

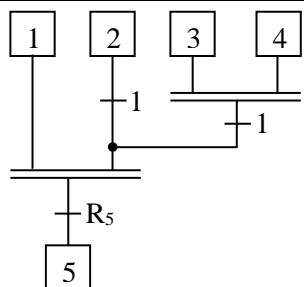
AS6 – GRAFCET ET PLC

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 E₅.

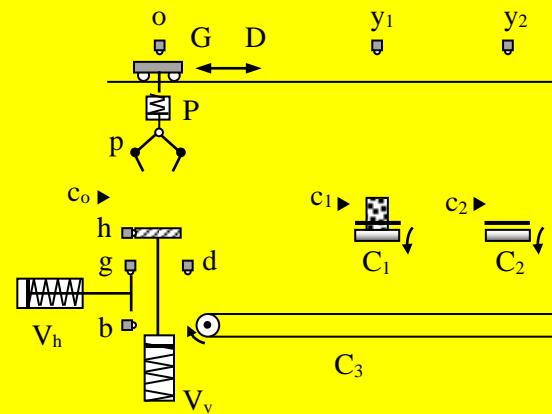


EXERCICE 6-2

Expliquer pourquoi on risque un dysfonctionnement si l'on remplace dans le grafcet de l'exemple d'application (B-transport, § 6-2) le front $\uparrow m_i$ par un appui continu sur m_i .

Supposons que juste avant l'arrivée du chariot à la station A_i pour se décharger, cette même station indique le besoin d'un autre chargement en poussant sur m_i . Si l'appui sur m_i se prolonge jusqu'à l'arrivée du chariot en A_i, le système comprend que A_i a libéré le chariot et le ramène immédiatement vers la trémie sans le décharger.

EXERCICE 6-3



Quand une pièce arrive à une butée à l'extrémité d'un convoyeur C₁ ou C₂, 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 C₃ pour l'évacuer.

Entrées. Le plateau et les convoyeurs C₁ et C₂ sont en face des boutons de fin de course o, y₁ et y₂. Les positions 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 c₁, c₂ ou c_o et la serrage de la pièce par la pince est détecté par un capteur p.

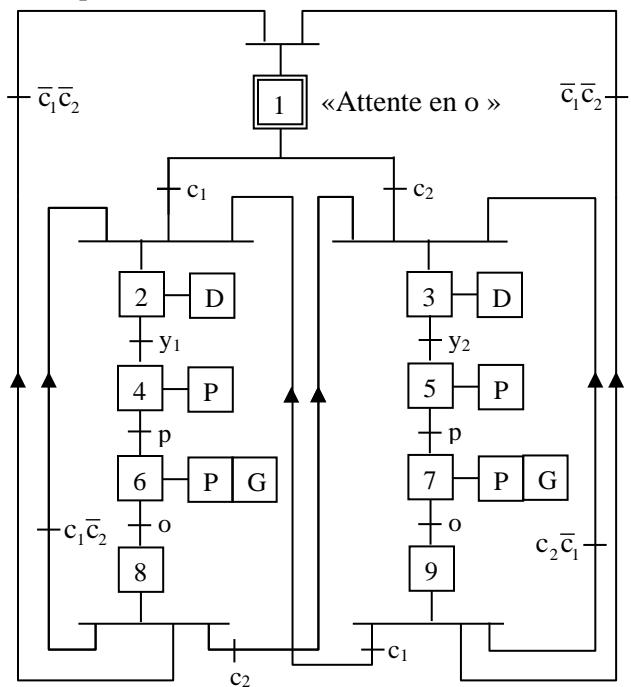
Actions. Un moteur déplace le chariot en tournant vers la droite (action D) ou vers la gauche (action G). La pince, le plateau et l'éjecteur sont activés par des vérins à simple effet P, V_v et V_h.

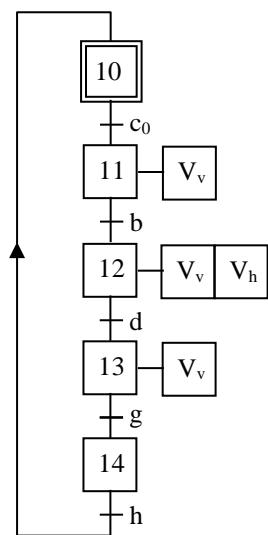
Priorité. Si deux pièces attendent sur les extrémités des deux convoyeurs, la pince prend la pièce du convoyeur qui n'a pas été servi la dernière fois.

Construire un grafcet pour ce système.

Ce système se subdivise en deux sous-systèmes : le transport des pièces et leur évacuation.

Transport.

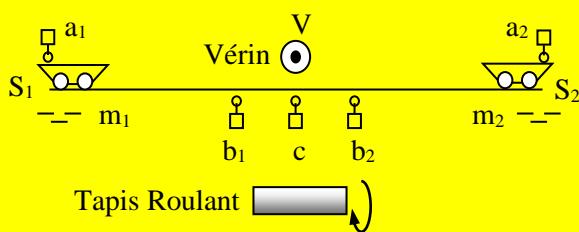


Évacuation

Nous avons supposé qu'à l'état initial, le chariot est en c_0 et attend la présence d'une pièce en c_1 ou c_2 (étape E_1) et que le piston vertical est dans sa position haute h et attend aussi la réception d'une pièce (étape E_{10}). Dans cette position, le plateau a le même niveau que les convoyeurs $C_1 C_2$.

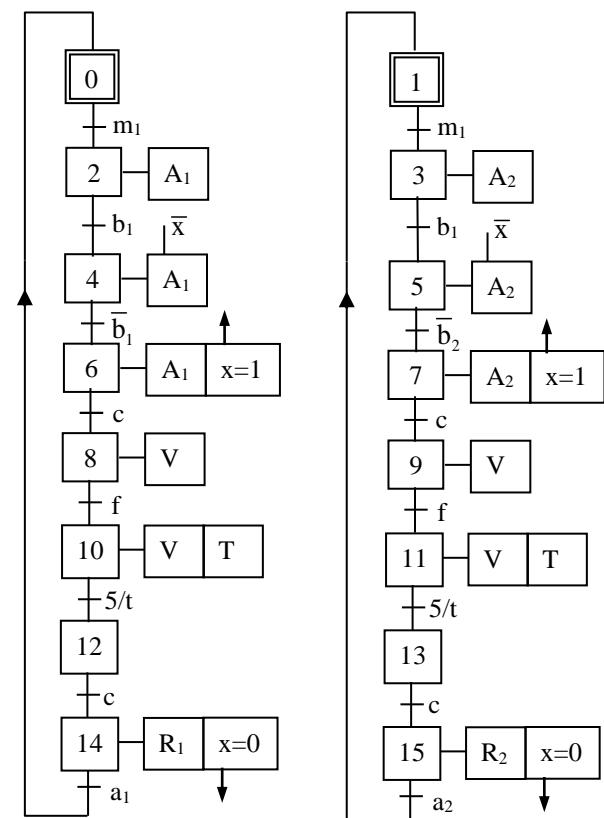
EXERCICE 6-4

Un poste de déchargement reçoit en c un chariot rempli à l'une des stations S_1 ou S_2 et le vide sur un tapis roulant en tournant la benne du chariot à l'aide d'un vérin V . Cette rotation libère le bouton c et un autre bouton f indique la fin de la rotation. Il faut 5 secondes de plus pour vider complètement la benne. Une fois vidée, la benne tourne dans le sens inverse, actionne c et le chariot retourne à sa station où il sera rempli de nouveau. En poussant sur le bouton m_i , le chariot se dirige de S_i vers le poste de déchargement. Si ce dernier est libre, le chariot avance jusqu'à c et s'il est occupé, le chariot attend en b_i .



Construire un grafcet pour ce système.

Nous subdivisons le système en deux sous-systèmes symétriques, l'un pour le chariot de la station S_1 et l'autre pour le chariot de la station S_2 . À l'état initial le chariot est à sa station et attend l'ordre pour se déplacer vers le poste de déchargement D . On pose A_i , $i = 1$ ou 2 , l'action qui avance le chariot de sa station S_i vers D et par R_i l'action qui recule le chariot vers sa station. Notons qu'un temporisateur T est nécessaire pour donner le temps à la benne de se vider complètement. D'autre part, on pose x_i , $i = 1$ ou 2 , l'état d'une mémoire qui indique si le poste de déchargement est occupé ou non.

**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 $R_2 = x_3 x_5$ de la divergence ET par $R_2 = x_3 + x_5$?

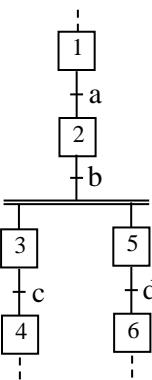
Si l'étape E_2 est active et son état suivant vérifie

$$x'_2 = x_1 a + x_2 (\overline{x_3} + \overline{x_5}),$$

il s'annule si $x_3 = 1$ ou $x_5 = 1$. Supposons que x_3 devient 1 avant x_5 . Comme la valeur suivante de ce dernier est donnée par

$$x'_5 = x_2 b + x_5 \bar{x}_6$$

et que x_2 et x_5 sont nuls, l'étape E_5 reste désactivée et on perd la simultanéité des deux branches de la convergence ET car, dans ce cas, seule évolue la branche de l'étape E_3 .



EXERCICE 6-6

a) Représenter le programme en échelle d'une bascule à retard D activée par le front montant d'une entrée de synchronisation h.

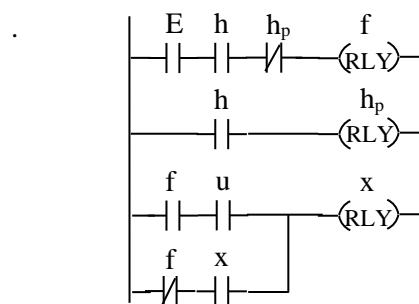
b) Même question pour une bascule JK.

a)

L'état de la bascule D prend la valeur de l'entrée u quand une variable d'activation E est égale à 1 et qu'une entrée h passe de 0 avant sa dernière lecture à 1 à sa dernière lecture (front montant). 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 du programme), l'état x de la bascule se modifie si et seulement si $f = Eh\bar{h}_p = 1$. D'où l'équation de l'état suivant de la bascule est

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

et il faut aussi retenir la nouvelle valeur de h pour qu'elle joue le rôle de h_p au prochain passage par cette bascule. Le programme est alors le suivant.

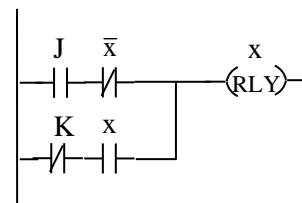


b)

Les deux branches de synchronisation f et h_p ne se modifient pas et peuvent être placées à n'importe quelle position dans un programme, en particulier, la branche de f peut être placée au début du programme et la branche de h_p à sa fin. Maintenant, comme l'état suivant d'une bascule JK est lié à l'état actuel x par

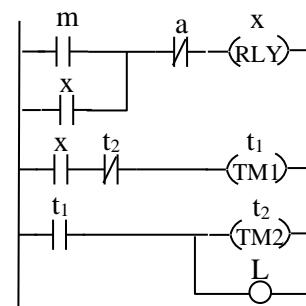
$$x' = J\bar{x} + \bar{K}x,$$

elle a les instructions en échelle suivantes.



EXERCICE 6-7

En poussant sur un bouton m une lampe L s'allume durant 2 sec et s'éteint durant 1 sec. Ce clignotement s'arrête en poussant sur un bouton a. Construire le programme en échelle de ce système.



Une mémoire d'état x indique la mise en marche ou l'arrêt du clignotement.

Quand $x = 1$, on obtient le clignotement (les pulsations alimentant la lampe) à l'aide de deux temporisateurs, T_1 de durée 1s et de sortie t_1 et T_2 de durée 2s et de sortie t_2 . En supposant que la période de l'horloge est $p = 0.1s$, les durées de T_1 et T_2 sont introduites dans la mémoire RAM de l'automate en associant respectivement à leurs adresses les nombres 10 et 20.

À l'instant où x s'active T_2 reste en arrêt pendant une 1 s durant la marche de T_1 (c.à.d. $T_1 = x\bar{t}_2$).

À l'instant $t = 1s$, T_1 se satire et t_1 devient 1. Ceci allume la lampe L et met T_2 en marche pendant 2s. Durant ce temps $t_2 = 0$ et $t_1 = 1$ (T_1 reste saturé).

À l'instant $t = 3s$, T_2 se satire et t_2 devient 1. Ceci annule t_1 à l'exécution suivante du programme (de durée très courte) ce qui annule t_2 , éteint la lampe et ramène le système à l'état initial pour répéter continuellement ce cycle jusqu'à l'appui sur a.

EXERCICE 6-8

- 1) Représenter le programme en échelle d'un comparateur 1 bit.
- 2) Soient « $A < B$ », « $A = B$ », « $A > B$ » les instructions qui comparent deux nombres binaires à $n - 1$ bits mémorisés dans 2 registres d'adresses A et B. Écrire le programme mnémonique d'un comparateur de deux nombres binaires à n bits.

1) Comparateur 1 bit

$$a < b \Leftrightarrow \bar{a}b = 1,$$

$$a = b \Leftrightarrow ab + \bar{a}\bar{b} = 1,$$

$$a > b \Leftrightarrow a\bar{b} = 1.$$

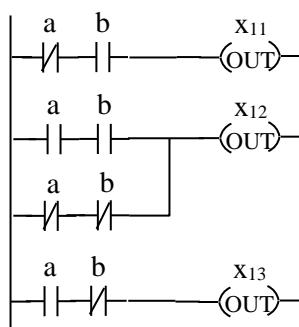
En posant

$$a < b \Leftrightarrow x_{11} = 1,$$

$$a = b \Leftrightarrow x_{12} = 1,$$

$$a > b \Leftrightarrow x_{13} = 1,$$

le programme en échelle est le suivant.



2)

Considérons les deux nombres à n bits

$$A_n = (a_n a_{n-1} \dots a_1)_2 \text{ et } B_n = (b_n b_{n-1} \dots b_1)_2$$

et soit $A = (a_{n-1} a_{n-2} \dots a_1)_2$ et $B = (b_{n-1} b_{n-2} \dots b_1)_2$.

On pose $x_{n1} = (a_n < b_n)$, $x_{n2} = (a_n = b_n)$, $x_{n3} = (a_n > b_n)$,
 $X_1 = (A < B)$, $X_2 = (A = B)$, $X_3 = (A > B)$,

On a :

$$X_{n1} = 1 \Leftrightarrow A_n < B_n$$

$$\Leftrightarrow (a_n < b_n) \text{ ou } (a_n = b_n \text{ et } A < B),$$

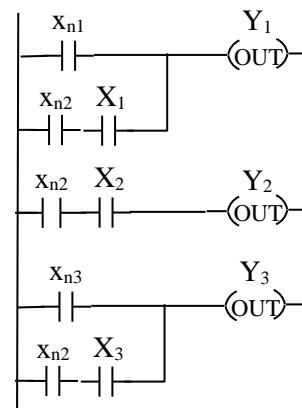
$$X_{n2} = 1 \Leftrightarrow A_n = B_n$$

$$\Leftrightarrow a_n = b_n \text{ et } A = B,$$

$$X_{n3} = 1 \Leftrightarrow A_n > B_n$$

$$\Leftrightarrow (a_n > b_n) \text{ ou } (a_n = b_n \text{ et } A > B).$$

Le programme en échelle de ces équivalences est le suivant.

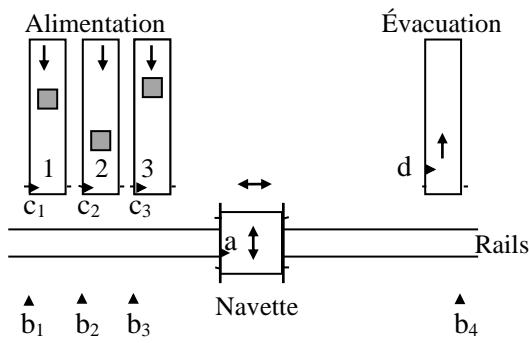


Les deux échelles précédentes conduisent au programme mnémonique suivant.

| <u>x_{n1}, x_{n2} et x_{n3}</u> | <u>X_1, X_2 et X_3</u> | <u>Y_1, Y_2 et Y_3</u> |
|---|--|--|
| LD NOT an | A < B | LD xn1 |
| AND bn | OUT X1 | LD xn2 |
| OUT xn1 | A = B | AND X1 |
| LD an | OUT X2 | OR LD |
| AND bn | A > B | OUT Y1 |
| LD NOT an | OUT X3 | LD xn2 |
| AND NOT bn | AND X2 | AND X2 |
| OR LD | OUT Y2 | OUT Y2 |
| OUT xn2 | LD xn3 | LD xn3 |
| LD an | LD xn2 | LD xn2 |
| AND NOT bn | AND X3 | AND X3 |
| OUT xn3 | OR LD | OR LD |
| | | OUT Y3 |

AUTRES EXERCICES ET COMPLÉMENTS

6.9 Une navette comportant un petit 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 convoyeur d'évacuation.

- 1) Définir dans un tableau les symboles des entrées (capteurs et boutons de commande) et des sorties (actionneurs).
- 2) Construire le grafset de ce système, représenter son programme en échelle et écrire sa traduction mnémonique.

a)

| Entrées | | Sorties | |
|----------------|--------------------------|----------------|------------------------|
| Nom | Définition | Nom | Définition |
| a | Pièce sur navette | G | Navette vers la gauche |
| b _i | Navette devant conv. i | D | Navette vers la droite |
| c _i | Pièce au bout du conv. i | M _i | Moteur conv. i |
| d | Pièce évacuée | A | Avance tapis navette |
| | | R | Recule tapis navette |

b) Le grafset de ce système sera subdivisé en deux grafsets partiels : le mouvement des convoyeurs d'alimentation selon leur priorité et le mouvement de la navette et de son tapis

Mouvement et priorité des convoyeurs 1, 2, et 3

Supposons d'abord qu'aucune pièce n'est présente à l'extrémité d'un convoyeur d'alimentation. La mémoire p_i de la priorité du convoyeur i est égale à 0 et les 3 convoyeurs sont en marche (étape 0).

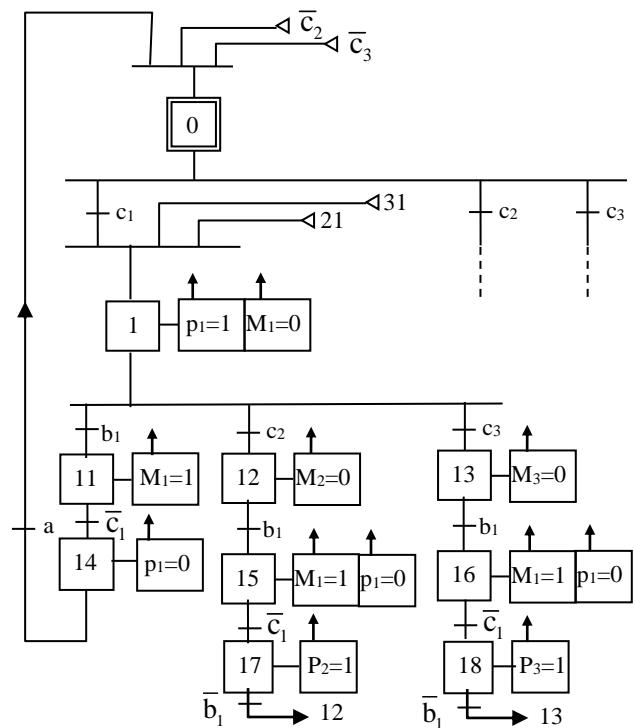
Le convoyeur qui amène la première pièce s'arrête et active une mémoire p_i, i = 1, 2 ou 3 pour dire que ce convoyeur est prioritaire.

Si la navette arrive devant i avant qu'une nouvelle pièce arrive à l'extrémité de l'un des deux autres convoyeurs j et k, le convoyeur i se remet en marche pour livrer sa pièce à la navette et retourner à l'étape du départ.

Si, avant que la navette retire la pièce de i, un convoyeur j ou k amène une nouvelle pièce, ce convoyeur s'arrête mais ne devient prioritaire que lorsque la première pièce quitte i.

Nous ne représentons que la partie relative au convoyeur 1, celles de 2 et de 3 étant symétriques.

Les flèches à la fin du diagramme suivant indiquent le passage de la priorité d'un convoyeur à un autre.

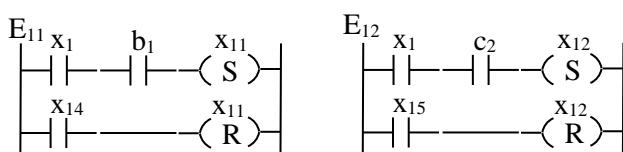
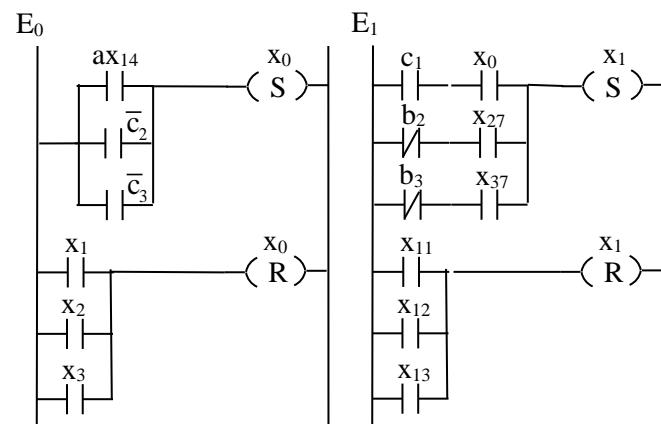


Programmation.

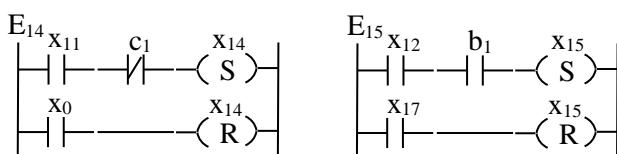
À chaque étape on associe l'état d'un loquet SR qui s'active au franchissement de la transition à son amont et se désactive après qu'une étape à son aval s'active.

Une transition est franchie quand sa réceptivité devient vraie en un moment où toutes les étapes immédiatement à son amont sont actives. Ce franchissement active toutes les étapes immédiatement à l'aval de la transition et désactive toutes les étapes immédiatement à son amont.

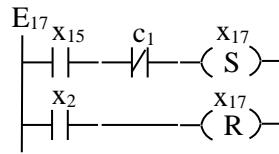
Une action se réalise si elle est associée à une étape active et si sa condition quand elle existe est égale à 1. En appliquant ces règles, on obtient le diagramme en échelle ci-dessous. Pour simplifier, nous omettons les entrées d'initialisation «in» et d'arrêt d'urgence «ad» et «aD». D'autre part, nous ne représentons que le loquet de l'étape initiale et de la partie relatives au convoyeur 1, celles des convoyeurs 2 et 3 étant symétriques. Enfin, pour plus de clarté, nous avons mis à part le programme de chaque étape ainsi que celui des actions.



L'échelle de E₁₃ se déduit de celui de E₁₂ en remplaçant 2 par 3 et 4 par 5.

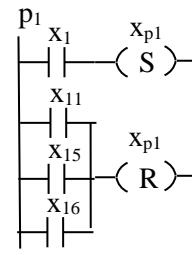
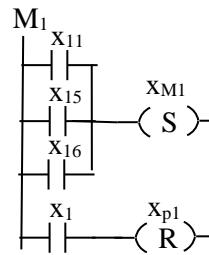


L'échelle de E₁₆ se déduit de celui de E₁₅ en remplaçant 2 par 3, 4 par 5 et 7 par 8.



L'échelle de E₁₈ se déduit de celle de E₁₇ en remplaçant 2 par 3, 5 par 6 et 7 par 8.

Les sorties du convoyeur 1



La partie du programme relative au convoyeur 2 se déduit de celle du convoyeur 1 et la partie relative au convoyeur 3 se déduit de celle du convoyeur 2 en remplaçant 1 par 2, 2 par 3 et 3 par 1. Cette permutation circulaire s'applique aussi au programme mnémonique suivant.

Étape E₀

LD x14
AND a
OR NOT c2
OR NOT c3
SET x0
LD x1
OR x2
OR x3
RESET x0

Étape E₁

LD c1
AND x0
LD NOT b2
AND x27
LD NOT b3
AND x37
OR LD
SET x1
LD x11
OR x12
OR x13
RESET x1

Étape E₁₁

LD x1
AND b1
SET x11
LD x14
RESET x11

Étape E₁₂

LD x1
AND c2
SET x12
LD x15
RESET x12

La mnémonique de E₁₃ se déduit de celui de E₁₂ en remplaçant l'indice 2 par 3 et l'indice 4 par 5.

| <u>Étape E₁₄</u> | <u>Étape E₁₅</u> |
|-----------------------------|-----------------------------|
| LD x11 | LD x12 |
| AND NOT c1 | AND b1 |
| SET x14 | SET x15 |
| LD x0 | LD x17 |
| RESET x14 | RESET x15 |

La mnémonique de E₁₆ se déduit de celui de E₁₅ en remplaçant 2 par 3, 5 par 6 et 7 par 8.

| <u>Étape E₁₇</u> | <u>La mnémonique de E₁₈ se déduit de celui de E₁₆ en remplaçant 2 par 3, 5 par 6 et 7 par 8.</u> |
|-----------------------------|--|
| LD x15 | |
| AND NOT c1 | |
| SET x17 | |
| LD x2 | |
| RESET x17 | |

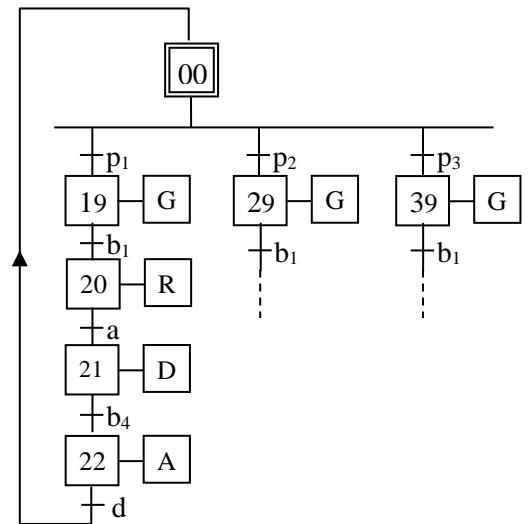
Les mnémoniques des sorties sont les suivants.

| <u>Étape M₁</u> | <u>Étape p₁</u> |
|----------------------------|----------------------------|
| LD x11 | LD x1 |
| OR x15 | SET p ₁ |
| Or x16 | LD x11 |
| SET M1 | OR x15 |
| LD x1 | OR x15 |
| RESET M1 | RESET p ₁ |

La navette et son tapis

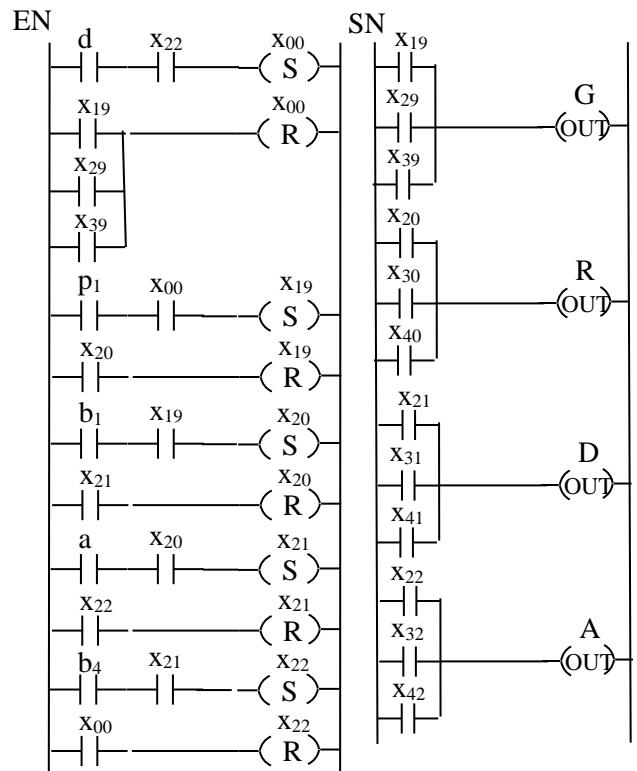
On suppose qu'initialement la navette est vide en face du convoyeur d'évacuation en attendant l'arrivée d'une pièce à l'extrémité de l'un des convoyeurs d'alimentation.

Quand p_i indique la présence d'une pièce à l'extrémité du convoyeur i = 1, 2 ou 3 (p_i = 1) et signal la priorité du convoyeur i, la navette se déplace à gauche pour s'arrêter en b_i. Le convoyeur i se remet alors en marche (grafset ci-dessus) et le tapis de la navette recule. Quand la pièce arrive en face du capteur a, le tapis s'arrête et la navette se déplace à droite jusqu'à b₄ où elle s'arrête et son tapis avance. À l'arrivée de la pièce en face du capteur d, on retourne à l'étape initiale qui sera transitoire si p_i = 1, i = 1, 2 ou 3. Nous représentons seulement la partie relative au cas p₁ = 1, les deux autres cas lui étant similaires.



Programmes

L'état de la navette EN et ses sorties SN sont définis par Les programmes en échelle suivants



| <u>EN₁</u> | <u>SN</u> |
|-----------------------|-----------|
| LD d | LD x19 |
| AND x22 | OR x29 |
| SET x00 | OR x39 |
| LD x19 | OUT G |
| OR x29 | LD x20 |
| OR x39 | OR x30 |
| RESET x00 | AND x40 |
| LD p1 | OUT R |
| AND x00 | LD x21 |
| SET x19 | OR x31 |
| LD x20 | OR x41 |
| RESET x19 | OUT D |
| LD b1 | LD x22 |
| AND x19 | OR x32 |
| SET x20 | OR x42 |
| LD x21 | OUT A |
| RESET x20 | |
| LD a | |
| AND x20 | |
| SET x21 | |
| LD x22 | |
| RESET x21 | |
| LD b4 | |
| AND x21 | |
| SET x22 | |
| LD x00 | |
| RESET x22 | |

6-10 Un ascenseur desservant 4 étages, 0, 1, 2 et 3, 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. Au rez-de-chaussée et au dernier étage il n'existe qu'un seul bouton d'appel.
- Quand la cabine est en mouvement (vers le haut ou vers le bas), elle s'arrête au premier niveau demandé par un bouton de destination ou par le bouton d'appel ayant le sens du mouvement de la cabine.
- À l'arrivée de la cabine à un niveau, les demandes vers ce niveau s'effacent, la porte s'ouvre pendant 30 secondes et la cabine ne peut repartir qu'après la fermeture de la porte.

• L'ascenseur satisfait toutes les demandes vers des niveaux situés à l'avant de son mouvement avant de se diriger dans le sens opposé. Si aucune demande n'est signalée la cabine reste au niveau où elle s'est arrêtée. Après cet arrêt, la cabine se dirige dans le sens de la première demande.

- 1) Dresser le tableau des capteurs et des boutons de commande (les entrées) et des actionneurs (les sorties) de ce système.
- 2) En supposant qu'on dispose d'encodeurs et de comparateurs, représenter le programme en échelle de cet ascenseur.

1)

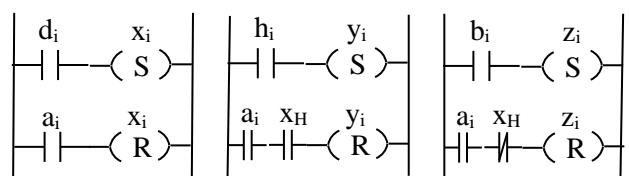
| Entrées | | Sorties | |
|---------|--------------------------------|---------|-------------------------|
| Nom | Définition | Nom | Définition |
| h_i | Appel du niveau i vers le haut | M | Montée |
| b_i | Appel du niveau i vers le bas | D | Desccente |
| d_i | Destination passager vers. i | O_i | Ouverture de la porte i |
| a_i | Arrivée de la cabine en i | F_i | Fermeture de la porte i |
| p_i | Porte fermée | T | Temorisateur |

2)

On subdivise le fonctionnement de l'ascenseur en deux parties : la mémorisation des demandes vers les niveaux, le mouvement de la cabine et la commande des portes.

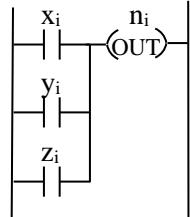
a) Mémorisation des demandes

On pose x_i l'état d'une mémoire activée par un bouton de destination d_i , par y_i l'état d'une étape activée par un bouton h_i d'appel vers le haut et par z_i l'état d'une étape activée par un bouton b_i d'appel vers le bas. Ces 3 mémoires sont désactivées par le bouton a_i d'arrivée de la cabine au niveau désirée i. Pour $i = 0, 1, 2, 3$, les échelles de ces mémoires sont les suivantes.

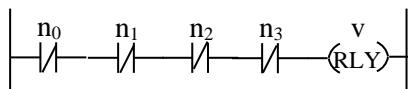


x_H est l'état d'une bascule qui s'active quand l'ascenseur monte et se désactive quand il descend. À

noter que y_4 et z_1 sont toujours nuls puisque les boutons h_4 et b_1 n'existent pas. L'adresse n_i , $i = 0, 1, 2$ et 3 , d'une sortie indiquant si le niveau i est demandé ou non par un bouton de destination ou un bouton d'appel se déduit du sous-programme suivant,

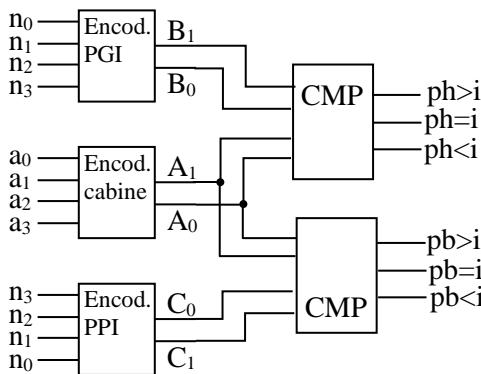


L'adresse v de l'échelon suivant indique que la cabine n'est pas demandée.



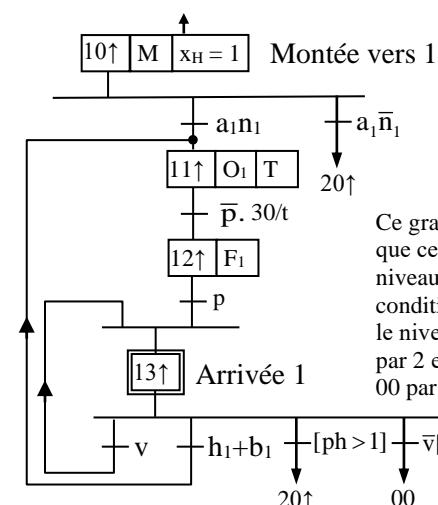
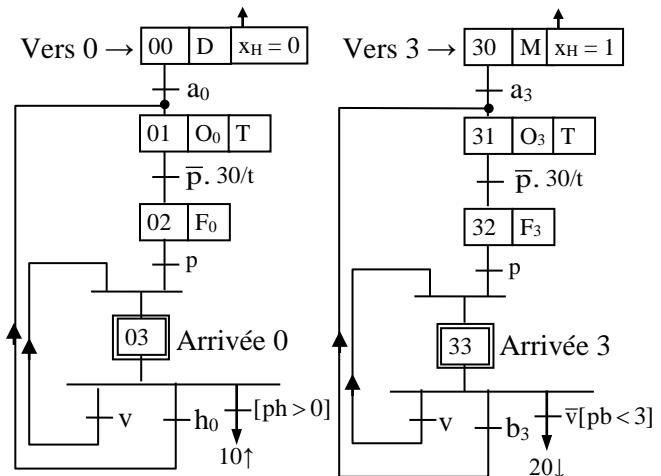
b) Mouvement de la cabine.

Le circuit externe suivant indique à l'automate si la position de la cabine est supérieure, égale ou inférieure à la position du plus haut (ph) ou du plus bas (pb) niveau demandé. Rappelons que la sortie binaire d'un encodeur est le plus grand indice (PGI) des entrées actionnées. Si l'ordre des entrées est dans le sens opposé au sens des indices de l'encodeur, il donne le numéro de l'entrée actionnée ayant le plus petit indice (PPI). a_i est le bouton d'arrêt de la cabine au niveau i . C'est à ce niveau que l'ascenseur décide de monter ou de descendre.

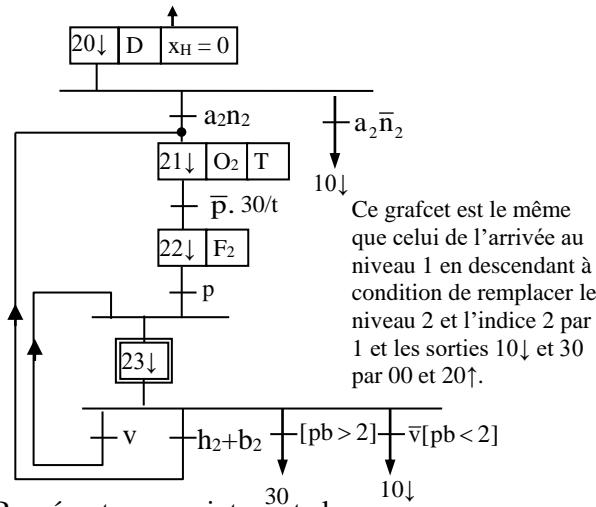
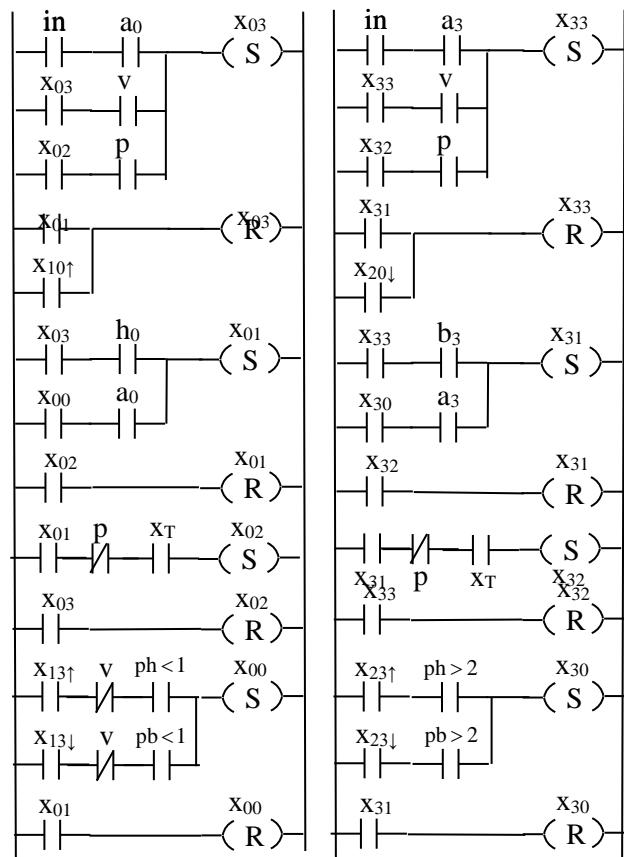
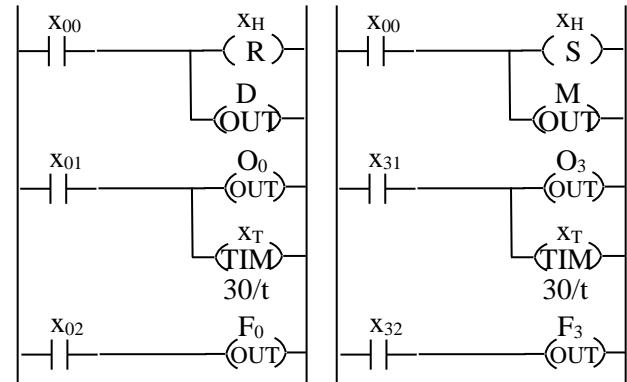
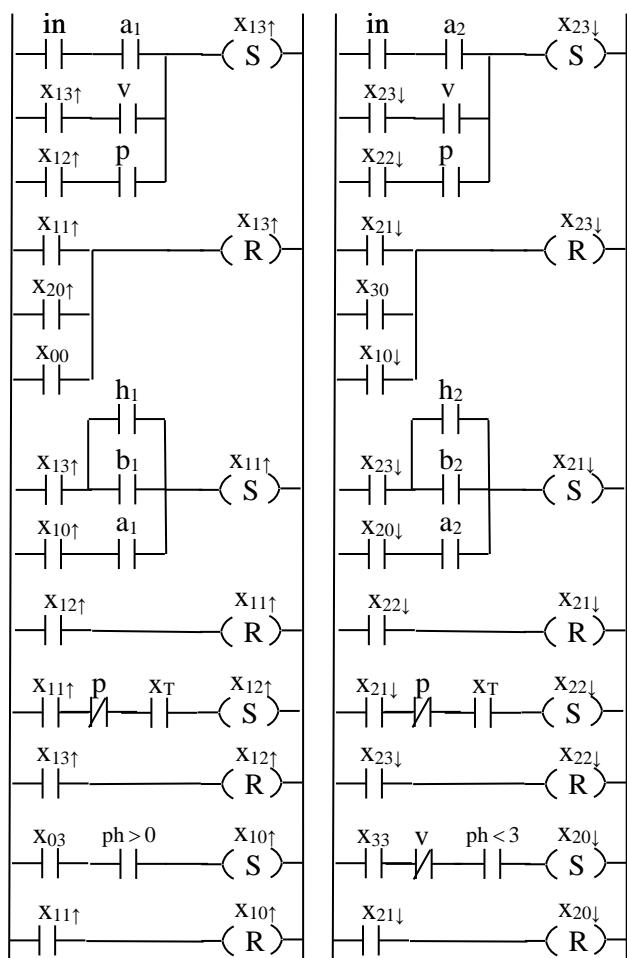


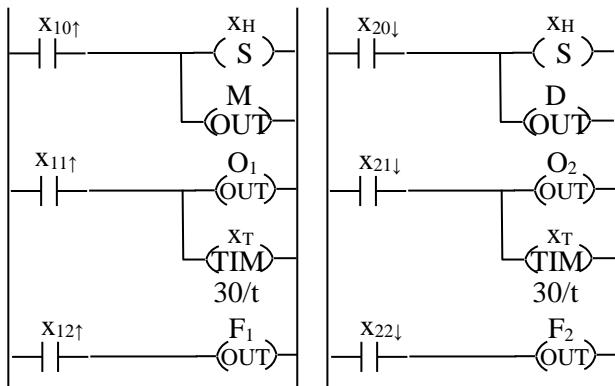
Souvent un PLC peut effectuer la comparaison entre les valeurs de deux registres ce qui permet de remplacer le circuit extérieur précédent par un sous-programme.

Nous passons maintenant au mouvement de la cabine en associant à chaque niveau i un sous-programme qui commence avant l'arrivée de la cabine à ce niveau jusqu'à son départ vers un autre niveau. Pour déterminer le sens du déplacement de la cabine à partir du niveau i , il est nécessaire de savoir si elle est arrivée en i en montant ou en descendant. Pour cette raison, nous allons distinguer entre ces deux possibilités mais il est évident qu'au niveau 0 la cabine ne peut que monter et au niveau 3 elle ne peut que descendre. Les grafcets suivants montrent l'évolution du système à partir de l'arrivée de la cabine au niveau $i = 0, 1, 2, 3$.



Ce grafcet est le même que celui de l'arrivée au niveau 2 en montant à condition de remplacer le niveau 1 et l'indice 1 par 2 et les sorties 20↑ et 00 par 30 et 10↓.

États aux niveaux 0 et 3Sorties aux niveaux 0 et 3.États au niveaux 1↑ et de 2↓

Sorties aux niveaux 1↑ et 2↓

Ces échelles sont les mêmes que celles de l'arrivée au niveau 2 en montant à condition de remplacer le premier

Ces échelles sont les mêmes que celles de l'arrivée au niveau 1 en descendant à condition de remplacer le premier indice 2 par 1 et les états $x_{10\downarrow}$ et x_{30} par x_{00} et $x_{20\uparrow}$.