Chapitre 6
LE GRAFCET ET SA MISE EN ŒUVRE

Pour des systèmes de complexité modérée, les méthodes de phases et d'étapes données au chapitre 4 ont l'avantage de minimiser le nombre d'états du système et de simplifier par conséquent le circuit ou le programme de son organe de commande. Cependant ces méthodes deviennent lourdes et difficilement interprétables quand il s'agit de systèmes industriels complexes comportant plusieurs sous-systèmes interconnectés dont le comportement de l'un dépend de l'état des autres. Le GRAFCET (Graphe Fonctionnel de Commande à Étapes et Transitions) est un moyen graphique clair permettant de décrire sans ambiguïté le fonctionnement de tels systèmes. Vu les prix actuellement modestes des circuits intégrés comparés au coût du travail humain, on donne dans la mise en œuvre du grafcet moins d'importance à l'économie en matériel qu'à la facilité de conception et de réparation.

Nous commençons par définir les symboles du grafcet et donner les règles d'évolution à travers ce graphe. Ces notions seront ensuite appliquées à deux systèmes industriels types puis nous montrerons la facilité et la clarté qu'apporte le grafcet à la réalisation câblée de l'organe de commande. Pour sa réalisation programmée, nous consacrerons la deuxième partie du chapitre aux principes de fonctionnement et de programmation d'un automate programmable industriel (PLC) accompagnés d'exemples de programmation sur cet automate.

6-1 DÉFINITIONS ET RÈGLES

Le grafcet est un diagramme qui montre, pour toute suite de vecteurs d'entrée possible, la séquence de *situations* par lesquelles passe un système séquentiel. Une situation est constituée d'une ou de plusieurs *étapes* de fonctionnement E_i durant chacune le système effectue des *actions* A_i bien définies. Une situation du système se caractérise par un vecteur d'état x dont la composante x_i vaut 1 ou 0 selon que l'étape E_i fait partie de cette situation ou non. Sa sortie est un vecteur y dont la composante y_i vaut 1 ou 0 selon que l'action A_i est en exécution ou non. Le système passe à une nouvelle étape dans la même situation ou dans une autre quand sa sortie ou son comportement vis-à-vis des entrées se modifie. La *transition* d'une étape E_i à une étape E_j se réalise à l'instant où se vérifie la condition $R_{ij}(u, x) = 1$, R_{ij} étant une proposition logique, appelée *réceptivité*, dépendante du vecteur d'entrées u et/ou de l'état x en cet instant. Les étapes au début du fonctionnement du système sont dites initiales ; elles sont activées dès la mise du système sous tension.

Conventions graphiques. Un grafcet est une représentation graphique des étapes et de leurs actions, des transitons et de leurs réceptivités ainsi que des liaisons connectant les étapes aux transitions.

a) *Étapes et actions*. Une étape ordinaire est représentée par un carré et une étape initiale par un double carré (fig. 6-1). Le carré d'une étape E_i (initiale ou ordinaire) est numéroté i et les symboles ou les noms des actions exécutées durant cette étape sont inscrits à l'intérieur de rectangles (étiquettes) placés à droite du carré représentant E_i.

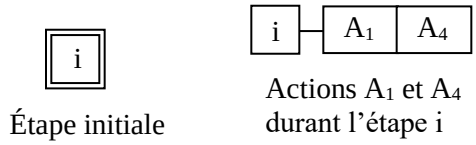


Fig. 6-1 Étapes et actions

En général, une action A est exécutée dès l'activation de l'étape i à laquelle elle est rattachée et on a :

$$x_i = 1 \Rightarrow A = 1.$$

Si l'action A rattachée à une étape active i ne s'exécute que lorsqu'une condition C est vraie on dit quelle est *conditionnelle*. Elle se représente par la figure 6-2 et on a :

$$x_i C = 1 \Rightarrow A = 1.$$

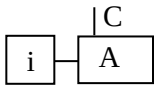
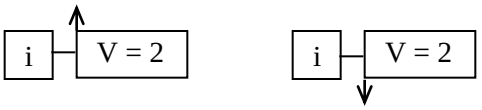


Fig. 6-2 Action conditionnelle

Lorsque l'action consiste à affecter à une ou plusieurs variables des valeurs déterminées et à conserver cette affectation après la désactivation de l'étape correspondante, on dit que cette action est *mémorisée* et se représente par l'une des figures 6-3a ou 6-3b selon que la mémorisation a lieu à l'instant d'activation de l'étape i ou à l'instant de sa désactivation.



a) À l'activation b) À la désactivation

Fig. 6-3 Action mémorisée

b) *Réceptivités* et *transitions*. La réceptivité R_{ij} (ainsi que la condition C d'une action conditionnelle) est une proposition logique à laquelle on associe la valeur 1 si elle est vraie et la valeur 0 si elle est fausse. Elle peut être

- une expression booléenne. Exemple : $a\bar{b} + x_2$.
- Un front montant ou descendant d'un signal s :

↑s n'est égal à 1 qu'à l'instant où s passe de 0 à 1,
↓s n'est égal à 1 qu'à l'instant où s passe de 1 à 0.

- Une relation entre crochets. Exemples :

[P > 6 bars], [Compteur = 7].

- Un intervalle de temporisation noté $t_1/t/t_2$ qui devient 1 après t_1 secondes de la montée de t (après ↑t) et redevient 0 après t_2 secondes de la descente de t (après ↓t). L'écriture t_1/t a la même signification que $t_1/t/0$. La variable t peut être une entrée ou une composante d'état. Exemple : $5/x_3$ devient 1 après 5 secondes de l'activation de l'étape 3 et redevient 0 dès la désactivation de cette étape.
- Une combinaison logique des cas précédents. Exemple : [V ≠ 16] + a(↑s) n'est égal à 1 que si V est différent de 16 ou que a est égal à 1 à l'instant de la montée de s.

La réceptivité R_{ij} s'écrit à droite d'un trait horizontal, appelé *transition*, placé entre les étapes E_i et E_j. Un arc joint E_i à la transition et un autre joint la transition à E_j. Ces arcs, appelés *liaisons*, sont constitués de segments verticaux ou horizontaux et ne comportent de flèches que lorsqu'ils sont dirigés de bas en haut ou de droite à gauche (fig. 6-4).

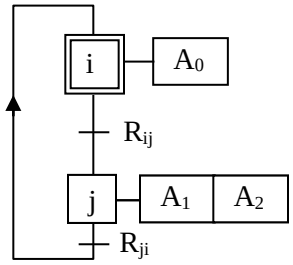
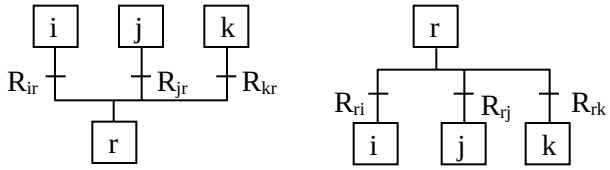


Fig. 6-4 Transitions, réceptivités et liaisons.

c) *Liaisons multiples*. Comme le montre la figure 6-5, une étape r se connecte à plusieurs transitions situées à son amont ou à son aval à travers une barre horizontale (ou parfois à travers un point). Pour éviter toute confusion, il convient de ne pas aligner deux segments situés des deux côtés de la barre.



(a) (b)

Fig. 6-5 Configurations OU

La configuration 6-5a est appelée *OU convergente*. Elle signifie que l'étape r s'active quand l'une des étapes à son amont est active et que la réceptivité entre cette étape et r se réalise (devient 1). Ceci se traduit par l'implication suivante :

$$x_i R_{ir} + x_j R_{jr} + x_k R_{kr} = 1 \Rightarrow x_r = 1.$$

La configuration 6-5b est appelée *OU divergente*. Elle signifie que si l'étape r est active, l'une des étapes à son aval s'active lorsque la réceptivité entre r et cette étape se réalise. Ceci se traduit par l'implication suivante :

$$x_r R_{rs} = 1 \Rightarrow x_s = 1, \quad s = i, j \text{ ou } k.$$

Comme le montre la figure 6-6, une transition se connecte à plusieurs étapes situées à son amont ou à son aval à travers une double barre horizontale. Pour éviter toute confusion, il convient de ne pas aligner deux segments situés des deux côtés de la double barre.

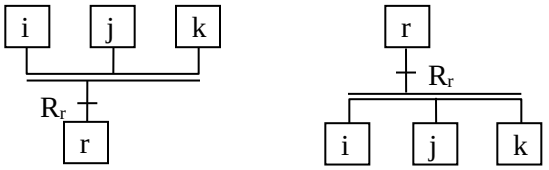


Fig. 6-6 Configurations ET

La configuration 6-6a est appelée *ET convergente*. Elle signifie que l'étape r s'active quand toutes les étapes à son amont sont actives (synchronisme) est que la réceptivité à son entrée se réalise. Ceci se traduit par l'implication suivante :

$$x_i x_j x_k R_r = 1 \Rightarrow x_r = 1.$$

La configuration 6-6b est appelée *ET divergente*. Elle signifie que si l'étape r est active, toutes les étapes à son aval s'activent (simultanéité) lorsque la réceptivité à la sortie de r se réalise. Ceci se traduit par l'implication suivante :

$$x_r R_r = 1 \Rightarrow x_i x_j x_k = 1.$$

Règles d'évolution. On appelle *situation* du système en un instant t l'ensemble des étapes actives en cet instant. Cette situation évolue en respectant les règles suivantes

Règle 1. La situation initiale est l'ensemble des étapes qui s'activent à la mise sous tension du système. Souvent, durant la situation initiale, le système informe l'utilisateur, par des lampes de signalisation ou par afficheur de texte, qu'un cycle est terminé et qu'il attend un ordre pour reprendre un nouveau cycle. Parfois le système peut refaire un cycle sans attendre un ordre de l'utilisateur.

Règle 2. Une transition est dite *validée* lorsque toutes les étapes immédiatement à son amont sont actives. Une transition est *franchie* si et seulement si elle est validée et que sa réceptivité est égale à 1.

À noter que si la transition n'est pas précédée par une double barre, il n'existe qu'une seule étape à l'amont de cette transition.

Règle 3. Le franchissement d'une transition entraîne l'activation simultanée de toutes les étapes immédiatement à son aval et la désactivation de toutes les étapes immédiatement à son amont.

À noter que si la transition n'est pas suivie d'une double barre, il n'existe qu'une seule étape à son aval.

Règle 4. Toutes les transitions simultanément franchissables sont simultanément franchies.

En toute rigueur, la simultanéité est physiquement impossible mais on admet que toutes les transitions sont franchies en un laps de temps négligeable.

Règle 5. Une étape s'active si elle est simultanément activée et désactivée.

L'exemple de la figure 6-7 montre qu'il est possible de rencontrer ce dernier cas bien qu'il soit rare.

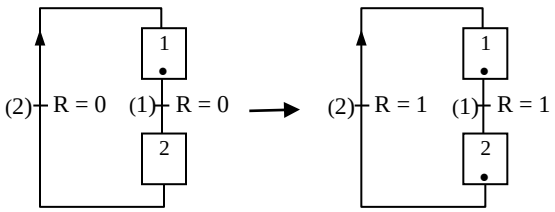


Fig. 6-7 Cas où s'applique la règle 5

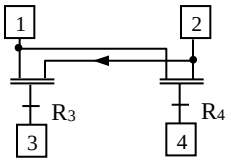
Au départ, seule l'étape 1 est active (le point indique que l'étape est active). Lorsque la réceptivité R devient 1, la transition (1) active l'étape 2 et désactive l'étape 1 et, simultanément, la transition (2) active 1 et désactive 2 d'où, d'après la règle 5, les deux étapes s'activent et resteront activées indépendamment de la valeur de R.

Exemples 6-1.

1) L'étape s appartenant à {3, 4} s'active si les étapes 1 et 2 sont toutes les deux actives et que la réceptivité R_s à l'entrée de l'étape s devient vraie. Ceci se traduit par l'implication

$$x_1x_2R_s = 1 \Rightarrow x_s = 1,\ s = 3 \text{ ou } 4,$$

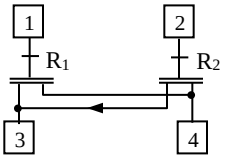
et par le grafcet suivant.



2) Les deux étapes 3 et 4 s'activent si l'une des étapes s = 1 ou s = 2 est active et que la réceptivité R_s à sa sortie devient vraie. Ceci se traduit par l'implication

$$x_sR_s = 1 \Rightarrow x_3x_4 = 1,\ s = 1 \text{ ou } 2$$

et par le grafcet suivant.



EXERCICE 6-1

Représenter le grafcet relatif à l'implication

$$x_1(x_2 + x_3x_4)R_5 = 1 \Rightarrow x_5 = 1.$$

R_5 étant à l'entrée de l'étape E_5.

6-2 EXEMPLES D'APPLICATON

A- Malaxage - Remplissage

La figure 6-8 représente un système de remplissage de bouteilles par un produit obtenu en dissolvant une matière granulée C dans un mélange de deux liquides A et B. Le déroulement du processus est le suivant.

- En poussant sur le bouton de marche m, les robinets R_A et R_C s'ouvrent. Quand le niveau dans le récipient L devient n_1, R_A se ferme et le robinet R_L s'ouvre pour vider le contenu de L dans le malaxeur.

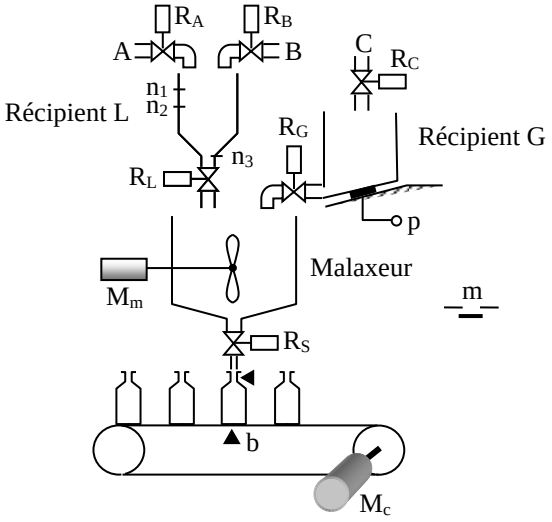


Fig. 6-8 Système de remplissage

R_L se referme à l'instant où le niveau devient inférieur à n_3. Quand le poids p dans le récipient G devient p_1, R_C se ferme et le robinet R_G s'ouvre pour vider le contenu de G dans le malaxeur. R_G se referme à l'instant où le poids de G devient inférieur à p_2.

- Après le vidage du liquide A et de la matière granulée C dans le malaxeur et la fermeture des deux robinets R_L et R_G, le moteur M_m tourne et en même temps le robinet R_B s'ouvre pour remplir le récipient L du liquide B jusqu'au niveau n_2. À ce moment, R_B se ferme et R_L s'ouvre jusqu'au niveau n_3. Le malaxage de A, B et C par le moteur M_m ne s'arrête qu'après le remplissage de N_1 bouteilles.
- Quand R_L se ferme après le vidage du liquide B dans le malaxeur, le moteur M_c du convoyeur se met en marche jusqu'à l'arrivée d'une bouteille vide sous le robinet R_S du malaxeur (détectée par un capteur b). Cependant, R_S ne s'ouvre pour remplir la bouteille que s'il a passé au moins 40 secondes depuis la dernière fermeture du robinet R_L.
- Quand le capteur c indique le remplissage de la bouteille, R_S se ferme et le moteur M_c se remet en rotation pour amener une nouvelle bouteille vide sous R_S et la remplir. Ceci se répète N fois ($N > N_1$) et quand N bouteilles sont ainsi remplies, le système retourne à sa situation initiale et attend la poussée sur m pour refaire un nouveau cycle.

Grafcet. La figure 6-9 représente le grafcet du système dont les entrées et les sorties sont définies dans les tableaux 6-1. Les commentaires entre guillemets au voisinage des étapes et des transitions

ne sont pas obligatoires mais servent à éclaircir le déroulement des opérations. Remarquer que les séquences {2, 3, 4} et {5, 6, 7} sont simultanées et il en est de même des séquences {8, 9} et {10}. D'autre part, l'action à l'étape 14 est à la fois mémorisée et conditionnelle.

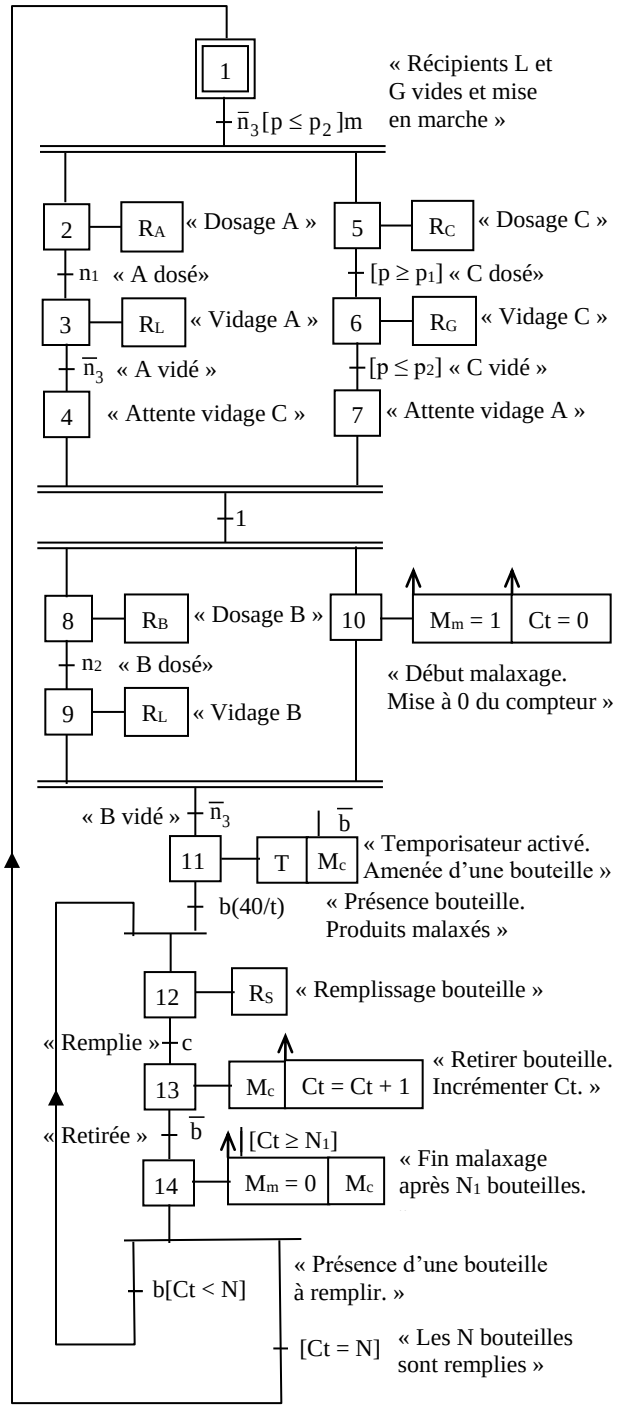


Fig. 6-9 Le grafcet d'un système de remplissage

Entrées			
Nom	Définition	Nom	Définition
m	mise en marche	p_1	Poids de la matière C
n_1	Niveau du liquide A	p_2	Poids de G vide
n_2	Niveau du liquide B	b	Bouteille présente
n_3	Niveau inférieur	c	Bouteille remplie
		t	Temps du temporisateur

Sorties			
Nom	Définition	Nom	Définition
R_A	Bobine du robinet A	R_S	Bobine du robinet S
R_B	Bobine du robinet B	M_m	Moteur malaxeur
R_C	Bobine du robinet C	M_c	Moteur convoyeur
R_L	Bobine du robinet L	T	Temporisateur
R_G	Bobine du robinet G	Ct	Compteur

Tableaux 6-1. Définitions des variables malaxeur exemple

B- <u>Transport</u> (voir exercice. 4-23).

Un chariot transporte un produit granulé d'une trémie aux stations A_i, i = 1, 2. Quand le chariot n'est pas en A_i, le contacte a_i est ouvert et une impulsion sur m_i indique le besoin de la station A_i d'un nouveau chargement. Quand le chariot est en A_i, le contacte a_i est fermé et une impulsion sur m_i envoie le chariot de la station A_i à la trémie. À l'arrivée du chariot en c, le volet V de la trémie s'ouvre pendant 40 secondes puis elle se ferme. Le chariot se dirige ensuite vers la station qui a indiqué son besoin pour un nouveau chargement en donnant la priorité à celle qui n'a pas été servie la dernière. Le chariot reste en c si aucune station ne le demande.

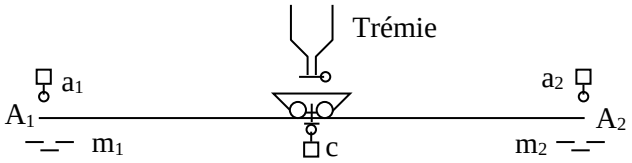


Fig. 6-10 Chariot desservant deux stations

<u>Grafcet.</u> Comme la demande d'une station d'un chargement ne dépend pas du mouvement du chariot, il convient de séparer le grafcet de cette demande de celui du mouvement du chariot. La figure 6-11 montre les 3 sous-grafcets du grafcet global du système ; les deux premiers pour les demandes des deux stations (qui sont indépendantes) et le troisième pour le mouvement du chariot. Les échanges d'informations entres les sous-grafcets se font moyennant les variables d'état. Nous supposons qu'à l'instant initial, le chariot sous la trémie est rempli et attend une demande et que les stations n'ont pas

encore indiqué leur besoin d'un chargement. L'étape 5 devient transitoire quand la dernière station servie n'a pas indiqué son besoin d'un chargement tandis que l'autre station a indiqué ce besoin. Le tableau 6-2 regroupe la signification des variables d'entrée et de sortie du système.

Remarque. Une boucle comportant moins que 3 étapes peut introduire un effet de course dont le remède sera donné plus loin.

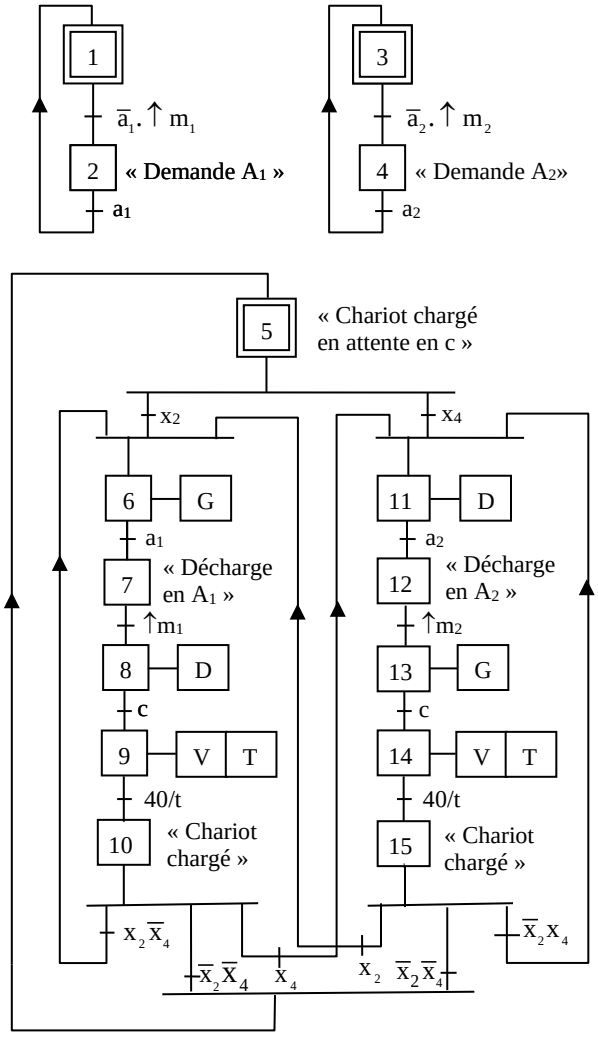


Fig. 6-11 Grafcet d'un système de transport

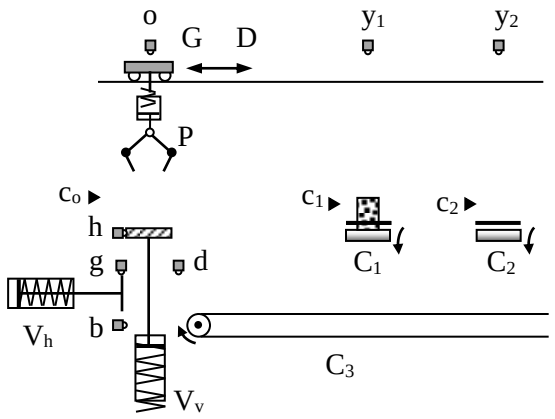
Entrées		Sorties	
Nom	Définition	Nom	Définition
a_i	Chariot en A_i	G	vers la gauche
m_i	Indicateur besoin	D	vers la droite
c	Chariot sous trémie	V	Ouverture volet
t	Temps du temporisateur	T	Temporisation

Tableau 6-2 Définitions des variables exemple B

EXERCICE 6-2

Expliquer pourquoi on risque un dysfonctionnement si l'on remplace dans le grafcet de l'exemple précédent le front $\uparrow m_i$ par un appui continu sur m_i.

EXERCICE 6-3



Quand une pièce arrive à une butée à l'extrémité d'un convoyeur C_1 ou C_2, un chariot muni d'une pince P transporte la pièce du convoyeur à un plateau horizontal fixé à l'axe d'un vérin V_v. Ce dernier descend le plateau avec la pièce au niveau d'un vérin V_h qui pousse la pièce sur un convoyeur C_3 pour l'évacuer.

Entrées. Le plateau et les convoyeurs C_1 et C_2 sont en face des boutons de fin de course o, y_1 et y_2. Les postions haute et basse du plateau et gauche et droite de l'éjecteur sont indiquées par les boutons h, b, g et d. La présence d'une pièce à l'extrémité d'un convoyeur ou sur le plateau est détectée par un capteur c_1, c_2 ou c_o et la fermeture de la pince par un capteur p.

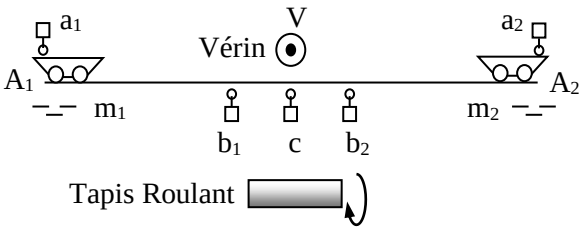
Actions. Un moteur déplace le chariot en tournant vers la droite (action D) ou vers la gauche (action G). La pince, le plateau et l'éjecteur sont activés par des vérins à simple effet P, V_v et V_h.

Priorité. Si deux pièces attendent sur les extrémités des deux convoyeurs C_1 et C_2, la pince prend la pièce du convoyeur qui n'a pas été servi la dernière fois.

Construire un grafcet pour ce système.

EXERCICE 6-4

Un poste de déchargement reçoit un chariot rempli à l'une des deux stations A_1 et A_2 pour le vider sur un tapis roulant en tournant la benne du chariot à l'aide d'un vérin V. En poussant sur le bouton m_i, le chariot se dirige de A_i vers le poste de chargement. Si ce dernier est libre, le chariot avance jusqu'à c et s'il est occupé, le chariot attend en b_i. Une fois vidé, le chariot retourne à sa station où il sera rempli de nouveau.



Construire un grafcet pour ce système.

6-3 ÉQUATIONS ET RÉALISATION CÂBLÉE

La réalisation d'un grafcet par câblage ou par programmation se déduit des équations des étapes qui lient l'état suivant x_i' de chaque étape E_i à l'état actuel x et au vecteur d'entrées u, $x_i' = f_i(x,u)$. Ces équations doivent satisfaire les 5 règles d'évolution mentionnées ci-dessus (section 6-1).

Séquence unique. Écrivons, l'équation d'une étape E_i intérieure à une séquence unique de plusieurs étapes (fig. 6-12).

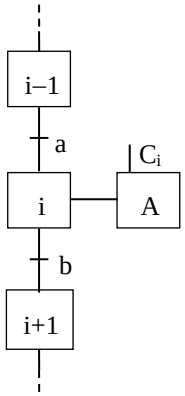


Fig. 6-12 Grafcet partiel

Une transition est franchie quand sa réceptivité a devient vraie en un moment où toutes les étapes immédiatement à son aval sont actives (règle 2). Ce franchissement active toutes les étapes à l'aval de la transition et désactive toutes les étapes immédiatement à son amont (règle 3). Par conséquent, l'étape E_i s'active au franchissement de la transition à son amont et se désactive à l'activation de l'étape E_{i+1}. En d'autres termes, si la variable d'état x_i est matérialisée par la sortie d'un loquet SR, on a :

$S_i = x_{i-1}a$ (Condition d'activation de E_i)
$R_i = x_{i+1}$ (Condition de désactivation de E_i)

et l'équation de l'étape E_i est

$$\begin{aligned} x_i' &= S_i + x_i\bar{R}_i \\ &= x_{i-1}a + x_i\bar{x}_{i+1} \end{aligned} \qquad (6\text{-}1)$$

Conformément à cette équation, E_i ne se désactive qu'après l'activation de E_{i+1}. De même, E_{i-1} ne se désactive qu'après l'activation de E_i. Ceci est important car si x_{i-1} s'annule avant que x_i devienne 1, d'après (6-1) x_i' restera nul et E_i ne s'activera pas. Pour cette raison, il faut éviter de poser $R_i = x_ib$ (franchissement de la transition avale de E_i) car ceci conduit à l'équation $x_i' = x_{i-1}a + x_i\bar{b}$ qui peut annuler x_i' avant que x_{i+1} devienne 1.

Remarquer aussi que (6-1) est l'équation d'un loquet à marche prioritaire dont la sortie x_i sera 1 (étape activée) si la condition d'activation S_i et la condition de désactivation R_i sont toutes les deux égales à 1 (règle 5).

Concernant la sortie, si l'action A conditionnée par C_i n'apparaît que durant l'étape E_i, son équation est $A = C_ix_i$ qui se réduit à $A = x_i$ quand l'action A n'est pas conditionnée ($C_i = 1$). Si cette action apparaît durant les étapes E_i, E_j, E_k, … avec les conditions C_i, C_j, C_k, … (dont certaines peuvent être égales à 1) on écrit :

$$A = C_ix_i + C_jx_j + C_kx_k + \cdots . \qquad (6\text{-}2)$$

Boucle à deux étapes. Dans l'équation (6-1) interviennent les états x_{i-1}, x_i et x_{i+1} de 3 étapes

successives différentes. Que se passe-t-il quand il s'agit d'une boucle à deux étapes seulement (pouvant appartenir à un grafcet comportant d'autres séquences ou boucles)?

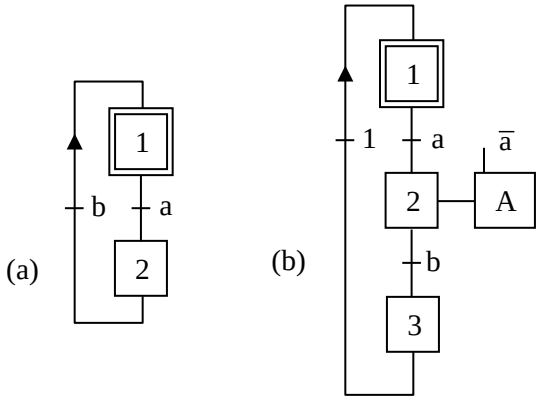


Fig. 6-13 Remède au Grafcet à 2 étapes

En appliquant (6-1) aux 2 étapes de la boucle 6-13a, on aura :

$$x_1' = x_2 b + x_1 \bar{x}_2, \quad x_2' = x_1 a + x_2 \bar{x}_1 . \qquad (6\text{-}3)$$

Pour éviter l'effet de course (entre x_i et $\bar{x}_i$), il faut ajouter à l'expression de x_1' le terme de jonction bx_1 et à l'expression de x_2' le terme de jonction ax_2 (voir section 3-2-4). Un autre remède (fig. 6-13b) consiste à ajouter une étape transitoire E_3 suivi d'une réceptivité toujours vraie (=1), ce qui ramène à des équations de la forme (6-1). En supposant que l'étape E_2 est accompagnée d'une action A conditionnée par $\bar{a}$, la figure 6-14 montre la réalisation électronique de la boucle 6-13b basée sur l'équation 6-1.

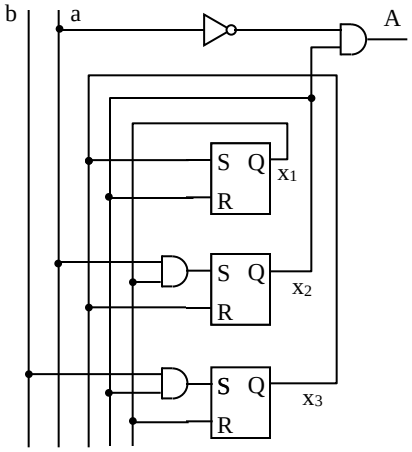


Fig. 6-14 Réalisation asynchrone d'une boucle

On peut aussi éviter l'aléa de continuité en utilisant des bascules synchronisées par une horloge H. Dans ce cas, comme les états de ces bascules ne se modifient qu'au front d'activation de H, on ne risque pas l'effet de course. La figure 6-15 représente la réalisation synchrone de la boucle à deux étapes 6-13a utilisant des bascules JK avec

$$\begin{aligned} J_1 &= x_2 b, \quad J_2 = x_1 a, \quad \text{(activation)}. \\ K_1 &= x_2, \quad K_2 = x_1, \quad \text{(désactivation)}. \end{aligned} \qquad (6\text{-}4)$$

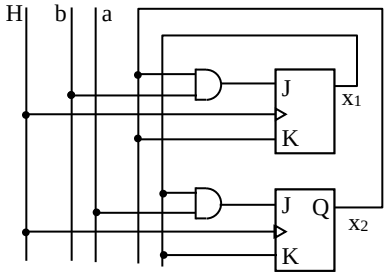


Fig. 6-15 Réalisation synchrone d'une boucle

Initialisation. Un signal « in » active les étapes initiales et désactive toutes les autres étapes quel que soit l'état précédent du système ou de ses entrées. « in » est généré par un bouton spécial après la mise sous tension du système ou quand on désire retourner à l'état initial. L'initialisation est introduite dans l'équation d'une étape initiale E_0 sous la forme

$$x_0' = (S_0 + \text{in}) + x_0 \bar{R}_0 \qquad (6\text{-}5)$$

et d'une étape ordinaire E_i sous la forme :

$$\begin{aligned} x_i' &= (S_i + x_i \bar{R}_i)\overline{\text{in}} \\ &= S_i \overline{\text{in}} + x_i \overline{R_i + \text{in}} \end{aligned} \qquad (6\text{-}6)$$

où S_0 et R_0 sont les conditions d'activation et de désactivation de E_0, S_i et R_i les conditions d'activation et de désactivation de E_i.

La figure 6-16 montre comment intervient le signal d'initialisation « in » dans le circuit d'une étape initiale et dans le circuit d'une étape ordinaire.

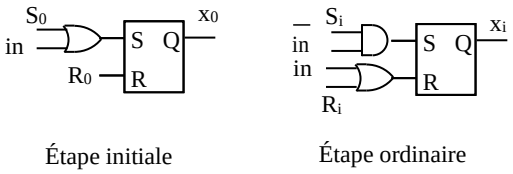


Fig. 6-16 Introduction du signal d'initialisation

Arrêt d'urgence. Un signal « ad », appelé arrêt doux, annule toutes les réceptivités et les conditions des actions (sauf éventuellement celles relatives à la sécurité : extincteur, frein, …). En d'autres termes, pour un arrêt doux, une réceptivité R_{ij} et une condition d'action C_i seront remplacées par

$$R_{ij}.\overline{ad}, \quad C_i.\overline{ad}.$$

De cette manière « ad » arrête les actions (qu'on désire arrêter) et fige le système dans sa dernière situation puisque les réceptivités ne peuvent plus modifier l'activité des étapes. Quand on annule « ad » le cycle repart de la situation où il s'est interrompu.

Un signal « aD », appelé arrêt dur, désactive toutes les étapes y compris les étapes initiales et par conséquent il arrête toutes les actions. Après un arrêt dur, le système doit être initialisé de nouveau. Les équations suivantes introduisent dans les équations des étapes l'initialisation et l'arrêt dur.

$$x_0' = \left[(S_0 + in) + x_0\overline{R}_0\right]\overline{aD} \quad \text{(étape initiale)},$$
$$x_i' = \left[S_i\overline{in} + x_i\overline{R_i + in}\right]\overline{aD} \quad \text{(étape ordinaire)}.$$

Généralement, on réserve au déclenchement des signaux d'arrêt d'urgence, « ad » et « aD », un grafcet séparé du grafcet principal. Par exemple, si un thermostat trm indique une surchauffe nécessitant un examen soigneux du système de refroidissement et un capteur de niveau crb indique un manque en carburant, le grafcet des arrêts d'urgence aura la configuration représentée par la figure 6-17 où rep informe le système que la réparation est terminée.

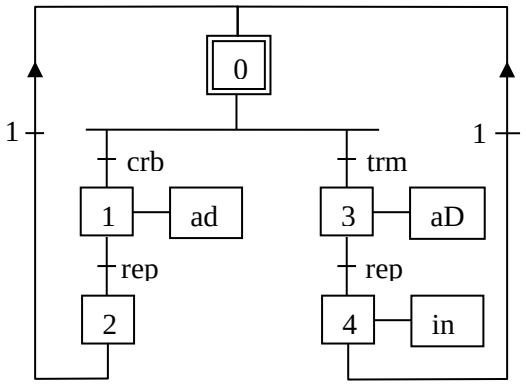


Fig. 6-17 Grafcet d'arrêt d'urgence

L'étape transitoire E_2 est introduite pour éviter l'aléa de continuité dans la boucle à deux étapes E_0 et E_1. En passant à l'étape E_2, le système repart de la situation où il s'est interrompu. De l'étape E_4 le système retourne aux étapes initiales du grafcet principal et du grafcet d'arrêts d'urgence.

Convergence et divergence. Le tableau 6-3 donne les conditions d'activation et de désactivation des étapes E_2, E_3 et E_5 pour chacun des graphes partiels divergents OU et ET de la figure 6-18.

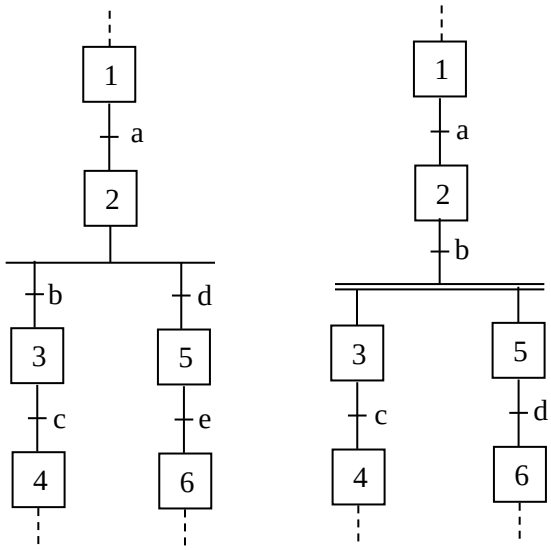


Fig. 6-18 Divergences OU et ET

	Divergence OU		Divergence ET	
	S	R	S	R
E_2	x_1a	$x_3 + x_5$	x_1a	x_3x_5
E_3	x_2b	x_4	x_2b	x_4
E_5	x_2d	x_6	x_2b	x_6

Tableau 6-3 Activation et désactivation des divergences

EXERCICE 6-5
Sachant que deux circuits n'ont jamais le même temps de propagation, quel risque de dysfonctionnement peut-on avoir si l'on remplace la condition $R_2 = x_3x_5$ de la divergence ET par $R_2 = x_3 + x_5$?

Le tableau 6-4 donne les conditions d'activation et de désactivation des étapes E_2, E_4 et E_5 pour chacun des grafcets partiels convergents OU et ET de la figure 6-19.

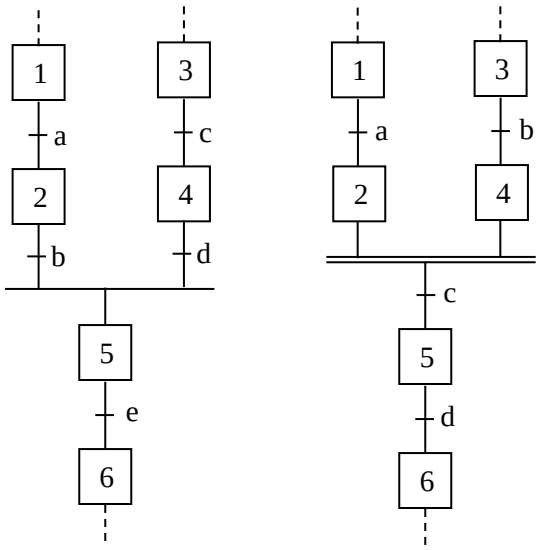


Fig. 6-19 Convergences OU et ET

	Convergence OU		Convergence ET	
	S	R	S	R
E_2	x_1a	x_5	x_1a	x_5
E_4	x_3c	x_5	x_3b	x_5
E_5	$x_2b + x_4d$	x_6	x_2x_4c	x_6

Tableau 6-3 Activation et désactivation des convergences

6-4 MISE EN ŒUVRE PAR PROGRAMMATION

La réparation ou la modification d'une réalisation câblée nécessite le remplacement d'une partie du circuit par un autre montage. Pour éviter ce travail souvent long, on préfère employer, pour commander des machines à fonctionnement relativement complexe, un automate programmable. Les langages de programmation de ces automates sont simples et leurs instructions sont facilement modifiables.

Nous représentons d'abord la structure d'un automate programmable et nous décrivons son fonctionnement avant de détailler son langage de programmation.

Structure et fonctionnement d'un automate. Un automate programmable est constitué d'une unité centrale de traitement (CPU) communiquant avec des registres d'entrées, de sorties, de variables internes (d'état) et avec une mémoire d'instructions non volatile ROM et une autre volatile RAM (fig. 6-20). Les instructions du programme sont introduites moyennant une console de programmation externe qui se branche à l'automate.

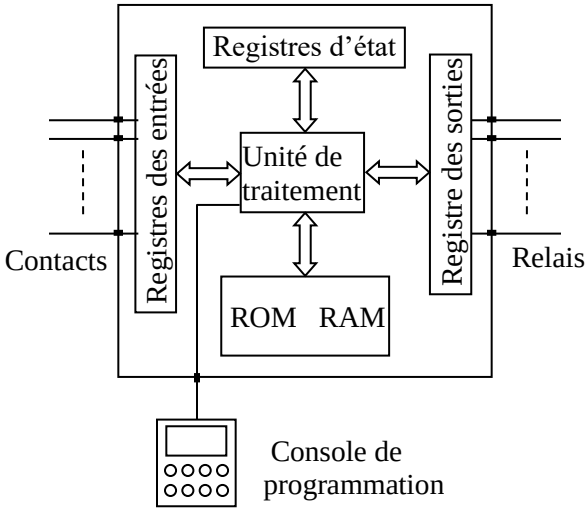


Fig. 6-20 Structure d'un automate programmable

Le fonctionnement de l'automate est géré par l'unité centrale de traitement qui répète continuellement deux fonctions :

a) Communication avec l'extérieur.
b) Exécution du programme.

La communication avec l'extérieur comporte les opérations suivantes :

1) Stockage dans les registres des entrées les informations binaires provenant des capteurs et des contacts de commande. À chaque entrée est associée une position (une bascule) dans ces registres.
2) Livraison du contenu du registre de sorties aux organes opératoires (relais, lampes témoins, convertisseurs, …). À chaque organe de sortie est associée une position du registre.

Une fois les données sont échangées avec l'extérieur, les instructions du programme enregistré dans la mémoire ROM sont exécutées l'une après l'autre comme suit :

1) Lecture et décodage d'une nouvelle instruction.
2) Lecture dans les registres d'entrée ou d'état les valeurs des variables adressées par l'instruction.
3) Calcul des variables de sortie ou d'état en effectuant les opérations indiquées par l'instruction sur les variables lues en 2).

4) Remise à jour des registres de sortie et d'état conformément aux résultats obtenus en 3).

Après l'exécution de la dernière instruction, l'automate communique de nouveau avec l'extérieur puis répète l'exécution du programme avec les nouvelles données. À noter que les registres d'état et de sorties se modifient au cours de l'exécution du programme tandis que les registre d'entrées ne se modifient qu'entre deux exécutions du programme.

6-4-1 Programmation en échelle.

C'est un langage graphique facile à comprendre par un technicien familier avec les circuits de commande électromagnétiques. Le programme ressemble à une échelle (ladder) constituée d'un ou de plusieurs échelons (rungs). Un échelon a la forme d'un circuit logique à contacts suivi de l'adresse où se mémorise la valeur de la fonction logique définie par ce circuit. À un contact est associée la valeur d'une position dans les registres d'entrées ou d'état ou le complément de cette valeur. L'adresse indique une position dans les registres d'état ou de sorties. Les échelons sont lus et exécutés successivement du haut en bas. Au départ, quand l'automate est mis sous tension, la valeur de toute bascule d'état est nulle et cette valeur ne se modifie que quand sa bascule est adressée par un échelon. La dernière valeur d'une bascule d'état est conservée au moment où l'automate recommence une nouvelle exécution du programme.

La figure 6-21 présente les symboles de base de la programmation en échelle.

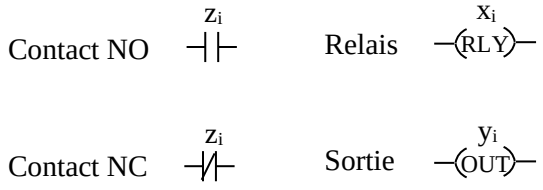


Fig. 6-21 Symboles d'un programme en échelle

À un contact NO (ouvert au repos) correspond la valeur d'une bascule d'entrée ou d'état d'adresse z_i et à un contact NC (fermé au repos) correspond le complément de cette valeur. La valeur de la fonction définie par le circuit à contact de l'échelon est transférée à une bascule d'état d'adresse x_i ou à une bascule de sortie d'adresse y_i. À noter que chaque fabricant adopte un format spécial pour les adresses et l'aspect de ses symboles peuvent légèrement différer de ceux représentés ci-dessus.

Exemple 6-2

1) *Bascule SR*. Sachant que l'équation de cette bascule est $x' = S + x\bar{R}$ et en admettant que son état x active une action A, le programme en échelle suivant (fig. 6-22) réalise cette fonction.

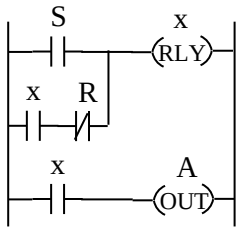


Fig. 6-22 Bascule RS et son action

Le contact S prend la valeur d'une donnée (entrée ou variable d'état) d'adresse S et le contact R prend le complément d'une donnée d'adresse R. Ces contacts peuvent être remplacés par des circuits réalisant les fonctions S et $\bar{R}$. D'autre part, il est important de savoir que la valeur d'un contact x est la dernière fournie par le relais x. Par conséquent, la valeur du contact x de l'échelon supérieur, pouvant provenir d'une exécution précédente du programme, n'est pas nécessairement égale à celle du contact x de l'échelon inférieur qui prend la dernière valeur du relais RLY.

2) *Bascule T*. L'état de cette bascule change de valeur quand T = 1 et qu'une entrée h passe de 0 avant sa dernière lecture à 1 à sa dernière lecture (front montant d'activation). Dans les autres cas, l'état de la bascule conserve sa valeur. En d'autres termes, en désignant par h_p la valeur précédente de h (h_p peut provenir de l'exécution précédente de programme), l'état x de la bascule se modifie si et seulement si $f = T\bar{h}_p = 1$. D'où l'équation de l'état suivant de la bascule est

$$x' = f.\bar{x} + \bar{f}.x$$

et son programme est représenté par la figure 6-23.

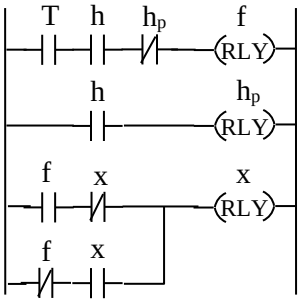


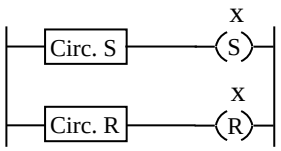
Fig. 6-23 Bascule T

Le deuxième échelon sert à mémoriser la valeur de h afin de l'utiliser par la suite en tant que h_p. Cet échelon peut être placé à n'importe quelle position et en particulier vers la fin du programme.

EXERCICE 6-6
a) Représenter le programme en échelle d'une bascule à retard D activée par le front montant d'une entrée h.
b) Même question pour une bascule JK.

Programmes préétablis. Les programmes des organes qui apparaissent fréquemment dans les applications sont souvent stockés dans une mémoire non volatile à l'intérieur de l'automate. Au lieu de répéter à chaque fois la programmation de l'organe, une représentation graphique spéciale dirige l'automate vers le programme de cet organe pour l'exécuter. Le symbole représentant un organe peut varier d'un type d'automate à un autre mais c'est toujours facile de comprendre sa signification. Par exemple, les figures 6-24a et 6-24b sont des représentations possibles des bascules SR et T dont les programmes ont été détaillés dans l'exemple 6-2.

a) Bascule SR



b) Bascule T

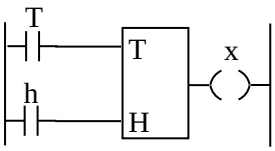


Fig. 6-24 Représentations de bascules

Dans la représentation 6-24a, la bascule SR est identifiée par les noms S et R des deux relais qui indiquent la même adresse x où l'état de la bascule sera mémorisé. Cet état s'active (mis à 1) quand la sortie du circuit S est 1 et se désactive (mis à 0) quand la sortie du circuit R est 1. Dans la représentation 6-24b, la bascule T est identifiée par les noms de ses entrées T et H et sa sortie est mémorisée à l'adresse x.

Exemple 6-3

Le tableau 6-4 donne les conditions d'activation et de désactivation des étapes du grafcet de la figure 6-25 tenant compte de la commande d'initialisation « in ».

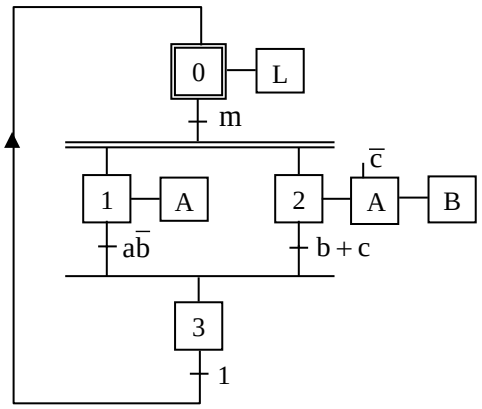


Fig. 6-25 Grafcet de l'exemple 6-3

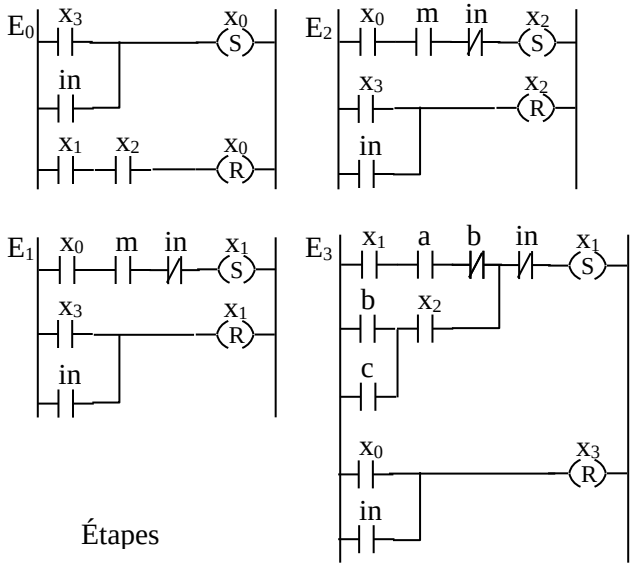
	S	R
E_0	$x_3 + in$	x_1x_2
E_1	$x_0\, m.\overline{in}$	$x_3 + in$
E_2	$x_0\, m.\overline{in}$	$x_3 + in$
E_3	$[x_1a\bar{b} + x_2(b+c)]\overline{in}$	$x_0 + in$

Tableau 6-4 Conditions d'évolution

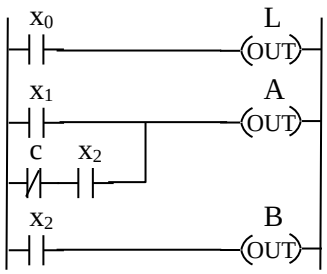
D'autre part, les équations des actions sont :

$$L = x_0, \quad A = x_1 + \bar{c}x_2, \quad B = x_2 .$$

La figure 6-26 montre la programmation de ce grafcet. Pour plus de clarté, nous avons mis à part le programme de chaque étape ainsi que celui des actions.

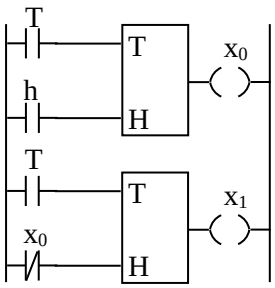


Actions



Exemple 6-4

Sachant que le bit le moins significatif x_0 d'un compteur binaire modulo 4 change de valeur à chaque monté de h et que le second bit x_1 change de valeur à chaque descente de x_0, il peut être réalisé par le programme suivant :



À côté d'autres organes, la plupart des automates comportent des programmes pour compter, temporiser et gérer des séquences d'opérations.

Compteur. La figure 6-25 est l'une des représentations d'un compteur ayant une entrée Ct de comptage, une autre entrée Rz de remise à zéro et une sortie SAT indiquant la saturation du compteur.

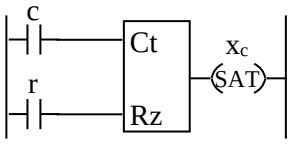


Fig. 6-25 Une représentation d'un compteur

Si Rz = 0, le compteur est incrémenté de 1 à chaque front montant de Ct c.à.d. quand la valeur précédente du contact c est 0 et sa valeur actuelle est 1. Quand le dénombrement arrive à une valeur maximum N préalablement introduite dans la mémoire RAM en une adresse spécifique à ce compteur, ce dernier cesse de compter et le relais SAT qui lui est rattaché active (met à 1) la bascule d'adresse x_c du registre d'état. Si Rz = 1, le compteur s'annule et l'entrée Ct ne l'affecte plus.

Certains compteurs peuvent être à rebours d'entrées Ct de comptage et Mx de remise à la valeur maximum N. Pour Mx = 0, un compteur de ce type se décrémente de 1 à chaque monté de Ct. Quand le dénombrement arrive à zéro, le compteur cesse de décompter et le relais SAT active sa bascule dans le registre d'état. Mx = 1 arrête le comptage et ramène le compteur à N.

D'autres compteurs sont bidirectionnels pouvant compter en direct ou à rebours. La figure 6-26 est une représentation d'un tel compteur. Elle est constituée de 3 échelons, le premier pour le comptage en direct, le deuxième pour le comptage à rebours et le troisième pour ramener le compteur à sa valeur de départ (0 ou N selon le compteur).

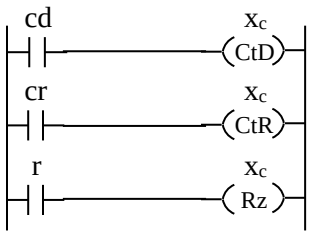


Fig. 6-26 Compteur bidirectionnel

Temporisateur. C'est un compteur qui compte les impulsions d'une horloge de période p (= 0.1 ou 0.01 seconde) incorporée dans l'automate. À la valeur maximum N, introduite dans la mémoire RAM par l'utilisateur, correspond la durée Np du temporisateur. Quand le nombre N d'impulsions est atteint, le comptage s'arrête et le relais du compteur active sa bascule dans le registre d'état.

On rencontre deux types de temporisateur, l'un est non conservatif représentés par la figure 2-27a et l'autre est conservatif représenté par la figure 2-27b.

a) Temporisateur non conservatif

b) Temporisateur conservatif

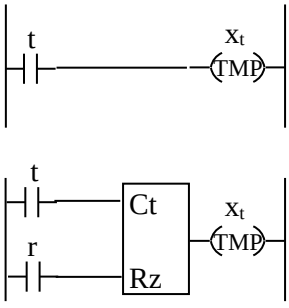


Fig. 6-27 Deux types de temporisateurs

Le temporisateur non conservatif est mis à 0 quand le contact t est égal à 0. Quand t devient 1, le compteur est incrémenté de 1 chaque p seconde jusqu'à la valeur maximum N où le comptage cesse et la bascule d'adresse x_t s'active. Selon les conventions du grafcet, x_t s'écrit Np/t pour dire que x_t devient 1 après Np secondes de la monté de t.

Pour un temporisateur conservatif, t = 0 arrête le comptage mais n'annule pas le dénombrement. Seule l'entrée r = 1 remet le temporisateur à 0. Tant que r = 0, le dernier dénombrement est conservé quand t = 0 et s'incrémente à partir de cette valeur quand t redevient 1. Comme pour le temporisateur non conservatif la bascule d'adresse x_t s'active quand le dénombrement total arrive à N.

EXERCICE 6-7
Construire un programme en échelle qui, quand un contact a est fermé, éclaire une lampe L durant 2 sec et l'éteint durant 1 sec.

Séquenceur. Cet organe est aussi un compteur ayant les spécificités suivantes :

1) Comme les compteurs, certains séquenceurs sont bidirectionnels pouvant compter en direct ou à rebours.
2) Contrairement aux compteurs, le dénombrement est cyclique :
   …, N – 2, N – 1, N, 0, 1, 2, … s'il est direct,
   …, 2, 1, 0, N, N – 1, N – 2, … s'il est à rebours.
3) À chaque valeur du séquenceur on peut associer un ou plusieurs contacts qui prennent la valeur 1 quand le dénombrement arrive à cette valeur.

La figure 6-28 représente un séquenceur bidirectionnel incrémenté de 1 à chaque monté du contact a et décrémenté de 1 à chaque monté du contact r. Il commande une bascule SR qui s'active quand le dénombrement arrive à 4 et se désactive quand le dénombrement arrive à 7.

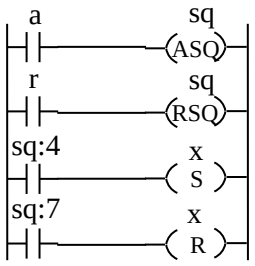
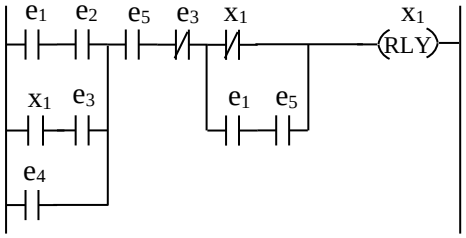


Fig. 6-28 Emploi d'un séquenceur

6-4-2 Programmation mnémonique.

Le programme en échelle doit être traduit en langage machine par un compilateur fourni avec l'automate ou manuellement par l'utilisateur en dressant une suite d'instructions chacune constituée d'une commande suivie d'un opérande représentant un contact, un relais ou tout autre élément. La commande indique à l'automate l'opération à effectuer sur l'opérande. Comme pour les symboles graphiques ou le format des adresses, les codes des commandes ou des opérandes ne sont pas standardisés et peuvent différer d'un automate à un autre. Nous adoptons dans cette section quelques codes qui rappellent leur rôle et nous laissons à l'utilisateur le soin de se référer au manuel de l'automate qu'il veut programmer.

Pour donner les principales règles de construction d'un programme mnémonique, considérons l'échelon de la figure 6-29.



. **Fig. 6-29** Échelon à plusieurs branches

Cet échelon est constitué de plusieurs branches qu'on peut grouper en blocs de (1) à (5) qui s'imbriquent par des connexions en parallèle ou en série comme le montre la figure 6-30.

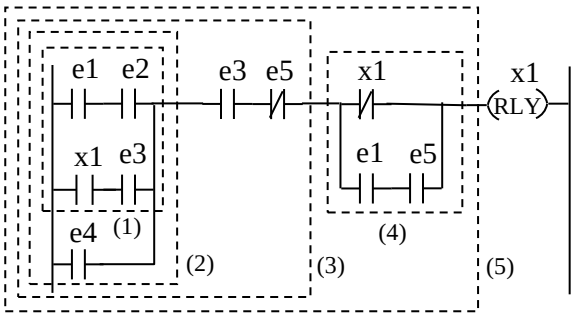


Fig. 6-30 Décomposition de l'échelon 6-29

On débute une branche par la commande LD (load, charger la valeur) ou LD NOT (charger le complément) suivie de l'adresse du premier élément de cette branche. Une partie de l'adresse spécifie la nature de l'élément (entrée, état, sortie) et l'autre

partie indique sa position dans le registre correspondant. La première instruction est donc « LD e1 » et pour compléter la première branche, on ajoute l'instruction « AND e2 ». Par la commande AND l'automate effectue l'opération ET entre la valeur de l'opérande e2 et le dernier résultat. De même, la deuxième branche se traduit par les deux instructions « LD x1 » et « AND e3 ». L'instruction « OR LD » connecte en parallèle les deux branches précédentes. Ainsi, pour constituer le bloc (1) on écrit successivement les instructions suivantes, chacune sur une ligne :

```
LD e1
AND e2
LD x1
AND e3
OR LD
```

On a maintenant la valeur de e1.e2 + x1.e3. Pour former le bloc (2) il suffit d'effectuer l'opération OU entre ce dernier résultat et la valeur de e4 en ajoutant l'instruction « OR e4 » aux précédentes. Ceci nous donne la valeur de e1.e2 + x1.e3 + e4. En effectuant l'opération AND entre le dernier résultat et l'entrée e3 puis l'opération AND NOT entre le résultat obtenu et e5, on obtient le bloc (3) d'expression algébrique $(e1.e2 + x1.e3 + e4)e3\bar{e}5$. Le bloc (4) d'expression $\bar{x}1 + e1.e5$ peut se définir par les 4 instructions « LD NOT x1 », « LD e1 », « AND e5 » et « OR LD » mais on économise une instruction si l'on commence par la branche inférieure car, par cette démarche, il suffit d'écrire les 3 instructions « LD e1 », « AND e5 » et « OR NOT x1 ». Il ne reste qu'à connecter en série les blocs (3) et (4) par l'instruction « AND LD » ce qui donne la valeur de

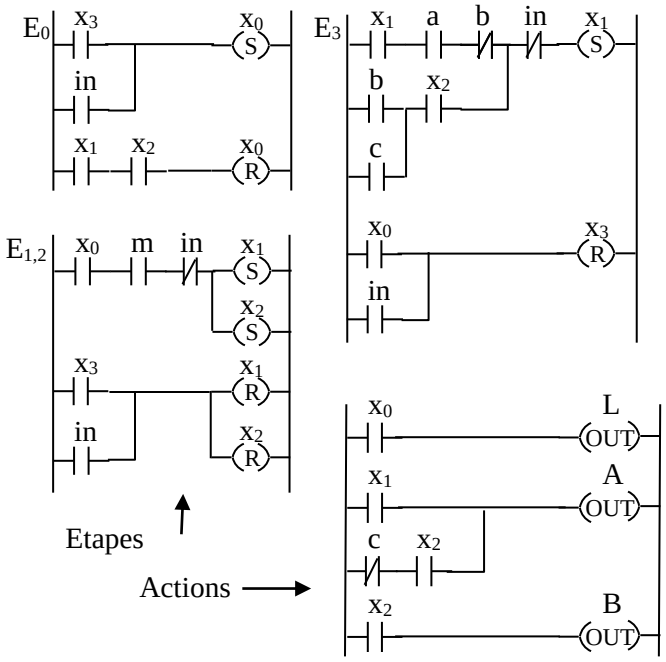
$$(e1.e2 + x1.e3 + e4)e3\bar{e}5(\bar{x}+e1.e5)$$

qu'on affecte à la sortie par l'instruction « OUT x1 ». Le programme mnémonique de l'échelon 6-29 est alors le suivant :

```
LD e1
AND e2
LD x1
AND e3
OR LD
OR e4
AND e3
AND NOT e5
LD e1
AND e5
OR NOT x1
AND LD
OUT x1
```

Exemple 6-4

Écrivons le programme mnémonique du grafcet de l'exemple 6-3. Pour la commodité, nous recopions ici son programme en échelle en fusionnant les étapes E_1 et E_2 puisqu'elles ont les mêmes conditions d'activation et de désactivation. Si le dernier résultat est 1, la commande SET x active la bascule d'adresse x et la commande RST x désactive cette bascule.



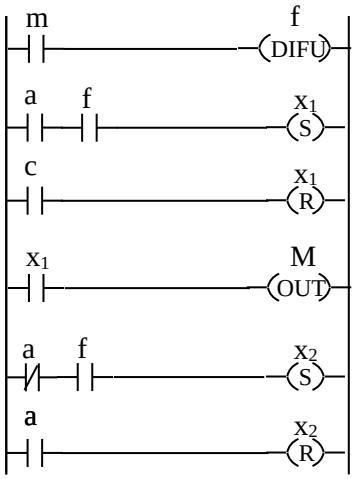
Étape E_0	Étape E_3	Actions
LD x3	LD x1	LD x0
OR in	AND a	OUT L
SET x0	AND NOT b	LD NOT c
LD x1	LD b	AND x2
AND x2	OR c	OR x1
RESET x0	AND x2	OUT A
	OR LD	LD x2
Étapes E_1 et E_2	AND NOT in	OUT B
LD x0	SET x3	
AND m	LD x0	
AND NOT in	OR in	
SET x1	RESET x3	
SET x2		
LD x3		
OR in		
RESET x1		
RESET x2		

Autres instructions. Nous terminons par des exemples faisant intervenir l'impulsion, le comptage et la temporisation.

Impulsion. La sortie de DIFU (differentiate up) devient 1 à l'instant où son entrée passe de la valeur précédente 0 à la valeur actuelle 1 (front montant). Au cycle suivant du programme, la sortie redevient 0 quel que soit la valeur de l'entrée puisque la valeur précédente de celle-ci est 1. Au passage de l'entrée de 0 à 1, DIFU produit donc une impulsion de très courte durée. DIFD (differentiate down) joue le même rôle que DIFU à part que l'impulsion à sa sortie apparait au passage de l'entrée de 1 à 0 (front descendant).

Exemple 6-5

Pour le système de la figure 6-10, une poussée sur m déplace le chariot si le bouton de fin de course a est fermé et demande le chariot si a est ouvert. Une demande involontaire aura donc lieu si m n'est pas relâché avant la libération de a par le chariot. Pour cette raison, l'appui sur m ne doit pas produire un signal continu mais une impulsion f. Comme on le voit sur le programme en échelle suivant, si a est fermé, l'impulsion f (notée ↑m en grafcet) active la variable x_1 qui met le moteur du chariot en marche et si a est ouvert, elle active la variable x_2 qui indique une demande. Un bouton de fin de course c désactive x_1 et arrête le chariot et la fermeture de a à l'arrivée du chariot à la station désactive la demande x_2.



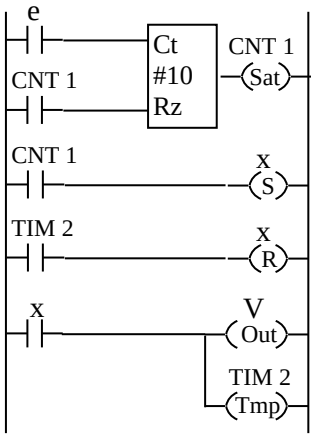
Ce programme en échelle se traduit par les instructions suivantes ;

```
LD m
DIFU f
LD a
AND f
SET x1
LD c
RESET x1
LD x1
OUT M
LD NOT a
AND f
SET x2
LD a
RESET x2
```

Compteur. L'instruction CNT est celle d'un compteur à rebours. Elle a une entrée d'impulsions et une autre de remise à la valeur maximum N. Ces entrées doivent être introduites juste avant l'instruction CNT. Celle-ci sera suivie sur la même ligne par le numéro i du compteur et sur la ligne suivante par la valeur de N. Sa sortie et une variable d'état, de même nom « CNT i » que le compteur, qui s'active quand le dénombrement des impulsions arrive à 0.

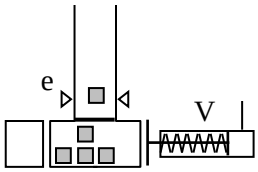
Temporisateur. L'instruction TIM est celle d'un temporisateur non-conservatif. Elle a une seule entrée d'activation qui doit être introduite juste avant l'instruction TIM. Celle-ci sera suivie sur la même ligne par le numéro j du temporisateur et sur la ligne suivante par le nombre N correspondant au temps de retard Np. Sa sortie et une variable d'état, de même nom « TIM j » que le temporisateur, qui s'active après Np secondes du passage de l'entrée d'activation de 0 à 1. Un compteur et un temporisateur ne doivent pas avoir le même numéro dans un même programme car le temporisateur n'est autre qu'un compteur.

Exemple 6-6




```
LD e
LD CNT 1
CNT 1
      #10
LD CNT 1
SET x
LD TIM 2
RESET x
LD x
OUT V
TIM 2
      #10
END
```

Ce programme commande le système représenté par la figure suivante.



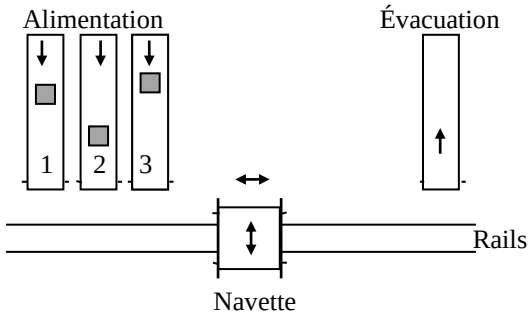
Le passage d'une pièce devant une photocellule produit une impulsion à l'entrée e. Après le passage de 10 pièces, le compteur CNT 1 active une bascule x qui avance un vérin à simple effet V et active un temporisateur TIM 2. Au cycle suivant du programme (après une très courte durée), le compteur est remis à 0 pour compter 10 nouvelles pièces. D'autre part, au bout d'une seconde (on suppose que la période p de l'horloge du temporisateur est de 0.1 seconde), la bascule se désactive ce qui recule le vérin et initialise le temporisateur.

EXERCICE 6-8

1) Représenter le programme en échelle d'un comparateur 1 bit.
2) En posant « cpr (A, B) » l'instruction qui compare deux nombres binaires à n – 1 bits, écrire le programme mnémonique d'un comparateur de deux nombres binaires à n bits.

AUTRES EXERCICES ET COMPLÉMENTS

6-9 Une navette comportant un tapis roulant transporte l'une après l'autre les pièces amenées par 3 tapis roulants d'alimentation pour les placer sur un tapis roulant d'évacuation. Quand une pièce arrive à son extrémité, le tapis d'alimentation s'arrête et attend son tour pour placer la pièce sur la navette. Cette dernière opération s'effectue par la rotation des deux tapis d'alimentation et de la navette jusqu'à l'arrivée de la pièce à la position correcte sur la navette. De même, pour décharger la pièce, le tapis de la navette se met de nouveau en marche mais en sens opposé jusqu'à l'arrivée de la pièce sur le tapis d'évacuation. Celle-ci est toujours en rotation.



Si plus qu'une pièce sont présentes aux extrémités des tapis d'alimentation, la navette transporte la première qui est arrivée. Si aucune pièce n'est présente sur ces tapis, la navette attend devant le tapis 1.

1) Définir dans un tableau les symboles des entrées (capteurs et boutons de commande) et des sorties (actionneurs).
2) Construire les grafcets des tapis d'alimentation, de leurs priorités et du mouvement de la navette. Représenter son programme en échelle et écrire sa traduction mnémonique.

6-10 Un ascenseur desservant n étages y compris le rez-de-chaussée fonctionne comme suit.

- Une demande vers un niveau est indiquée par un bouton d'appel situé à ce niveau ou par un bouton de destination placé à l'intérieur de la cabine. Le bouton d'appel peut être vers le haut ou vers le bas selon qu'un passager désire monter ou descendre. Le bouton d'appel au rez-de-chaussée ou au dernier étage peut être considéré comme étant à la fois pour monter et pour descendre.
- Quand la cabine est en mouvement (vers le haut ou

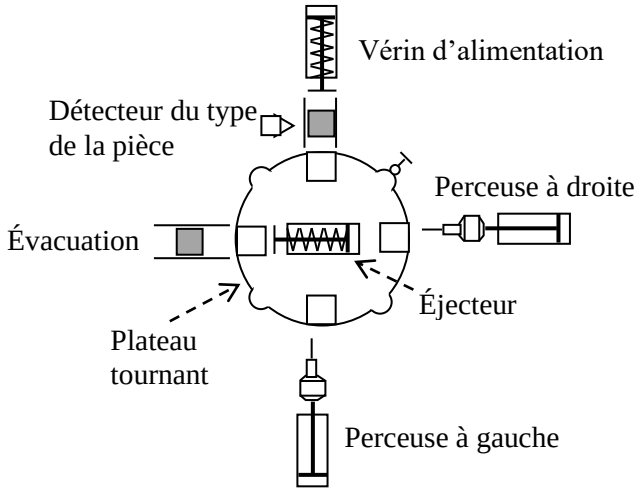
vers le bas), elle s'arrête au premier niveau demandée soit par un bouton de destination soit par le bouton d'appel ayant le même sens que le mouvement.

- À l'arrivée de la cabine à un niveau, les demandes vers ce niveau, de destination ou d'appel dans le sens du mouvement, s'effacent, la porte s'ouvre pendant 20 secondes et la cabine ne peut repartir qu'après la fermeture de la porte.
- L'ascenseur satisfait d'abord toutes les demandes vers des niveaux situés à l'avant de son mouvement puis dessert ensuite les demandes vers des niveaux situés en arrière. Si aucune demande n'est signalée la cabine reste au niveau où elle s'est arrêtée.

1) En supposant que les boutons d'appel ou de destination sont suivis d'encodeurs et que l'automate est muni de compteurs bidirectionnels, de temporisateurs et de comparateurs, dresser le tableau des capteurs et des boutons de commande (les entrées) et des actionneurs (les sorties) de ce système.
2) Représenter le grafcet de cet ascenseur, son programme en échelle et donner la traduction mnémonique de ce programme.

6-11 La figure suivante représente un dispositif permettant de percer 3 types de pièces :

- d : à percer à droite,
- g : à percer à gauche,
- dg : à percer à droite et à gauche.



Un capteur détecte le type de la pièce qui arrive devant le vérin d'alimentation. Quand la rotation du plateau est possible, un moteur le tourne jusqu'à la position convenable. Quand une pièce arrive devant l'éjecteur, elle est poussée dans un canal d'évacuation. Les vérins d'éjection et d'alimentation sont à simple effet tandis que ceux des perceuses sont à double effet.

1) Définir dans un tableau les symboles des entrées (capteurs et boutons de commande) et des sorties (actionneurs).
2) Représenter les grafcets de ce système ainsi que le circuit électronique de commande.