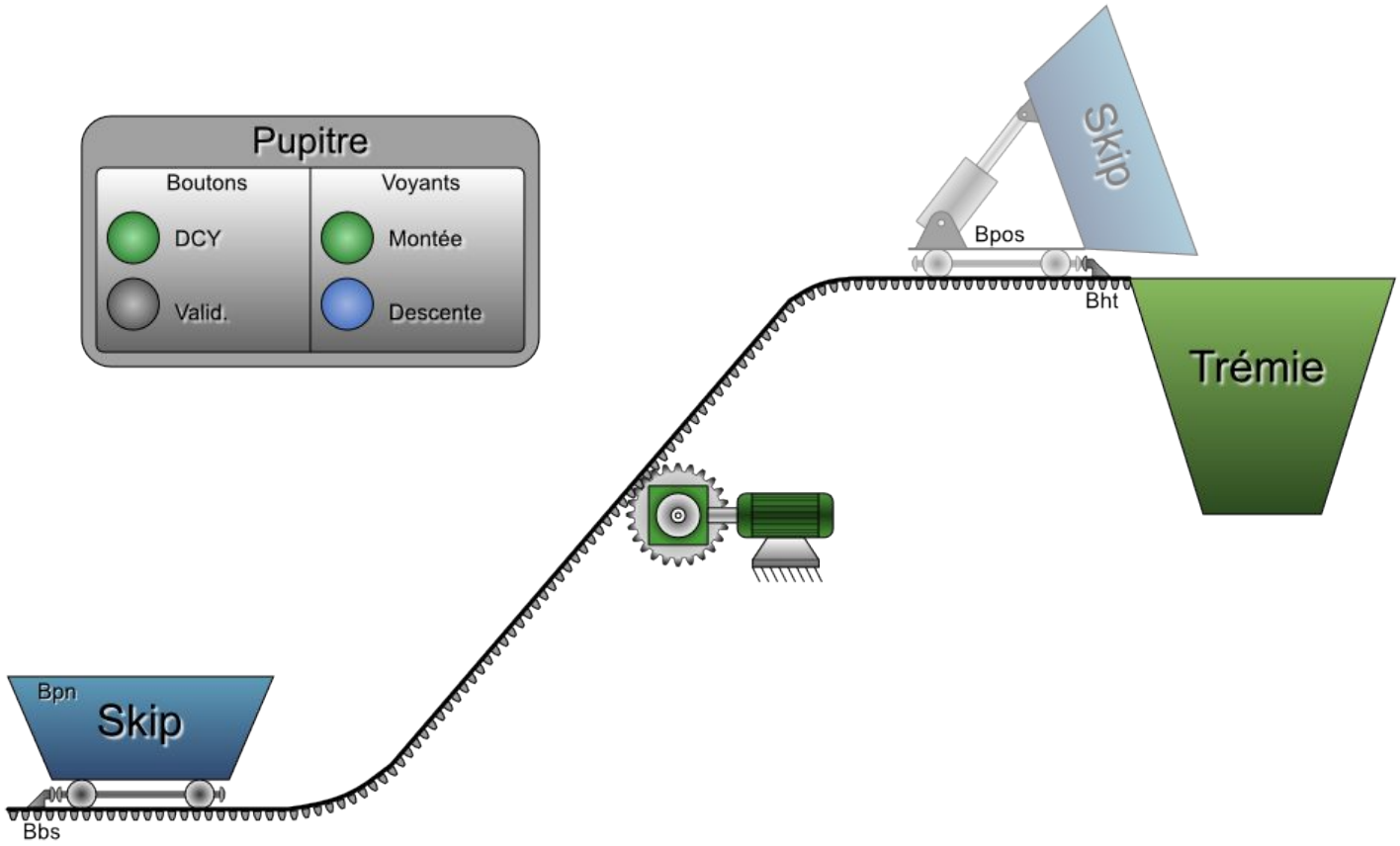
PROBLÉMATIQUE DE MAINTENANCE

Une entreprise d'exploitation minière désire automatiser une benne monte charge « SKIP ».

Dans une première analyse, l'étude du GRAFCET point de vue commande a été réalisée.

Vous allez devoir rédiger le programme automate permettant de réaliser le cycle de fonctionnement du Skip.

I. Principe de fonctionnement :



II. Description du cycle :

Lorsque la benne est pleine (Bpn), en bas du rail (Bbs), l'opérateur peut la faire monter (moteur électrique asynchrone triphasé M1) vers la trémie en appuyant sur le bouton dcy.

Une fois la benne en haut (Bht), un vérin (double effet 1A) permet de la vidanger, lorsque celle-ci est vidanger, l'opérateur doit valider l'opération à l'aide d'un bouton poussoir (Valid.).

Le vérin rentre et remet la benne en position (Bph), lorsque la benne a basculé en position horizontale, celle-ci redescend (Bbs).

Deux voyants signaleront la phase de montée et de descente de la benne.

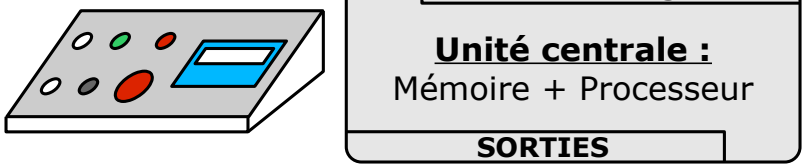
Remarque : le système de déverrouillage de la trappe de la trémie n'est pas pris en compte dans le cycle.

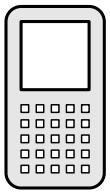
NOTIONS DE BASE

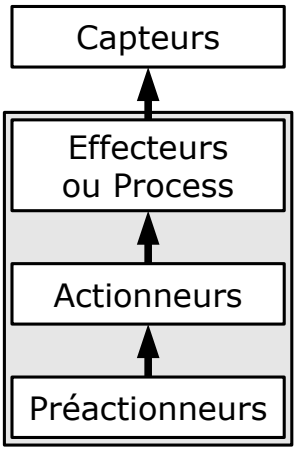
I. Structure d'un Automate Programmable Industriel (A.P.I.) :

Un automate se décompose en quatre sous-ensembles :

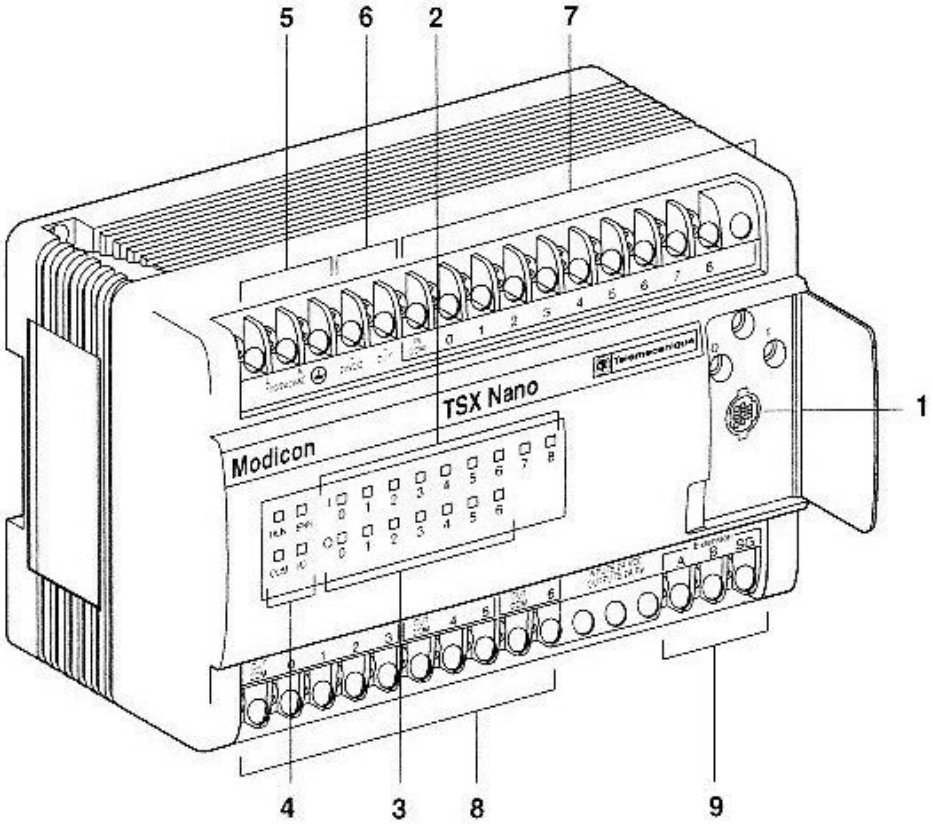
- Les Entrées ;
- Les Sorties ;
- La mémoire où sont stockées les instructions du programme utilisateur ;
- Le processeur qui lit les informations d'entrées et commande les sorties en fonction des instructions du programme utilisateur.




Console de programmation



II. Exemple d'automate : TSX Nano :



1 Raccordement terminal de programmation (Console FTX 117, ordinateur)

2 Visualisation de l'état des _____

3 Visualisation de l'état des _____

4 Visualisation de l'état de l'automate : RUN, ERR, COM, I/O

5 Raccordement de _____

6 Alimentation capteurs : =24V/150mA

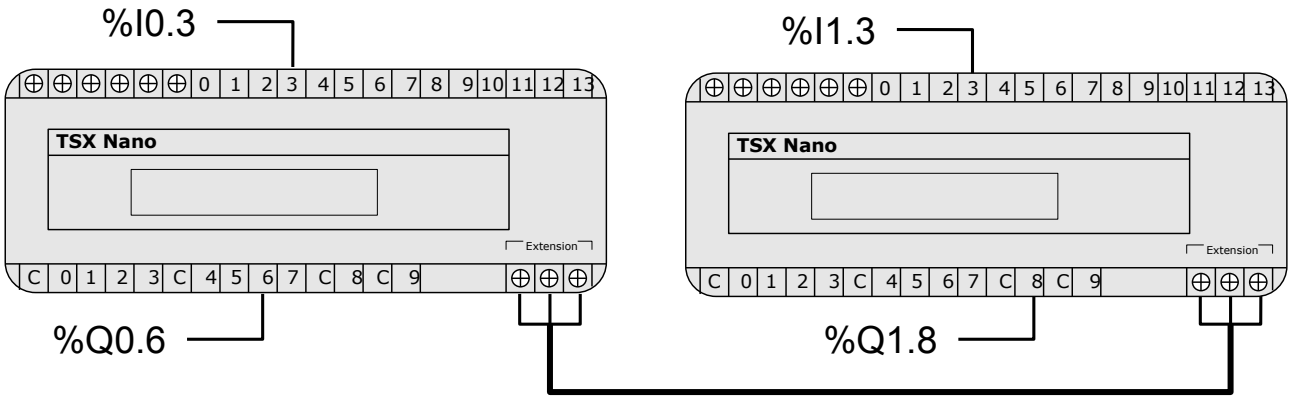
7 Raccordement des _____

8 Raccordement des _____

III. Adressage Entrées /Sorties :

III.1) L'adressage des Entrées / Sorties :

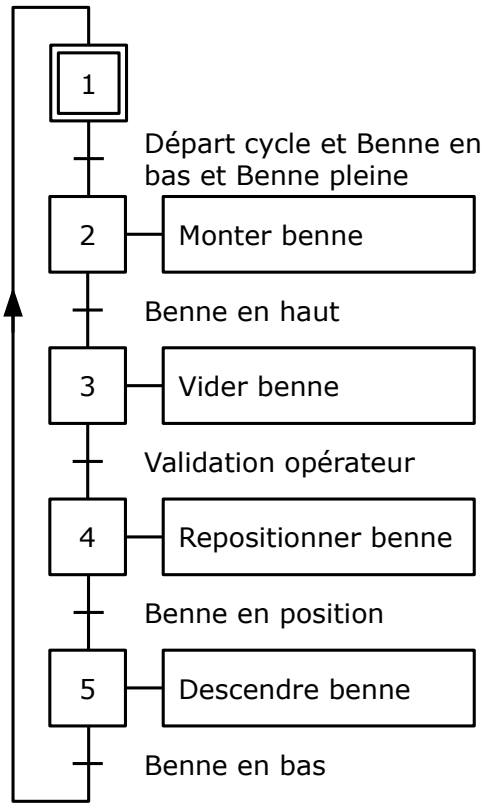
Automate de base **Extension d'Entrées/Sorties**



%	I ou Q	0 ou 1	.	i
symbole	I = _____ Q = _____	0 = _____ _____ 1 = _____ _____	point	i = _____ _____
%	**I**	**0**	**.**	**3**
%	**Q**	**1**	**.**	**6**
%	**I**	**1**	**.**	**0**

DOSSIER TECHNIQUE

I. GRAFCET point de vue Procédé (ou système)



II. Adressage des Entrées :

	Repère	Entrée
Départ cycle	Dcy	%I0.0
Validation opérateur	Bp Valid.	%I0.1
Benne pleine	Bpn	%I0.2
Benne en Bas	Bbs	%I0.3
Benne en Haut	Bht	%I0.4
Benne en position	Bpos	%I0.5

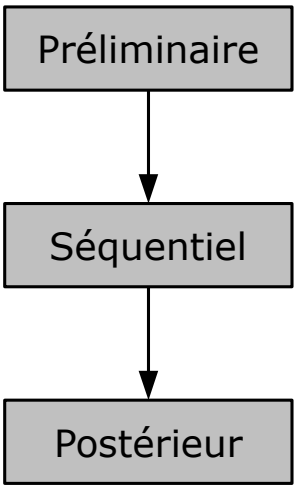
III. Adressage des Sorties :

	Préactionneur/Voyant		Tension commande	Sortie
	Repère	Pilote		
Montée Benne	KM1		24VAC	%Q0.0
Descente Benne	KM2		24VAC	%Q0.1
Vider Benne	1V1	1Y1.14	24VAC	%Q0.2
Repositionner Benne		1Y1.12	24VAC	%Q0.3
Voyant « Montée »	H1		24VAC	%Q0.4
Voyant « Descente »	H2		24VAC	%Q0.5

INTRODUCTION À LA PROGRAMMATION

I. Introduction à la programmation :

1. Le programme automate dépend strictement du GRAFCET point de vue API.
2. Il peut être rédigé à l'aide de deux formes de langage:
   - Le langage Liste d'instructions « _____ » ;
   - Le langage à contact « _____ ».
3. La saisie du programme s'effectue à l'aide d'un ordinateur ou d'une console de programmation :
   - directement connecté à l'automate ;
   - non connecté à l'automate et transféré ensuite.
4. Le programme peut être divisé en trois parties distinctes :



Le traitement **PRÉLIMINAIRE** permet de traiter:

- les reprises secteur,
- les défaillances,
- le prépositionnement du graphe, ...

Le traitement **SÉQUENTIEL** décrit la structure du GRAFCET :

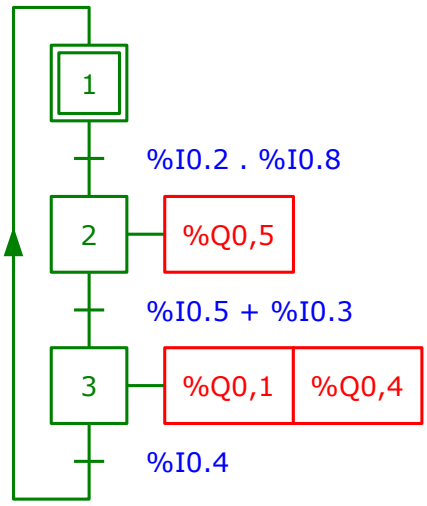
- _____
- _____ et _____ associées

Le traitement **POSTÉRIEUR** permet d'activer les _____ associées aux étapes.

II. Éléments principaux de programmation :

Exemple :

GRAFCET API **Langage LIST**



Traitement séquentiel

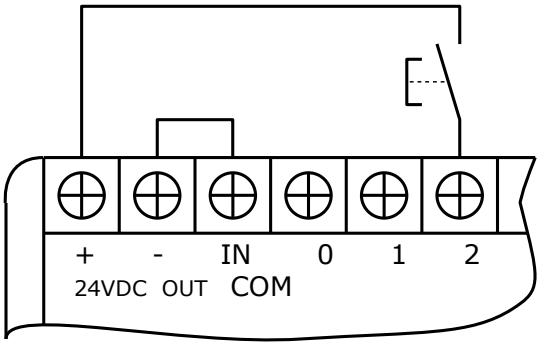
Traitement postérieur

N°	Code instruction	Opérande
000		
001		
002		
003		
004		
005		
006		
007		
008		
09		
010		
011		
012		
013		
014		
015		
016		
017		

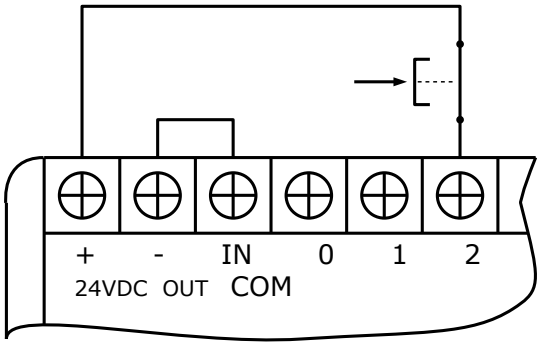
Remarque : L'instruction ST est de type monostable. Dès que les conditions nécessaires à la mise à 1 d'un bit par une bobine ST ne sont plus validées, ce bit retombe à 0.

Le bit d'entrée %I0.2 est considéré à 1 lorsque le contact associé à son entrée est fermé.

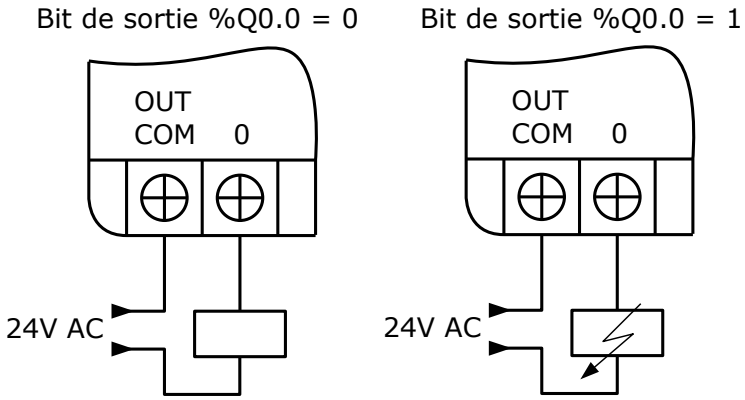
Bit d'entrée %I0.2 = 0



Bit d'entrée %I0.2 = 1



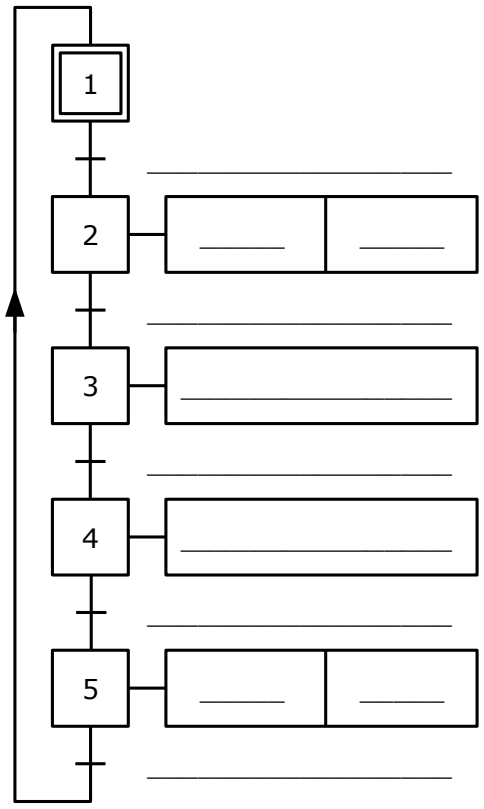
Le bit de sortie %Q0.0 est mis à 1 par le programme. Lorsque ce bit est à 1, il permet d'alimenter le récepteur qui est raccordé à la sortie associée.



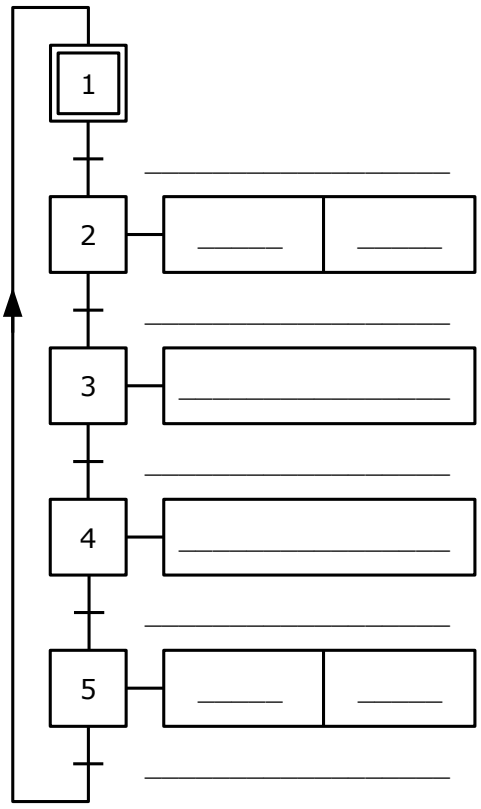
TRAVAIL DEMANDÉ

I. Déduire le GRAFCET point de vue API à partir du GRAFCET point de vue Commande :

I.1) GRAFCET Point de vue Partie Commande



I.2) GRAFCET Point de vue API



II. Rédiger le programme en langage « LIST » :

II.1) Programmation :

N°	Code instruction	Opérande
000		
001		
002		
003		
004		
005		
006		
007		
008		
009		
010		
011		
012		
013		
014		
015		
016		
017		
018		
019		
020		
021		
022		
023		
024		
025		
026		
027		
028		
029		
030		
031		
032		
033		
034		
035		

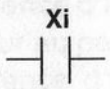
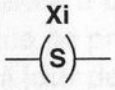
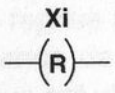
II.2) Correction :

N°	Code instruction	Opérande
000		
001		
002		
003		
004		
005		
006		
007		
008		
009		
010		
011		
012		
013		
014		
015		
016		
017		
018		
019		
020		
021		
022		
023		
024		
025		
026		
027		
028		
029		
030		
031		
032		
033		
034		
035		

2.3 Instructions Grafcet

2.3-1 Description

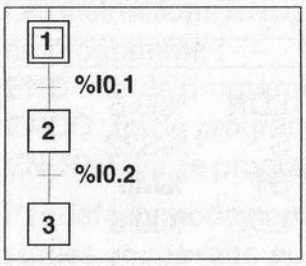
Les instructions Grafcet du langage PL7 permettent de traduire un Grafcet graphique de façon simple.
Le langage PL7 comprend 62 étapes maximum, y compris la ou les étapes initiales. Le nombre d'étapes actives simultanément n'est limité que par le nombre d'étapes.
Le tableau ci-dessous regroupe l'ensemble des instructions et objets permettant de programmer un Grafcet.

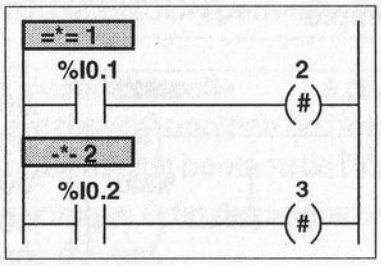
Représentation graphique	Transcription en langage PL7	Rôle
 Etape initiale Transition Etape	=*= i	Début de l'étape initiale (1)
	# i	Activation de l'étape i après désactivation de l'étape en cours
	-*- i	Début de l'étape i et validation de la transition associée (1)
	#	Désactivation de l'étape en cours sans activation de tout autre étape
	#Di	Désactivation de l'étape en cours et de l'étape i spécifiée.
	=*= POST	Début du traitement postérieur et fin du traitement séquentiel
	%Xi	Bit associé à l'étape i, peut être testé n'importe où dans le programme mais ne peut être écrit que dans le traitement préliminaire (nombre d'étapes maxi : 62).
	LD %Xi, LDN %Xi, AND %Xi, ANDN %Xi OR %Xi, ORN %Xi XOR %Xi, XORN %Xi	Test de l'activité de l'étape i
	S %Xi	Activation de l'étape i
	R %Xi	Désactivation de l'étape i

(1) la première étape =*=i ou -*-i écrite indique le début du traitement séquentiel donc la fin du traitement préliminaire.

Exemples

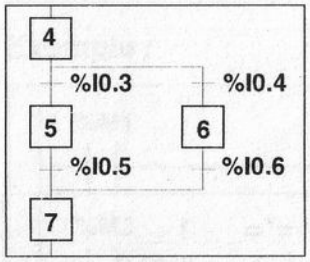
Séquence linéaire

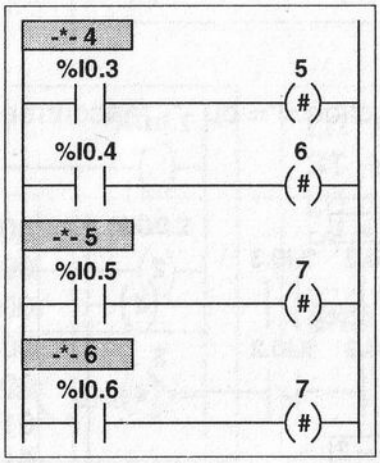





```
=*=1
LD %I0.1
#  2
-*- 2
LD %I0.2
#  3
```

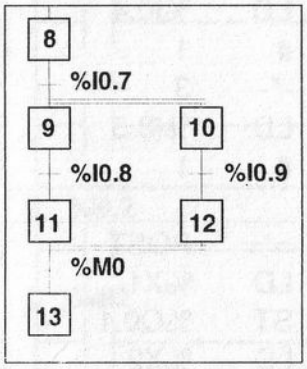
Aiguillage



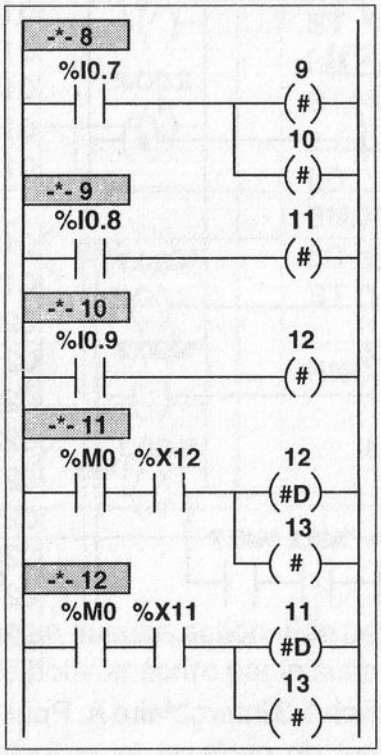



```
-*- 4
LD %I0.3
#  5
LD %I0.4
#  6
-*- 5
LD %I0.5
#  7
-*- 6
LD %I0.6
#  7
```

Séquences simultanées



Note :
Pour qu'un Grafcet soit opérationnel, il est nécessaire de déclarer au minimum une étape intiale =*=i ou de prépositionner le graphe dans le traitement préliminaire à l'aide du bit système %S23. Voir exemple annexe A.10 intercalaire G.




```
-*- 8
LD %I0.7
#  9
#  10
-*- 9
LD %I0.8
#  11
-*- 10
LD %I0.9
#  12
-*- 11
LD %M0
AND %X12
#D 12
#  13
-*- 12
LD %M0
AND %X11
#D 11
#  13
```