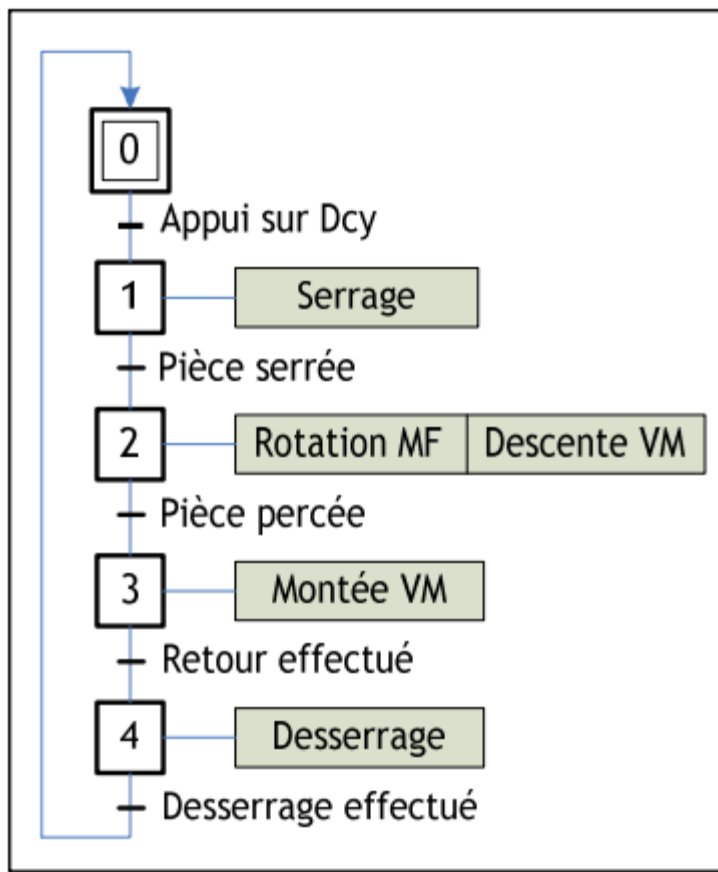


Solution du TD N°4 (3ème partie)

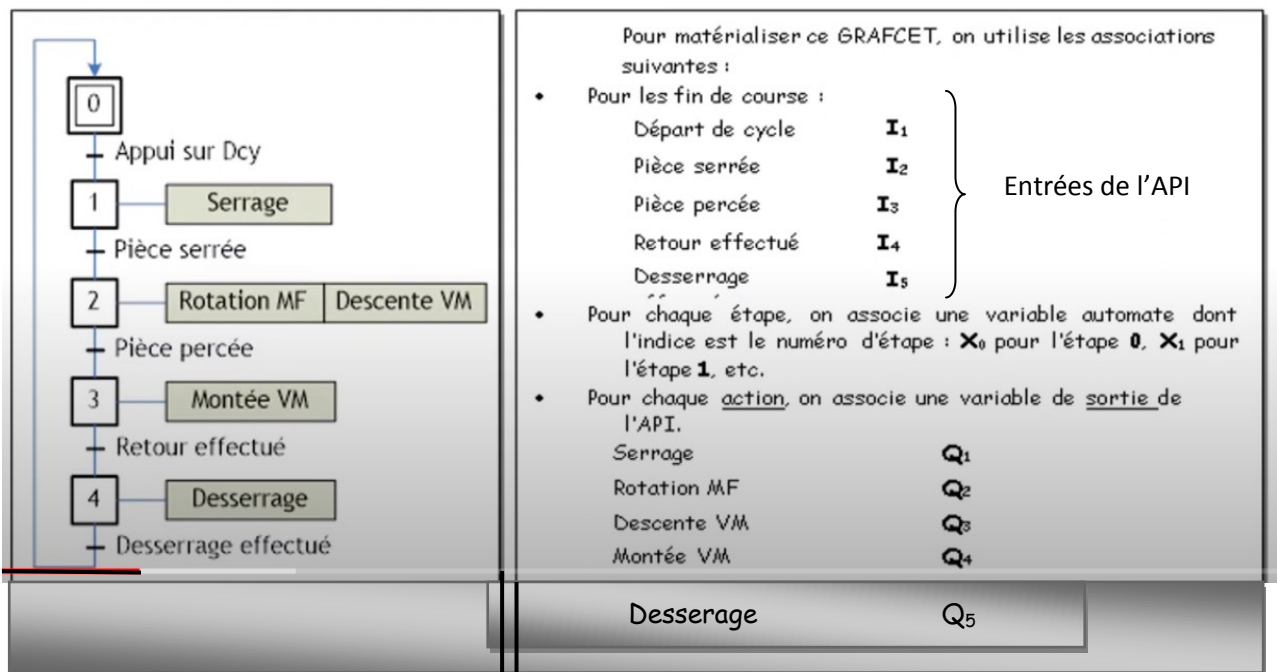
I- Langage de programmation Ladder ou à contacts

Solution de l'Exercice 1 :

Le grafcet de la perceuse automatisée



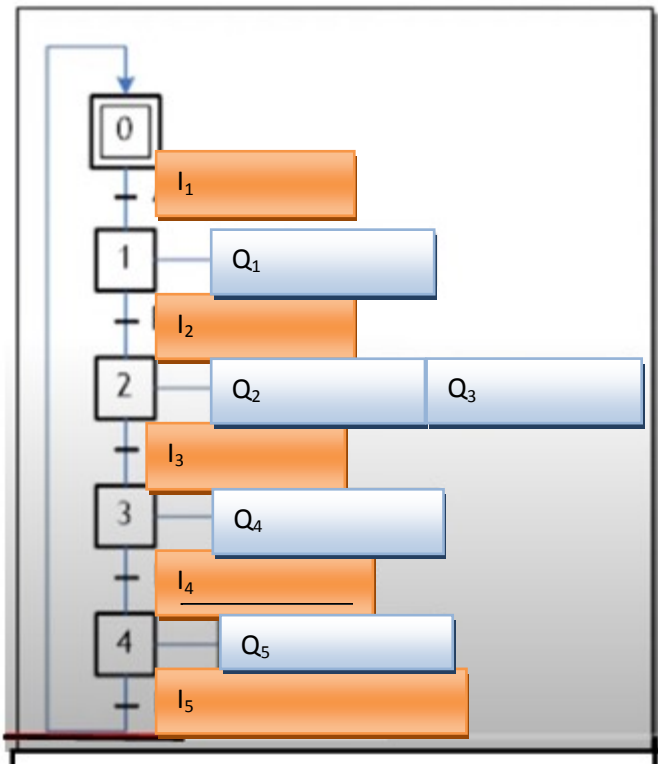
On traduit le grafcet en Ladder



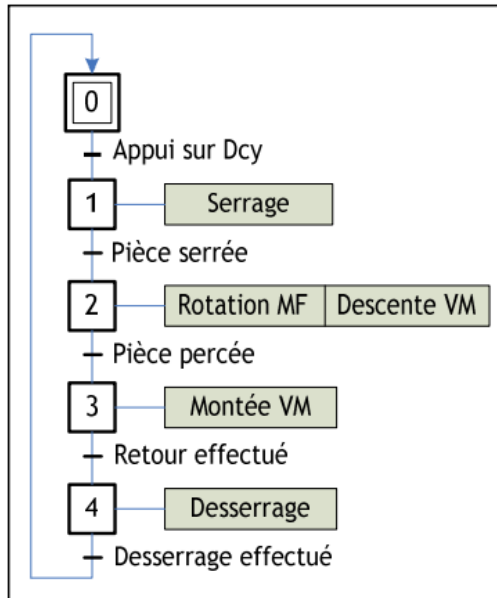
Pour coder le Grafcet en API, on procède comme suit :

- Les receptivités dans le Grafcet sont remplacés par les entrées API
- Les actions dans le Grafcet sont remplacées par les sorties API

Le grafcet codé API est donné par ou les receptivités sont remplacées par entrées API et les actions par les sorties API



Grafcet codé API



Pour matérialiser ce GRAFCET, on utilise les associations suivantes :

- Pour les fin de course :

Départ de cycle	% I0.1
Pièce serrée	% I0.2
Pièce percée	% I0.3
Retour effectué	% I0.4
Desserrage effectué	% I0.5
- Pour chaque étape, on associe une variable automate dont l'indice est le numéro d'étape : X0 pour l'étape 0, X1 pour l'étape 1, etc.
- Pour chaque action, on associe une variable de sortie de l'API.

Serrage	% Q1.1
Rotation MF	% Q1.2
Descente VM	% Q1.3
Montée VM	% Q1.4
Desserrage	% Q1.5

Les étapes :

Equations d'activation (Set) :

Pour l'étape 1 :

$$A1 = I_1 X_0$$

Pour l'étape 2 :

$$A2 = I_2 X_1$$

Pour l'étape 3 :

$$A3 = I_3 X_2$$

Pour l'étape 4 :

$$A4 = I_4 X_3$$

Pour l'étape 0 :

$$A0 = I_5 X_4 + CI$$

Equations de désactivation(Reset) :

Pour l'étape 1 :

$$D1 = X_2$$

Pour l'étape 2 :

$$D2 = X_3$$

Pour l'étape 3 :

$$D3 = X_4$$

Pour l'étape 4 :

$$D4 = X_0$$

Pour l'étape 0 :

$$D0 = X_1$$

Les actions

$$Q_1 = X_1$$

$$Q_2 = X_2$$

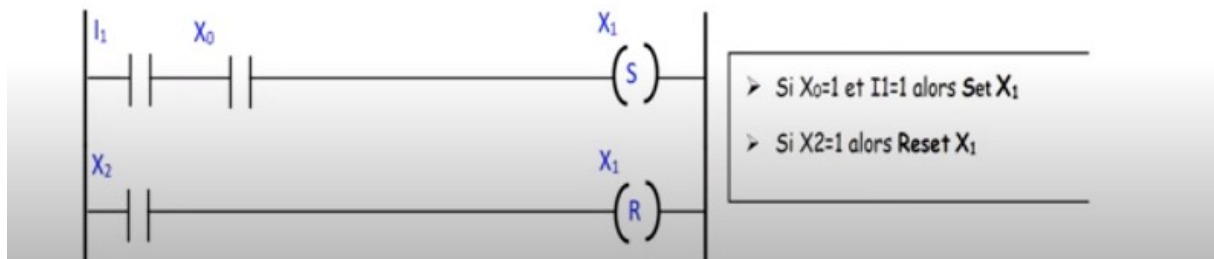
$$Q_3 = X_2$$

$Q_4 = X_3$

$Q_5 = X_4$

Les Etapes

Pour l'étape 1



Pour l'étape 2



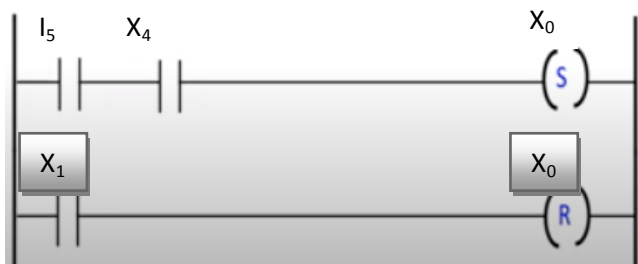
Pour l'étape 3



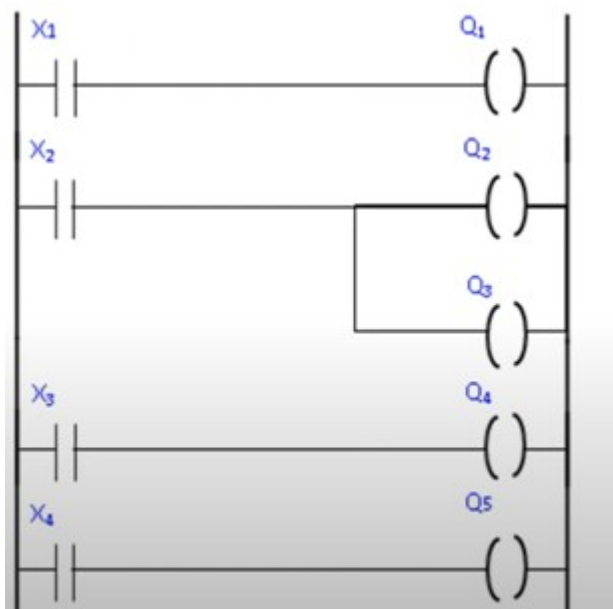
Pour l'étape 4

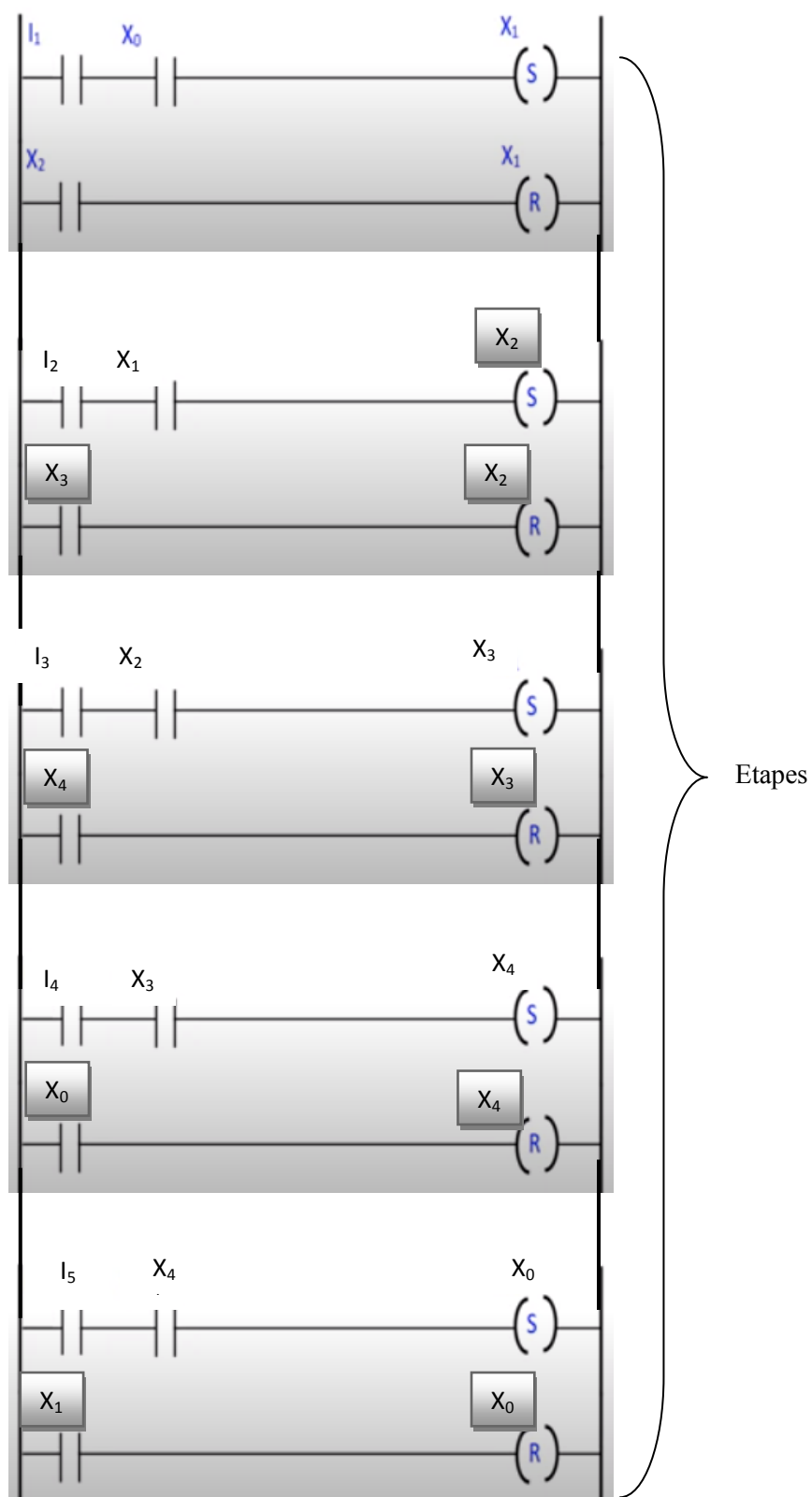


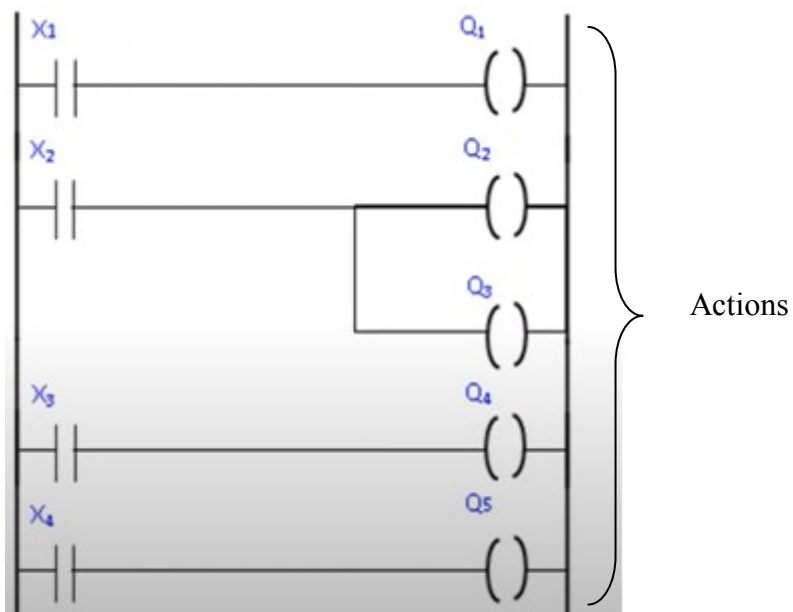
Pour l'étape 0



Les actions(traitement postérieur activation des sorties)



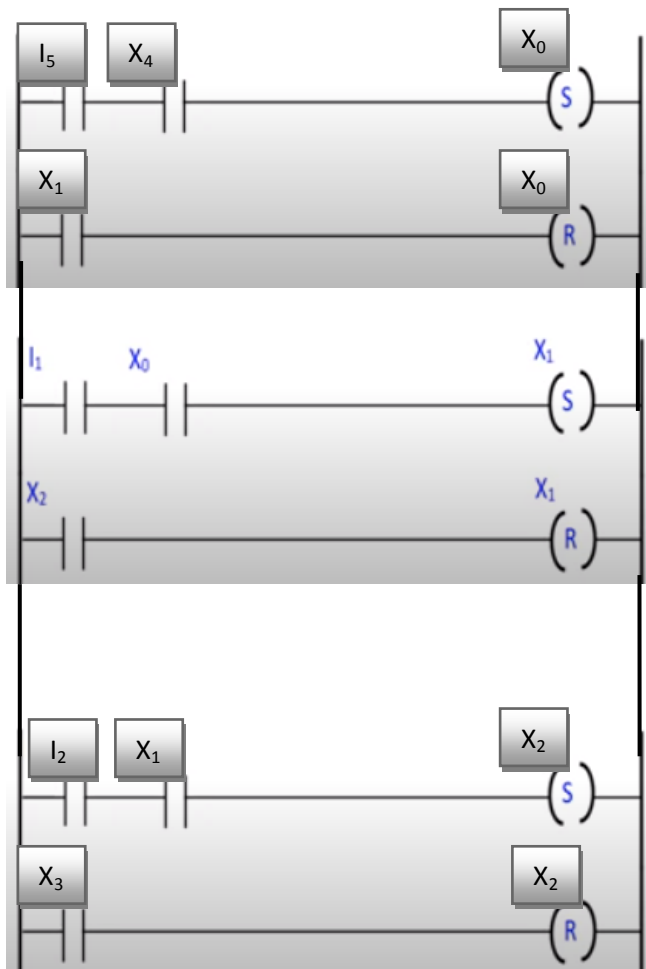


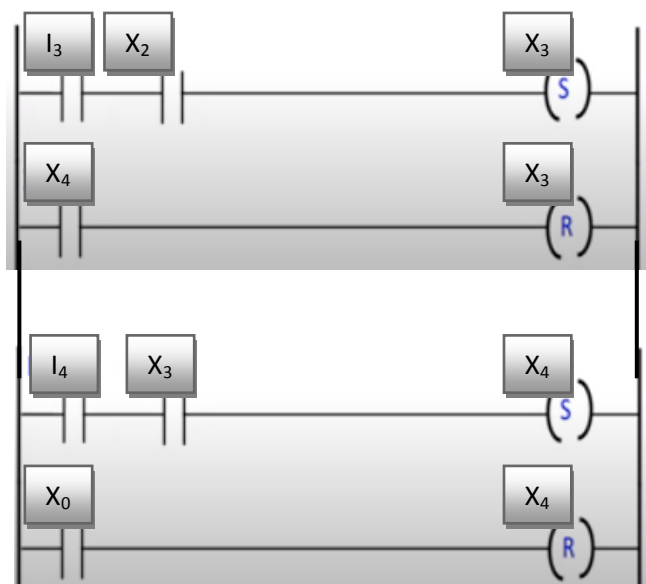


II- Liste d'instructions "IL" :

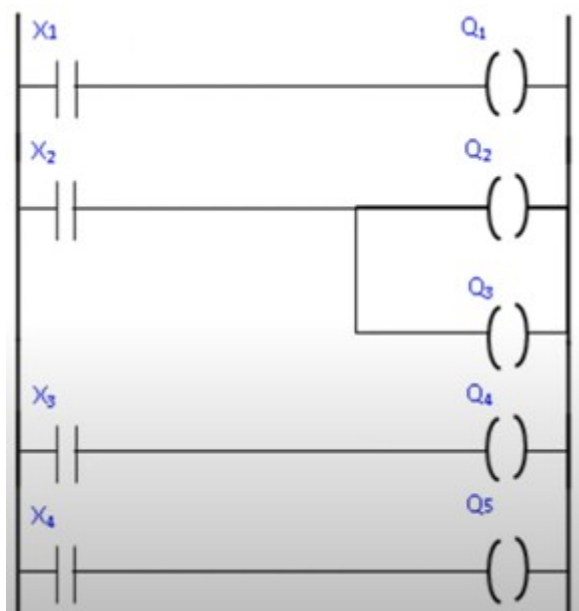
Solution de l'Exercice 2 :

Les étapes





Les actions



Le programme suivant est une mise en œuvre du GRAFCET de l'exemple de la perceuse en langage IL

Etapes

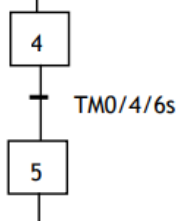
```
LD%I0.5
AND %X4
S %X0
R %X1
LD%I0.1
AND %X0
S %X1
R %X2
LD%I0.2
AND %X1
S %X2
R %X3
LD%I0.3
AND %X2
S %X3
R %X4
```

Actions: traitement postérieur

```
LD    % X1
S     % Q1.1
LD    % X2
S     % Q1.2
S     % Q1.3
LD    % X3
S     % Q1.4
LD    % X4
S     % Q1.5
END
```

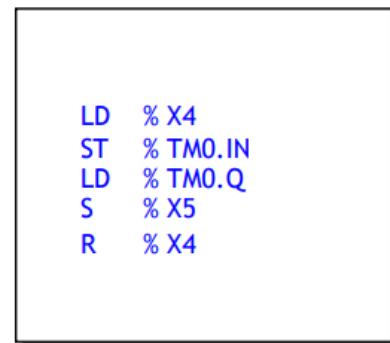
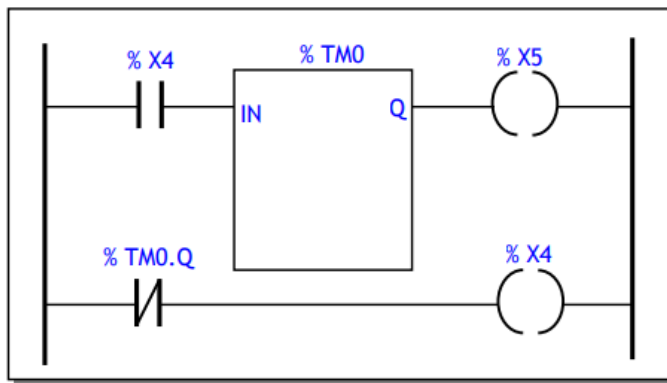
Solution de l'Exercice 3 :

Exemple :



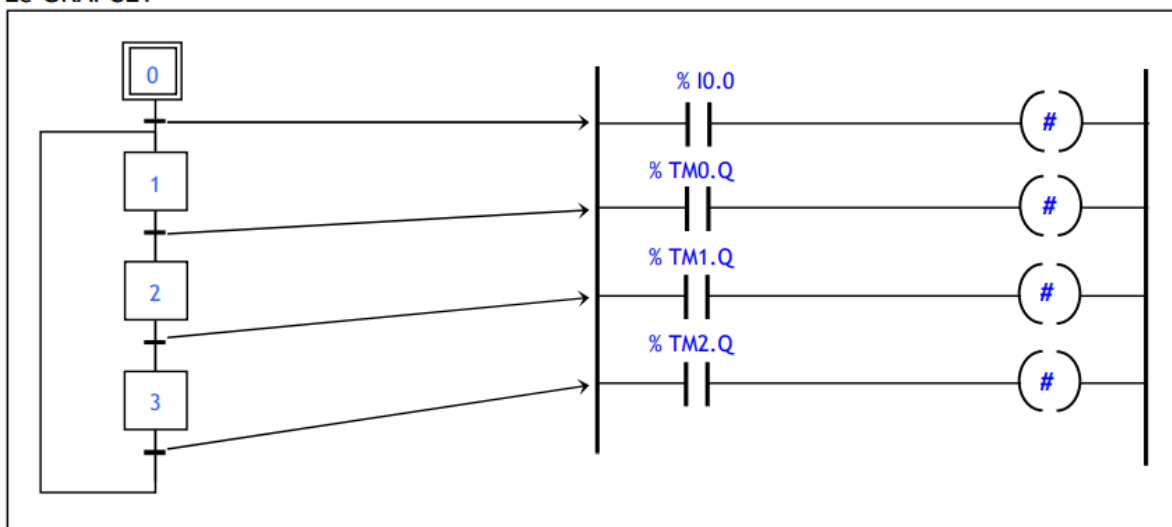
- Le timer TM0 est lancé par la variable d'étape X4.
- Quand la sortie du timer TM0.Q passe à 1 il doit activer l'étape X5 et désactiver l'étape X4.
- La désactivation de X4 permet d'initialiser le timer pour le préparer pour une nouvelle utilisation.

L'interprétation de ceci en LADDER et en IL se fait comme suit :

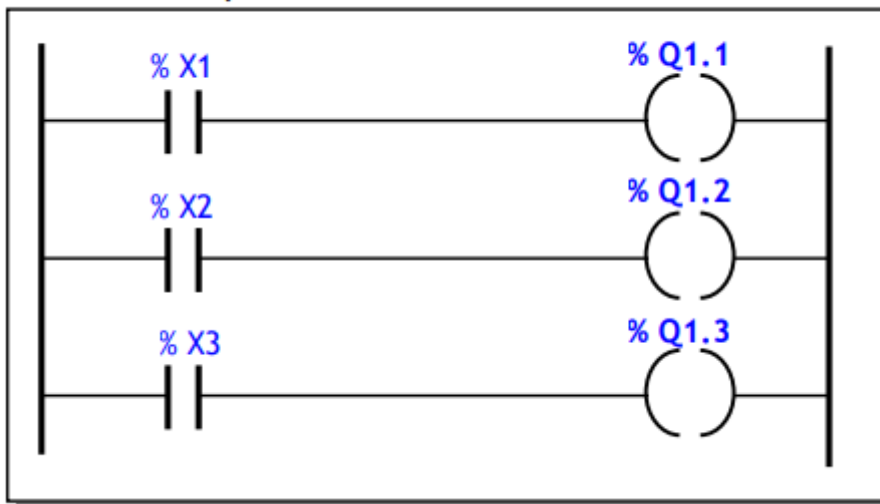


On propose dans ce qui suit la correction de l'application en traduisant les réceptivités et le postérieur en LADDER.

Le GRAFCET



Traitement postérieur



Préliminaire : Initialisation des compteurs

