

## VIII . MISE EN OEUVRE

### ETUDE DE CAS N° 3 : TRANSFERT LINEAIRE DE CELLULE

#### 1 - MISE EN SITUATION

##### 1 - 1. Contexte

Une usine produit des pompes à la demande des clients : les séries sont faibles voire à l'unité, les morphologies et les tailles des pièces sont très variées. Les opérations d'usinage sont très diverses. Néanmoins, l'entreprise a mis en place *une gamme limitée de pièces standards regroupées en 10 familles* : elles sont personnalisées lors de leur usinage.

##### 1 - 2. Description des palettes

En vue de leur usinage, les pièces (corps, couvercles, etc. ) sont maintenues sur des palettes standards adaptées à toutes les machines de la cellule ainsi qu'au système de transitique.

Chaque palette est équipée d'une étiquette magnétique. Des informations peuvent y être inscrites lorsqu'elles passent devant une tête d'enregistrement : numéro de pièce transportée par la palette, horodatage des événements, nom du client et numéro de la commande, etc. Ces informations peuvent être lues lors du passage de l'étiquette devant une tête de lecture, particulièrement pour déterminer le numéro du programme d'usinage à lancer au moment où une palette entre dans une machine, ou bien pour sélectionner la machine de destination lors du transport d'une palette comme on le verra plus loin.

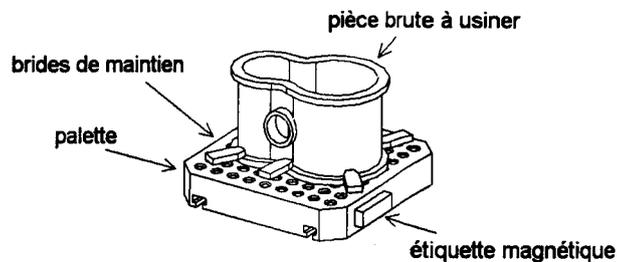


figure C-16 : exemple de palette

##### 1 - 3. Composition de la cellule

La figure C-17 montre schématiquement la composition de la cellule : 6 centres d'usinage autonomes équipés chacun d'un magasin d'outils pré-réglés et d'un système d'échange de

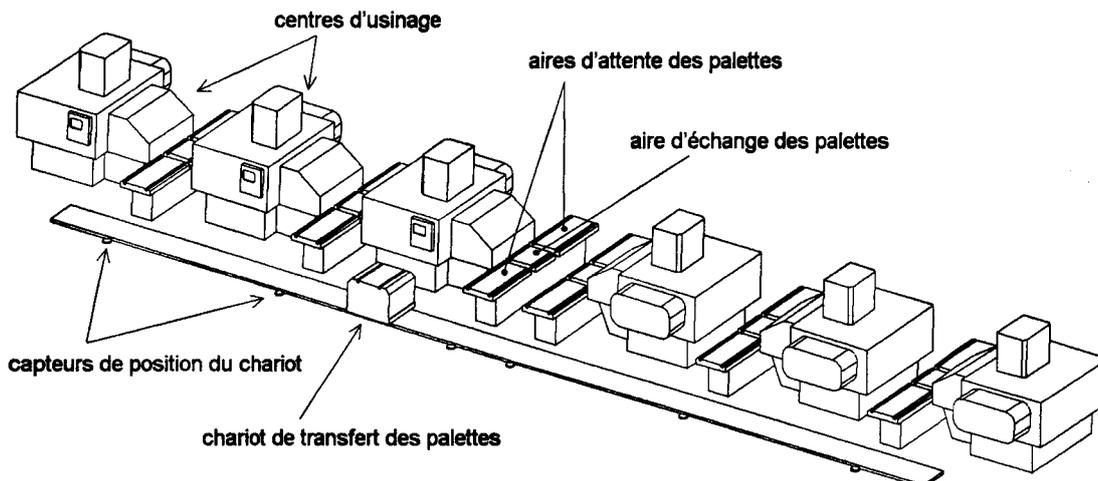


figure C-17

palettes, un chariot linéaire pour le transport des palettes et des aires d'attente pour les palettes en amont et en aval de chaque machine.

## 2 - FONCTIONNEMENT DE LA CELLULE

### 2 - 1. Flux physique

La cellule est scindée en deux groupes de 3 machines :

- les palettes sont préparées en amont du premier groupe de machines,
- une première phase d'usinage est réalisée sur l'une des machines du premier groupe (opérations de type fraisage, perçage... ne nécessitant que des outils standards),
- les palettes sont ensuite traitées par l'une des machines du deuxième groupe (opérations spéciales de type fraisage profond, alésage... nécessitant des outils spéciaux),
- elles sont ensuite dirigées vers l'atelier suivant.

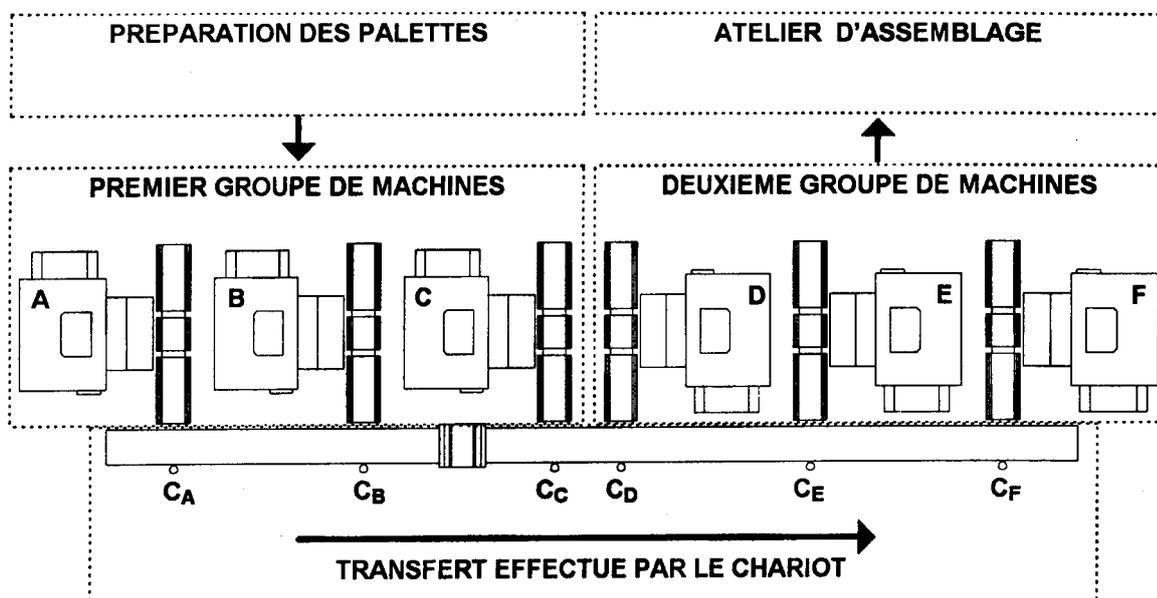


figure C-18 : cellule vue de dessus

### 2 - 2. Contraintes liées aux méthodes de fabrication

#### - En amont du premier groupe de machines

L'ordre dans lequel les opérateurs préparent les palettes est déterminé par la gestion de production (commandes des clients, disponibilité des bruts, gestion des stocks, etc.) Les palettes sont orientées indifféremment vers l'une des machines A, B ou C selon leur taux d'occupation. Plusieurs palettes peuvent être en attente devant chaque machine.

#### - Le premier groupe de machines

Ces machines sont toutes capables de réaliser la première phase de tous les types de pièces. A chaque nouveau cycle d'usinage, le numéro du programme est automatiquement sélectionné par la machine grâce aux informations contenues dans les étiquettes magnétiques.

#### - En aval du premier groupe de machines

Une palette qui quitte une machine après usinage est temporairement stockée sur l'aire d'attente. Jusqu'à 3 palettes par machine peuvent ainsi être mises en attente avant leur transfert vers le deuxième groupe de machines.

### - Le chariot

La fonction du chariot est de transporter les palettes des machines A-B-C vers les machines D-E-F. Il ne peut transporter qu'une seule palette à la fois. Il comporte également un dispositif de lecture-écriture des étiquettes magnétiques.

*L'ordre dans lequel les palettes doivent être prises en compte par le chariot doit permettre d'équilibrer le taux de charge des machines A, B et C et éviter en particulier la saturation des stocks en aval de ces machines.*

### - En amont du deuxième groupe de machines

Un stock maximum de 3 palettes par machine est prévu. Les machines D-E-F sont équipées pour réaliser la deuxième phase d'usinage, plus délicate : elles sont davantage spécialisées et les outils sont spécifiques à des opérations bien déterminées. Ainsi, *la sélection de la machine est faite individuellement pour chaque type de pièce par le Bureau des Méthodes afin de garantir l'adéquation entre les moyens utilisés et les opérations à réaliser.*

### - Le deuxième groupe de machines

Comme pour les machines du premier groupe, les étiquettes magnétiques permettent le lancement automatique du programme adéquat à chaque nouveau cycle d'usinage.

### - En aval du deuxième groupe de machines

Les palettes sont orientées vers l'atelier suivant dès que l'usinage est terminé.

## 3 - PRINCIPES TECHNIQUES

### 3 - 1. Architecture informatique

#### a) Vue d'ensemble de la cellule

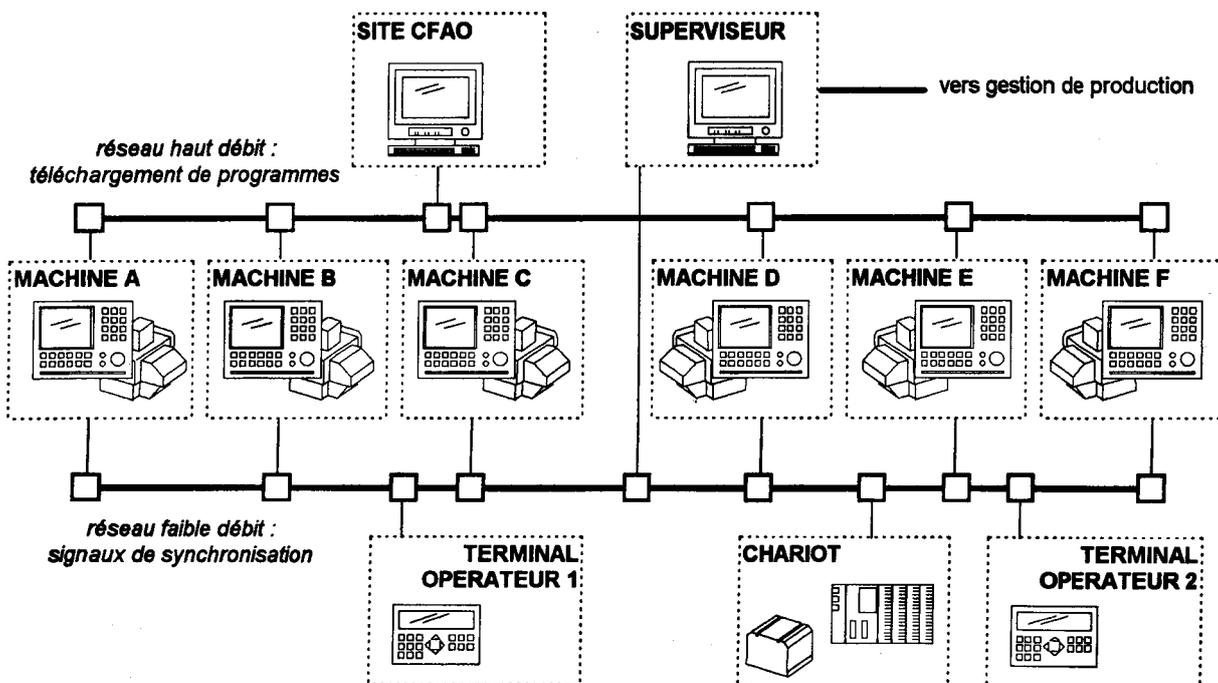


figure C-19

- Les échanges des palettes sont gérés par chaque commande de machine (directeur de commande numérique et automate intégré à la machine),
- les chargements et déchargements des palettes sur le chariot ainsi que les déplacements de celui-ci sont commandés par un automate programmable,

- deux terminaux répartis à des points stratégiques de la cellule permettent certaines fonctions de pilotage,
- un superviseur, également relié à la gestion de production, permet de piloter et de contrôler l'ensemble de la cellule,
- les machines dialoguent en temps réel avec le site de création des programmes d'usinage (FAO) afin de rafraîchir leur mémoire à chaque palette.

#### b) Interface entre les machines et le chariot

Grâce à l'interconnexion de toutes les commandes par le réseau, les machines A, B et C peuvent émettre chacune un code spécifique dès la fin des cycles. Ces informations permettront au chariot de gérer les stocks en aval de ces machines. La taille de ces informations est limitée du fait qu'il s'agit de signaux de synchronisation.

### 4 - ETUDE DU FONCTIONNEMENT SEQUENTIEL

Le cas étudié dans ce chapitre a spécialement été sélectionné en raison de la simplicité de l'analyse des fonctions et de la recherche des séquences qui constituent les deux premiers volets de la méthode, étudiés aux chapitres V et VI. Ces points seront donc traités sans difficulté pour aborder rapidement les éléments de base qui permettent à un système d'évoluer d'une manière flexible.

L'objet de ce livre limite l'étude de cette cellule à la seule gestion des déplacements du chariot en vue de la programmation de son automate. Il sera cependant nécessaire de prendre en compte certains éléments de son environnement.

#### 4 - 1. Situation initiale

La fonction principale du chariot étant de faire transiter les palettes du premier vers le deuxième groupe de machines, on peut considérer qu'un cycle est terminé lorsque le transit est effectué. La situation initiale est donc la suivante :

- chariot en position devant l'une des machines D, E ou F,
- chariot vide.

#### 4 - 2. Analyse des fonctions

En référence au chapitre V, l'inventaire des fonctions à réaliser par le chariot est le suivant :

transporter
identifier
charger
décharger
préparer

*figure C-20*

##### - Charger et décharger

Ces fonctions sont assurées par un mécanisme unique, solidaire du chariot. Le chargement s'effectue à la sortie des machines A-B-C et le déchargement aux stocks d'entrée des machines D-E-F. Ce mécanisme n'étant pas défini dans le cadre de cette étude, les séquences correspondantes seront traitées par des grafjets de tâche notés *charger* et *décharger*.

##### - Préparer

Il s'agit du déplacement du chariot à vide vers la gauche en vue de se présenter devant l'une des machines du premier groupe.

Il est important de noter que la destination du chariot (la machine devant laquelle il doit se présenter) ne fait pas partie de la fonction, mais que l'obtention de cette condition constitue l'événement qui produira la fin du déplacement.

Le comportement réel (accélération, décélération, positionnement précis) n'est pas concerné par cette étude, aussi cette action sera-t-elle directement notée GAUCHE.

**- Transporter**

C'est le déplacement vers la droite du chariot : celui-ci transporte une palette. De même que pour préparer, l'endroit de destination n'est pas à inclure dans la fonction. Et également par soucis de l'essentiel, cette action sera notée DROITE.

**- Identifier**

Lors du chargement d'une palette sur le chariot, l'étiquette magnétique est lue. Ceci permet en particulier de reconnaître le type de pièce en présence. On rappelle que les pièces ont été définies en un nombre limité de familles et qu'une personnalisation plus poussée est réalisée au moment de l'usinage (§ 1 - 1). En conséquence, seule l'appartenance à telle ou telle famille est déterminante pour sélectionner la machine de destination du chariot. Considérons pour l'instant cette fonction comme une tâche subalterne.

**4 - 3. Séquence**

Le chariot effectue simplement des allers-retours entre les deux groupes de machines. Partant de la situation initiale, l'enchaînement des fonctions est purement linéaire dans ce cas. L'épure du grafcet de commande est immédiate (figure C-21). Les réceptivités t2-3 et t5-0 sont respectivement les comptes-rendus de la fin des tâches charger et décharger. Il reste à élaborer les réceptivités t0-1, t1-2, t3-4 et t4-5.

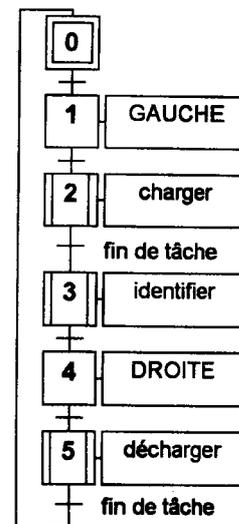


figure C-21

**5 - RECEPTIVITES**

**5 - 1. Chargement au premier groupe de machines**

La productivité de la cellule est garantie si les stocks en aval des machines n'arrivent pas à saturation. Un problème similaire a été traité par une gestion de compteurs au chapitre VI (cellule double). On propose ici un cahier des charges plus serré en précisant que l'ordre dans lequel les palettes sont évacuées des machines soit également celui dans lequel elles sont prises en charge par le chariot.

**a ) Structure de données**

Une structure de données de type file d'attente permet de satisfaire à cette contrainte : il s'agit d'empiler les messages de fin d'usinage émis par les machines au fur et à mesure de leur réception et de les dépiler dans le même ordre à chaque nouveau cycle de transit. La lecture du code courant indique la machine à décharger. Dans l'exemple de la figure C-22, les machines ont évacué des palettes dans l'ordre C → A → A → B.

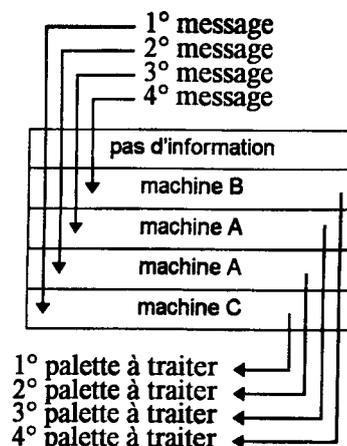


figure C-22

### b) Evolution de la file d'attente

Si la file d'attente est dans l'état de la figure C-22 à la mise en service du chariot, un cycle de transit pour la palette issue de la machine C démarre. Au cycle suivant, la première palette en attente devant la machine A devra être traitée : comme il est inutile de conserver l'information *machine C* et pour éviter la saturation de la file d'attente, celle-ci se décale entièrement d'un pas. Ainsi, à chaque cycle, c'est la même adresse qui détermine l'endroit où charger une palette (le pointeur est fixe). La figure C-23 montre l'état de la file d'attente dans ces 2 situations successives.

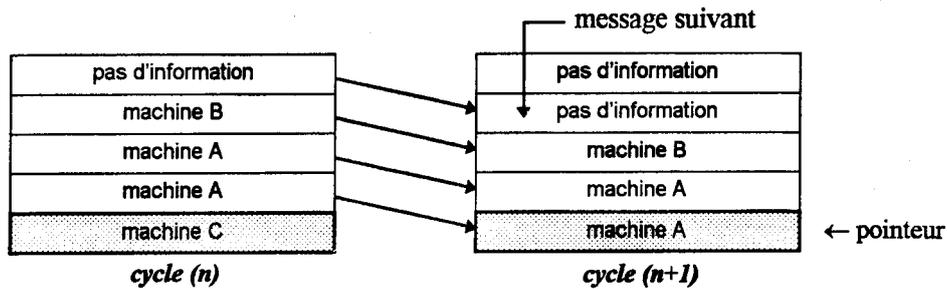


figure C-23

Le décalage de la file d'attente constitue une nouvelle fonction qui peut être réalisée :

- soit en intercalant une étape supplémentaire dans le grafcet,
- soit en la combinant avec une autre action.

Il suffit de veiller à ce que l'information adéquate soit disponible lorsque le chariot exécute la fonction *préparer* et qu'elle soit ensuite régénérée pour le cycle suivant.

Par ailleurs, l'une des machines peut être amenée à évacuer une nouvelle palette à n'importe quel instant : le message qui en résulte est alors empilé.

### c) Démarrage et positionnement du chariot

- En début de campagne, la file d'attente ne contient aucune information de sorte que le chariot conserve son état de repos.
- A condition que le chariot soit en service, un quelconque message présent au niveau du pointeur permet d'engager un cycle (évolution  $X_0 \rightarrow X_1$ ) :

la réceptivité  $t_0-1$  est donc : « pointeur » différent de « pas d'information »

- Pendant le déplacement vers la gauche, le premier capteur rencontré est  $C_C$  (voir la figure page 132). Si et seulement si le pointeur indique « machine C », alors le chariot doit s'arrêter suite à l'évolution du grafcet  $X_1 \rightarrow X_2$ . Sinon le déplacement vers la gauche doit se poursuivre ( $X_1$  reste actif). Une situation similaire se produit en  $C_B$  puis en  $C_A$ . La condition d'arrêt du chariot (conséquence de l'évolution du grafcet) s'écrit sous forme logique de la manière suivante :

réceptivité  $t_1-2$  :  
 OU [ (« pointeur » = « machine A » ) ET « capteur  $C_A$  » ]  
 OU [ (« pointeur » = « machine B » ) ET « capteur  $C_B$  » ]  
 OU [ (« pointeur » = « machine C » ) ET « capteur  $C_C$  » ]

## 5 - 2. Déchargement au deuxième groupe de machines

On rappelle que chaque famille de pièces est orientée individuellement vers l'une des machines D, E ou F en fonction des gammes opératoires élaborées par le Bureau des Méthodes. Il est impératif de mettre à disposition des agents de production une interface qui permet la saisie des gammes sans risque d'erreur.

### a) Dialogue entre l'opérateur et la cellule

#### - Terminaux programmables (voir aussi la figure C-19)

Une première façon d'instruire la cellule des gammes opératoires est offerte par les terminaux dont les touches de fonction et la fenêtre de visualisation sont programmables.

D'autres fonctions de pilotage, non abordées dans cette étude, leur sont également assignées : réglage des paramètres d'asservissement du chariot, validation ou inhibition de chaque machine, etc.

Un système de navigation par arborescence permet d'accéder aux paramètres comme illustré dans l'exemple de la figure C-24 : la manipulation décrite aura comme effet d'orienter les pièces de type 03 vers la machine E lorsque celles-ci se présentent.

Par ailleurs, un système de mots de passe à plusieurs niveaux permet l'accès sélectif aux différents menus : un agent de maintenance pourra par exemple régler les paramètres d'asservissement à l'exclusion de toute autre intervention, le paramétrage des gammes étant réservé aux agents de production.

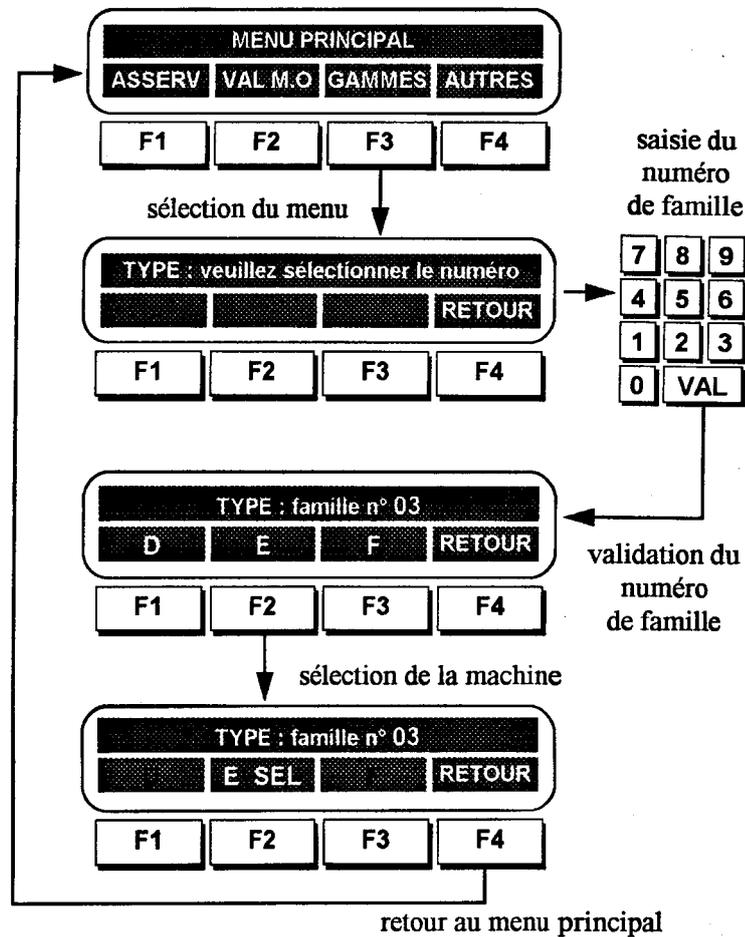


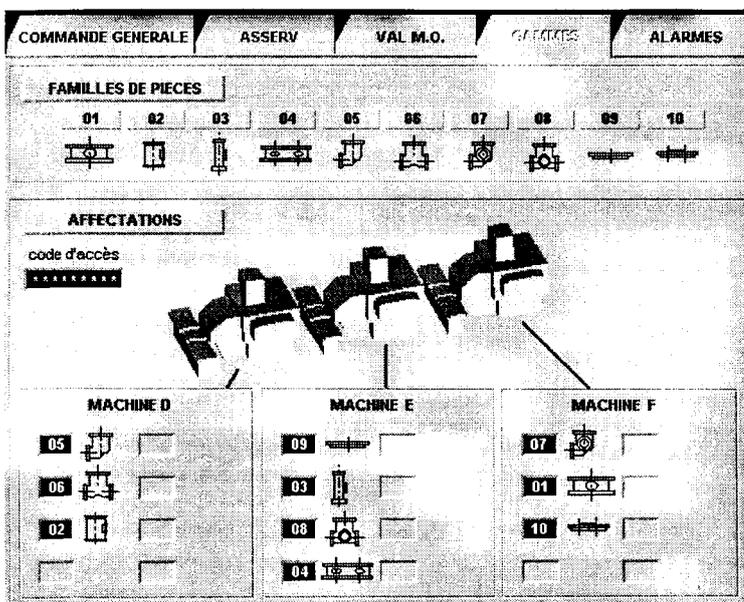
figure C-24

#### - Supervision

Le réglage des affectations pièces-machines peut également se faire grâce au superviseur. Il offre davantage de convivialité et une meilleure appréciation d'ensemble :

- des objets graphiques rendent la reconnaissance des familles et des machines plus aisée,
- l'affectation se fait simplement en désignant l'une après l'autre : la famille (dans la partie

figure C-25



supérieure de l'écran) puis la machine,

- l'écran affiche pour chaque machine toutes les pièces qui lui sont affectées (partie inférieure de l'écran),
- l'édition se fait entièrement sur une seule page, ce qui évite la recherche dans des menus.

## b) Structure de données

Les réglages décrits ci-dessus doivent évidemment être transmis à l'automate de commande du chariot par l'intermédiaire du réseau. Il est nécessaire de structurer les données correspondantes aussi bien du côté des terminaux opérateurs et du superviseur que du côté de l'automate. La nécessité de cohérence entre ces différents partenaires a été démontrée au chapitre VII : cette étude se limite dorénavant au seul automate.

### - Mémoire initiale

Le type de structure de données adapté au cas présent est la *recette*. Il s'agit certes d'une recette limitée à un tableau de 10 x 1 entrées puisque *l'information source* est le numéro de famille délivré par l'étiquette magnétique lors du chargement de la palette et que *l'information calculée* est le numéro de la machine associée.

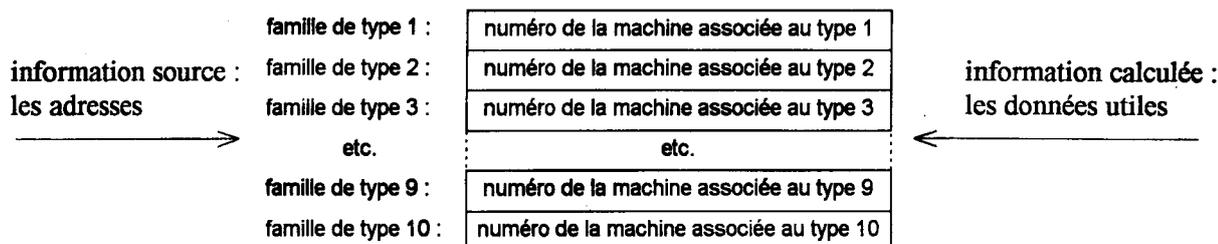


figure C-26

Les numéros de famille constituent les 10 *adresses* de la recette : elles permettent de pointer les *données utiles* à chaque cycle de transit. Les données utiles définissent l'endroit d'arrêt du chariot quand il se déplace vers la droite (fonction *transporter*) : lorsqu'il y a correspondance entre le numéro de machine et le numéro du capteur où se trouve le chariot, celui-ci doit s'arrêter (le grafctet devant donc évoluer X4 → X5). Il en résulte que la réceptivité t4-5 s'écrit sous forme logique de la manière suivante :

réceptivité t4-5 :    [[« donnée pointée » = « machine D »] ET « capteur C<sub>D</sub> »]  
                           OU    [[« donnée pointée » = « machine E »] ET « capteur C<sub>E</sub> »]  
                           OU    [[« donnée pointée » = « machine F »] ET « capteur C<sub>F</sub> »]

Cette expression est incomplète car la donnée pointée a une adresse variable qui dépend du type de pièce.

### - Mémoire de travail

La mise à jour de la recette se fait lorsque l'opérateur valide les modifications apportées aux affectations. Or cette validation peut intervenir à n'importe quel moment du cycle. Supposons que la machine E soit la destination de la palette à transférer : la fonction *transporter* est lancée et le chariot se déplace vers la droite. Arrivé en position C<sub>D</sub>, la réceptivité proposée ci-dessus est évidemment fautive, ce qui implique que le grafctet n'évolue pas et que le chariot poursuit son déplacement. C'est bien le but recherché. Mais si la recette est alors mise à jour et qu'elle prévoit maintenant la machine C pour la palette en cours, cette réceptivité ne pourra plus être vraie. En conséquence, le grafctet n'évoluera plus. Le chariot se perdra en fin de course droite et le système sera bloqué. Il est donc

nécessaire de stabiliser la fonction *transporter* dès son activation. Pour remédier à cela on prévoit une mémoire supplémentaire, appelée *mémoire de travail*. Le rafraîchissement de ce registre constitue une fonction supplémentaire à ajouter sur le grafcet immédiatement à la suite de *identifier*. La stabilité et la cohérence du cycle sont ainsi garanties une fois la palette prise en charge par le chariot : les modifications apportées à la recette seront réellement effectives au cycle de transit suivant.

Par la même occasion, la donnée pointée est maintenant une adresse invariable. La réceptivité t4-5 est modifiée comme suit :

réceptivité t4-5 : [(« mém. travail » = « machine D ») ET « capteur C<sub>D</sub> »]  
 OU [(« mém. travail » = « machine E ») ET « capteur C<sub>E</sub> »]  
 OU [(« mém. travail » = « machine F ») ET « capteur C<sub>F</sub> »]

## 6 - ELEMENTS DE PROGRAMMATION

### 6 - 1. Format des données

Une phase essentielle dans la préparation du grafcet au niveau programmation consiste à procéder au choix des formats des données et de leurs adresses.

#### - Numéro de famille

Ce numéro variant de 1 à 10, un octet est suffisant. On peut également utiliser un mot de 16 bits selon la préférence de l'utilisateur ou les spécificités de l'automate.

Step5    P17  
 numéro de famille : MB30    %MW30    type

*figure C-27*

#### - File d'attente

- Le nombre total de palettes qui peuvent physiquement être stockées en aval des machines A-B-C détermine la taille de la file d'attente : 9 positions sont nécessaires et suffisantes.
- Le nombre de machines et le type de codage retenu déterminent le format minimal des positions. Le choix doit ensuite se porter sur une suite d'octets, de mots ou de double-mots en fonction essentiellement des plages d'utilisation des paramètres (figure C-28-a).

	Step5	P17		Step5	P17		
position 9 :	MB58	%MW58		famille 1 :	MB61	%MW61	
position 8 :	MB57	%MW57		famille 2 :	MB62	%MW62	
position 7 :	MB56	%MW56		famille 3 :	MB63	%MW63	
position 6 :	MB55	%MW55		famille 4 :	MB64	%MW64	
position 5 :	MB54	%MW54		famille 5 :	MB65	%MW65	
position 4 :	MB53	%MW53		famille 6 :	MB66	%MW66	
position 3 :	MB52	%MW52		famille 7 :	MB67	%MW67	
position 2 :	MB51	%MW51		famille 8 :	MB68	%MW68	
pointeur :	MB50	%MW50	machine_appel	famille 9 :	MB69	%MW69	
				famille 10 :	MB70	%MW70	
				mémoire travail :	MB80	%MW80	machine_dest

- a - (file d'attente)

- b - (recette)

*figure C-28*

**- Recette**

La taille du tableau a déjà été définie. Bien que ce ne soit pas obligatoire, on verra qu'il est intéressant, de même que pour la file d'attente, de choisir des adresses consécutives. La figure C-28-b propose des adresses pour les deux langages.

**- Assignment de symboles**

Afin de faciliter l'écriture et la maintenance des programmes, la plupart des logiciels de programmation permettent l'utilisation de symboles associés aux adresses : on procède en général par configuration de tables d'assignation. Dans le cas étudié, les symboles qui correspondent aux adresses indicées 30, 50 et 80 en font partie.

machine	code
aucune	0
A	1
B	2
C	3
D	11
E	12
F	13

**- Codification des machines**

Les codes associés aux machines sont exprimés selon le tableau ci-contre. Par ailleurs, les logiciels des terminaux opérateurs et du superviseur doivent garantir que toutes les positions de la recette ont bien été saisies avant de permettre le démarrage de la cellule.

**6 - 2. Grafset au niveau programmation**

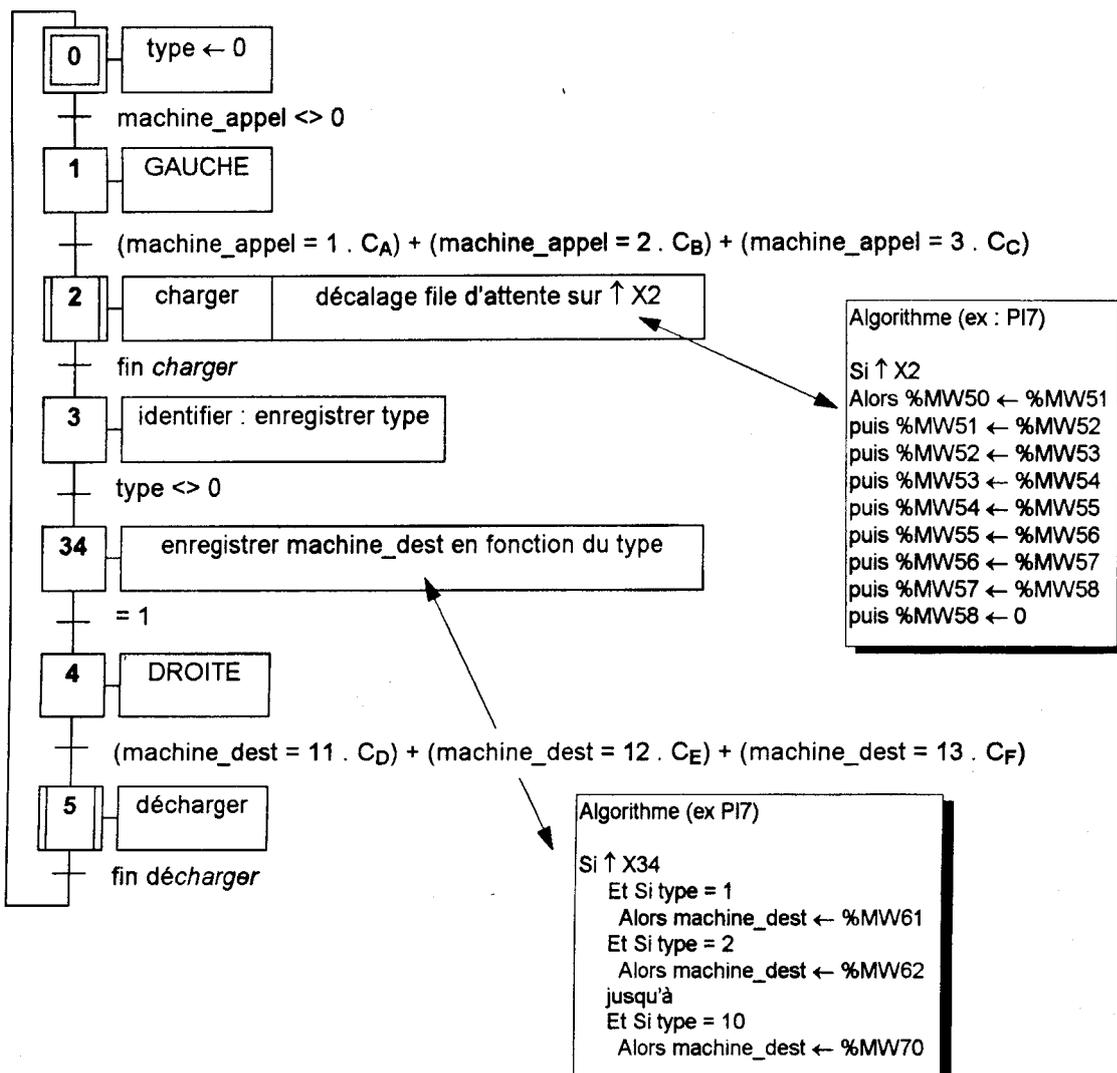


figure C-29

## 6 - 3. File d'attente

### a) Décalage

#### - Unicité du décalage

Lors de la programmation du décalage, il faut bien veiller à ce qu'il ne soit effectué qu'une seule fois pour que son évolution ne se fasse que d'un seul pas.

Selon l'automate, la technique sera différente :

- En Step5, qui n'intègre pas les fronts en tant que fonctions de base, on peut par exemple activer un bit dès la fin du décalage et n'autoriser celui-ci que si le bit est désactivé. Cette désactivation sera programmée sur X0 par exemple.
- En P17-Micro, il suffit de programmer le décalage sur front montant de l'étape X2 grâce à la fonction *PI* : *action à l'activation*.

#### - Remise à zéro de la file d'attente

La mise à zéro systématique de %MW58 permet la purge progressive de la file d'attente lorsque toutes les positions étaient utilisées avant le décalage.

#### - Sens de programmation

Si par inadvertance l'ordre de transfert des variables était inversé lors de la programmation (%MW58 ← 0, puis %MW57 ← %MW58, etc.) la file d'attente serait immédiatement vidée de tout son contenu par copies successives de la valeur 0 dans toutes les adresses.

#### - Utilisation d'instructions de base

Les opérations de transfert de variables conduisent à une liste d'instructions de longueur proportionnelle à la taille de la file d'attente.

Ci-contre un exemple en Step5 : les instructions L (load) et T (transfer) dupliquent successivement le contenu de l'adresse *n* à l'adresse *n-1*.

#### - Utilisation d'instructions avancées

Les instructions avancées sont d'utilisation plus aisée.

L'exemple ci-dessous est écrit en P17-Micro : ROR\_ARW est l'instruction de décalage dans le sens souhaité, 1 est le pas du décalage, %MW50 est l'adresse de début (les numéros d'adresses doivent se suivre) et 9 est la taille du registre.

```
L MB51
T MB50
```

```
L MB52
T MB51
```

```
L MB53
T MB52
```

etc

```
L MB57
T MB58
```

\*\*\*

```
L KF0
T MB58
```

```
ROR_ARW (1,%MW50:9)
%MW58 := 0
```

*exemple C-30*

### b) Ajout d'une nouvelle valeur

Il s'agit d'enregistrer le message émis par l'une des machines. La position à laquelle doit être inscrite cette information est la première qui est à zéro en partant du pointeur.

#### - Utilisation d'instructions de base

*Attention, bien que cette question puisse être traitée par Grafcet, il est conseillé de tracer un organigramme.* En effet, l'opération à réaliser est un calcul et il vaut mieux réserver le Grafcet à sa vocation première : l'exécution du programme sera beaucoup plus rapide.

L'algorithme de recherche est lancé sur réception d'un message émis par une machine : des tests successifs déterminent la première adresse rencontrée dont le contenu est 0 pour y inscrire le numéro de machine. La file d'attente ne peut déjà être saturée puisque dans ce cas les trois stocks seraient également tous saturés, ce qui implique qu'aucune machine ne serait en mesure d'évacuer une palette, et *à fortiori* d'émettre un message.

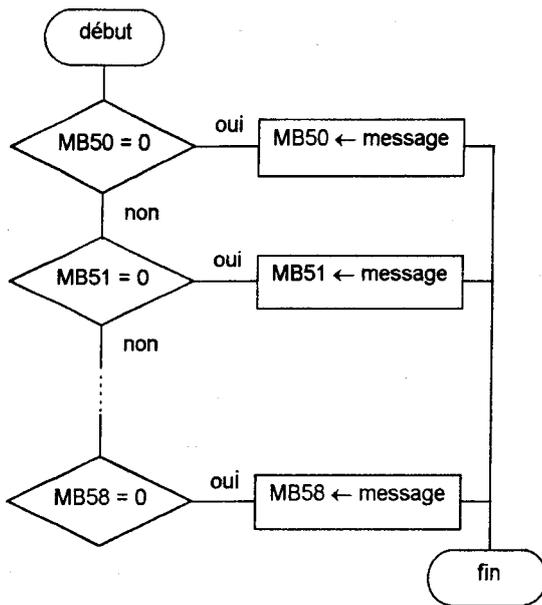
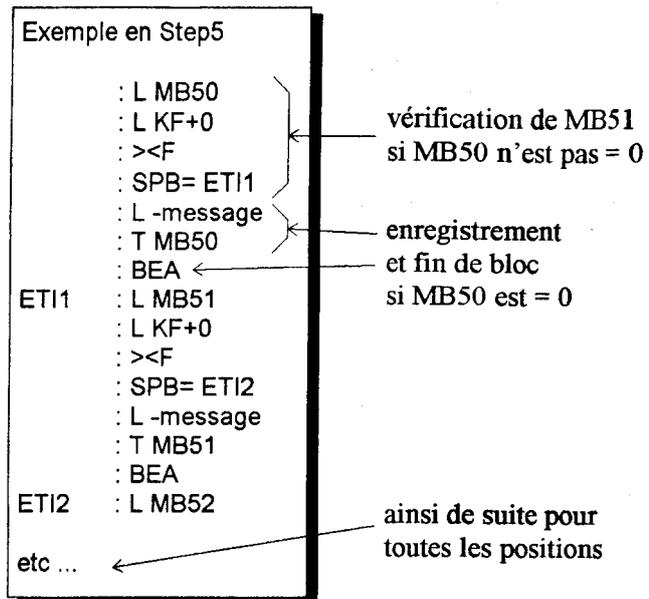


figure C-31

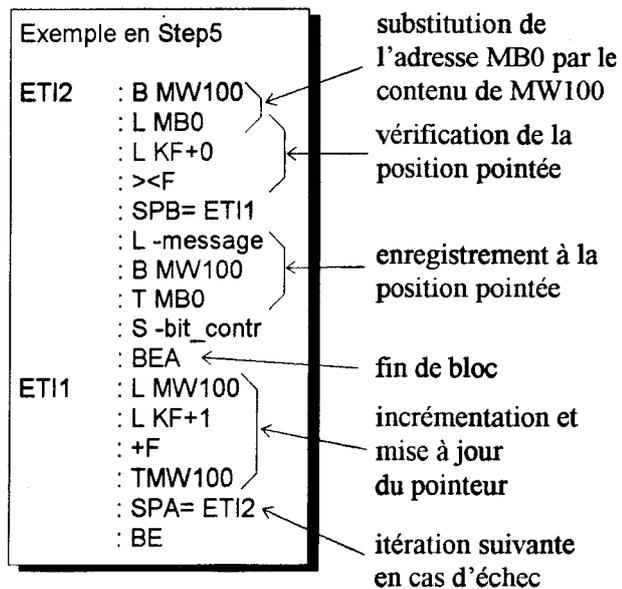


**- Utilisation d'instructions avancées**

• L'instruction de substitution B permet en langage Step5 de réduire considérablement la taille du programme, surtout si la file d'attente comporte un grand nombre de positions. Une boucle itérative scrute les différentes positions les unes après les autres et *s'interrompt naturellement* quand le message a été enregistré.

*note 1* : le pointeur MW100 doit être positionné à 50 avant chaque exécution de ce bloc de programme.

*note 2* : le bit de contrôle -bit\_contr est positionné à *un* par l'instruction S et est mis à *zéro* lorsqu'apparaît un nouveau message. Ce bloc n'est appelé que si ce bit = 0 pour éviter l'empilement multiple du message.



• Les langages normalisés de type PL7-Micro offrent des instructions plus compactes :

Exemple en PI7-Micro

```

%MW10 := FIND_EQW(%MW50:9,%KW27)
%MW50[%MW10] := message
  
```

Dans cet exemple, la première ligne opère la recherche et l'adresse calculée est stockée dans le mot %MW10. FIND\_EQW réalise la recherche de la première position égale à une valeur donnée, %MW50 est le début du tableau de mots et 9 sa longueur, %KW27 est une constante quelconque qui aura préalablement été configurée à 0, qui est l'objet de la recherche. La deuxième ligne enregistre le message dans le mot d'adresse 50 augmenté du contenu de %MW10.

La deuxième ligne enregistre le message dans le mot d'adresse 50 augmenté du contenu de %MW10.

• Des registres spéciaux peuvent être utilisés, comme par exemple le bloc fonction %R intégré à PI7-Micro. Les différents paramètres sont réglés par configuration (mode de fonctionnement, variables d'évolution...) la programmation étant alors relativement limitée.

Tous ces programmes traduisent chacun à sa manière l'algorithme de la figure C-31. Ils doivent être appelés à partir d'un programme hiérarchiquement supérieur qui détecte la réception d'un nouveau message et qui doit s'assurer que cet enregistrement ne s'effectue qu'une seule fois.

## 6 - 4. Recette

### a ) Mise à jour de la recette

Le rafraîchissement de la recette n'est pas synchronisé avec le cycle du chariot, aussi cette fonction n'apparaît-elle pas sur le grafcet. Un sous-programme exécuté en continu copie dans la recette les informations reçues des terminaux et du superviseur par l'intermédiaire du réseau.

### b ) Mise à jour de la mémoire de travail

#### - Utilisation d'instructions de base

« La variable *machine\_dest* prend la valeur du mot pointé par la variable *type*, qui dépend du numéro de famille ».

La fonction ainsi définie peut se traduire de différentes manières dont voici deux exemples :

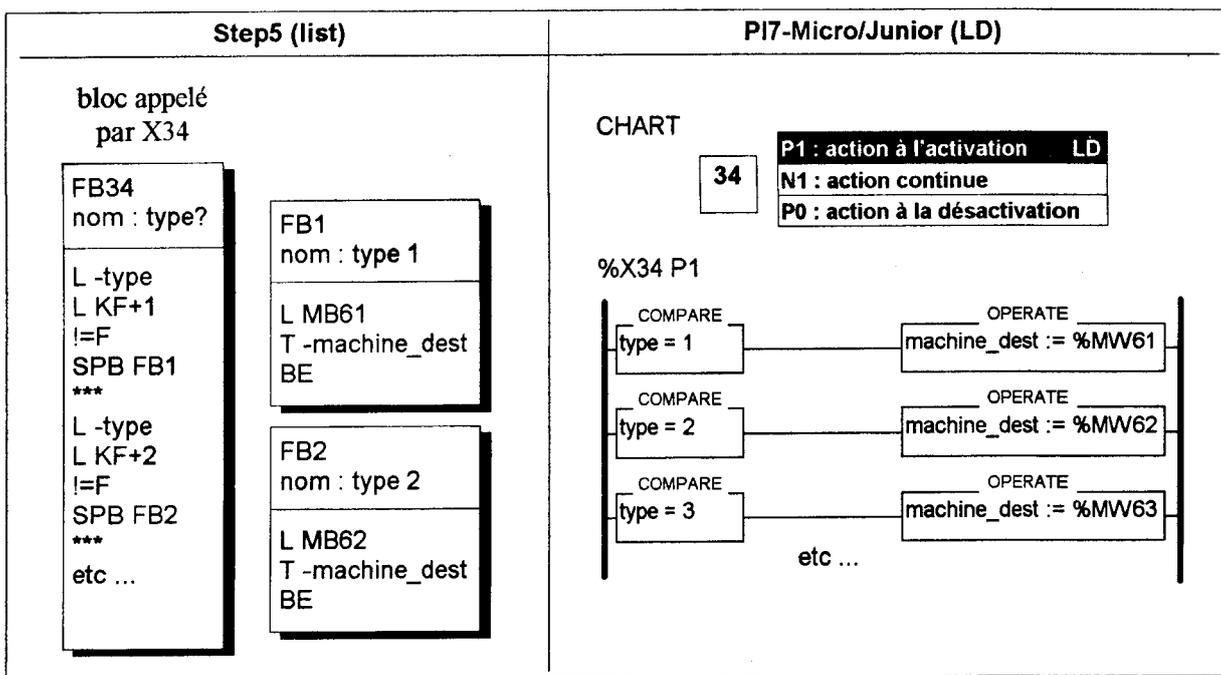


figure C-32

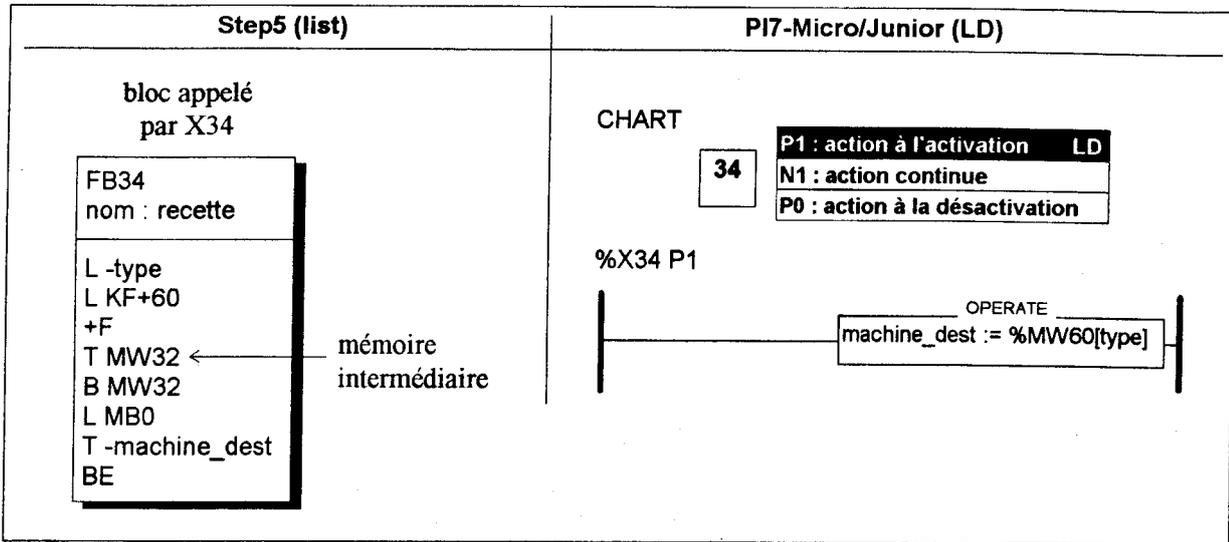
Cette programmation est d'abord très simple et le résultat est très explicite mais s'avère trop lourde lorsque le nombre de données à gérer est important.

#### - Utilisation d'instructions avancées

La fonction à réaliser peut être exprimée d'une autre façon : « l'opération consiste à affecter à la variable *machine\_dest* la valeur du mot dont l'adresse correspond au numéro de famille augmenté de 60 (famille 1  $\leftrightarrow$  adresse 61) ». Cette manière de définir la fonction se traduit dans les programmes de la figure C-33.

Notons que les adresses de la recette doivent être consécutives. Contrairement à la gestion de la file d'attente, il n'y a aucun inconvénient à ce que la duplication de la donnée utile dans la mémoire de travail se fasse plusieurs fois à chaque appel de cette fonction.

Grâce aux instructions avancées, le nombre d'instructions est limité, et ce quelle que soit la taille du tableau de données : les programmes de la figure C-33 sont directement exploitables si le nombre de familles est augmenté, même de manière importante. Dans d'autres cas, seules des modifications mineures seraient alors à apporter au programme (modification des valeurs numériques 9 et 58 dans l'exemple C-30 page 141).



*figure C-33*