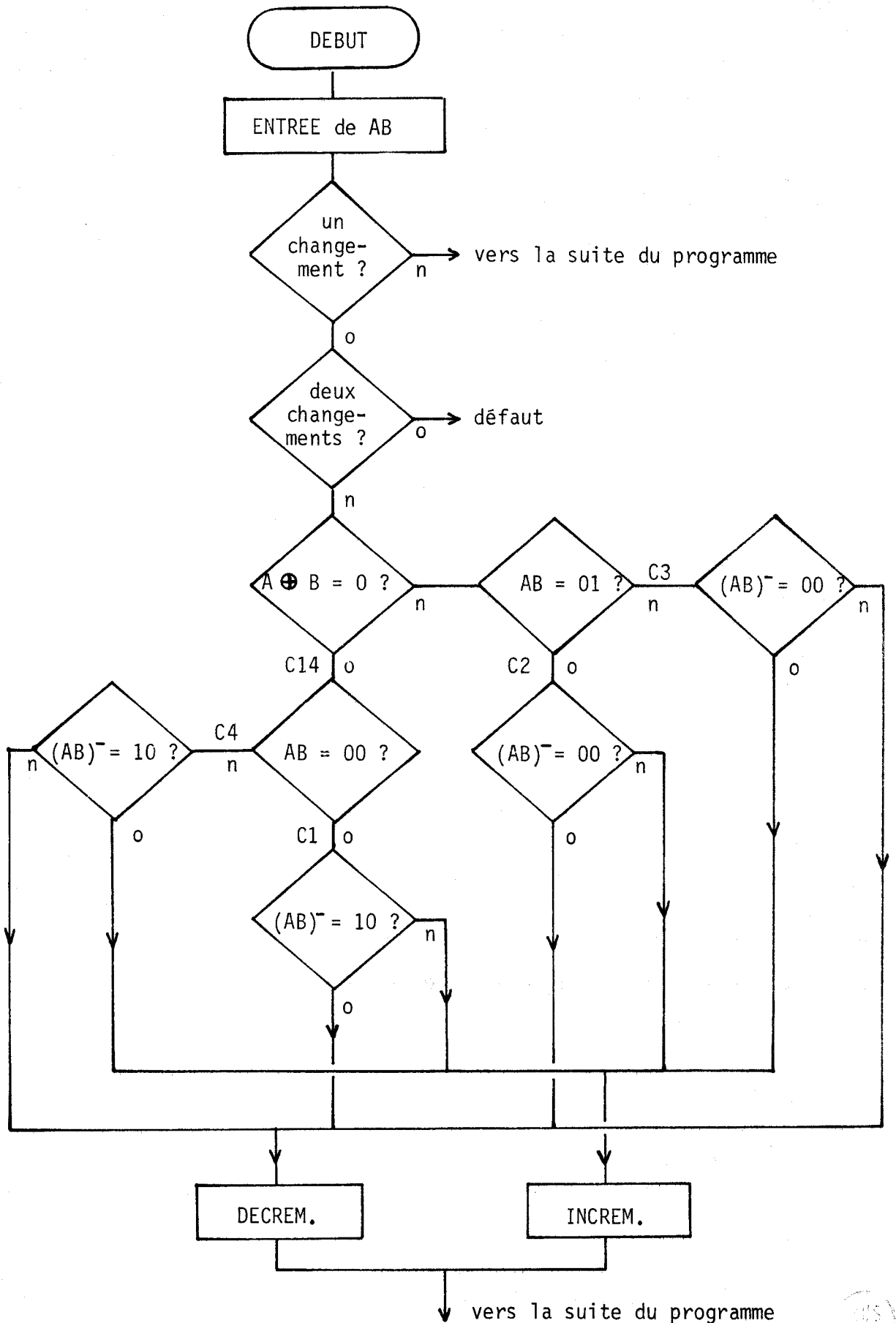


Figure 3.5.

3.2.3. Détermination des arrêts

Arrivé à un niveau, on examine s'il y a une demande d'arrêt parcours. Si c'est le cas, on commande l'arrêt. Sinon, on examine s'il y a une demande d'arrêt "bout de course". Pour cela, on décale dans PN (appelée POS1 dans le programme) le bit de position dans le sens de la marche, puis on recherche si, dans ces conditions, le sens change. Si c'est le cas, on a un arrêt "bout de course" qui est alors trouvé par la procédure normale de la détermination des arrêts parcours.

Cette application suit la troisième solution indiquée dans le chapitre précédent pour déterminer l'arrêt "bout de course".

3.2.4. Contrôles des auxiliaires

Par rapport à la commande câblée, nous avons essayé d'ajouter quelques contrôles d'auxiliaires mais nous nous sommes aperçu que la programmation de ces contrôles devenait vite difficile. Cette démarche par organigramme est pénible et n'est pas faite méthodiquement. Si on touche à un point de l'organigramme pour y ajouter un élément, aussitôt d'autres branches ne se décrivent plus correctement. Nous avons donc abandonné cette voie pour mettre au point une méthode systématique qui est exposée au chapitre suivant.

o o
o

Ces premières réalisations ont été nécessaires pour juger des difficultés de mise en application de principes étudiés sur papier, et pour se pencher sur le détail du comportement de l'ascenseur et de ses annexes. La réalisation sur Picolog a permis de nous familiariser avec les contrôles programmés et de nous rendre compte que la description de machines logiques par organigrammes n'est pas aisée.

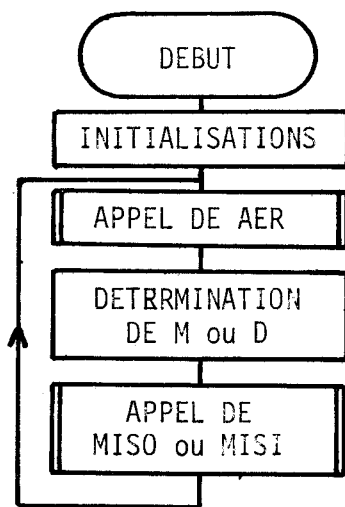
ANNEXE 3.1.

COMMANDE PROGRAMMEE D'ASCENSEUR

SUR MICROCALCULATEUR PICOLOG 80

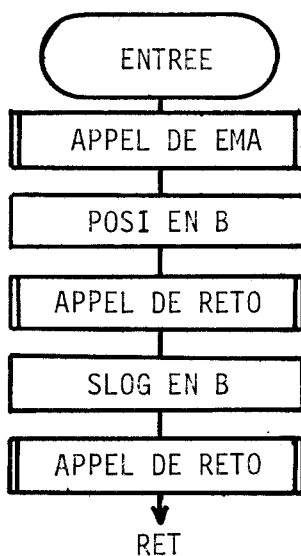
ORGANIGRAMME GENERAL

PROGRAMME PRINCIPAL



Le but du programme principal consiste seulement à déterminer le sens de parcours. Il ne s'exécute qu'à la mise sous tension et avant chaque départ. Il a l'effet d'une bascule SR et mémorise le sens de parcours (Montée M ou descente D).

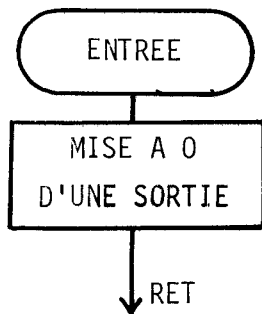
SOUS PROGRAMME AER



Ce premier sous-programme fait d'abord appel à un autre sous-programme important, le sous-programme EMA qui gère les entrées, la marche et l'arrêt. Ensuite il retourne les mots donnant la position et la somme logique des demandes. Ce qui permettra ensuite au programme principal de déterminer le sens de parcours.

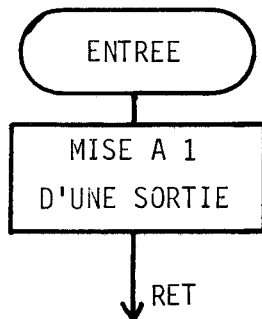


SOUS-PROGRAMME MISO



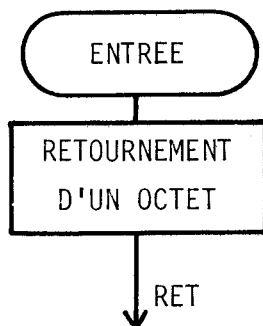
Ce sous-programme met en sortie, une variable à zéro.

SOUS-PROGRAMME MIS1



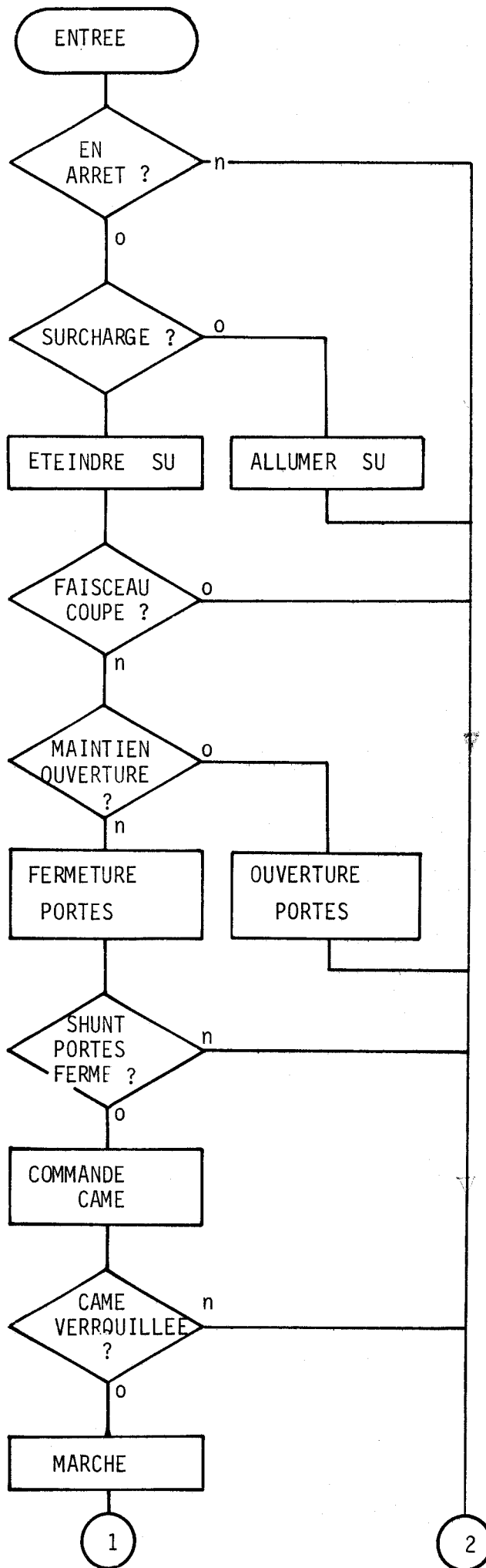
Ce sous-programme met, en sortie, une variable à un.

SOUS-PROGRAMME RETO



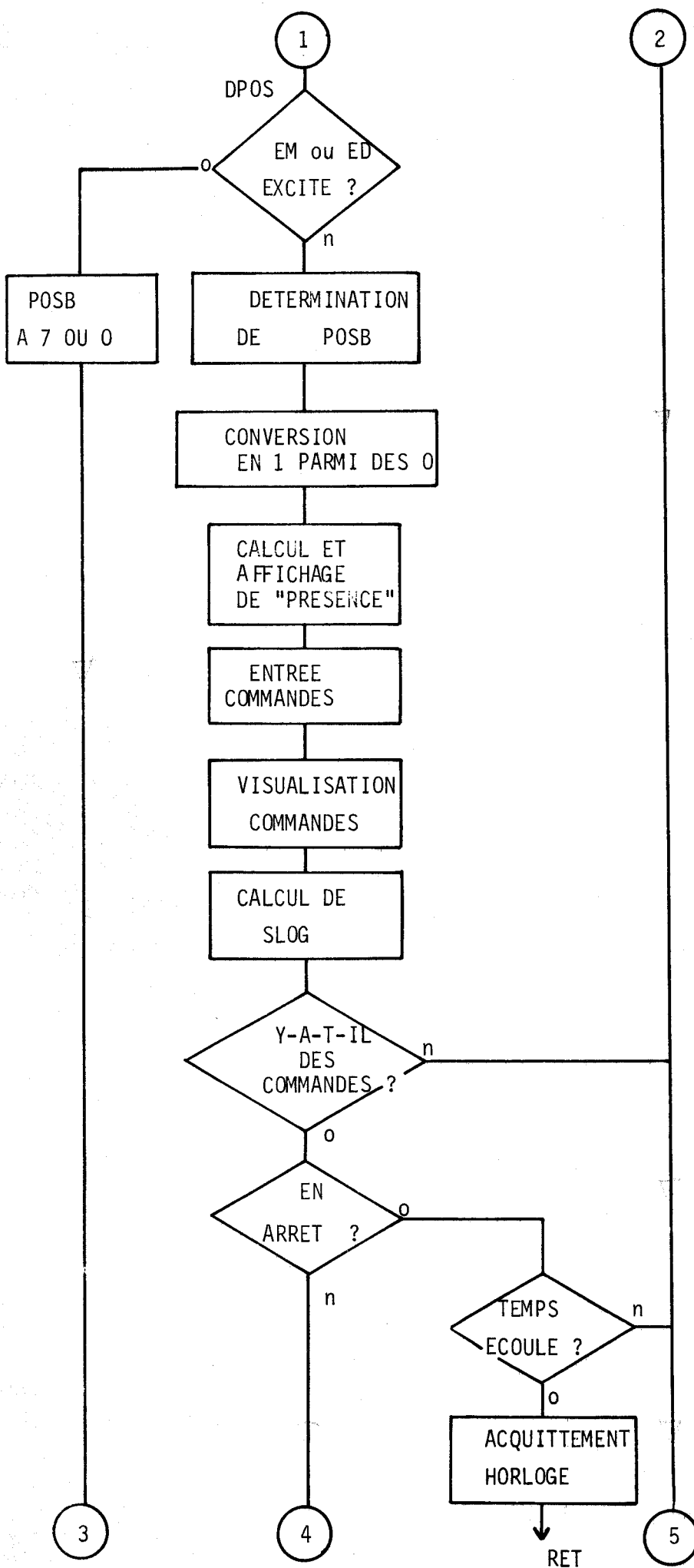
Ce sous-programme inverse la position des bits dans un octet.

SOUS PROGRAMME EMA



On commence à voir dans ce sous-programme si on est à l'arrêt ou en marche. Si on est à l'arrêt, il faut prévoir, à ce stade, la possibilité de partir. On examine alors si toutes les conditions de départ sont satisfaites. Elles ne le seront pas au premier examen. On passe alors à DPOS pour continuer à enregistrer les entrées. On reviendra ensuite dans cette chaîne après être passé (de nouveau et sans nécessité) par le programme principal.

Il faut remarquer que ces vérifications sont faites de façon sommaire : si, par exemple, la surcharge apparaît après la fermeture des portes, les passagers sont coincés.



Si un des contacts extêmes EM ou ED est excité, on affiche 0 ou 7 et on s'arrête.

La position de la cabine est déterminée par comptage d'un compteur POSB à chaque changement de l'état des capteurs A et B.

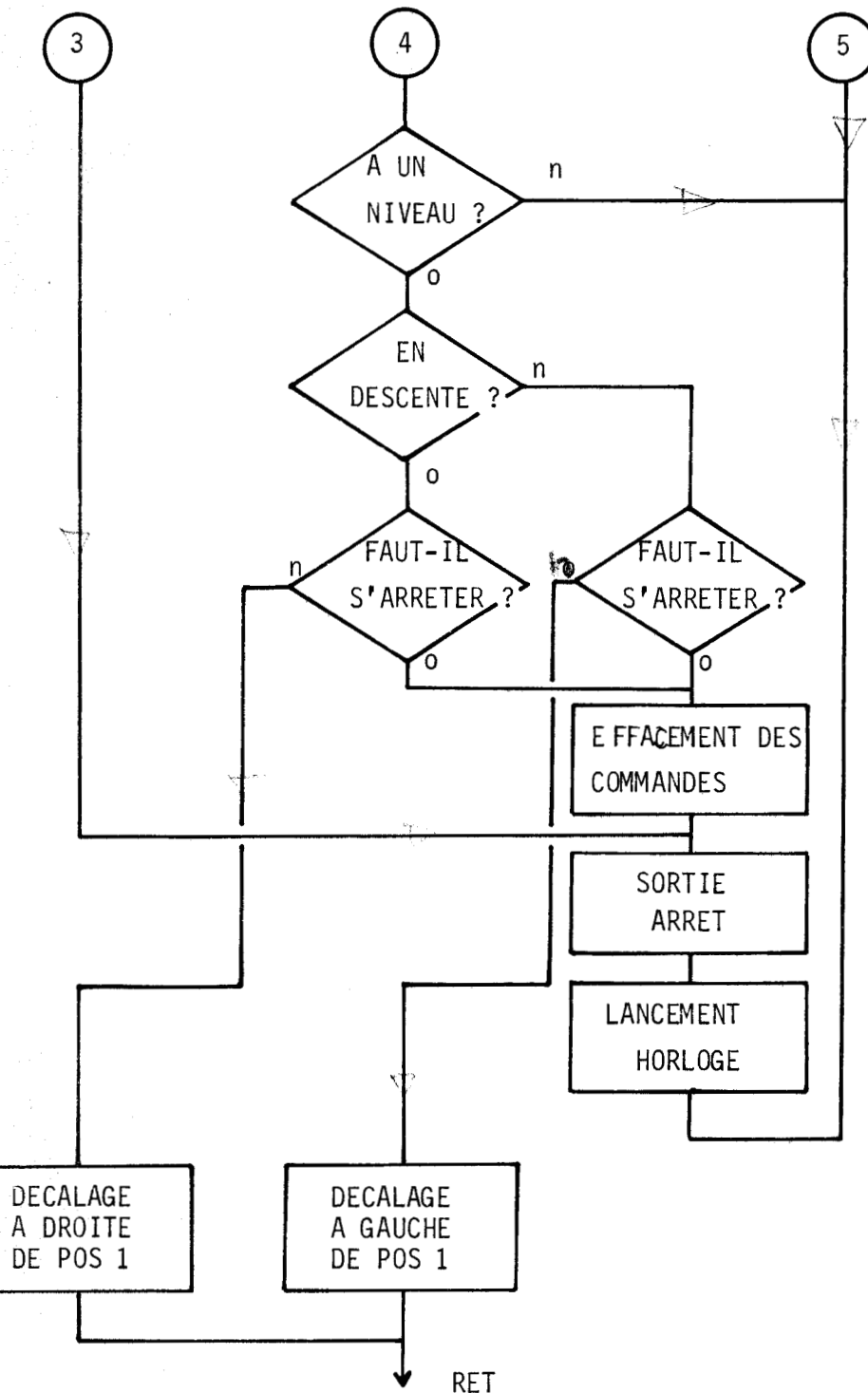
On affiche la position pour les usagers.

Les envois-cabine, les appels-montée et les appels-descente sont enregistrées par multiplexage. Toutes ces commandes sont de même visualisées par multiplexage.

On calcule la somme logique des commandes.

Si l'ascenseur n'est pas sollicité, on tourne dans la boucle ci-dessus.

Une fois arrivé à un étage, l'ascenseur doit attendre quelques instants avant d'obéir aux demandes suivantes. On teste l'état de l'horloge lancée au moment de l'arrêt.



L'arrêt n'est possible que si la cage est à hauteur d'un palier.

Est-on en descente ou en montée ?

Y-a-t-il des demandes d'arrêt au palier (ordre de la cabine ou appel de palier dans le sens de la marche) ?

Si oui, on s'arrête

On lance une temporisation.

Si non, on décale le 1 de POS1 de un rang dans le sens de la marche avant de repartir dans le sous-programme AER.

```

/PROGRAMME  A S C E N S E U R - D O U B L E C O M P A R A I S O N -
/              SUR  P I C O L O G
/              H. G H E S T E M  -  J U I N  1 9 7 7
/
/              M E T H O D E  E M P L O Y E E  :  O R G A N I G R A M M E  G E N E R A L

```

```

001 017      M1PR= '17    1
002 017      M2AM= '17    2
004 017      M4AD= '17    4
010 017      M10OC= '17   10
005 017      M5DI= '17    5
003 017      SLQG= '17    3
006 017      P0SB= '17    6
007 017      P0SI= '17    7
011 017      CAPT= '17   11

```

```

/PROGRAMME PRINCIPAL :  D O U B L E C O M P A R A I S O N

```

```

/      UTILISE : A,B,C,D,H,L
/      APPELLE : AER, MISO, MISI

```

```

000 010      * '10    0
10 000 061 040 017  LXI SP '17  40

```

```

/INITIALISATIONS

```

```

10 003 046 017      LHI 17
10 005 056 005      LLI 5
10 007 066 000      LMI 0
10 011 056 010      LLI 10
10 013 066 377      LMI 377

```

```

/D O U B L E C O M P A R A I S O N

```

```

10 015 315 067 010 TSTS,  CAL AER
10 020 170          LA B
10 021 272          CP D
10 022 171          LA C
10 023 332 032 010  JTC DESC
10 026 275          CP L
10 027 332 042 010  JTC MONT
10 032 016 375      DESC,  LCI 375
10 034 315 113 010  CAL MISO
10 037 303 015 010  JMP TSTS
10 042 016 002      MONT,  LCI 2
10 044 315 122 010  CAL MISI
10 047 315 067 010  CAL AER
10 052 171          LA C
10 053 275          CP L
10 054 170          LA B
10 055 332 042 010  JTC MONT
10 060 272          CP D
10 061 332 032 010  JTC DESC
10 064 303 042 010  JMP MONT

```

```

/S O U S - P R O G R A M M E  A E R  :  A P P E L  E M A
/              R E T O U R N .  D E  P O S I  E T  S L O G

```

```

/      UTILISE : A,B,C,D,L
/      APPELLE : EMA, RETO

```

```

10 067 315 160 010 AER,  CAL EMA
10 072 072 007 017      LDA P0SI
10 075 127            LD A      /P0SI EN D

```

10 076 107	LB A
10 077 315 132 010	CAL RETØ
10 102 151	LL C /PØSI RETØURNEE EN L
10 103 072 003 017	LDA SLØG
10 106 107	LB A /SLØG EN B
10 107 315 132 010	CAL RETØ/SLØG RETØURNEE EN C
10 112 311	RET

/SØUS-PRØGRAMME MISO : MISE A 0

/ DØNNEES EN C SØUS FØRME DE 0
/ UTILISE : A,C,L,M

10 113 056 005	MISO,	LLI 5
10 115 176		LA M
10 116 241		ND C
10 117 303 126 010		JMP MIS1+4

/SØUS-PRØGRAMME MIS1 : MISE A 1

/ DØNNEES EN C SØUS FØRME DE 1
/ UTILISE : A,C,L,M

10 122 056 005	MIS1,	LLI 5
10 124 176		LA M
10 125 261		ØR C
10 126 167		LM A
10 127 323 013		ØUT 13
10 131 311		RET

/SØUS-PRØGRAMME RETØ : RETØURNEMENT

/ UTILISE : A,B,C,H

/LE CØNTENU DU REGISTRE B ETANT
/ B7. B6. B5. B4. B3. B2. B1. B0
/ØN PLACE DANS LE REGISTRE C
/ B0. B1. B2. B3. B4. B5. B6. B7
/LE REGISTRE H EST UTILISE EN CØMPTEUR
/PUIS EST REMIS A 17

10 132 016 000	RETØ,	LCI 0
10 134 046 010		LHI 10
10 136 171		LA C
10 137 007		RLC
10 140 117		LC A
10 141 170		LA B
10 142 346 001		NDI 1
10 144 261		ØR C
10 145 117		LC A
10 146 170		LA B
10 147 017		RRC
10 150 107		LB A
10 151 045		DCH
10 152 302 136 010		JFZ RETØ+4
10 155 046 017		LHI 17
10 157 311		RET

PAUSE

/S0US-PR0GRAMME EMA : ENTREES-MARCHE-ARRET

/ UTILISE : A,B,C,D,L
 / H D0IT ETRE A 17
 / APPELLE : MISO, MISI, C0AN,
 / EMDC, CSLG

10 160 072 005 017	EMA,	LDA MSDI/EST-0N EN ARRET ?
10 163 017		RRC
10 164 322 220 010		JFC DBU /0UI ALLER AU DEBUT
10 167 346 001		NDI 1 /EST-0N EN M0NT. 0U EN DESC. ?
10 171 302 206 010		JFZ ENM
10 174 072 007 017		LDA P0S1/EN DESC.
10 177 007		RLC
10 200 062 007 017		STA P0S1
10 203 303 337 010		JMP DP0S
10 206 072 007 017	ENM,	LDA P0S1/EN M0NT.
10 211 017		RRC
10 212 062 007 017		STA P0S1
10 215 303 337 010		JMP DP0S
10 220 076 001	DBU,	LAI 1 /MISE EN R0UTE H0RL.
10 222 323 017		0UT 17
10 224 333 006		INP 6
10 226 127		LD A /ENTREE SAUVEE EN D
10 227 346 004		NDI 4
10 231 302 306 010		JFZ SRCH
10 234 016 277		LCI 277
10 236 315 113 010		CAL MISO
10 241 172		LA D /FAISCEAU C0UPE ?
10 242 346 010		NDI 10
10 244 302 337 010		JFZ DP0S
10 247 172		LA D /MAINTIEN 0UVERTURE ?
10 250 346 100		NDI 100
10 252 302 316 010		JFZ 0UVP
10 255 016 020		LCI 20 /FERM. P0RTES
10 257 315 122 010		CAL MISI
10 262 172		LA D /SHUNT P0RTES ?
10 263 346 020		NDI 20
10 265 312 337 010		JTZ DP0S
10 270 016 010	CCA1,	LCI 10 /C0MM. CAME A 1
10 272 315 122 010		CAL MISI
10 275 172		LA D /VERR0UILLAGE CAME ?
10 276 346 040		NDI 40
10 300 302 326 010		JFZ DEP
10 303 303 337 010		JMP DP0S
10 306 016 100	SRCH,	LCI 100
10 310 315 122 010		CAL MISI
10 313 303 337 010		JMP DP0S
10 316 016 357	0UVP,	LCI 357
10 320 315 113 010		CAL MISO
10 323 303 337 010		JMP DP0S
10 326 016 001	DEP,	LCI 1
10 330 315 122 010		CAL MISI
10 333 303 337 010		JMP DP0S
10 336 311		RET
	/ENTREE P0SITION	
10 337 072 011 017	DP0S,	LDA CAPT
10 342 127		LD A /(AB)N-1 EN D
10 343 056 006		LLI 6 /P0UR DEFINIR M (P0SB)

/TESTS DE EM ET ED

10 345 333 002		INP 2
10 347 117		LC A /ENTREE SAUVEE EN C
10 350 346 010		NDI 10 /EM A 1 ?
10 352 302 370 010		JFZ MA8
10 355 171		LA C
10 356 346 004		NDI 4 /ED A 1
10 360 312 375 010		JTZ COMP
10 363 066 020		LMI 20
10 365 303 311 011		JMP ARRE
10 370 066 200	MA8,	LMI 200
10 372 303 311 011		JMP ARRE
	/POSITION EN BINAIRE NAT.	
10 375 171	COMP,	LA C
10 376 346 003		NDI 3
11 000 062 011 017		STA CAPT
11 003 272		CP D /DETECTION DE 1 CHANGEMENT
11 004 312 106 011		JTZ UPA0
11 007 117		LC A /AB SAUVES EN C
11 010 057		CMA /DETECTION DE 2 CHANGEMENTS
11 011 272		CP D
11 012 312 000 076		JTZ '76 0 /RETOUR AU MONITEUR
11 015 171		LA C
11 016 247		ND A
11 017 352 033 011		JTP C14
11 022 376 001		CPI 1
11 024 172		LA D
11 025 312 051 011		JTZ C2
11 030 303 061 011		JMP C3
11 033 376 000	C14,	CPI 0
11 035 172		LA D
11 036 312 071 011		JTZ C1
11 041 376 002		CPI 2
11 043 312 101 011		JTZ INCR
11 046 303 105 011		JMP DECR
11 051 376 000	C2,	CPI 0
11 053 312 105 011		JTZ DECR
11 056 303 101 011		JMP INCR
11 061 376 000	C3,	CPI 0
11 063 312 101 011		JTZ INCR
11 066 303 105 011		JMP DECR
11 071 376 002	C1,	CPI 2
11 073 312 105 011		JTZ DECR
11 076 303 101 011		JMP INCR
11 101 064	INCR,	INM
11 102 303 106 011		JMP UPA0
11 105 065	DECR,	DCM
	PAUSE	

/CONV. DE P0SB EN 1 PARMi DES 0

11 106 176	UPAO,	LA M	
11 107 346 017		NDI 17	
11 111 302 136 011		JFZ CALPR	
11 114 176		LA M	
11 115 346 360		NDI 360	
11 117 017		RRC	
11 120 017		RRC	
11 121 017		RRC	
11 122 017		RRC	
11 123 107		LB A	
11 124 076 200		LAI 200	
11 126 005	ZZZ,	DCB	
11 127 007		RLC	
11 130 302 126 011		JFZ ZZZ	
11 133 062 007 017		STA P0SI	
	/CALCUL	PRES.	
11 136 176	CALPR,	LA M	
11 137 346 017		NDI 17	
11 141 376 010		CPI 10	
11 143 302 153 011		JFZ DIF	
11 146 076 360		LAI 360	
11 150 303 170 011		JMP CHAR	
11 153 176	DIF,	LA M	
11 154 322 164 011		JFC PLUS	
11 157 346 360		NDI 360	
11 161 303 170 011		JMP CHAR	
11 164 346 360	PLUS,	NDI 360	
11 166 306 020		ADI 20	
11 170 062 001 017	CHAR,	STA M1PR	
	/ENTREE	COMMANDES	
11 173 056 002	COMM,	LLI 2	/L VA PRENDRE SUCCESSIVEMENT /LES VALEURS 2-4-10
11 175 315 023 012	AUTR,	CAL EMDC	
11 200 175		LA L	
11 201 007		RLC	
11 202 157		LL A	
11 203 376 020		CPI 20	
11 205 302 175 011		JFZ AUTR	
	/VISUALISATION	PRES. ET COMMANDES	
11 210 056 010		LLI 10	
11 212 176	VISU,	LA M	
11 213 323 014		OUT 14	
11 215 175		LA L	
11 216 323 015		OUT 15	
11 220 017		RRC	
11 221 157		LL A	
11 222 076 000		LAI 0	
11 224 323 015		OUT 15	
11 226 322 212 011		JFC VISU	
	/PAS DE	COMMANDE ?	
11 231 315 035 012		CAL CSLG	
11 234 376 000		CPI 0	
11 236 312 337 010		JTZ DP0S	
11 241 072 005 017		LDA M5DI	/A L'ARRET ?
11 244 017		RRC	
11 245 322 357 011		JFC TH	/OUI
11 250 072 006 017		LDA P0SB	/A L'ETAGE ?
11 253 346 017		NDI 17	
11 255 302 337 010		JFZ DP0S	


```

11 260 072 007 017      LDA PØS1
11 263 127              LD A /PØS1 EN D
11 264 072 005 017      LDA M5DI
11 267 346 002          NDI 2 /EN DESC. ?
11 271 302 304 011      JFZ ARRM/VERS ARRET EN MØNTEE
11 274 056 004          ARRD, LLI 4
11 276 315 373 011      CAL CØAN
11 301 303 311 011      JMP ARRE/SØRTIE ARRET
11 304 056 002          ARRM, LLI 2
11 306 315 373 011      CAL CØAN
11 311 016 346          ARRE, LCI 346 /ARRET + RET. CAME + ØUV. PØRTE
11 313 315 113 010      CAL MISO
11 316 076 001          LAI 1 /MISE EN RØUTE HØRL.
11 320 323 017          ØUT 17
11 322 303 210 011      JMP VISU-2
11 325 063              PACØ, INX SP
11 326 063              INX SP
11 327 072 005 017      LDA M5DI
11 332 346 002          NDI 2
11 334 302 347 011      JFZ TM
11 337 072 007 017      LDA PØS1/EN DESC.
11 342 017              RRC
11 343 062 007 017      STA PØS1
11 346 311              RET
11 347 072 007 017      TM, LDA PØS1/EN MØNT.
11 352 007              RLC
11 353 062 007 017      STA PØS1
11 356 311              RET
11 357 333 006          TH, INP 6
11 361 017              RRC
11 362 322 337 010      JFC DPØS
11 365 257              XR A
11 366 323 017          ØUT 17
11 370 323 016          ØUT 16 /ACQU. HØRL.
11 372 311              RET

```

/SØUS-PRØGRAMME CØAN : COINCIDENCE ET ANNULATION

/ UTILISE : A,D,L,M

```

11 373 176          CØAN, LA M
11 374 242          ND D
11 375 312 011 012      JTZ ARRØC
12 000 256          XR M
12 001 167          LM A
12 002 056 010          LLI 10
12 004 172          LA D
12 005 057          CMA
12 006 246          ND M
12 007 167          LM A
12 010 311          RET
12 011 056 010          ARRØC, LLI 10
12 013 176          LA M
12 014 242          ND D
12 015 312 325 011      JTZ PACØ
12 020 256          XR M
12 021 167          LM A
12 022 311          RET

```

/SØUS-PRØGRAMME EMDC : ENTREE DES AM, AD ET ØC

/ UTILISE : A,L,M

12 023 175	EMDC,	LA L
12 024 057		CMA
12 025 323 014		ØUT 14
12 027 333 007		INP 7
12 031 057		CMA
12 032 266		ØR M
12 033 167		LM A
12 034 311		RET

/SØUS-PROGRAMME CSLG : CØNSTITUTION DE SLØG

/ UTILISE : A,B,H,L,M

12 035 056 002	CSLG,	LLI 2
12 037 006 000		LBI 0
12 041 176	SUIT,	LA M
12 042 260		ØR B
12 043 107		LB A
12 044 175		LA L
12 045 007		RLC
12 046 157		LL A
12 047 376 020		CPI 20
12 051 302 041 012		JFZ SUIT
12 054 170		LA B
12 055 062 003 017		STA SLØG
12 060 311		RET

\$

CHAPITRE IV

C O N T R O L E P R O G R A M M É

Au sujet des anciens francs et des nouveaux francs

-Et à votre télévision, qu'est-ce qu'ils disent ?

-La aussi, ça dépend. Ils parlent plutôt en centimes, surtout pour les hold-up. A la campagne aussi. Mais le gouvernement, lui, c'est en nouveaux francs, en francs, quoi, si tu veux, surtout pour les impôts...

Finalement, tu vois, tout le monde a le choix.

Michel GABRIELLI

Nous avons vu que l'emploi de l'organigramme n'est pas facile à mettre en oeuvre pour résoudre les problèmes de contrôle d'auxiliaires. Ce contrôle d'auxiliaires entrant dans le cadre plus général des machines logiques industrielles, il convient dans ce chapitre de définir les moyens et les méthodes qui en facilitent la synthèse.

4.1. LES AUTOMATES PROGRAMMABLES INDUSTRIELS

Un automate programmable peut commander un ascenseur à blocage et même un ascenseur à mémoire. Les entrées sont en nombre suffisant et sont protégées de l'influence des parasites. L'élaboration du programme n'est pas immédiate mais paraît possible. Seulement le prix de revient de l'installation serait prohibitif.

Un automate est un système évolué d'utilisation commode dans la commande de machines logiques. Cette solution est retenue pour des réalisations en peu d'exemplaires. Les commandes d'ascenseurs, par contre, se fabriquent par

milliers d'exemplaires. On doit donc utiliser des composants de base, de prix peu élevé, quitte à passer au départ beaucoup de temps pour l'étude et la mise au point.

On sait, d'autre part, qu'il existe plusieurs catégories de commandes d'ascenseur, qu'on appelle manoeuvres. Les constructeurs sont en outre sollicités pour des cas d'espèces : par exemple, le chef de service d'un hôpital a priorité dans l'utilisation de l'ascenseur; on est donc en contradiction avec l'optique de fabrication "grande série" citée plus haut. Il est nécessaire de trouver une solution intermédiaire entre l'appareil souple mais cher et la carte figée à bas prix.

4.2. LE MICROCALCULATEUR

4.2.1. Solutions diverses

Le microcalculateur traite normalement des mots, en l'occurrence des octets pour les microcalculateurs que nous utilisons. Dans le contrôle courant, chaque entrée ou sortie n'est constituée que d'une seule variable. Par exemple, l'appui d'un bouton B met en marche une lampe L. On peut donc affecter l'un des bits d'un octet M1 à B et un bit d'un autre octet M2 à L. Pour allumer la lampe, il faut donc entrer le mot M1 puis rechercher le bit correspondant à B et voir s'il est à 1. Ensuite, dans M2, le bit correspondant à L est mis à 1 sans toucher aux autres. Cette façon de faire devient lourde lorsque le nombre de variables d'entrée et de sortie est grand. On simplifie quelquefois le travail en affectant un octet entier à une entrée ou à une sortie. Alors les possibilités de la mémoire sont sous-utilisées.

Si le problème à traiter ne comporte que du traitement logique sur des bits, la solution est d'utiliser le processeur à un bit. Ce processeur fait les opérations logiques courantes sur un bit.

Il peut travailler avec de nombreuses entrées et de nombreuses sorties grâce à un multiplexeur d'entrée et à un démultiplexeur de sortie (figure 1.1.).

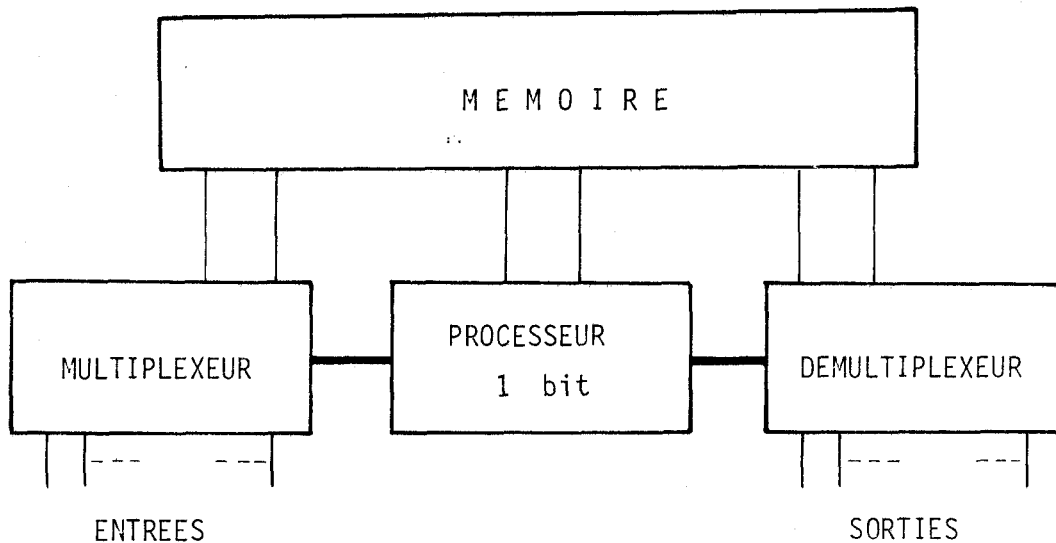


Figure 4.1.

4.2.2. Solution adoptée

L'énoncé de la solution que nous avons adoptée s'exprime facilement :

Nous voulons utiliser le microcalculateur avec un langage évolué qui permette la programmation directe des réseaux de Pétri, mis sous la forme de graphes d'état, tout en gardant la possibilité d'utiliser également les ressources de calculateur en langage assembleur.

Cette solution comporte deux particularités. Elle laisse d'abord la possibilité de travailler en assembleur. Il est utile par exemple de pouvoir évaluer une expression combinatoire, de faire du comptage, de calculer des prédicats, de créer des temporisations. Cette idée a aussi été retenue par certains constructeurs. Ainsi Hewlett-Packard propose des calculateurs en langage BASIC avec accès possible aux instructions de base de la machine.

Nous supposons ensuite que les spécifications de la machine à commander sont décrites par un ensemble de graphes d'état. Ces graphes peuvent être dépendants les uns des autres. Une variable d'évènement de l'un peut être une variable d'action d'un autre. Ce contrôle effectué à partir des graphes d'état a été appelé contrôle "fractionné" par opposition au

contrôle "global" où on part d'une représentation ne comportant qu'un seul graphe (15).

4.3. AUTOMATE PROGRAMMABLE A PARTIR DE GRAPHS D'ETAT

4.3.1. Microcalculateur

La solution retenue a été étudiée sur un calculateur MOSTEK 88 et mise en oeuvre sur une carte bâtie avec le microprocesseur MOSTEK 88. Cette carte, construite par AUTINOR a été détournée de son application première qui est la commande d'ascenseur, pour devenir le support d'un automate universel.

4.3.2. Description de la commande

Si la représentation d'automatismes séquentiels aboutit à des graphes comportant des actions simultanées, il faut décomposer ces graphes en graphes d'état. Il existe plusieurs procédures de décomposition.

La première, tout à fait générale, s'appuie sur les travaux de MM TOULOTTE ET PARSY (14), (9). Ces auteurs ont montré qu'un graphe quelconque peut être décomposé en sous-réseaux, chacun des sous-réseaux possédant un marqueur au plus. Certains sous-réseaux contiennent des transitions qui ne proviennent pas de places ou qui n'aboutissent pas à des places. Il suffit de relier ces transitions par des places d'attente pour obtenir des graphes d'état.

Une deuxième procédure permet d'éliminer des branches parallèles. L'ensemble des branches parallèles est remplacé par une macro-place. Le graphe comportant cette place devient un graphe d'état maître. Chacune des branches parallèles est alors décrite par un graphe d'état secondaire. Dans le graphe d'état maître, une action de lancement des graphes secondaires est écrite au niveau de la transition qui précède la macro-place. La fin de parcours des graphes secondaires est signalée par des sémaphores dont le produit logique est l'évènement permettant de démarquer la macro-place.

4.3.3. Exemple

Pour illustrer nos propos, nous prenons un exemple en dehors de l'ascenseur pour montrer l'universalité des applications.

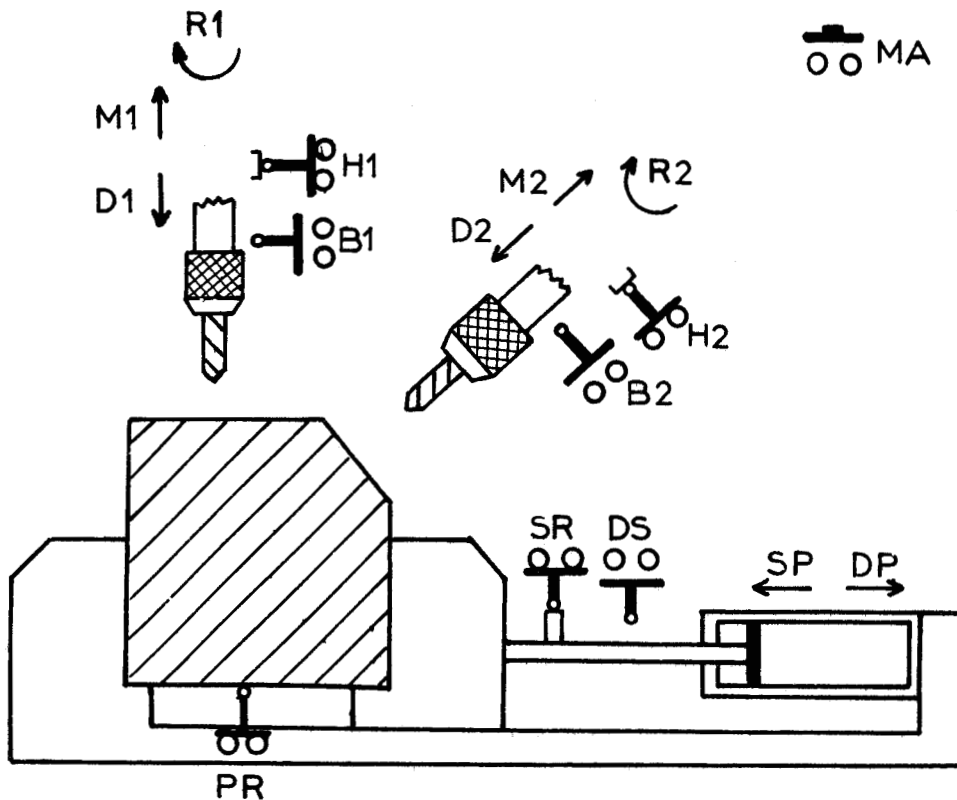


Figure 4.2.

Soit l'ensemble de perçage de la figure 4.2.

- le serrage ne peut s'effectuer que si une pièce est présente dans l'étau ($PR = 1$) et que si l'ordre de marche est donné ($MA = 1$).
- dès que le serrage de la pièce est détecté ($SR = 1$), les deux unités de perçage sont mises en service.
- le desserrage n'est autorisé que lorsque les deux perceuses ont regagné leur position de repos ($H1.H2 = 1$).
- le cycle recommence après chargement d'une nouvelle pièce.

Le fonctionnement de cette machine est donné par le réseau unique de la figure 4.3a. Celle-ci comprend d'abord une branche unique décrivant le serrage. Cette opération étant terminée (événement SR = 1), les deux opérations de perçage, indépendantes l'une de l'autre, sont décrites par deux branches parallèles et sont suivies par deux marqueurs qui peuvent progresser à des rythmes différents. Les deux perceuses doivent être en position haute pour permettre le desserrage (place 9).

La figure 4.3b correspond à la décomposition suivant la première procédure signalée plus haut. La place 10 est la place d'attente. En principe cette place est quittée si la place 2 est marquée et si l'évènement SR est présent. Il faut voir cependant que le traitement des réseaux est fait séquentiellement par l'automate. Si le premier réseau de la figure 4.3b est traité avant le deuxième, le marqueur de la place 2 avance d'abord en place 3 lors de l'arrivée de l'évènement SR. On peut dire que "place 2 marquée et SR présent" correspond au marquage de la place 3. On fait donc apparaître une variable intermédiaire RE, action correspondant au marquage de la place 3 et évènement pour quitter le place 10. De même l'évènement RF correspond au marquage de la place 10 et permet d'aboutir à la place 9.

La figure 4.3c décrit exactement les mêmes opérations en séparant les fonctions :

- serrage et desserrage de l'étau
- commande de la perceuse 1
- commande de la perceuse 2

Elle correspond à la décomposition du graphe d'origine suivant la deuxième procédure. Quelques nouvelles variables intermédiaires, apparaissent également dans ce cas qui relient les réseaux entre eux. Le réseau-maître contient la macro-place 3. L'action de lancement des deux déroulements simultanés est RD tandis que F1 et F2 sont les sémaphores.

Quelle que soit la méthode de décomposition utilisée, il faut remarquer que les variables intermédiaires sont remises à zéro par le réseau utilisateur de ces variables.

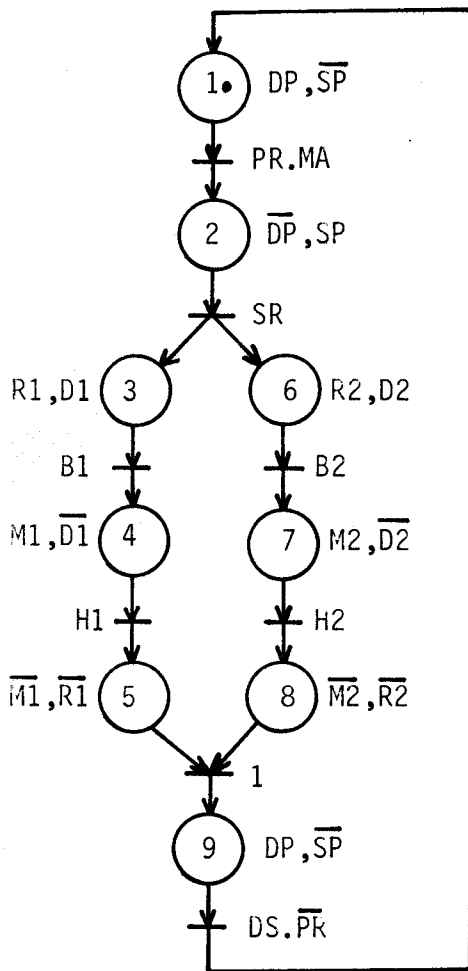


Figure 4.3a

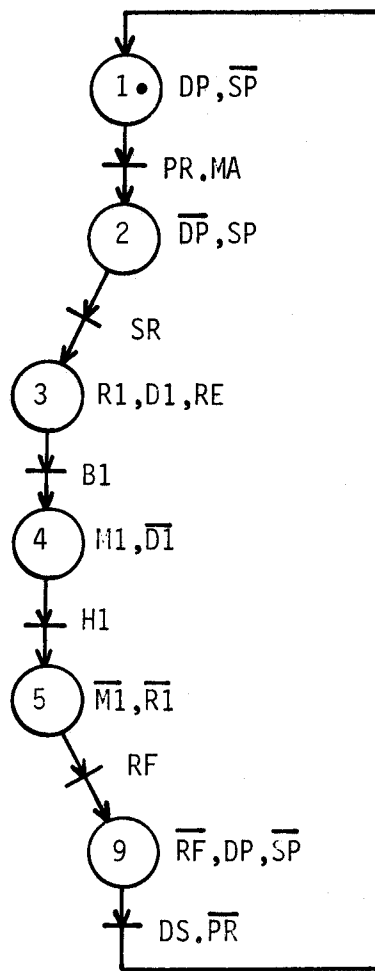


Figure 4.3b

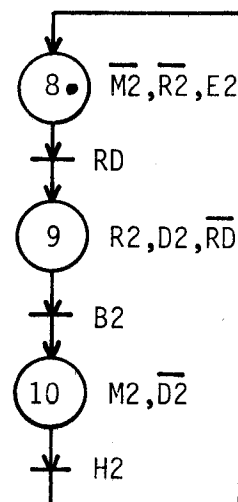
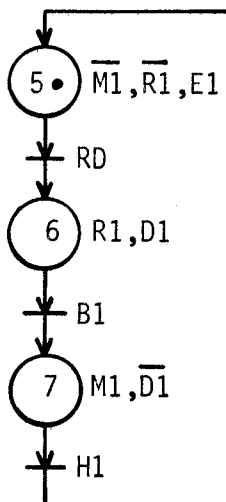
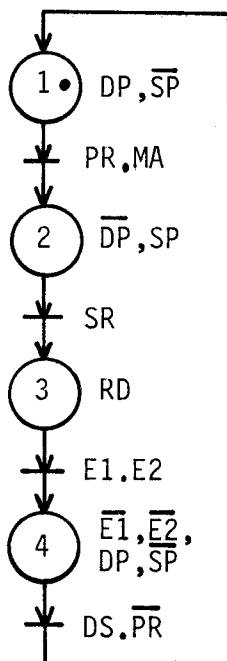
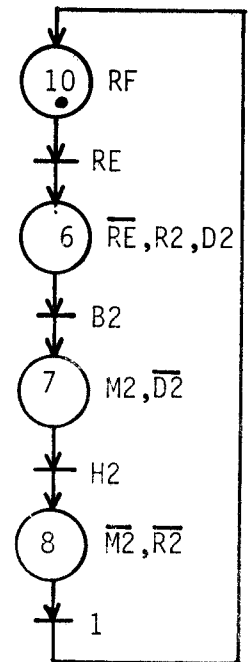


Figure 4.3c

4.3.4. Suivi des réseaux de Pétri

4.3.4.1. Table de structure

Les réseaux de Pétri indiquent le fonctionnement de la machine commandée. Le calculateur de commande suit donc l'évolution des marqueurs sur les réseaux. Ces réseaux sont mémorisés dans une table appelée table "Structure" sous la forme d'une suite de pas. Un pas est constitué d'une place et de ses transitions déférentes avec indication des places successeurs. Pour chacun des pas, on met donc successivement (figure 4.4.)

- l'adresse de l'action à effectuer au franchissement d'une transition afférente à la place concernée
- le nombre de transitions déférentes
- et à la suite, pour chaque transition déférente :
 - * l'adresse de l'évènement associé
 - * l'incrément à ajouter à l'adresse présente de la table, pour aboutir à la place suivante dans le cas où l'évènement est vrai.

PAS 1

PAS 2

Adresse de l'action
Nombre de transitions
Adresse du premier évènement
Incrément pour le pas suivant
Adresse du second évènement
Incrément pour le pas suivant
⋮
Adresse du dernier évènement
Incrément pour le pas suivant
⋮
⋮

Figure 4.4. : Table de structure

4.3.4.2. Table des évènements

Un évènement se présente toujours comme un produit logique de variables élémentaires, affirmées ou complémentées. Ces variables peuvent être extérieures (variables d'entrée du système) ou intermédiaires (fin de temporisation ou variable d'action d'un autre réseau, par exemple). Une table "évènements" contient la liste de tous les évènements de l'ensemble des réseaux.

4.3.4.3. Table des actions

Une action est un ensemble de variables élémentaires, extérieures (variables de sortie), ou intermédiaires qui doivent être mises à 0 ou à 1 suivant le cas. Une table "actions" contient la liste de toutes les actions de l'ensemble des réseaux.

4.3.4.4. Table des variables

Enfin, le détail des variables d'entrée, de sortie et intermédiaires est contenu dans la table des variables.

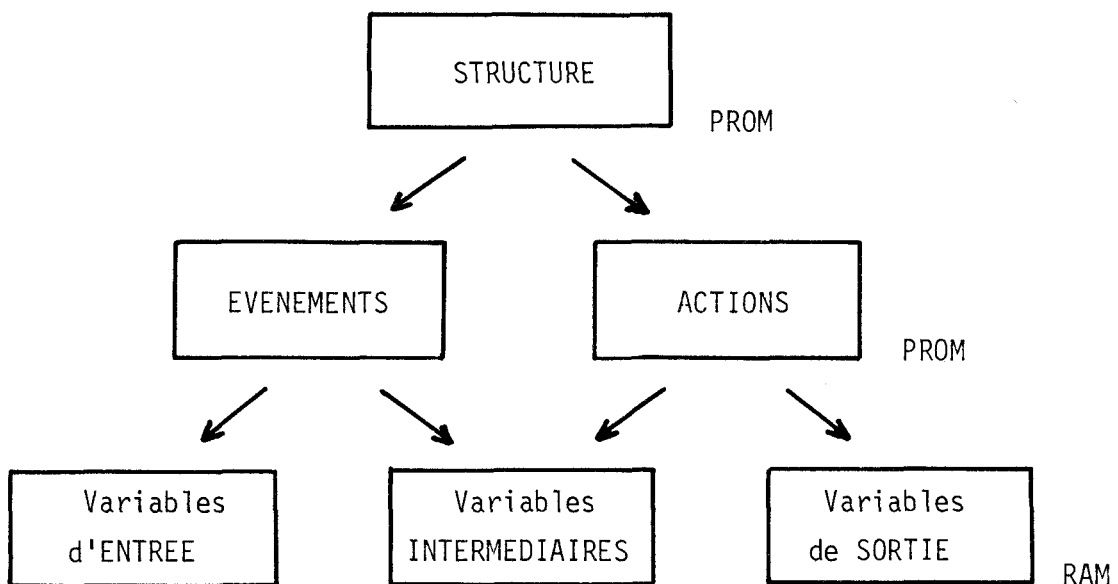


Figure 4.5.

Nous avons donc utilisé un adressage indirect à deux niveaux (figure 4.5.). Dans la table de structure, on indique les adresses des événements et des actions. Dans les tables des événements et des actions, on indique l'adresse de chacune des variables. Ce procédé d'adressage indirect permet une grande souplesse d'utilisation. Les pas peuvent être placés dans un ordre quelconque dans la table de structure. Les événements, les actions et les variables élémentaires peuvent être placés également dans un ordre quelconque dans les tables. Les modifications éventuelles consistent en des changements d'adresse. Les ajouts peuvent être faits en fin de tables.

Un programme "Aide", à l'étude, permettra de remplir les tables de structure, d'événements et d'actions directement à partir des réseaux, en mode conversationnel, sur un périphérique écran-clavier.

4.3.4.5. Sous-programme "Evolution"

Le suivi de chacun des graphes est effectué par un même sous-programme que nous avons appelé "Evolution". Celui-ci examine si l'un ou l'autre des événements associés aux différentes transitions validées, est vrai. Il fixe les sorties à chaque modification du marquage. Ce sous-programme a donc accès aux tables définies précédemment.

4.3.5. Organisation générale

Outre le suivi des réseaux de Pétri, le microcalculateur doit effectuer les entrées et sorties des variables élémentaires ainsi que des opérations diverses (évaluation d'expressions combinatoires, temporisations etc). Entrées, sorties et opérations diverses sont programmées en assembleur.

Si un seul microcalculateur exécutait toutes les opérations, il le ferait séquentiellement selon le schéma de la figure 4.6, dans lequel on a inclus à chaque cycle les opérations de sortie et d'entrée des variables extérieures. On imagine qu'un seul microcalculateur ne parvienne pas à traiter un problème complexe, ou bien qu'il serait trop lent. On peut alors séparer

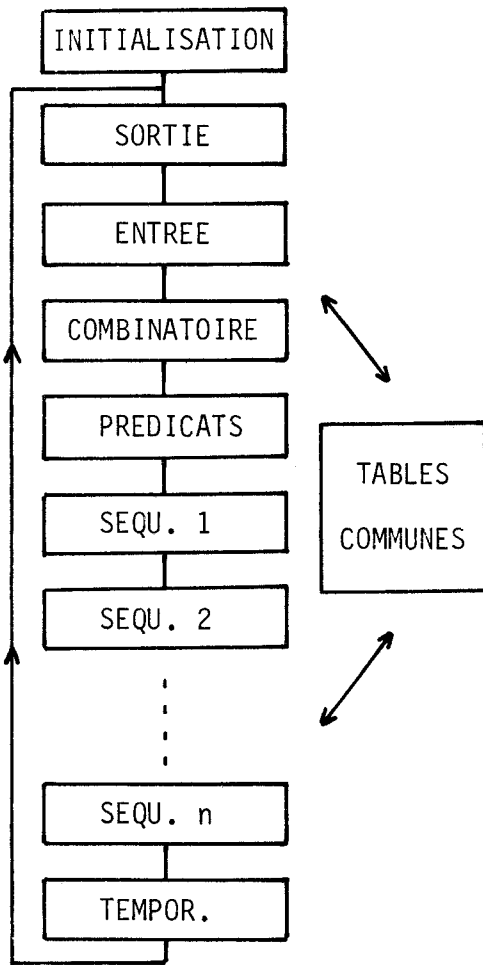


Figure 4.6

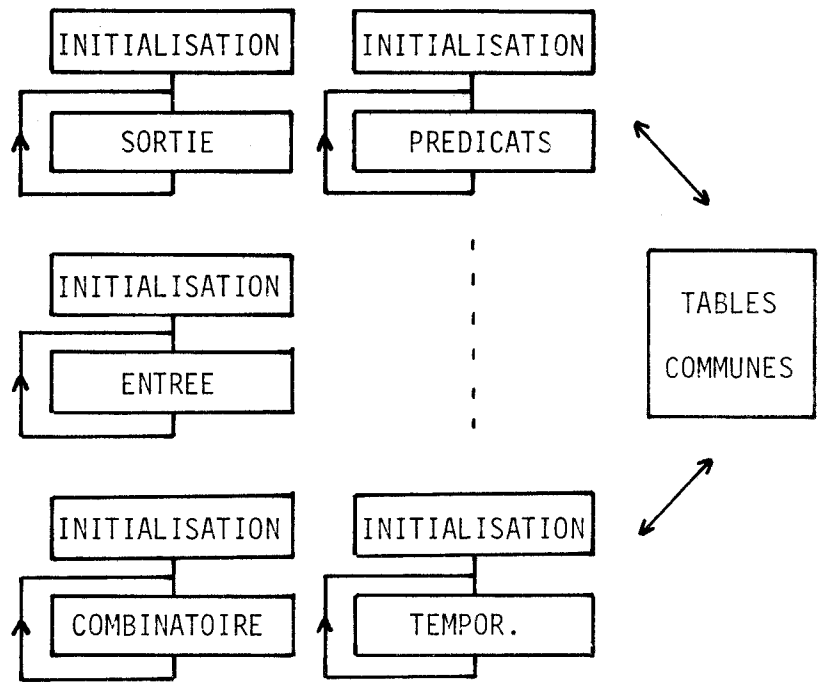


Figure 4.7

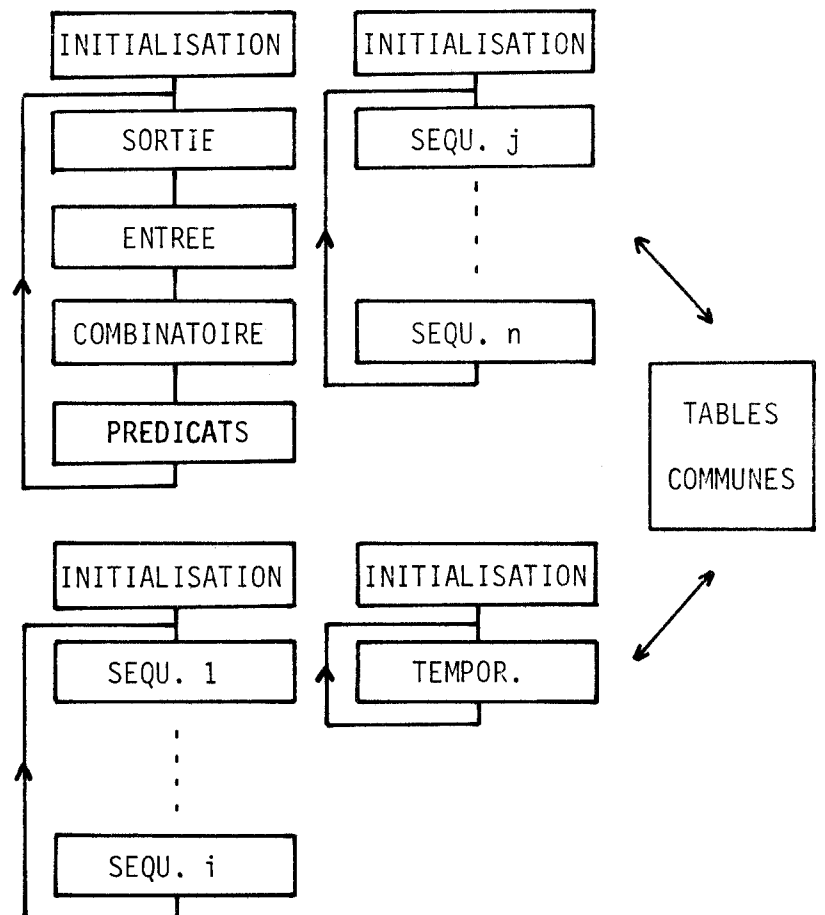


Figure 4.8



toutes les opérations qui seront exécutées chacune par un microcalculateur (figure 4.7). Cette solution est une vue de l'esprit. En pratique, on adoptera plutôt une solution intermédiaire, où, suivant la nature du problème, on regroupera certaines opérations (figure 4.8). Quelle que soit la solution adoptée, chaque sous-système a accès à des tables communes qui contiennent les variables d'entrée et de sortie ainsi que les variables intermédiaires qui n'ont aucun intérêt pour le monde extérieur. Le contrôle des opérations à effectuer par un même microcalculateur est géré par un programme que nous avons appelé "Superviseur", écrit également en assembleur.

En conclusion, le programme "Aide" permet au départ d'entrer en mémoire les réseaux de Pétri. Le programme "Superviseur" agence les modules, traite les fonctions annexes et fait appel au sous-programme "Evolution" pour suivre l'évolution des marqueurs des réseaux de Pétri.

4.3.6. Mise en oeuvre

4.3.6.1. Les tables

Les variables élémentaires sont rangées, suivant leur nature, dans trois buffers du "bloc-notes" du MOSTEK F8. On peut ainsi placer 64 variables d'entrée, 64 variables de sortie et 64 variables intermédiaires.

Dans une réalisation définitive, les événements sont placés en mémoire morte, à la suite, de la manière suivante :

- la 1ère mémoire contient le nombre de variables ;
- la 2ème mémoire contient l'adresse et l'état de la 1ère variable ;
- la 3ème mémoire contient l'adresse et l'état de la 2ème variable ;
- etc

L'adresse et l'état de chacune des variables sont codés de la façon indiquée à la figure 4.9.

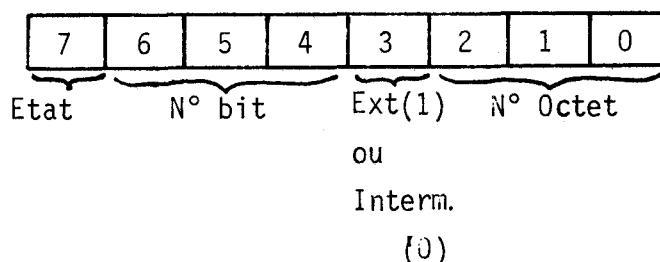


Figure 4.9.

La variable peut-être trouvée dans le buffer des variables d'entrée (variables extérieures) ou dans le buffer des variables intermédiaires. Le bit 3 permet de choisir l'un ou l'autre des buffers :

- buffer des variables extérieures si le bit est à 1,
- buffer des variables intermédiaires si le bit est à 0.

L'adresse de la variable, dans le buffer, est fournie par les bits 2,1 et 0 (N° de l'octet) et 6,5 et 4 (N° du bit de l'octet).

Enfin le bit 7 donne l'état dans lequel la variable doit être :

- état affirmé, si le bit est à 1,
- état nié, si le bit est à 0.

Les actions sont placées de la même manière en mémoire morte.

Enfin, la table de structure est construite également en mémoire morte de la façon expliquée, au § 4.3.4.1. Précisons que :

- les adresses des actions et événements sont indiquées en clair sur deux octets
- l'incrément est noté en "complément à 2" en décomptant le saut à effectuer à partir de la mémoire suivante.

Les figures 4.10 à 13 donnent ces différentes tables pour l'exemple décrit plus haut.

4.3.6.2 Les programmes

Le listing du sous-programme "Evolution" cœur de l'automate, se trouve, avec commentaires à la fin du chapitre.

Dans la cas d'utilisation d'un seul microcalculateur, le programme "Superviseur", après l'initialisation tourne dans une boucle.

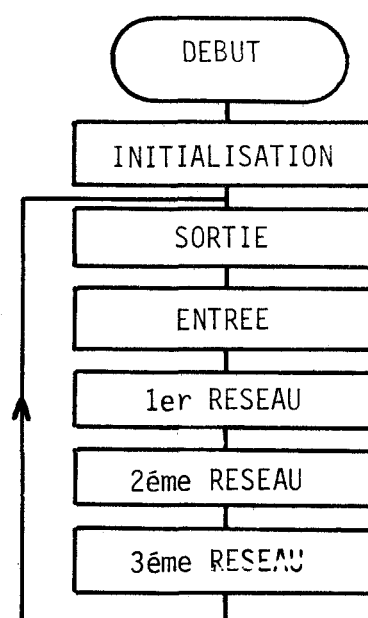


Figure 4.14.

N° des places	Codes	Adresses
①	4 E 2 0 0 1 4 E 0 0 0 0	4 E 5 0 2 3 5
②	4 E 2 3 0 1 4 E 0 3 0 0	6 8 9 B
③	4 E 2 6 0 1 4 E 0 5 0 0	C E F 6 1
④	4 E 3 D 0 1 4 E 0 8 E 8	2 4 5 7
⑤	4 E 2 8 0 1 4 E 0 B 0 0	8 A B D
⑥	4 E 2 C 0 1 4 E 0 D 0 0	E 7 0 1 3
⑦	4 E 2 F 0 1 4 E 0 F E E	4 6 7 9
⑧	4 E 3 2 0 1 4 E 0 B 0 0	A C D F
⑨	4 E 3 6 0 1 4 E 1 1 0 0	8 0 2 3 5
⑩	4 E 3 A 0 1 4 E 1 3 E E	6 8 9 B

Figure 4.10 Table de structure



Evénements	Codes	Adresses
PR,MA	0 2	4 E 0 0
	9 8	1
	8 8	2
SR	0 1	3
	A 8	4
E1.E2	0 2	5
	9 0	6
	A 0	7
DS. \overline{PR}	0 2	8
	F 8	9
	1 8	A
RD	0 1	B
	8 0	C
B1	0 1	D
	B 8	E
H1	0 1	F
	C 8	1 0
B2	0 1	1
	D 8	2
H2	0 1	3
	E 8	4

Figure 4.11. Table des évènements

Actions	Codes	Adresses
$\overline{DP}, \overline{SP}$	0 2	4 E 2 0
	8 8	1
	1 8	2
$\overline{SP}, \overline{DP}$	0 2	3
	9 8	4
	0 8	5
RD	0 1	6
	8 0	7
	0 3	8
E1, $\overline{M1}$, \overline{RT}	9 0	9
	2 8	A
	6 8	B
R1,D1	0 2	C
	E 8	D
	B 0	E
M1, \overline{DT}	0 2	F
	A 8	3 0
	3 8	1
E2, $\overline{M2}$, $\overline{R2}$	0 3	2
	A 0	3
	4 8	4
\overline{RD} ,R2,D2	7 8	5
	0 3	6
	0 0	7
M2, $\overline{D2}$	F 8	8
	D 8	9
	0 2	A
E1, \overline{EZ} ,DP, \overline{SP}	C 8	B
	5 8	C
	0 4	D
	1 0	E
	2 0	F
	8 8	4 0
	1 8	1

Figure 4.12 Table des actions



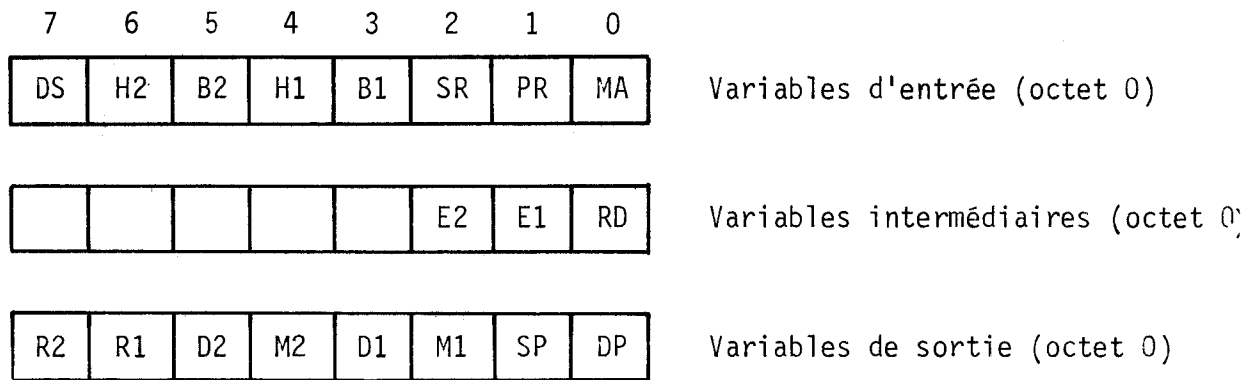


Figure 4.13 Table des variables élémentaires

La figure 4.14 montre la solution obtenue pour l'exemple de la perceuse. Dans la boucle, le programme :

- exécute les sorties
- effectue les entrées
- appelle le sous-programme "Evolution" pour les trois réseaux successivement.

Les sorties sont placées en début de boucle afin que la machine commandée reçoive ses commandes initiales dès le démarrage du programme.

Les entrées ne sont mises à jour et les sorties ne sont effectuées qu'une seule fois par cycle et dans leur ensemble. Cette procédure évite des aléas.

0 0
0

Nous avons envisagé la possibilité de faire du multiprocessing, c'est-à-dire exploiter plusieurs processeurs qui utilisent des mémoires communes. Une solution matérielle peut-être trouvée suivant le problème traité. Pour des réalisations de grande envergure, il existe des systèmes tout faits, par exemple le système MULTIBUS d'INTEL où chaque processeur est programmé comme travaillant seul.

ANNEXE 4.1

```

0000      *
0001      *
0002      *SOUS-PROGRAMME "EVOLUTION"
0003      *-----
0004      *
0005      *      UN EVENEMENT EST UN PRODUIT LOGIQUE DE VARIABLES
0006      *ELEMENTAIRES , AFFIRMEES OU COMPLEMENTEES. CES
0007      *VARIABLES PEUVENT ETRE EXTERIEURES (VARIABLES D'ENTREE
0008      *DU SYSTEME) , OU INTERMEDIAIRES.
0009      *
000A      *      CE SOUS-PROGRAMME EXAMINE SI L'UN OU L'AUTRE DES
000B      *EVENEMENTS ASSOCIES AUX DIFFERENTES TRANSITIONS
000C      *VALIDEES , EST VRAI.
000D      *      SI AUCUN EVENEMENT N'EST VRAI , ON RETOURNE AU
000E      *PROGRAMME PRINCIPAL. SI UN EVENEMENT EST VRAI , ON
000F      *PASSE A LA PLACE SUIVANTE ET ON EFFECTUE LES ACTIONS
0010      *CORRESPONDANTES : ON MET A 1 OU A 0 DES VARIABLES QUI
0011      *PEUVENT ETRE EXTERIEURES (VARIABLES DE SORTIE DU
0012      *SYSTEME) , OU INTERMEDIAIRES.
0013      *
0014      *      CE SOUS-PROGRAMME A ACCES :
0015      *--A LA TABLE "STRUCTURE" , QUI CONTIENT LA SUCCESSION DE
0016      *PLACES ET DES TRANSITIONS, AVEC L'ADRESSE DES ACTIONS
0017      *ET DES EVENEMENTS ASSOCIES.
0018      *      (EN FROM)
0019      *--AUX TABLES "ACTIONS" ET "EVENEMENTS" , QUI CONTIENNENT
001A      *LE DETAIL DES VARIABLES QUI CONSTITUENT CHACUNE DES
001B      *ACTIONS OU CHACUN DES EVENEMENTS.
001C      *      (EN FROM)
001D      *--AUX 3 TABLES DE VARIABLES ELEMENTAIRES :
001E      *      VARIABLES D'ENTREE
001F      *      VARIABLES DE SORTIE
0020      *      VARIABLES INTERMEDIAIRES
0021      *      (DANS LE SCRATCHPAD)
0022      *
0023      *REGISTRES UTILISES : 0,1,2,3
0024      *
0025      ORG      H'5000'
0026      ENT      EQU      5      BUFFER DES VAR. D'ENTREE
0027      SOR      EQU      6      BUFFER DES VAR. DE SORTIE
0028      INT      EQU      7      BUFFER DES VAR. INTERM.
0029      *
002A      *TESTS DES EVENEMENTS
002B      5000 08      LR      K,P
002C      5001 11      LR      H,DC
002D      5002 2C      XDC      SAUV. DE L'ADRESSE DE
002E      *      "NBRE. DE TRANS." EN DCI
002F      5003 10      LR      DC,H
0030      5004 16      LM
0031      5005 50      LR      0,A      NBRE DE TRANSITIONS EN 0
0032      5006 16      EVI      LM
0033      5007 06      LR      QU,A
0034      5008 16      LM
0035      5009 07      LR      QL,A
0036      500A 11      LR      H,DC      ADRESSE DE "DEPLACEMENT" EN H
0037      500B 0F      LR      DC,Q      BRANCHEMENT SUR LA TABLE
0038      *      DES EVENEMENTS
0039      500C 16      LM
003A      500D 51      LR      I,A      NBRE DE VAR. A TESTER EN I

```

0038		*	TEST D'UNE VARIABLE			
003C 500E 16	EV2	*	LM			
003D 500F 52		*	LR	2,A	ADR. ET ETAT DE LA VARIABLE EN 2	
003E		*				
003F		*	*-----*			
0040		*	7	6	5	4
0041		*				3
0042		*				2
0043		*				1
0044		*				0
0045		*	*-----*			
0046 5010 77		*	ETAT	NO.	BIT	EXT (1)
0047 5011 F2		*				NO. OCTET
0048 5012 08		*				OU
0049 5013 42		*				INTERM (0)
004A 5014 15		*				
004B 5015 81 04		*	LIS	7	MASQ. NO OCTET	
004C 5017 65		*	NS	2		
004D 5018 90 02		*	LR	15,A		
004E 501A 67	EV3	*	LR	A,2		
004F 501B 20 70	EV4	*	SL	4		
0050 501D F2		*	BP	EV3		
0051 501E 14		*	LISU	ENT	VAR. D'ENTREE	
0052 501F 1F		*	BR	EV4		
0053 5020 53		*	LISU	INT	VAR. INTERM.	
0054 5021 71	EV5	*	LI	H'70'	MASQ. NO. BIT	
0055 5022 33		*	NS	2		
0056 5023 84 04		*	SR	4		
0057	*	*	INC			
0058 5025 13		*	LR	3,A		
0059 5026 90 FB		*	LIS	1		
005A 5028 FC	EV6	*	DS	3		
005B 5029 42		*	BZ	EV6	UN 1 DANS A AU RANG DU BIT	
005C 502A 94 02		*			A TESTER	
005D 502C 18		*	SL	1		
005E 502D 21 FF	EV7	*	BR	EV5		
005F 502F 81 11		*	NS	5	LE BIT EST-IL A 1 OU A 0 ?	
0060	*	*	LR	A,2		
0061 5031 31		*	BZ	EV7	BIT A 1	
0062	*	*	COM			
0063	*	*	NI	H'FF'		
0064 5032 94 08		*	BP	EV8	LA VAR. N'EST PAS	
0065 5034 10		*			DANS LE BON ETAT	
0066	*	*	DS	1	LA VAR. EST DANS L'ETAT VOULU.	
0067	*	*			TOUTES LES VAR. ONT-ELLES ETE	
0068 5035 16		*	BZ	EV2	TESTEES ?	
0069 5036 8E		*	LR	DC,H	NON. VOIR LES AUTRES	
006A 5037 16		*			OUI. EVENEMENT VRAI	
006B 5038 06		*			DC RENIS SUR "TABLE STRUCTURE"	
006C 5039 16		*			A L'ADRESSE DE "DEPLACEMENT"	
006D 503A 07		*	LM			
006E 503B 2C		*	ADC		DC A LA PLACE SUIVANTE	
006F	*	*	LM			
0070 503C 0F		*	LR	QU,A		
0071	*	*	LM			
0072 503D 16		*	LR	QL,A		
		*	XDC		SAUV. DE L'ADRESSE DE	
		*			"NBRE. DE TRANS." EN DCI	
		*	LR	DC,Q	BRANCH. DUR LA TABLE DES	
		*			ACTIONS	

0073	503E	51		LR	1,A	
0074	503F	90	0B	BR	EVA	
0075	5041	30		DS	0	TOUTES LES TRANSITIONS ONT-ELLES
0076				*		EIE EXAMINEES ?
0077	5042	94	03	BNZ	EV9	NON
0078	5044	20		XDC		OUI
0079	5045	00		PK		AUCUN EVENEMENT N'EST VRAI,
007A				*		SORTIE DU PROGRAMME
007B	5046	10		EV9	LR	DC,H
007C	5047	71			LIS	1
007D	5048	8E			ADC	
007E	5049	90	0C		BR	EVI
007F				*		
0080				*EXECUTION DES ACTIONS : MISE A 0 OU A 1		
0081				*	DE VAR. DE SORTIE OU INTERM.	
0082				*		
0083	504B	16		EVA	LM	
0084	504C	52		LR	2,A	ADR. ET ETAT DE LA VAR. EN 2
0085	504D	77		LIS	7	
0086	504E	F2		NS	2	
0087	504F	0B		LR	15,A	
0088	5050	42		LR	A,2	
0089	5051	15		SL	4	
008A	5052	81	04	BP	EV8	
008B	5054	66		LISU	SOR	VAR. DE SORTIE
008C	5055	90	02	BR	EVC	
008D	5057	67		EV8	LISU	INT
008E	5058	20	70	EVC	LI	H'70'
008F	505A	F2			NS	2
0090	505B	14			SR	4
0091	505C	1F			INC	
0092	505D	53			LR	3,A
0093	505E	71			LIS	1
0094	505F	33		EVD	DS	3
0095	5060	84	04		BZ	EVE
0096	5062	13			SL	1
0097	5063	90	FB		BR	EVD
0098	5065	53		EVE	LR	3,A
0099	5066	18			COM	
009A	5067	FC			NS	5
009B	5068	5C			LR	S,A
009C	5069	42			LR	A,2
009D	506A	21	FF		NI	H'FF'
009E	506C	81	04		BP	EVF
009F	506E	43			LR	A,3
00A0	506F	EC			XS	5
00A1	5070	5C			LR	S,A
00A2	5071	31		EVF	DS	1
00A3	5072	94	08		BNZ	EVA
00A4				*		
00A5	5074	2C			XDC	
00A6	5075	0C			PK	
00A7					END	
00						

CHAPITRE V

M I S E E N O E U V R E G L O B A L E

Aussitôt, les uns escaladèrent le mur,
et les autres pénétrèrent par la solide porte.
Homère . L'Iliade

Dans ce dernier chapitre, nous incorporons à la commande d'ascenseur, l'automate étudié au chapitre précédent. Le programme total de commande contient donc deux parties : une partie étudiée de manière habituelle par organigrammes et programmée à partir des instructions de base du microprocesseur, et une autre partie traitée par réseaux de Pétri. Les communications entre ces deux sous-ensembles sont établies au moyen de variables intermédiaires. La mise en oeuvre est faite à partir d'un seul microprocesseur MOSTEK F8 pour un ascenseur à 24 niveaux. Le programme se trouve en annexe de ce chapitre.

5.1. ORGANISATION GENERALE

L'introduction de l'automate permet, comme prévu, d'aboutir à un programme général constitué d'une suite de modules. Cet essai a été effectué en simulation. Un panneau de boutons donne la possibilité d'appliquer toutes les entrées nécessaires : appels, envois, contacts divers.... Grâce à une visualisation sur écran de toutes les variables qui entrent en jeu dans la commande, ainsi que des pointeurs des tables de structure, on peut suivre continûment l'évolution de la commande.

Examinons d'abord l'ensemble du programme donné par la figure 5.1. Après l'initialisation, qui se fait au départ, on arrive dans la boucle des modules. L'état des variables de sortie est, dans un premier

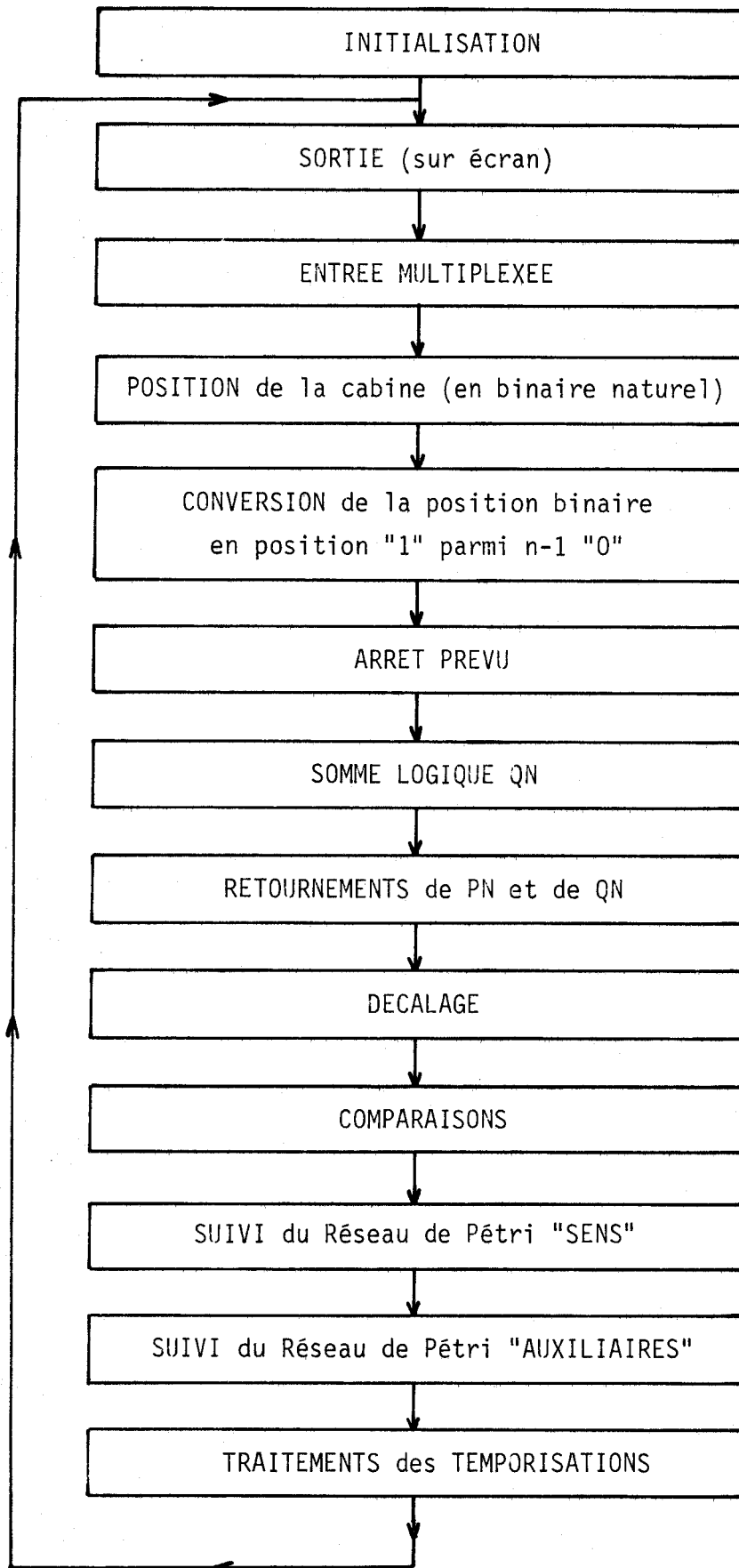


Figure 5.1 Organigramme général

temps, écrit sur l'écran puis on procède aux entrées des demandes de service et des états des différents contacts de contrôle. La position de la cabine est déterminée en incrémentant ou décrémentant un registre du "bloc-notes" suivant les variations des signaux A et B émis par le capteur de position. On obtient ainsi une position en binaire naturel POSB. Cette position est aussitôt convertie en un "1" parmi 23 "0" dans un mot de longueur 24, le rang du "1" correspondant au numéro du niveau (mot PN).

Puis on examine si une condition d'arrêt "parcours" est remplie, c'est à dire si, au niveau atteint, on a un envoi-cabine ou un appel-palier dans le sens de la marche. Si au moins l'une des conditions est remplie, on met à 1 une variable intermédiaire DA, demande d'arrêt, qui sera prise en compte plus loin dans le programme.

On ne cherche donc pas directement les possibilités d'arrêt "bout de course". L'obtention de ces arrêts se fait dans l'agencement des modules. En effet, après avoir établi QN, somme logique de toutes les demandes et avoir retourné la position PN et la somme QN, on procède immédiatement au décalage de PN dans le sens de la montée à condition que la cabine monte effectivement, ou de PN retourné, dans le sens de la descente, à condition que la cabine descende. La position retournée est notée PR. La somme retournée des demandes est notée QR. La double comparaison vient ensuite et s'applique donc

à QN et PN éventuellement décalé

et à QR et PR éventuellement décalé.

Cette façon de procéder est une application du principe exposé au chapitre 2 pour la détermination des arrêts "bout de course".

Prenons un exemple. L'ascenseur monte pour un appel-descente du 18ème étage. Il n'y a plus de demande pour aller au-delà. On décale PN vers les niveaux les plus élevés. On ne touche pas à PR. Tant que l'ascenseur n'est pas au 18ème étage, PN décalé n'est jamais supérieur à QN. Le sens ne change pas. A l'arrivée au 18ème étage, le programme enregistre cette position et voit qu'il n'y a pas d'arrêt-parcours. Il décale PN qui devient plus grand que QN. Le sens change et au cycle suivant du programme, l'arrêt est obtenu. On est en descente et on y demeure.

L'arrêt est donc dans ce cas obtenu avec un retard correspondant à l'exécution d'un balayage complet du programme. Ce retard, de l'ordre

quelques millisecondes, est négligeable devant le temps que met la cabine pour s'arrêter.

Pour terminer la boucle, on fait deux fois appel au sous-programme "Evolution" pour suivre deux réseaux de Pétri : l'un pour déterminer le sens de parcours et la demande de service, l'autre pour contrôler tous les auxiliaires. Enfin le dernier module met en oeuvre les temporisations demandées.

Reprenons dans le détail quelques points particuliers.

5.2. VISUALISATION

Le bloc-notes (scratchpad) du microprocesseur F8 est une mémoire RAM de 64 octets adressés par l'ISAR (Indirect Scratchpad Address Register). Ce bloc-notes est divisé en 8 buffers de 8 octets chacun notés de 0 à 7. Les deux premiers buffers (0 et 1) constituent les registres de programmation et servent tout au long du programme à des fins diverses. Les autres buffers reçoivent des informations explicitées ci-après et qui sont visualisées en binaire suivant la disposition générale de la figure 5.2.

Les buffers 2, 3 et 4 ont été accolés pour former des mots de longueur 24, nécessaires pour la commande d'un ascenseur à 24 niveaux (figure 5.3). On mémorise d'abord le mot EC des envois-cabine, le mot AM des appels-montée et le mot AD des appels-descente. On y place ensuite la somme logique QN, le mot retourné QR, la position PX, qui est la position PN si on descend ou la position PN décalée si on monte, et enfin la position PY, qui est la position PR si on monte ou la position PR décalée si on descend. Il reste encore un mot noté DV, disponible, non utilisé dans notre programme.

Les variables d'entrée sont rangées dans le buffer 5, les variables de sortie dans le buffer 6 et les variables intermédiaires dans le buffer 7. Comme le nombre des variables de sortie et intermédiaires n'est pas important, les buffers 6 et 7 sont également utilisés pour garder les adresses des pointeurs de tables et la position binaire POSB de la cabine.

La visualisation de ces buffers 2 à 7 a lieu à chaque cycle de de programme, ce qui permet de contrôler pas à pas la commande et d'en vérifier le bon fonctionnement.

Buffer 2	Buffer 3	Buffer 4
Buffer 5	Buffer 6	Buffer 7

Figure 5.2

EC	Instantanés des divers mots de longueur 24		
AM			
AD			
QN			
QR			
PX			
PY			
DV			
00	Variables d'entrée	Variables de sortie	Variables intermédiaires
01			
02		Position binaire Pointeur table "Position"	Pointeurs des réseaux
03			
04			
05			
06			
07			

Figure 5.3

5.3. RESEAUX DE PETRI

Notre programme comporte donc deux réseaux de Pétri. Le détail des variables d'entrée, de sortie et intermédiaires, utiles pour ces réseaux, est indiqué à la figure 5.4.

Le premier réseau (figure 5.5) représente la bascule qui commande la montée ou la descente à partir des résultats CN et CR des deux comparateurs. Une troisième place permet de savoir que l'ascenseur n'est plus sollicité. La figure 5.6 donne les contenus et adresses de la table de structure, de la table des événements et de la table des actions.

Le deuxième réseau, plus important, doit commander effectivement le moteur de traction et tous les auxiliaires, en fonction des informations fournies par le programme standard ou par les différents capteurs. Le réseau dessiné (figure 5.7) est un réseau simplifié qui fait comprendre l'esprit de la méthode. Les tables correspondantes sont indiquées à la figure 5.8.

Nous reproduisons également aux figures 5.10a, 5.10b et 5.10c un réseau de Pétri étudié pour veiller à la sécurité de fonctionnement d'un ascenseur à blocage. C'est un graphe d'état qui tient compte de toutes les exigences de la manoeuvre. La compréhension totale n'est pas aisée car bon nombre de variables qui apparaissent sur ce réseau sont, soit élaborées, soit traitées par la partie de programme classique.

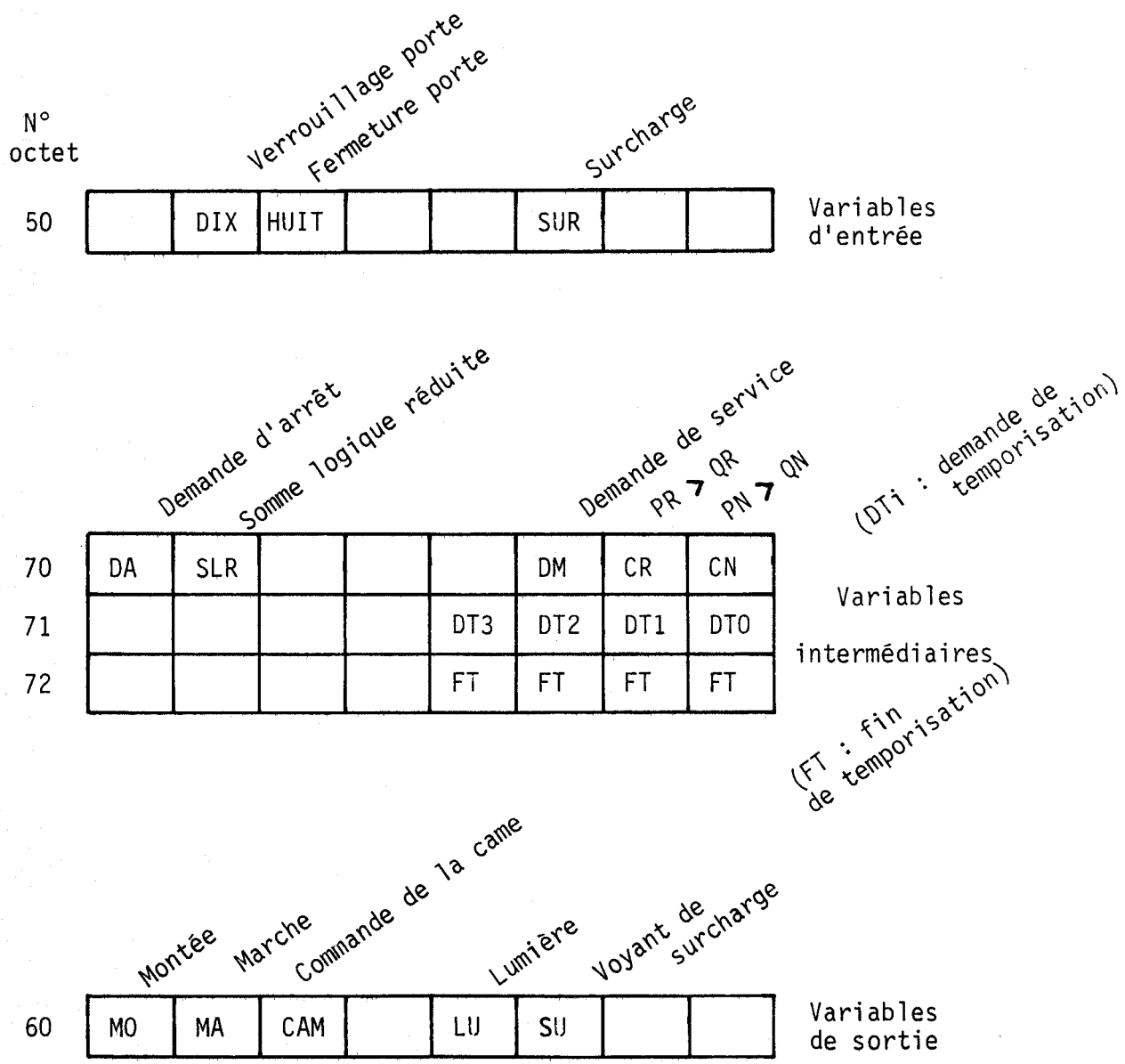


Figure 5.4 Implantation des variables

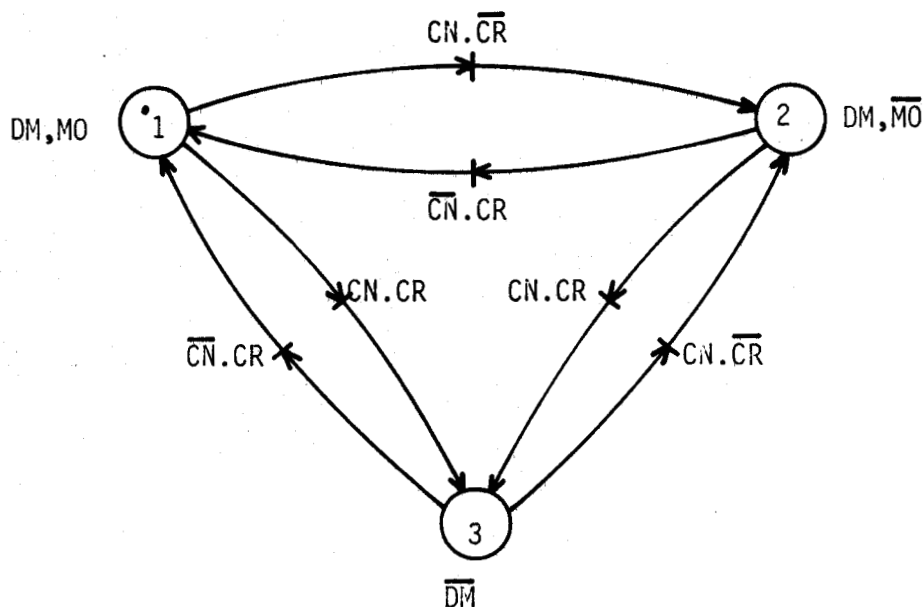


Figure 5.5. Choix du sens

N° des places	Codes	Adresses	Evénements	Codes	Adresses
①	4E	4E00	CN, CR	02	4E20
	30	1		80	1
	02	2		10	2
	4E	3	CR, CN	02	3
	20	4		00	4
	03	5		90	5
	4E	6	CN, CR	02	6
	26	7		80	7
	09	8		90	8
②	4E	9			
	33	A			
	02	B			
	4E	C			
	23	D			
	F1	E			
	4E	F			
	26	10			
③	00	1			
	4E	2			
	36	3			
	02	4			
	4E	5			
	23	6			
	E8	7			
	4E	8			
	20	9			
	EE	A			

Table des événements

Actions	Codes	Adresses
DM, MO	02	4E30
	A0	1
	F8	2
DM, M0	02	3
	A0	4
	78	5
DM	01	6
	20	7

Tables des actions

Table de structure



Figure 5.6 Tables du réseau de la figure 5.5

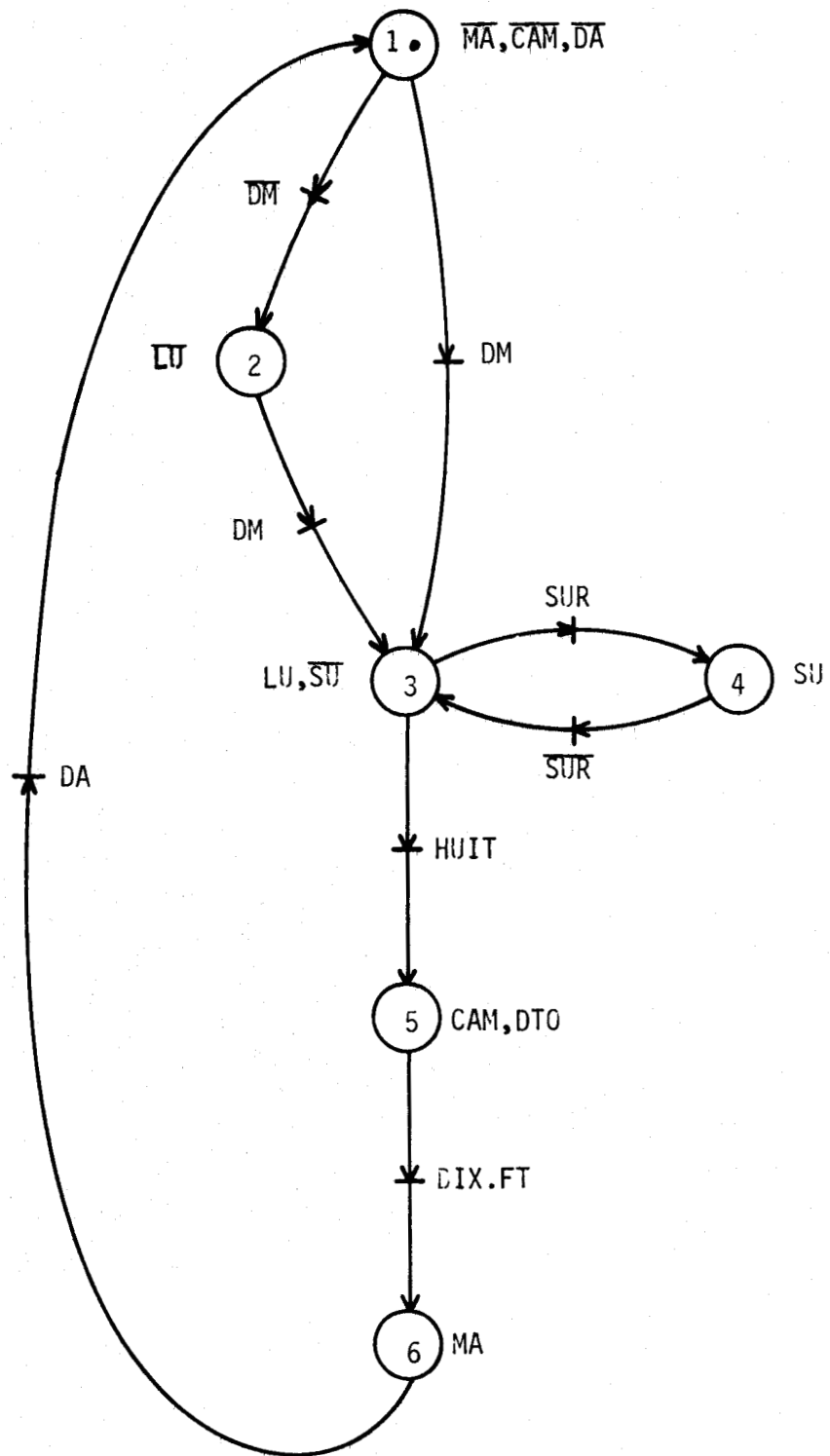


Figure 5.7 Commande du moteur et des auxiliaires.



N° des places	Codes	Adresses	Evènements	Codes	Adresses
①	4E	4E40	DM	01	4E70
	90	1		A0	1
	02	2	DM	01	2
	4E	3		20	3
	70	4	SUR	01	4
	09	5		A8	5
	4E	6	SUR	01	6
	72	7		28	7
	00	8	HUIT	01	8
	4E	9		D8	9
	94	A	DIX, FT	02	A
	01	B		E8	B
	4E	C		92	C
	70	D	DA	01	D
	00	E		F0	E
	4E	F			
②	96	50			
	02	1			
	4E	2			
	74	3			
	03	4			
	4E	5			
	78	6			
	06	7			
	4E	8			
	99	9			
	01	A			
	4E	B			
	76	C			
	F1	D			
	4E	E			
	9B	F			
③	01	60			
	4E	1			
	7A	2			
	00	3			
	4E	4			
	9E	5			
	01	6			
	4E	7			
	7D	8			
	D6	9			
④					
⑤					
⑥					

Table des évènements

Actions	Codes	Adresses
MA, CAM, DA	03	4E90
	68	1
	58	2
	70	3
LU	01	4
	38	5
LU, SU	02	6
	B8	7
	28	8
SU	01	9
	A8	A
CAM, DTO	02	B
	D8	C
	91	D
MA	01	E
	E8	F

Table des actions

Table de structure

Figure 5.8 : Tables du réseau de la figure 5.7.



VARIABLES D'ENTREE

(proviennent de l'extérieur de la carte)

SIX	: sécurité cabine
HUIT	: shunt portes
DIX	: verrouillage portes
MDN	: sécurité de fonctionnement du contacteur montée-descente
GVN	: sécurité de fonctionnement du contacteur grande vitesse
PVN	: sécurité de fonctionnement du contacteur petite vitesse
GV	: en grande vitesse (\overline{GV} : en petite vitesse)
M	: en montée (\overline{M} : en descente)
EA	: annulation d'un ordre
OSTH	: option sonde thermique sur le moteur de traction
OTHV	: option sonde thermique sur le moteur de ventilation
OSU	: option surcharge
CLO	: option clignotant
OIG	: option intégrateur
MASN	: défaut sur la masse
STH	: sonde thermique traction
THV	: sonde thermique ventilation
SUR	: surcharge
IP	: intégrateur
RESET	: retour aux conditions initiales

VARIABLES INTERMEDIAIRES

(n'apparaissent pas à l'extérieur de la carte)

LT0 à LT7	: lancements de temporisation de diverses durées
FT	: fin de temporisation
UNP	: variables d'entrée actualisées
CC	: commun cabine
CP	: validation des boutons-poussoirs paliers
SP	: suspension de service des paliers
ASU	: autorisation surcharge
CLD	: clignotage autorisé
PEA	: EA actionné
ORD1	: ordre passé
CM	: orientation montée
CD	: orientation descente
GUM	: indication de mouvement
RECA	: recalage
SCMT	: défaut de fonctionnement de la came
CSL	: appareil en mouvement
CEX	: changement des contacts extrêmes
DEFI	: défaut définitif
DCA	: défaut sens de parcours en descente
MCA	: défaut sens de parcours en montée

Figure 5.9a Tableau des variables (1ère partie)

VARIABLES DE SORTIE

VARIABLES INTERNES

commandent des éléments à l'extérieur de
la carte)

(restent dans le réseau de Pétri)

	: commande de l'éclairage	COUP	: coupure chaîne de sécurité
AM	: commande de la came	PIP	: changement d'état de IP
L	: commande du clignotant	CVI	: passage de GV à PV
U	: visualisation défaut surcharge	SCM0	: la came a été commandée 1 fois sans succès
EFi	: i = 0 à F (en hexadécimal) défaut N°i	SCM1	: la came a été commandée 2 fois sans succès
DEF	: suppression des défauts		

Figure 5.9b : Tableau des variables (2ème partie)

Les tableaux des figures 5.9a et 5.9b donnent la liste des variables d'entrée, de sortie et intermédiaires qui apparaissent dans ce réseau. Parmi les variables intermédiaires, on trouve quelques variables internes : variables élaborées et reprises par le réseau lui-même. Voici quelques indications sur les trois parties, correspondant aux trois figures, de ce réseau.

- Figure 5.10a : Cette partie correspond à l'état de repos de l'ascenseur on y teste la possibilité d'apparition de défauts (pertes de courant par la masse, dépassement de température dans le moteur de traction ou dans le moteur de ventilation, défaut de sécurité de la cabine). L'apparition d'un défaut interrompt le service de l'ascenseur, qui reprend à la disparition du défaut.

- Figure 5.10b : La cabine a été sollicitée, la porte s'est refermée. On teste un défaut éventuel de surcharge, puis on commande la came et on vérifie le verrouillage de la porte.

- Figure 5.10c : L'appareil est en mouvement. On examine si le sens de parcours réel correspond au sens de déplacement commandé. On vérifie, par le test de la fermeture régulière du "contact d'intégration", la bonne adhérence des câbles sur la poulie d'entraînement. On regarde si le passage de la grande à la petite vitesse s'effectue correctement. Enfin on vérifie à tout moment

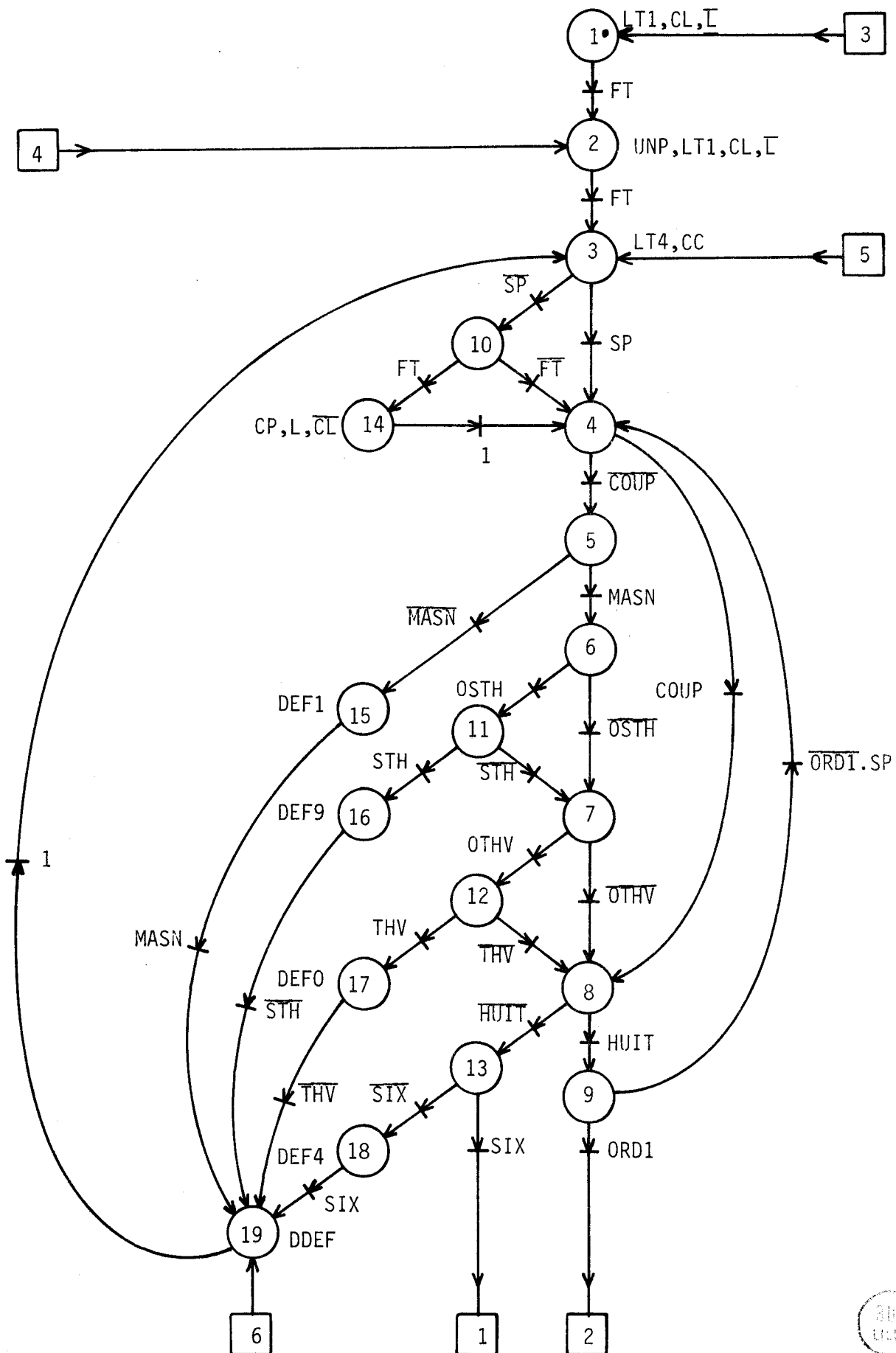


Figure 5.10a

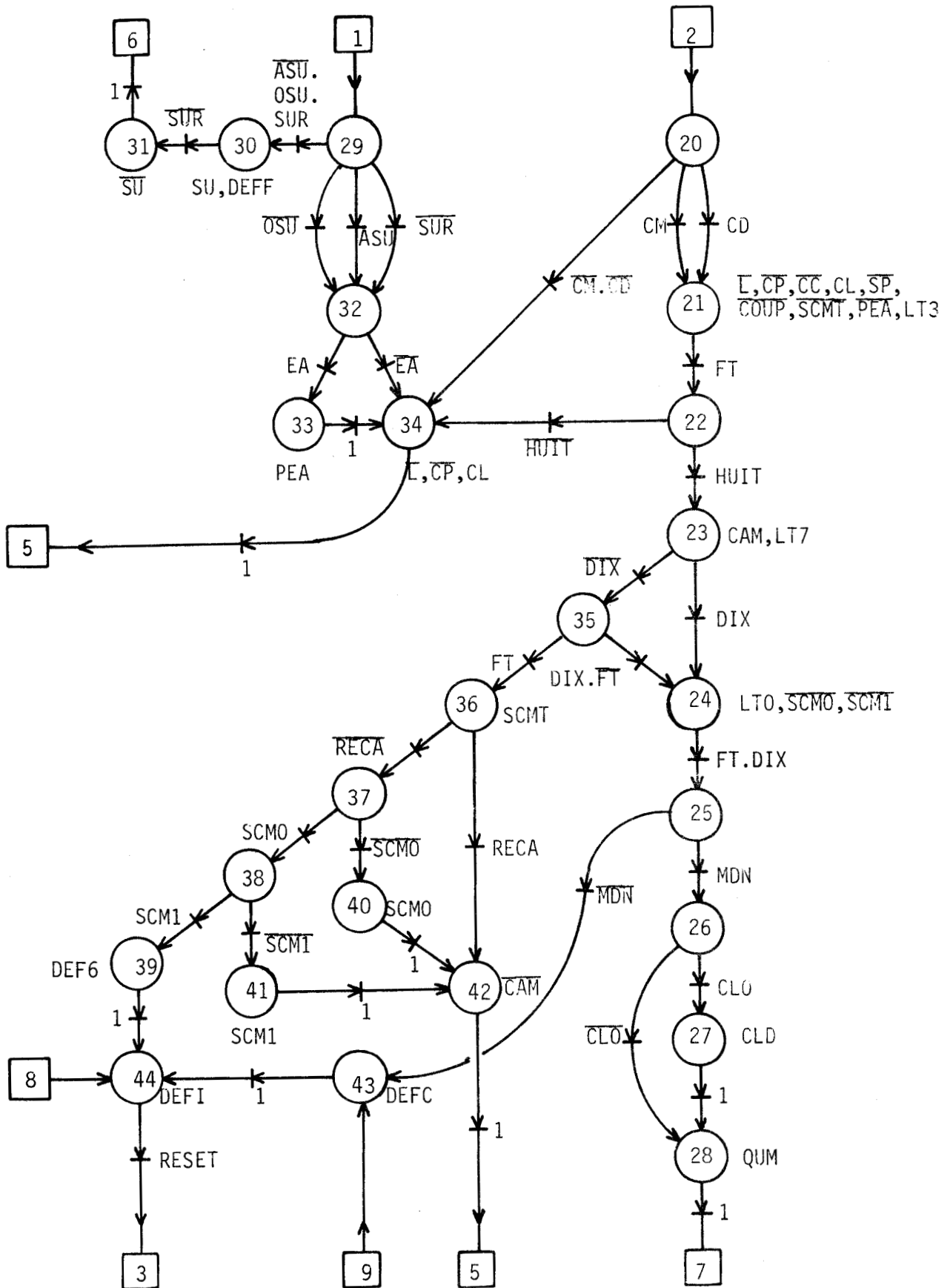


Figure 5.10b



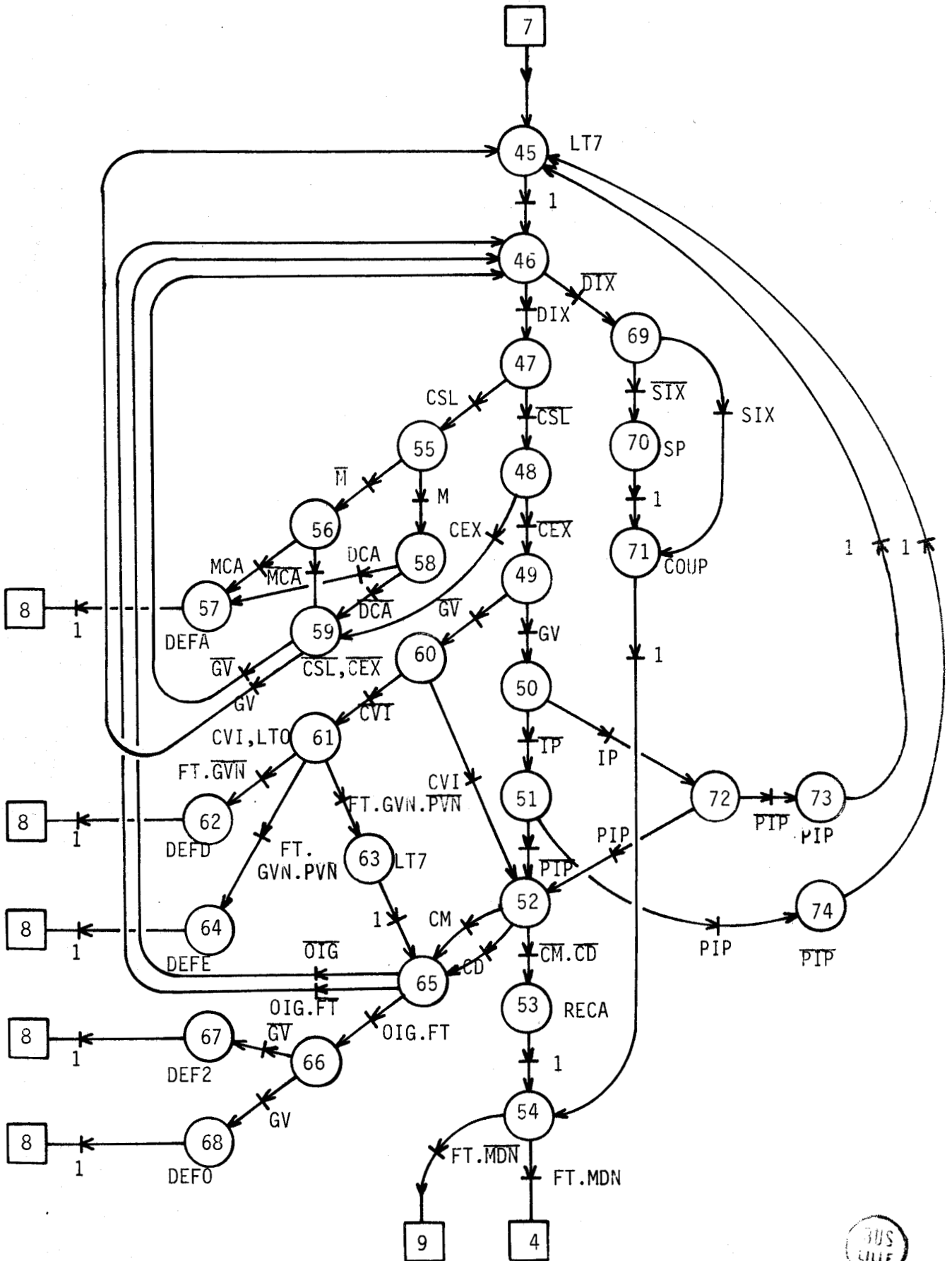


Figure 5.10c

s'il n'y a pas ouverture de la chaîne des contacts de sécurité. Les défauts de la 2ème et de la 3ème partie aboutissent habituellement à l'arrêt définitif de la cabine. Il est alors nécessaire que le service d'entretien intervienne pour la remise en route (Il dispose d'un bloc de visualisation qui indique le numéro du défaut).

5.4. TEMPORISATIONS

En programmation habituelle, les temporisations créent des difficultés. Il est facile certes de lancer une temporisation, à condition qu'elle soit de courte durée. Mais dans l'attente de la fin de cette temporisation, signalée par interruption, on continue habituellement de contrôler le fonctionnement de la machine commandée. Il faut donc toujours prévoir dans le programme la possibilité d'être interrompu sans gêne et, lorsque l'interruption arrive, faire les sauvegardes nécessaires.

Dans le réseau de Pétri, la commande de temporisation est une variable intermédiaire qui apparaît comme toute variable d'action. La fin de temporisation est de même une variable intermédiaire qui intervient au niveau des transitions comme toute autre variable d'évènement.

Dans notre programme final, nous disposons d'un seul organe de temporisation et il est possible de programmer plusieurs durées. Dans le cas traité, quatre durées sont utilisées. Les demandes de temporisations sont notées DT0, DT1, DT2 et DT3. La fin des temporisations est unique ; elle est notée FT. Le programme "Temporisations" examine donc s'il y a une demande de temporisation. Si c'est le cas, il charge le registre 4 à une valeur lue dans un tableau suivant la longueur de temporisation demandée. Le temps maximum de temporisation est :

$$3,9525 \times 255 = 1008 \text{ ms}$$

et le registre 4 est chargé à 255. Si on désire des temporisations plus longues, ou des temps intermédiaires, on a la possibilité de mettre plusieurs demandes de temporisations à la suite dans le graphe (figure 5.11).

Quand une temporisation est en cours, le registre 4 est décrémenté par interruption,

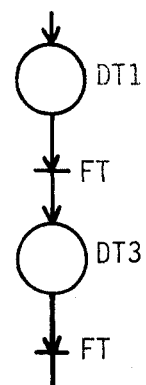


Figure 5.11

alors que le programme général tourne. Il ne faut donc pas utiliser ce registre 4 dans le programme général. On peut interdire les interruptions à certains endroits du programme, où celles-ci ne seraient pas les bienvenues. On interdit surtout les interruptions dans le cas d'appel à un sous-programme pour éviter d'avoir à faire les sauvegardes et restitutions nécessaires en cette circonstance. Mais, étant donné que le programme est constitué d'une suite de modules, il y a peu d'appels de sous-programmes. On voit donc que les précautions à prendre ne sont pas excessives.

o o

o

Une originalité de cette commande programmée est d'allier deux procédés de description de fonctionnement du système. La description par organigramme conduit à une programmation classique. La description par réseau de Pétri entraîne l'utilisation d'un logiciel plus universel. La programmation classique permet de suivre, avec toute la richesse du jeu d'instructions du microprocesseur, des points spécifiques. Le logiciel de suivi de réseaux procure, dans le domaine qui lui est propre, un gain de temps d'étude et une facilité de modification.

ANNEXE 5.1

```

00000
00001
00002
00003
00004
00005
00006
00007
00008
00009
0000A
0000B
0000C
0000D
0000E
0000F
00010
00011
00012
00013
00014
00015
00016
00017
00018
00019
0001A
0001B
0001C
0001D
0001E
0001F
00020
00021
00022
00023
00024
00025
00026
00027
00028
00029
0002A
0002B
0002C
0002D
0002E
0002F
00030
00031
00032
00033
00034

4504 FE
4505 6D
4506 2E
4507 13
4508 00
4509 01
450A 03
450B 02
450C 00
450D 67
450E 6F
450F 20 42
4511 5E
4512 20 4E
4514 5E
4515 20 02
4517 5E
4518 20 4E
451A 5E
451B 66
451C 6F
451D 20 08
451F 5E
4520 20 45
4522 5E
4523 20 FF
4525 27 13
4527 73
4528 27 12
452A 90 21

*
*
*ASCENSEUR - MANOEUVRE COLLECTIVE - 24 NIVEAUX
*DOUBLE COMPARAISON - PETRI
*H. GHESTEM - DEC 1978
*MOSTEK F8
*
*****
*
*INITIALISATIONS
*****
*
PPD EQU H'4508' POINTEUR POSITION
PSS EQU H'4E02' POINTEUR 'SENS'
PRU EQU H'4E42' POINTEUR 'AUXILIAIRES'
*
*OR6 H'4504'
*TEMPS TEMPORISATIONS
DC I,H'FE'
DC I,H'6D'
DC I,H'2E'
DC I,H'13'
*DONNEES POUR ENTREE POSB
DC I,H'00'
DC I,H'01'
DC I,H'03'
DC I,H'02'
DC I,H'00'
*MISE EN PLACE DES POINTEURS
LISU 7
LISL 7
LI PRU POINTEUR 'AUXILIAIRES' EN 0'77'
LR D,A
LI PRU
LR D,A
LI PSS POINTEUR 'SENS' EN 0'75'
LR PSS
LI PSS
LR D,A
LISU 6
LISL 7
LI PPD POINTEUR POSITION EN 0'67'
LR D,A
LI PPD
LR D,A
*
*INITIALISATION TIMER
LI H'FF' ARRET TIMER
OUT H'13'
LIS 3 MOT DE CONTROLE DU TIMER
OUT H'12'
BR FVI
*

```

```

0035      *VISUALISATION
0036      *=====
0037      *
0038      *REGISTRES UTILISES : 0,1,2,6,7
0039      *
003A      TU      EQU      6
003B      TL      EQU      7
003C      *
003D      *PORT UTILISE : 1 (BIT 7 A 0 : DEM. DE VISU)
003E      *
003F      *SOUS-PROGR.
0040      *
0041      TT      EQU      H'0728' IMPRESSION D'UN CARACT.
0042      *VOC      VISU. D'UN OCTET
0043      *
0044      *CARACTERES ASCII
0045      *
0046      CR      EQU      H'0D'
0047      LF      EQU      H'0A'
0048      SP      EQU      H'20'
0049      *
004A      *TEXTES A AFFICHER, MIS EN MEMOIRE
004B      *
004C      452C 45 43 41 TXT1      DC      16,C'ECAMADQNRXPYDV'
           452F 4D 41 44
           4532 51 4E 51
           4535 52 50 58
           4538 50 59 44
           453B 56
004D      453C 30 30 30 TXT2      DC      16,C'0001020304050607'
           453F 31 30 32
           4542 30 33 30
           4545 34 30 35
           4548 30 36 30
           454B 37
004E      *
004F      *VISUALISATION DE EC,AM,AD,QN,QR,PN,PR,DV
0050      *
0051      454C 70      FVI      LIS      0
0052      454D B1      OUTS     1
0053      454E A1      INS      1
0054      454F 91 FC      BM      FVI      PAS DE DEM. DE VISU
0055      4551 7F      LIS      H'F'
0056      4552 50      LR      0,A
0057      4553 70      LIS      0
0058      4554 56      LR      TU,A
0059      4555 2A 45 2C      DCI      TXT1
005A      4558 16      VISI      LM      1ER. CAR.
005B      4559 57      LR      TL,A
005C      455A 1A      DI
005D      455B 28 07 28      PI      TT
005E      455E 16      LM      2EME. CAR.
005F      455F 57      LR      TL,A
0060      4560 28 07 28      PI      TT
0061      4563 20 20      LI      SP
0062      4565 57      LR      TL,A
0063      4566 28 07 28      PI      TT
0064      4569 40      LR      A,0      ISAR
0065      456A 1F      INC
0066      456B 50      LR      0,A
0067      456C 0B      LR      IS,A
0068      456D 28 45 C5      PI      VOC

```

0069	4570	63		LISU	3	
006A	4571	28	45 C5	PI	VOC	
006B	4574	64		LISU	4	
006C	4575	28	45 C5	PI	VOC	
006D	4578	7D		LIS	CR	RETOUR A LA LIGNE
006E	4579	57		LR	TL,A	
006F	457A	28	07 28	PI	TT	
0070	457D	7A		LIS	LF	
0071	457E	57		LR	TL,A	
0072	457F	28	07 28	PI	TT	
0073	4582	8F	D5	BR7	VISI	
0074	4584	20	0A	LI	LF	
0075	4586	57		LR	TL,A	
0076	4587	28	07 28	PI	TT	
0077				*		
0078				*VISUALISATION DES BUFFERS 5,6 ET 7		
0079				*		
007A	458A	20	27	LI	H'27'	
007B	458C	50		LR	0,A	
007C	458D	70		LIS	0	
007D	458E	56		LR	TU,A	
007E	458F	2A	45 3C	DCI	TXT2	
007F	4592	16		LM		1ER. CAR
0080	4593	57		LR	TL,A	
0081	4594	28	07 28	PI	TT	
0082	4597	16		LM		2EME. CAR.
0083	4598	57		LR	TL,A	
0084	4599	28	07 28	PI	TT	
0085	459C	20	20	LI	SP	
0086	459E	57		LR	TL,A	
0087	459F	28	07 28	PI	TT	
0088	45A2	40		LR	A,0	ISAR
0089	45A3	1F		INC		
008A	45A4	50		LR	0,A	
008B	45A5	0B		LR	IS,A	
008C	45A6	28	45 C5	PI	VOC	
008D	45A9	66		LISU	6	
008E	45AA	28	45 C5	PI	VOC	
008F	45AD	67		LISU	7	
0090	45AE	28	45 C5	PI	VOC	
0091	45B1	7D		LIS	CR	RETOUR A LA LIGNE
0092	45B2	57		LR	TL,A	
0093	45B3	28	07 28	PI	TT	
0094	45B6	7A		LIS	LF	
0095	45B7	57		LR	TL,A	
0096	45B8	28	07 28	PI	TT	
0097	45BB	8F	D6	BR7	VISI	
0098	45BD	20	0A	LI	LF	
0099	45BF	57		LR	TL,A	
009A	45C0	28	07 28	PI	TT	
009B				*		
009C	45C3	90	29	BR	EN0	
009D				*		
009E				*SOUS-PROGRAMME VOC : VISU D'UN OCTET		
009F				*		
00A0				*REGISTRES UTILISES : 1 : DONNEE		
00A1				*	2 : DECOMPTEUR DE BITS	
00A2				*		
00A3	45C5	08		VOC	LR	K,P
00A4	45C6	78		LIS	8	
00A5	45C7	52		LR	2,A	
00A6	45C8	4C		LR	A,S	

00A7	45C9	51		LR	I,A
00A8	45CA	21	FF	NI	H'FF'
00A9	45CC	91	05	VOC1	VOC2
00AA	45CE	20	30	LI	C'0'
00AB	45D0	90	03	BR	VOC3
00AC	45D2	20	31	VOC2	LI
00AD	45D4	57		VOC3	LR
00AE	45D5	28	07 28	PI	TT
00AF	45D8	20	20	LI	SP
00B0	45DA	57		LR	TL,A
00B1	45DB	28	07 28	PI	TT
00B2	45DE	32		DS	2
00B3	45DF	84	06	BZ	VOC4
00B4	45E1	41		LR	A,I
00B5	45E2	13		SL	I
00B6	45E3	51		LR	I,A
00B7	45E4	90	E7	BR	VOC1
00B8	45E6	20	20	VOC4	LI
00B9	45E8	57		LR	TL,A
00BA	45E9	28	07 28	PI	TT
00BB	45EC	0C		PK	
00BC				*	
00BD				*ENTREES MULTIPLEXEES DE EC, AM, AD, EN	
00BE				*****	
00BF				*	
00C0				*REGISTRES UTILISES : 0 : SAUVEGARDES	
00C1				1 : ENTREE OCTET-DONNEE	
00C2				*	
00C3				*PORTS UTILISES :	
00C4				PS	ERU H'10' SORTIE MULTIP.
00C5				PE	ERU H'11' ENTREE MULTIP.
00C6				PI	ERU I BIT 6 : CHOIX DE L'UN DES
00C7				*	DEUX BLOCS D'ENTREES
00C8				*	
00C9	45ED	71		EN0	LIS I
00CA	45EE	50		LR	0,A
00CB	45EF	28	46 68	PI	ENM
00CC	45F2	62		LISU	2
00CD	45F3	21	40	NI	H'40'
00CE	45F5	94	07	BNZ	A22
00CF	45F7	69		LISL	I
00D0	45F8	28	46 74	PI	OU
00D1	45FB	90	05	BR	ENI
00D2	45FD	6A		A22	LISL 2
00D3	45FE	28	46 74	PI	OU
00D4	4601	72		ENI	LIS 2
00D5	4602	50		LR	0,A
00D6	4603	28	46 68	PI	ENM
00D7	4606	63		LISU	3
00D8	4607	21	40	NI	H'40'
00D9	4609	94	07	BNZ	A32
00DA	460B	69		LISL	I
00DB	460C	28	46 74	PI	OU
00DC	460F	90	05	BR	EN2
00DD	4611	6A		A32	LISL 2
00DE	4612	28	46 74	PI	OU
00DF	4615	74		EN2	LIS 4
00E0	4616	50		LR	0,A
00E1	4617	28	46 68	PI	ENM
00E2	461A	64		LISU	4
00E3	461B	21	40	NI	H'40'
00E4	461D	94	07	BNZ	A42

00E5 461F 69		LISL	1	
00E6 4620 28 46 74		PI	OU	
00E7 4623 90 05		BR	EN3	
00E8 4625 6A	A42	LISL	2	
00E9 4626 28 46 74		PI	OU	
00EA 4629 78	EN3	LIS	8	
00EB 462A 50		LR	0.A	
00EC 462B 28 46 68		PI	ENM	
00ED 462E 68		LISL	0	
00EE 462F 21 40		NI	H'40'	
00EF 4631 94 07		BNZ	E30	
00F0 4633 62		LISU	2	
00F1 4634 28 46 74		PI	OU	
00F2 4637 90 05		BR	EN4	
00F3 4639 63	E30	LISU	3	
00F4 463A 28 46 74		PI	OU	
00F5 463D 20 10	EN4	LI	H'10'	
00F6 463F 50		LR	0.A	
00F7 4640 28 46 68		PI	ENM	
00F8 4643 64		LISU	4	
00F9 4644 21 40		NI	H'40'	
00FA 4646 94 07		BNZ	E47	
00FB 4648 68		LISL	0	
00FC 4649 28 46 74		PI	OU	
00FD 464C 90 05		BR	EN5	
00FE 464E 65	E47	LISU	5	
00FF 464F 68		LISL	0	
0100 4650 41		LR	A.I	VAR. EN NON MEMORISEES
0101 4651 50		LR	S.A	
0102 4652 20 20	EN5	LI	H'20'	
0103 4654 50		LR	0.A	
0104 4655 28 46 68		PI	ENM	
0105 4658 1B		EI		
0106 4659 65		LISU	5	
0107 465A 21 40		NI	H'40'	
0108 465C 94 06		BNZ	D27	
0109 465E 69		LISL	1	
010A 465F 41		LR	A.I	
010B 4660 50		LR	S.A	
010C 4661 90 1A		BR	PST	
010D 4663 6A	D27	LISL	2	
010E 4664 41		LR	A.I	
010F 4665 50		LR	S.A	
0110 4666 90 15		BR	PST	
0111	*			
0112	*SOUS- PROGRAMMES			
0113	*			
0114 4668 40	ENM	LR	A.0	
0115 4669 27 10		OUT	PS	
0116 466B 70		CLR		
0117 466C 27 11		OUT	PE	
0118 466E B1		OUTS	PI	
0119 466F 26 11		IN	PE	ENTREE OCTET-DONNEE
011A 4671 51		LR	1.A	SAUVE EN 1
011B 4672 A1		INS	PI	ENTREE OCTET-CHOIX
011C 4673 1C		POP		
011D	*			
011E 4674 41	OU	LR	A.I	
011F 4675 EC		XS	S	
0120 4676 50		LR	0.A	X RANGEE 0
0121 4677 41		LR	A.I	
0122 4678 FC		NS	S	AND EN A

0123 4679 E0	XS	0	
0124 467A 5C	LR	S.A	OU EN S
0125 467B 1C	POP		
0126	*		
0127	*ENTREE POSITION (POSB)		
0128	*=====		
0129	*		
012A	*MISE EN PLACE DU POINTEUR DC POUR 'POSITION'		
012B 467C 66	PST	LISU	6
012C 467D 6F		LISL	7
012D 467E 4E		LR	A.D
012E 467F 07		LR	QL.A
012F 4680 4C		LR	A.S
0130 4681 06		LR	QU.A
0131 4682 0F		LR	DC.Q
0132	*		
0133	*REG UTILISES : H		
0134	*		
0135	*PROGRAMME		
0136 4683 66		LISU	6 POSB EN 0'65'
0137 4684 6D		LISL	5
0138 4685 70		CLR	ENTREE DIRECTE DE AB (PORT 1)
0139 4686 B1		OUTS	1
013A 4687 A1		INS	1
013B 4688 21 03		NI	3
013C 468A 8D		CM	
013D 468B 84 19		BZ	PST3
013E 468D 8D		CM	
013F 468E 94 06		BNZ	PST1
0140 4690 4C		LR	A.S
0141 4691 1F		INC	
0142 4692 5C		LR	S.A
0143 4693 90 11		BR	PST3
0144 4695 3C	PST1	DS	S
0145 4696 11		LR	H.DC
0146 4697 38		DS	H'B'
0147 4698 3B		DS	H'B'
0148 4699 3B	PST2	DS	H'B'
0149 469A 4B		LR	A.H'B'
014A 469B 21 FB		NI	H'FB'
014B 469D 22 08		DI	H'08'
014C 469F 5B		LR	H'B',A
014D 46A0 10		LR	DC.H
014E 46A1 4C		LR	A.S VISUAL. DE POSB (PORT 0)
014F 46A2 B0		OUTS	0
0150 46A3 90 04		BR	SVG
0151 46A5 11	PST3	LR	H.DC
0152 46A6 90 F2		BR	PST2
0153	*		
0154	*SAUVEGARDE DU POINTEUR DE 'POSITION'		
0155	*		
0156 46A8 66	SVG	LISU	6
0157 46A9 6F		LISL	7
0158 46AA 0E		LR	Q.DC
0159 46AB 03		LR	A.QL
015A 46AC 5E		LR	D.A
015B 46AD 02		LR	A.QU
015C 46AE 5C		LR	S.A

```

015D
015E
015F
0160
0161
0162
0163
0164 46AF 6D
0165 46B0 70
0166 46B1 62
0167 46B2 5C
0168 46B3 63
0169 46B4 5C
016A 46B5 64
016B 46B6 5C
016C 46B7 66
016D 46B8 4C
016E 46B9 50
016F 46BA 21 00
0170 46BC 84 12
0171 46BE 25 40
0172 46C0 84 08
0173 46C2 62
0174 46C3 1A
0175 46C4 28 46 D6
0176 46C7 90 1F
0177 46C9 63
0178 46CA 28 46 D6
0179 46CD 90 19
017A 46CF 64
017B 46D0 28 46 D6
017C 46D3 1B
017D 46D4 90 12
017E
017F
0180
0181
0182
0183 46D6 40
0184 46D7 21 38
0185 46D9 12
0186 46DA 12
0187 46DB 12
0188 46DC 1F
0189 46DD 50
018A 46DE 71
018B 46DF 30
018C 46E0 84 04
018D 46E2 13
018E 46E3 90 FB
018F 46E5 5C
0190 46E6 1C
0191
0192
0193
0194
0195
0196
0197 46E7 66
0198 46E8 68
0199 46E9 4C
019A 46EA 51

```

```

*
*CONVERSION POSB EN UN PARMi N
*=====
*
*REG. UTILISE : 0
*Sous-PROGR. APPELE : BU : CONV. SUR UN OCTET
*
LISL 5
CLR MISE A ZERO
LISU 2
LR S.A DE 0'25'
LISU 3
LR S.A DE 0'35'
LISU 4
LR S.A DE 0'45'
LR A.S
LR 0.A
NI H'00' RECH. DE LA TRANCHE DE NIVEAU
BZ CBUI
CI H'40'
BZ CBU2
LISU 2 EN 0'25'
DI
PI BU
BR APP
CBU2 LISU 3 EN 0'35'
PI BU
BR APP
CBUI LISU 4 EN 0'45'
PI BU
EI
BR APP
*
*Sous-PROGRAMME BU : CONVERSION SUR UN OCTET
*
*REG. UTILISE : 0
*
BU LR A.0
NI H'38'
SR 1
SR 1
SR 1
INC
LR 0.A
LIS 1
BUI DS 0
BZ FBU
SL 1
BR BUI
FBU LR S.A
POP
*
*ARRET PARCOURS PREVU
*=====
*
*REG. UTILISE : 1
*
APP LISU 6 BIT M/D DANS 1
LISL 0
LR A.S
LR 1.A

```

019B 46EB 1A	DI		
019C 46EC 62	LISU	2	
019D 46ED 28 46 FA	PI	DAO	
019E 46F0 63	LISU	3	
019F 46F1 28 46 FA	PI	DAO	
01A0 46F4 64	LISU	4	
01A1 46F5 28 46 FA	PI	DAO	
01A2	*		
01A3 46F8 90 27	BR	SL	
01A4	*		
01A5	*SOUS-PROG. DAO : DEMANDE D'ARRET DANS UN OCTET		
01A6	*	ET RAZ DES DEMANDES	
01A7	*	SI DEM. D'ARRET: BIT 7 DE 0'70' A 1	
01A8	*	REG. UTILISE : 0	
01A9	*		
01AA 46FA 6D	DAO	LISL	5
01AB 46FB 4C		LR	A,5
01AC 46FC 50		LR	0,A PN EN 0
01AD 46FD 68		LISL	0
01AE 46FE FC		NS	5 INTERS. PN-EC EN A
01AF 46FF 94 12		BNZ	RAZ
01B0 4701 41		LR	A,1
01B1 4702 21 80		NI	H'80'
01B2 4704 84 07		BZ	DESC
01B3 4706 69		LISL	1 EN MONTEE
01B4 4707 40		LR	A,0
01B5 4708 FC		NS	5 INTERS. PN-AM
01B6 4709 94 08		BNZ	RAZ
01B7 470B 1C		POP	
01B8 470C 6A	DESC	LISL	2
01B9 470D 40		LR	A,0
01BA 470E FC		NS	5 INTERS. PN-AD
01BB 470F 94 02		BNZ	RAZ
01BC 4711 1C		POP	
01BD 4712 6A	RAZ	LISL	2
01BE 4713 40	RAZI	LR	A,0
01BF 4714 18		COM	
01C0 4715 FC		NS	5
01C1 4716 5E		LR	D,A
01C2 4717 8F FB		BR7	RAZI
01C3	*MISE A 1 DU BIT 7 DE 0'70'		
01C4 4719 67		LISU	7
01C5 471A 68		LISL	0
01C6 471B 4C		LR	A,5
01C7 471C 22 80		OI	H'80'
01C8 471E 5C		LR	S,A
01C9 471F 1C		POP	
01CA	*		
01CB	*ELABORATION DE LA SOMME LOGIQUE ON		
01CC	*=====		
01CD	*		
01CE	*TEST DE SLG		
01CF	*		
01D0 4720 67	SL	LISU	7
01D1 4721 68		LISL	0
01D2 4722 4C		LR	A,5
01D3 4723 21 40		NI	H'40'
01D4 4725 84 11		BZ	SLN SOMME LOG NORMALE
01D5 4727 62		LISU	2
01D6 4728 4C		LR	A,5
01D7 4729 6B		LISL	3
01D8 472A 5C		LR	S,A EC2 EN ONE

01D9 472B 63	LISU	3	
01DA 472C 68	LISL	0	
01DB 472D 4C	LR	A,S	
01DC 472E 6B	LISL	3	
01DD 472F 5C	LR	S,A	ECI EN QNI
01DE 4730 64	LISU	4	
01DF 4731 68	LISL	0	
01E0 4732 4C	LR	A,S	
01E1 4733 6B	LISL	3	
01E2 4734 5C	LR	S,A	EC0 EN QN0
01E3 4735 90 1F	BR	RET	
01E4	*		
01E5 4737 62	SLN LISU	2	
01E6 4738 1A	DI		
01E7 4739 28 47 46	PI	SLO	
01E8 473C 63	LISU	3	
01E9 473D 28 47 46	PI	SLO	
01EA 4740 64	LISU	4	
01EB 4741 28 47 46	PI	SLO	
01EC 4744 90 10	BR	RET	
01ED	*		
01EE	*SOUS-PROG. DE SOMME LOG. SUR 3 OCTETS		
01EF	*	REG. UTILISES : 0,1	
01F0	*		
01F1 4746 68	SLO LISL	0	
01F2 4747 4D	LR	A,I	UNE DONNEE EN A
01F3	*		UNE DONNEE EN S
01F4 4748 EE	XS	D	
01F5 4749 50	LR	0,A	OX RANGE EN 0
01F6 474A 4D	LR	A,I	
01F7 474B FD	NS	I	AND EN A
01F8 474C E0	XS	0	OU EN A
01F9 474D 51	LR	I,A	OU EN I
01FA	*ON RECOMMENCE : UNE DONNEE EN I		
01FB	*	UNE DONNEE EN S	
01FC 474E EC	XS	S	
01FD 474F 50	LR	0,A	
01FE 4750 41	LR	A,I	
01FF 4751 FD	NS	I	
0200 4752 E0	XS	0	
0201 4753 5D	LR	I,A	
0202 4754 1C	POP		
0203	*		
0204	*RETOURNEMENT DE QN ET DE PN		
0205	*=====		
0206	*		
0207 4755 6B	RET LISL	3	RETOURNEMENT DE QN
0208 4756 28 47 5F	PI	RETG	
0209 4759 6D	LISL	5	RETOURNEMENT DE PN
020A 475A 28 47 5F	PI	RETG	
020B 475D 90 2D	BR	DPP	
020C	*		
020D	*SOUS-PROGRAMME DE RETOURNEMENT D'ENSEMBLE DES OCTETS		
020E	*	REG. UTILISE : 2	
020F 475F 08	RETG LR	K,P	
0210 4760 62	LISU	2	
0211 4761 4C	LR	A,S	
0212 4762 52	LR	2,A	0'23' EN 2
0213 4763 64	LISU	4	
0214 4764 4D	LR	A,I	0'43' EN A
0215 4765 62	LISU	2	
0216 4766 51	LR	I,A	

0217	4767	28	47	78	PI	RETO	
0218	476A	64			LISU	4	
0219	476B	42			LR	A.2	
021A	476C	51			LR	I.A	
021B	476D	28	47	78	PI	RETO	
021C	4770	4E			LR	A.D	POUR DECREMENTER S
021D	4771	63			LISU	3	
021E	4772	4D			LR	A.I	
021F	4773	51			LR	I.A	
0220	4774	28	47	78	PI	RETO	
0221	4777	0C			PK		
0222					*		
0223					*SOUS-PROG. DE RETOURNEMENT D'UN OCTET		
0224					* REG. UTILISE : I		
0225					* DONNEE EN I - RESULTAT EN S		
0226					*		
0227	4778	70			RETO	CLR	
0228	4779	5C			LR	S.A	REG. S A 0
0229	477A	78			LIS	8	
022A	477B	50			LR	0.A	REG. 0 A 8
022B	477C	4C			RTI	A.S	
022C	477D	13			SL	I	
022D	477E	5C			LR	S.A	DECAL. G. DE S
022E	477F	41			LR	A.I	
022F	4780	21	01		NI	I	IER. BIT DE I
0230	4782	EC			XS	S	
0231	4783	5C			LR	S.A	DANS S
0232	4784	41			LR	A.I	
0233	4785	12			SR	I	
0234	4786	51			LR	I.A	DECAL. DR. DE I
0235	4787	30			DS	0	
0236	4788	94	F3		BNZ	RTI	
0237	478A	1C			POP		
0238					*		
0239					*DECALAGE A GAUCHE DE PN OU DE PR		
023A					*=====		
023B					*		
023C					*LES RESULTATS SONT MIS DANS LES REGISTRES D'ORIGINE		
023D					*		
023E	478B	66			DPP	LISU	6 TEST DE MO
023F	478C	68				LISL	0
0240	478D	4C				LR	A.S
0241	478E	21	80			NI	H'80'
0242	4790	84	07			BZ	DDS EN DESC.
0243	4792	6D				LISL	5
0244	4793	28	47	9F		PI	DEC DECAL. DE PN
0245	4796	90	2E			BR	CPN
0246					*		
0247	4798	6E			DDS	LISL	6
0248	4799	28	47	9F		PI	DEC DECAL. DE PR
0249	479C	1B				EI	
024A	479D	90	27			BR	CPN
024B					*		
024C					*SOUS-PROGR. DECALAGE		
024D					*		
024E	479F	64			DEC	LISU	4
024F	47A0	4C				LR	A.S
0250	47A1	21	FF			NI	H'FF'
0251	47A3	94	10			BNZ	DECI
0252	47A5	63				LISU	3
0253	47A6	4C				LR	A.S
0254	47A7	21	FF			NI	H'FF'

Address	Hex Data	Label	Comment
02555	47A9 94 11	BNZ	DEC2
02556	47B8 62	LISU	2
02557	47AC 4C	LR	A,5
02558	47AD 13	5L	1
02559	47AE 94 14	BNZ	DEC4
0256A	47B0 20 80	LI	H'80'
0256B	47B2 90 10	BR	DEC4
0256C	47B4 13	5L	1
0256D	47B5 5C	LR	A,5
0256E	47B6 94 0D	BNZ	DEC3
0256F	47B8 63	LISU	3
02570	47B9 90 06	BR	DEC3
02571	47BB 13	5L	1
02572	47BC 5C	LR	A,5
02573	47BD 94 06	BNZ	DEC3
02574	47BF 62	LISU	2
02575	47C0 4C	LR	A,5
02576	47C1 24 01	HI	1
02577	47C3 5C	LR	A,5
02578	47C4 1C	POP	
02579			
0258A			
0258B			
0258C			
0258D			
0258E			
0258F			
02590			
02591			
02592			
02593			
02594			
02595			
02596			
02597			
02598			
02599			
0259A			
0259B			
0259C			
0259D			
0259E			
0259F			
025A0			
025A1			
025A2			
025A3			
025A4			
025A5			
025A6			
025A7			
025A8			
025A9			
025AA			
025AB			
025AC			
025AD			
025AE			
025AF			
025B0			
025B1			
025B2			
025B3			
025B4			
025B5			
025B6			
025B7			
025B8			
025B9			
025BA			
025BB			
025BC			
025BD			
025BE			
025BF			
025C0			
025C1			
025C2			
025C3			
025C4			
025C5			
025C6			
025C7			
025C8			
025C9			
025CA			
025CB			
025CC			
025CD			
025CE			
025CF			
025D0			
025D1			
025D2			
025D3			
025D4			
025D5			
025D6			
025D7			
025D8			
025D9			
025DA			
025DB			
025DC			
025DD			
025DE			
025DF			
025E0			
025E1			
025E2			
025E3			
025E4			
025E5			
025E6			
025E7			
025E8			
025E9			
025EA			
025EB			
025EC			
025ED			
025EE			
025EF			
025F0			
025F1			
025F2			
025F3			
025F4			
025F5			
025F6			
025F7			
025F8			
025F9			
025FA			
025FB			
025FC			
025FD			
025FE			
025FF			
02600			
02601			</


```

02CF      *SAUVEGARDE DU POINTEUR
02D0      *
02D1 4830 67      LISU      7
02D2 4831 6D      LISL      5
02D3 4832 0E      LR        Q,DC
02D4 4833 03      LR        A,QL
02D5 4834 5E      LR        D,A
02D6 4835 02      LR        A,QU
02D7 4836 5C      LR        S,A
02D8      *
02D9      *CONTROLE DES AUXILIAIRES
02DA      *=====
02DB      *
02DC      *MISE EN PLACE DU POINTEUR
02DD      *
02DE 4837 67      LISU      7
02DF 4838 6F      LISL      7
02E0 4839 4E      LR        A,D
02E1 483A 07      LR        QL,A
02E2 483B 4C      LR        A,S
02E3 483C 06      LR        QU,A
02E4 483D 0F      LR        DC,Q
02E5      *
02E6      *APPEL 'EVOLUTION'
02E7      *
02E8 483E 28 50 00      PI      EVOL
02E9      *
02EA      *SAUVEGARDE DU POINTEUR
02EB      *
02EC 4841 1B      EI
02ED 4842 67      LISU      7
02EE 4843 6F      LISL      7
02EF 4844 0E      LR        Q,DC
02F0 4845 03      LR        A,QL
02F1 4846 5E      LR        D,A
02F2 4847 02      LR        A,QU
02F3 4848 5C      LR        S,A
02F4      *
02F5      *TEMPORISATIONS
02F6      *=====
02F7      *
02F8      *RECHERCHE SI DEMANDES TEMPORISATIONS
02F9 4849 67      LISU      7
02FA 484A 69      LISL      1
02FB 484B 7F      LIS      H'F'
02FC 484C FC      NS        S
02FD 484D 84 19      BZ      SORT      PAS DE DEMANDE
02FE      *
02FF      *TRAITEMENT TEMPO
0300      * REG. UTILISES : 0,1
0301      * 4 (DECOMPTEUR)
0302 484F 2B      TIMI      NOP
0303 4850 15      SL        4
0304 4851 50      LR        Q,A      SAUV. BIT-TEMPS DEMANDE
0305 4852 78      LIS      8
0306 4853 51      LR        I,A
0307 4854 40      LR        A,Q
0308 4855 31      TMII      DS      I      CONV. POS BIT-TEMPS EN BN
0309 4856 13      SL        I
030A 4857 94 FD      BNZ      TMII
030B 4859 41      LR        A,I      CHARG. DE DC
030C 485A 07      LR        QL,A
030D 485B 20 46      LI      H'46'
030E 485D 06      LR        QU,A

```



030F 485E 0F	LR	DC,Q	
0310 485F 16	LM		CHARG. DU DECOMPTEUR DU
0311	*		TINI
0312 4860 54	LR	4,A	
0313 4861 70	CLR		RAZ DE 0'71' ET '72'
0314 4862 50	LR	I,A	
0315 4863 50	LR	S,A	
0316 4864 27 13	OUT	H'13'	CHARG. TIMER
0317 4866 18	EI		
0318 4867 29 45 40 SORT	JMP	FVI	
0319	*		
031A	*INTERRUPTIONS TIMER		
031B	*		
031C	*	REG. UTILISES :	4 (DECOMPTEUR)
031D	*		6 ET 7 (SAUVEGARDES)
031E	ORG	H'4100'	
031F 4100 1E	LR	J,W	
0320 4101 34	DS	4	
0321 4102 94 0F	BNZ	ITM	
0322 4104 56	LR	6,A	SAUVEGARDES
0323 4105 0A	LR	A,IS	
0324 4106 57	LR	7,A	
0325 4107 20 FF	LI	H'FF'	ARRET TIMER
0326 4109 27 13	OUT	H'13'	
0327 410B 67	LISU	7	SORTIE
0328 410C 6A	LISL	2	
0329 410D 7F	LIS	H'F'	
032A 410E 50	LR	S,A	
032B 410F 47	LR	A,7	
032C 4110 0B	LR	IS,A	
032D 4111 46	LR	A,6	
032E 4112 1D	ITM	LR	W,J
032F 4113 1B	EI		
0330 4114 1C	POP		
0331	*		
0332	END		
00			

A22 45FD A32 4611 A42 4625 APP 46E7 BU 46D6

C O N C L U S I O N

L'évolution des techniques permet de réaliser des commandes d'ascenseurs plus performantes et plus sûres. Seulement il est nécessaire de mettre au point des méthodes pour utiliser de façon optimale les nouveaux composants et réduire les temps d'étude. Ce travail apporte quelques éléments de solution dont certains ont été appliqués industriellement. Le lecteur de ces lignes empruntera tôt ou tard, en France ou à l'étranger, un ascenseur commandé par Réseaux de Pétri.

En dehors des ascenseurs, nous constatons que les machines industrielles de type logique sont de plus en plus commandées par des systèmes programmés. L'automate classique est souvent utilisé. On commence à employer le microprocesseur qui joue d'abord le rôle de l'automate et qui ensuite, de part son universalité, permet de traiter des points spécifiques, comme la transmission de signaux, la tenue d'un journal de bord, etc..... On aboutit alors à des solutions peu onéreuses en matériel. Pour parfaire ces solutions, le logiciel doit être conçu méthodiquement. On peut donc reprendre la structure proposée au chapitre 5. Une première partie du programme est constituée par le suivi de Réseaux de Pétri au moyen de logiciel standard et une seconde partie par le traitement d'opérations particulières au moyen de logiciel écrit sur mesure.

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

(1) AUTINOR

Notice d'utilisation de la manoeuvre programmée d'ascenseur A100

(2) J.L. BOUSSIN

"Synthèse et analyse des automatismes logiques"

Document EDF - 1977

(3) COMITE EUROPEEN DE NORMALISATION

"Règles de sécurité pour la construction et l'installation des ascenseurs et monte-charges" - 1978

(4) E. DACLIN - M. BLANCHARD

"Synthèse des systèmes logiques"

CEPADUES-Edition - 1976

(5) P. GRESSIN

"Les ascenseurs dans les grands ensembles"

(Revue Générale de l'Electricité - avril 1978)

(6) LEANORD

Notices d'utilisation du microcalculateur PICOLOG 80

(7) M. LILEN

" Du microprocesseur au microordinateur"

Editions Radio - 1976

(8) MOSTEK

"F8 Development system documentation"

(9) J.P. PARSY

"Méthode de décomposition d'un réseau de Pétri pour implantation sur automate programmable"

DEA - Juillet 1978

(10) J.L. POKOSKI

"Software analyses for combinatorial logic"

Revue "Computer design" Juin 1978

(11) J. SESIA

"Ascenseurs et monte-charges dans le bâtiment"

Moniteur des travaux publics

(12) S. THELLIEZ

"Pratique séquentielle et réseaux de Pétri"

Eyrolles - 1978

(13) J.M. TOULOTTE et G. MANESSE

"Programmable controllers and generalized Pétri nets"

MECO 1978 - ATHENES

(14) J.M. TOULOTTE et G. MANESSE

" Contrôleurs programmables et microprocesseurs"

MECO 1977 - ZURICH

(15) G. DELORY et H. GHESTEM

"PROGRAMMED DRIVING SEQUENTIAL MACHINES

BY WHOLE OR FRACTIONAL CONTROL"

Symposium IMACS 1978 - VIENNE