Automatisation Industrielle

Niveau 1

3.Programmation d'un Automate Programmable Industriel

Syllabus de la séance J4

Automatisation Industrielle – Niveau 1

3.Programmation d'un Automate Programmable Industriel (J4)

- 3.1.Passage du GRAFCET aux équations logiques
- 3.2.les bases d'un code LD (LADDER Diagram)

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

A. Introduction

Les langages de programmation des API sont de natures diverses étant donné la diversité, des utilisateurs pouvant les utiliser.

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

B. Le langage LADDER (Ladder Diagram)

Le langage des API d’origine américaine utilise le symbolisme classique des schémas à relais accompagné de blocs graphiques préprogrammés pour réaliser des fonctions d’automatisme (calculs, temporisation, compteur,.....).

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

B. Le langage LADDER (Ladder Diagram)

C'est une suite de réseaux qui seront parcourus séquentiellement. Les entrées sont représentées par des interrupteurs **-| |-** ou **-|/|-** si entrée inversée, les sorties par des bobines **-( )-** ou des bascules **-(S)- -(R)-**Il y a également d'autre opérations :

- l'inverseur **- | NOT | -**,
- l'attente d'un front montant **-(P)-** ou descendant **-(N)-**.

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

B. Le langage LADDER (Ladder Diagram)

Les sorties sont obligatoirement à droite du réseau On doit évidemment identifier nos **E/S**, soit directement par leur code (**Ia.b** / **Qa.b**), ou avec leur libellé en clair défini dans la table des mnémoniques.

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

B. Le langage LADDER (Ladder Diagram)

On relie les éléments en série pour la fonction **ET**, en parallèle pour le **OU**. On peut utiliser des bits internes (peuvent servir en bobines et interrupteurs), comme on utilise dans une calculatrice une mémoire pour stocker un résultat intermédiaire (**Ma.b**).

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

B. Le langage LADDER (Ladder Diagram)

On peut aussi introduire des éléments plus complexes, en particulier les opérations sur **bits** comme par exemple une bascule **SR** (priorité déclenchement), **RS** (priorité enclenchement), **POS** et **NEG** pour la détection de fronts... on trouvera d'autres fonctions utiles, les compteurs, les temporisateurs et le registre à décalage. On peut également utiliser des fonctions plus complexes (calculs sur mots par exemple)

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

B. Le langage LADDER (Ladder Diagram)

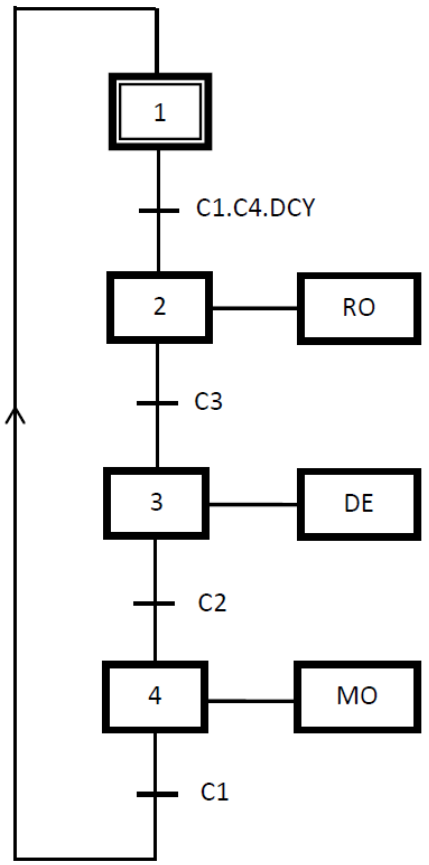
La déclaration d'une entrée ou sortie donnée à l'intérieur d'un programme s'appelle l'adressage. Les entrées et sorties des API sont la plupart du temps regroupées en groupes de huit sur des modules d'entrées ou de sorties numériques. Cette unité de huit est appelée **octet**. Chaque groupe reçoit un numéro que l'on appelle l'adresse d'octet. Afin de permettre l'adressage d'une entrée ou sortie à l'intérieur d'un octet, chaque octet est divisé en huit bits. Ces derniers sont numérotés de 0 à 7. On obtient ainsi l'adresse du bit..L'API représenté ici a les octets d'entrée 0 et 1 ainsi que les octets de sortie 0 et 1.

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

C. Adressage des entrèes/Sorties

Reprenons l’exemple du grafcet suivant :

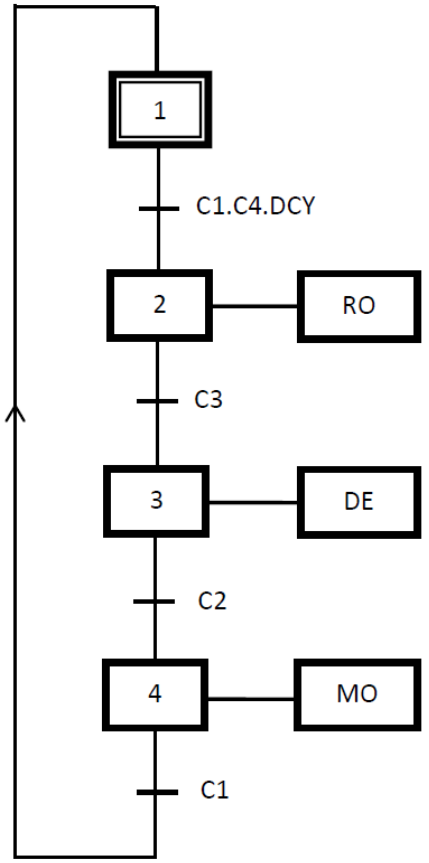


3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

C. Adressage des entrèes/Sorties

Reprenons l’exemple du grafcet suivant :

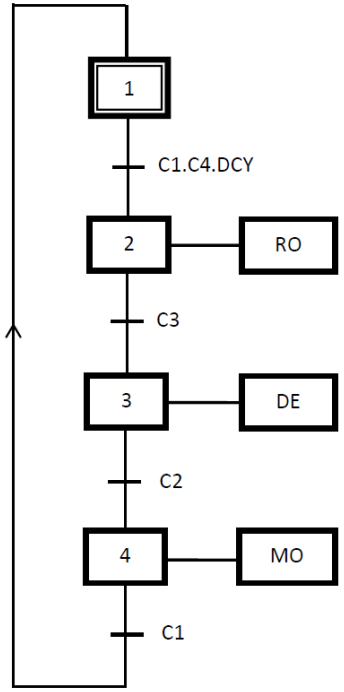


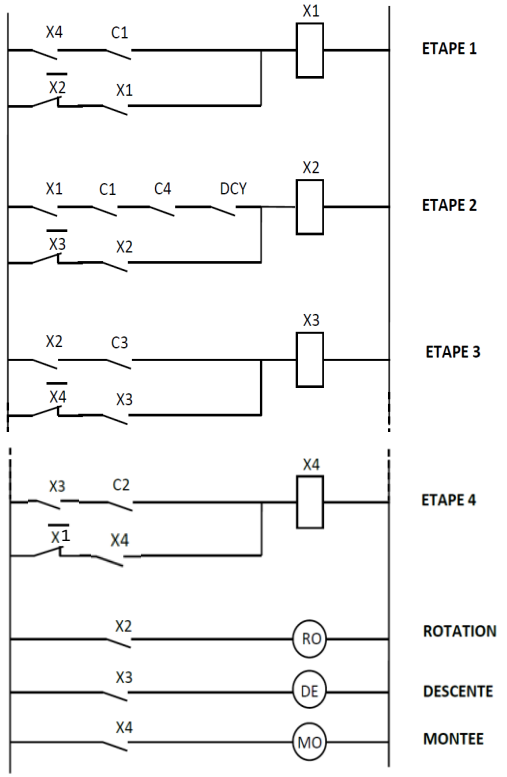
3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

C. Adressage des entrèes/Sorties

Reprenons l'exemple du grafcet suivant :



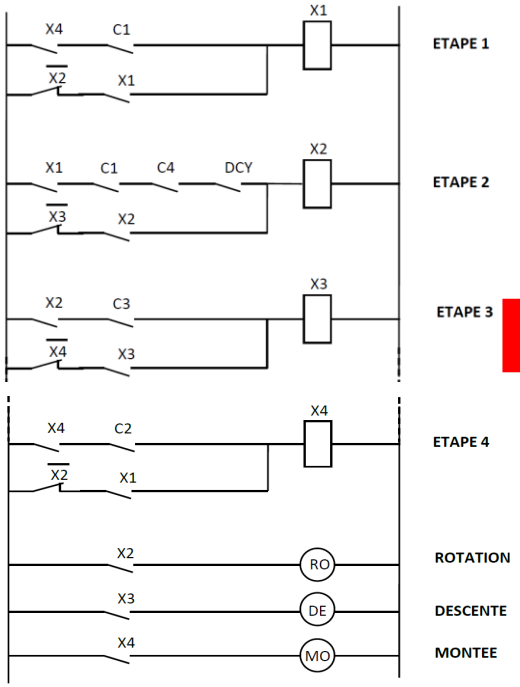


3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

C. Adressage des entrèes/Sorties

Reprenons l'exemple du grafcet suivant :



Nom	Type de données	Adresse
ETAPE 1	Bool	M0.1
ETAPE 2	Bool	M0.2
ETAPE 3	Bool	M0.3
ETAPE 4	Bool	M0.4
C1	Bool	I0.0
C2	Bool	I0.1
C3	Bool	I0.2
C4	Bool	I0.3
DCY	Bool	I0.4
RO	Bool	Q0.0
DE	Bool	Q0.1
MO	Bool	Q0.2

← *Etape 1 de type logique (Bool) affecté à la mémoire M0.1*

← *Le capteur C1 est de type logique et affecté à l'adresse I0.0*

← *La sortie DE est de type logique et affecté à l'adresse Q0.0*

Tableau 1 : table de variables mnémoniques

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

C. Adressage des entrèes/Sorties

Reprenons l'exemple du grafcet suivant :

Nom	Type de données	Adresse
ETAPE 1	Bool	M0.1
ETAPE 2	Bool	M0.2
ETAPE 3	Bool	M0.3
ETAPE 4	Bool	M0.4
C1	Bool	I0.0
C2	Bool	I0.1
C3	Bool	I0.2
C4	Bool	I0.3
DCY	Bool	I0.4
RO	Bool	Q0.0
DE	Bool	Q0.1
MO	Bool	Q0.2

← *Etape 1 de type logique (Bool) affecté à la mémoire M0.1*

← *Le capteur C1 est de type logique et affecté à l'adresse I0.0*

← *La sortie DE est de type logique et affecté à l'adresse Q0.0*

Tableau 1 : table de variables mnémoniques

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

C. Adressage des entrèes/Sorties

Reprenons l'exemple du grafcet suivant :

Par exemple, pour adresser la 5ème entrée du **DCY** en partant de la gauche, on définit l'adresse suivante :

I0.4 **I** indique une adresse de type entrée, **0**, l'adresse d'octet et **4**, l'adresse de bit. Les adresses d'octet et de bit sont toujours séparées par un point.

Pour adresser la 3ème sortie, par exemple, on définit l'adresse suivante :

Q0.2 **Q** indique une adresse de type Sortie, **0**, l'adresse d'octet et **2**, l'adresse

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

C. Adressage des entrèes/Sorties

Reprenons l'exemple du grafcet suivant :

Par exemple, pour adresser la 5ème entrée du **DCY** en partant de la gauche, on définit l'adresse suivante :

I0.4 **I** indique une adresse de type entrée, **0**, l'adresse d'octet et **4**, l'adresse de bit. Les adresses d'octet et de bit sont toujours séparées par un point.

Pour adresser la 3ème sortie, par exemple, on définit l'adresse suivante :

Q0.2 **Q** indique une adresse de type Sortie, **0**, l'adresse d'octet et **2**, l'adresse de bit. Les adresses d'octet et de bit sont toujours séparées par un point.

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

C. Adressage des entrèes/Sorties

Reprenons l'exemple du grafcet suivant :

Remarque : *L'adresse du bit de la dixième sortie est un **1** car la numérotation commence à zéro.*

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

D. Application

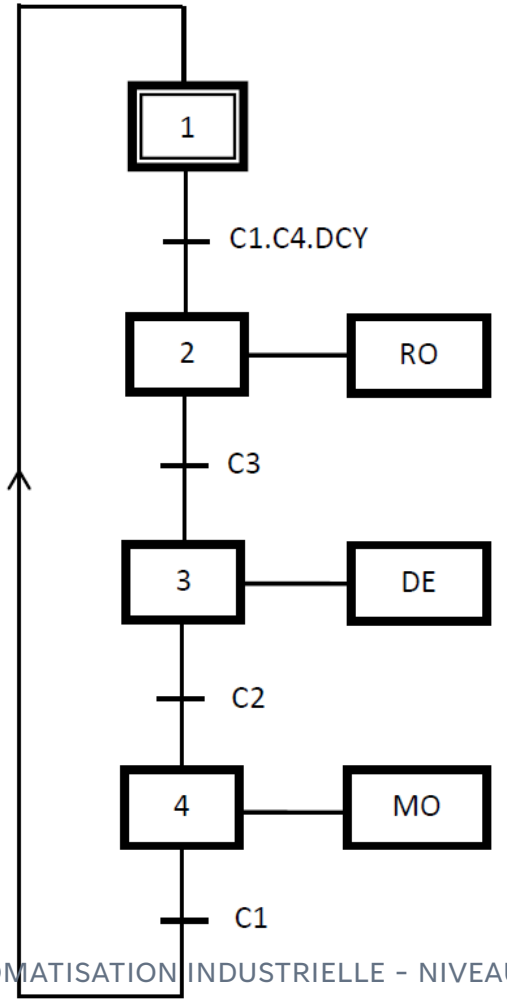
Dans l’exemple précédant et suivant la table mnémonique d’affectation le programme en LADER de la première étape est :

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

D. Application

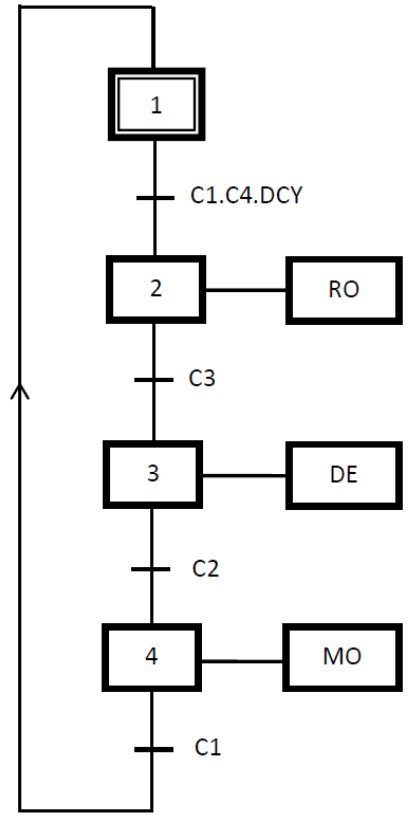
Soit le GRAFCET suivant :



3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

D. Application

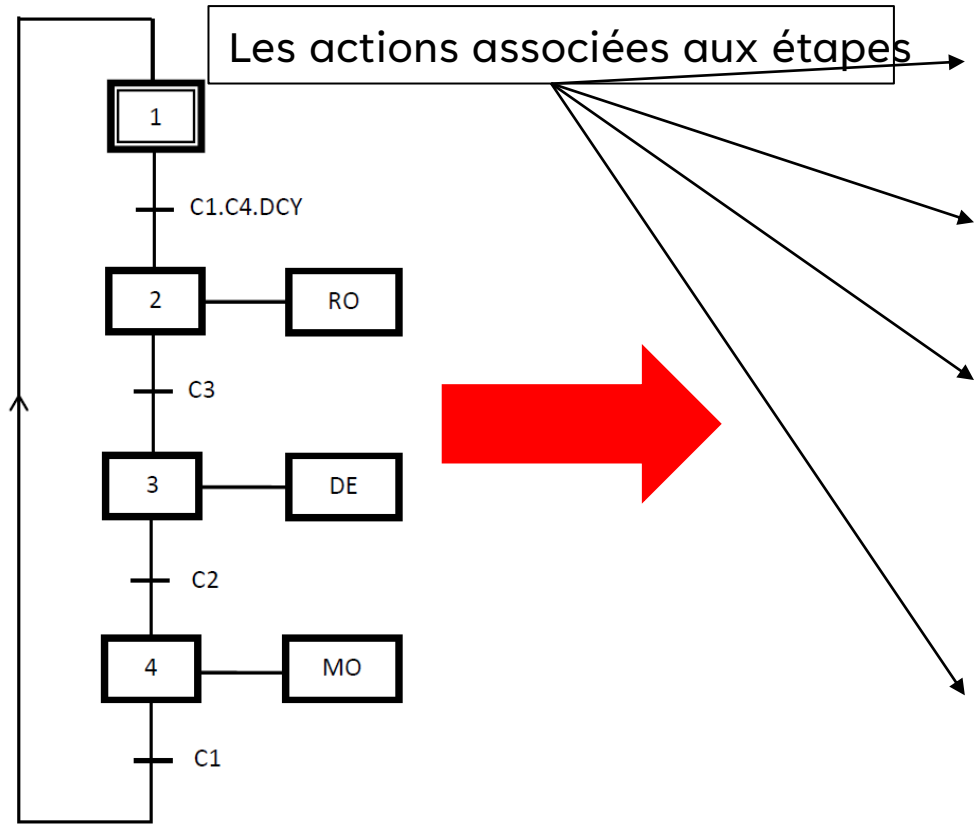


ETAPE X1	ENC : X4.C1
	RAZ : X2
	EQU : X4.C1+/X2.X1
ETAPE X2	ENC : X1.C1.C4.DCY
	RAZ : X3
	EQU : X1.C1.C4.DCY + /X3.X2
ETAPE X3	ENC : X2.C3
	RAZ : X4
	EQU : X2.C3 + /X4.X3
ETAPE X4	ENC : X3.C2
	RAZ : X1
	EQU : X3.C2 + /X1.X4

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

D. Application



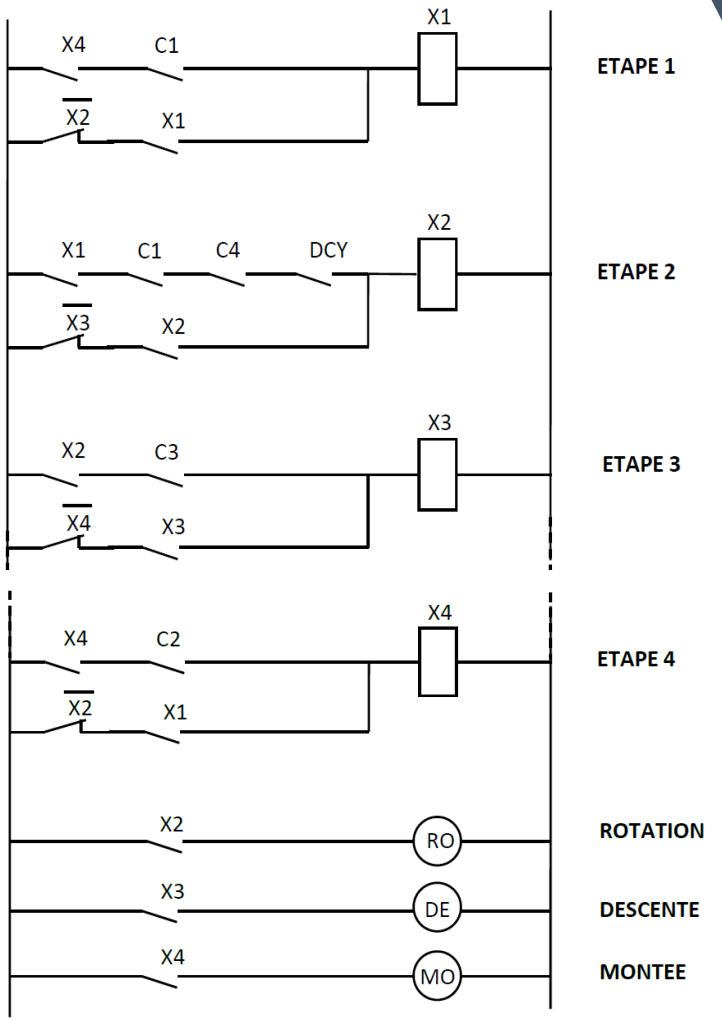
-	ETAPE X1	ENC : X4.C1
		RAZ : X2
		EQU : X4.C1+/X2.X1
RO	ETAPE X2	ENC : X1.C1.C4.DCY
		RAZ : X3
		EQU : X1.C1.C4.DCY + /X3.X2
DE	ETAPE X3	ENC : X2.C3
		RAZ : X4
		EQU : X2.C3 + /X4.X3
MO	ETAPE X4	ENC : X3.C2
		RAZ : X1
		EQU : X3.C2 + /X1.X4

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

D. Application

-	ETAPE X1	ENC : X4.C1
		RAZ : X2
		EQU : X4.C1+/X2.X1
RO	ETAPE X2	ENC : X1.C1.C4.DCY
		RAZ : X3
		EQU : X1.C1.C4.DCY + /X3.X2
DE	ETAPE X3	ENC : X2.C3
		RAZ : X4
		EQU : X2.C3 + /X4.X3
MO	ETAPE X4	ENC : X3.C2
		RAZ : X1
		EQU : X3.C2 + /X1.X4

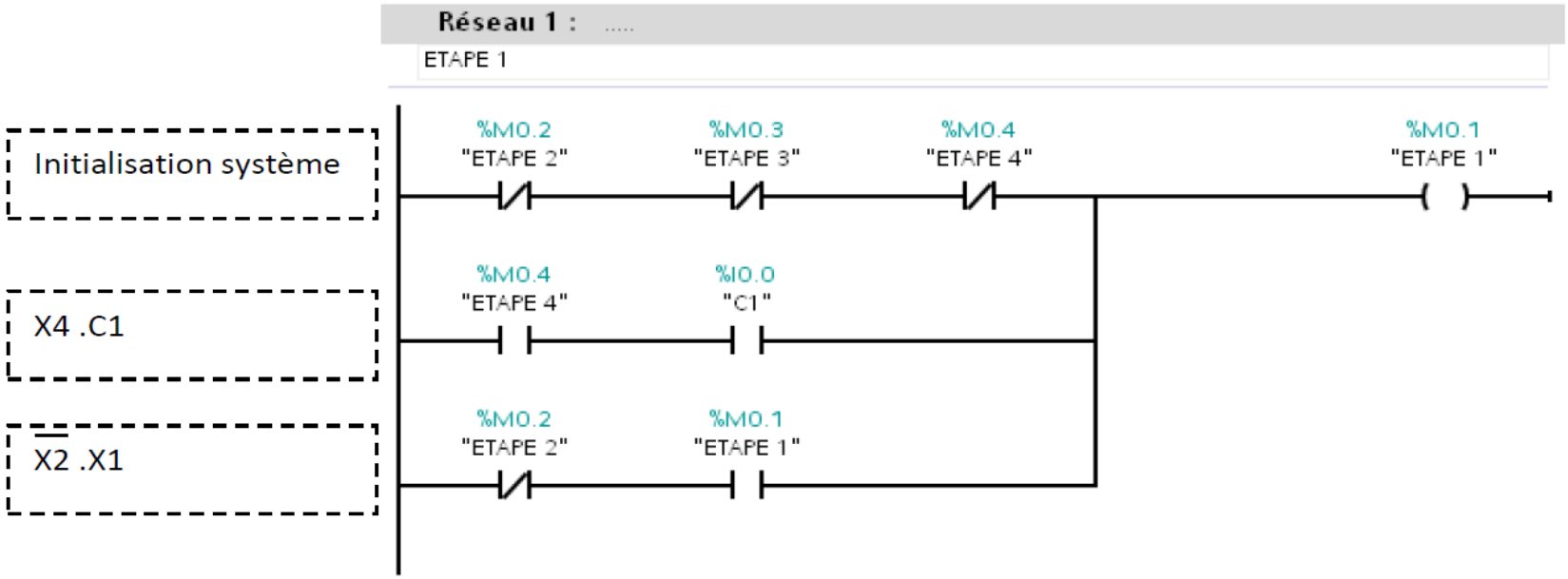


3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

D. Application

$$X1 = X4.C1 + \overline{X2}\,.\,X1$$



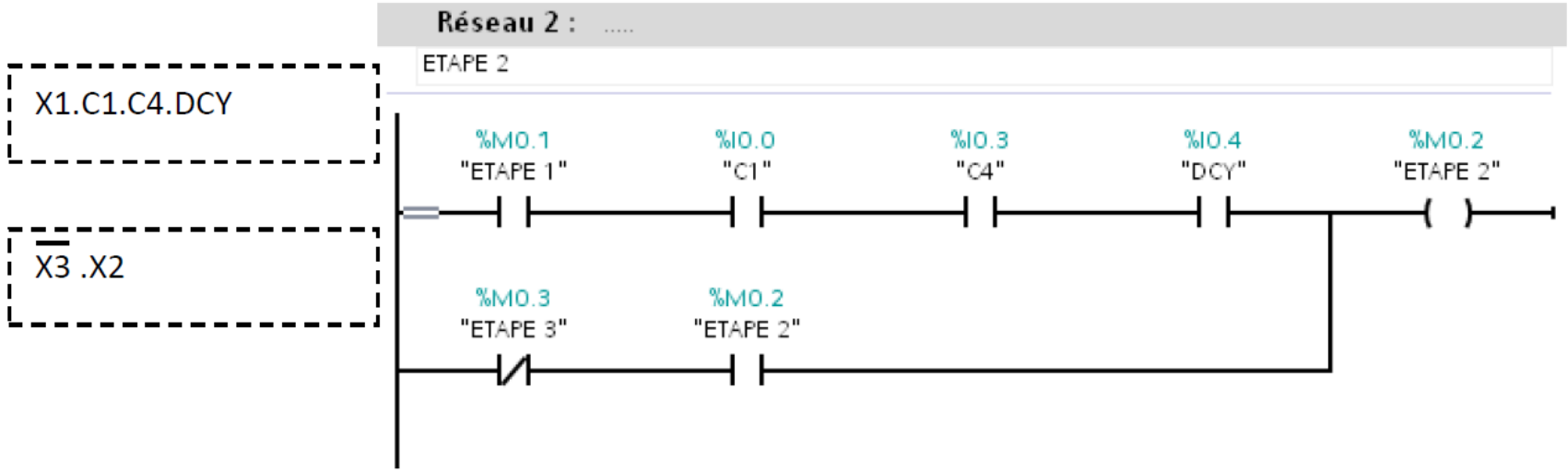
3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

D. Application

Et ainsi pour l'étape 2 est :

$$X2 = X1.C1.C4.DCY + \overline{X3}.X2$$



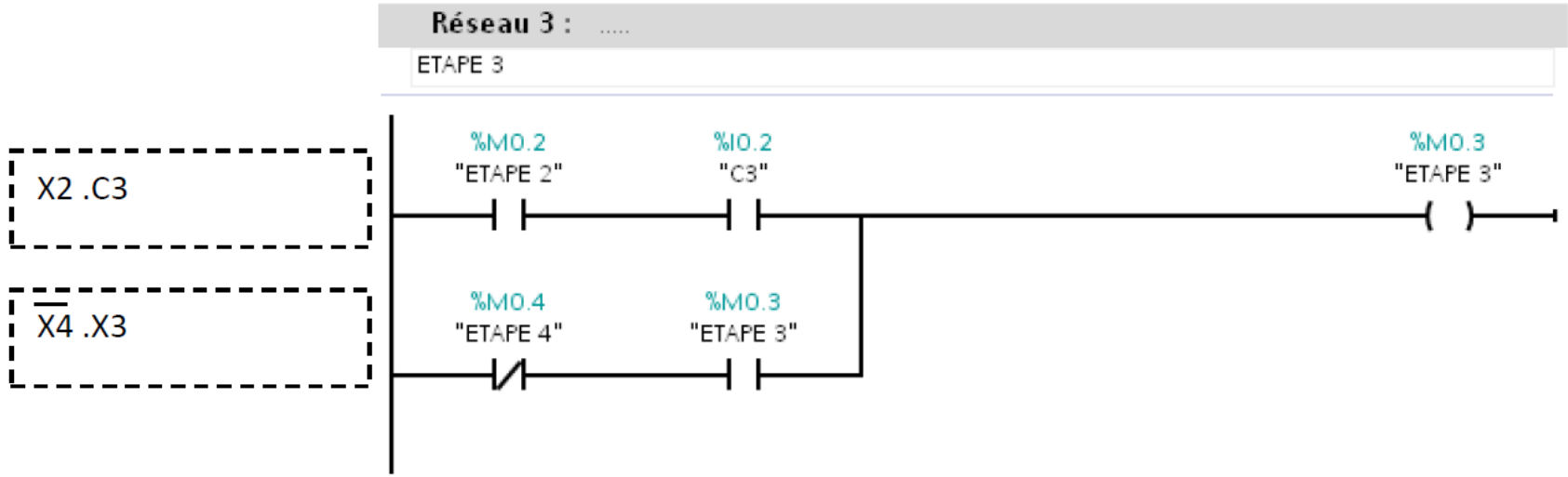
3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

D. Application

L'étape 3 :

$$X3 = X2.C3 + \overline{X4}\,.\,X3$$

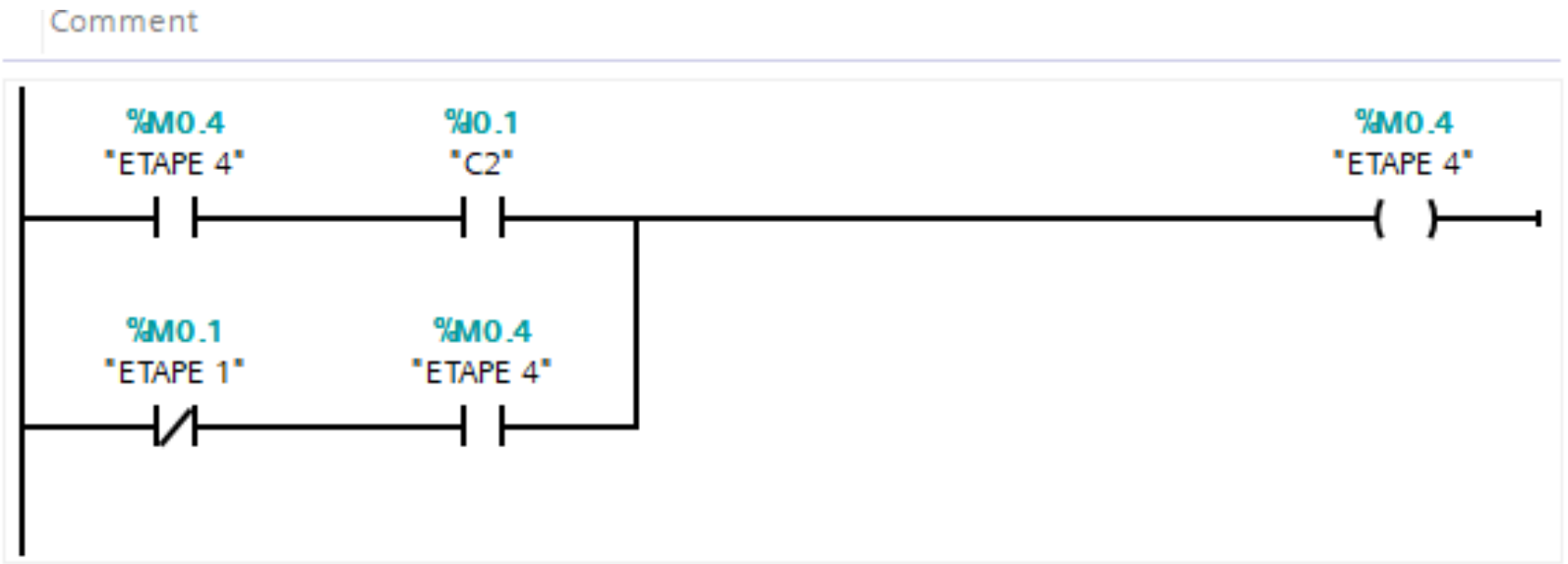


3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

D. Application

X4 = X3.C2 + /X1.X4



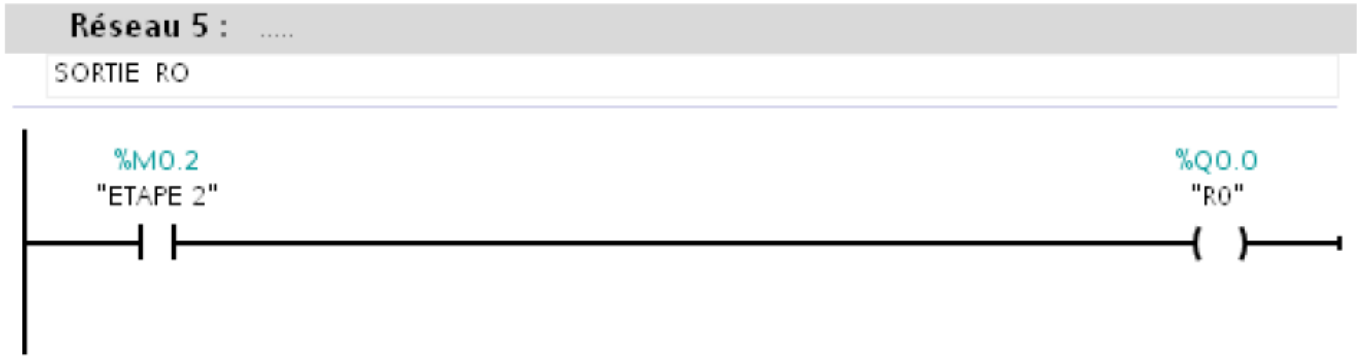
3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

D. Application

Pour la programmation des sorties

R0 : est actionné uniquement à l'étape 2



3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

D. Application

DE : est actionné uniquement à l'étape 3

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

D. Application

MO : est actionné uniquement à l'étape 4

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

D. Application

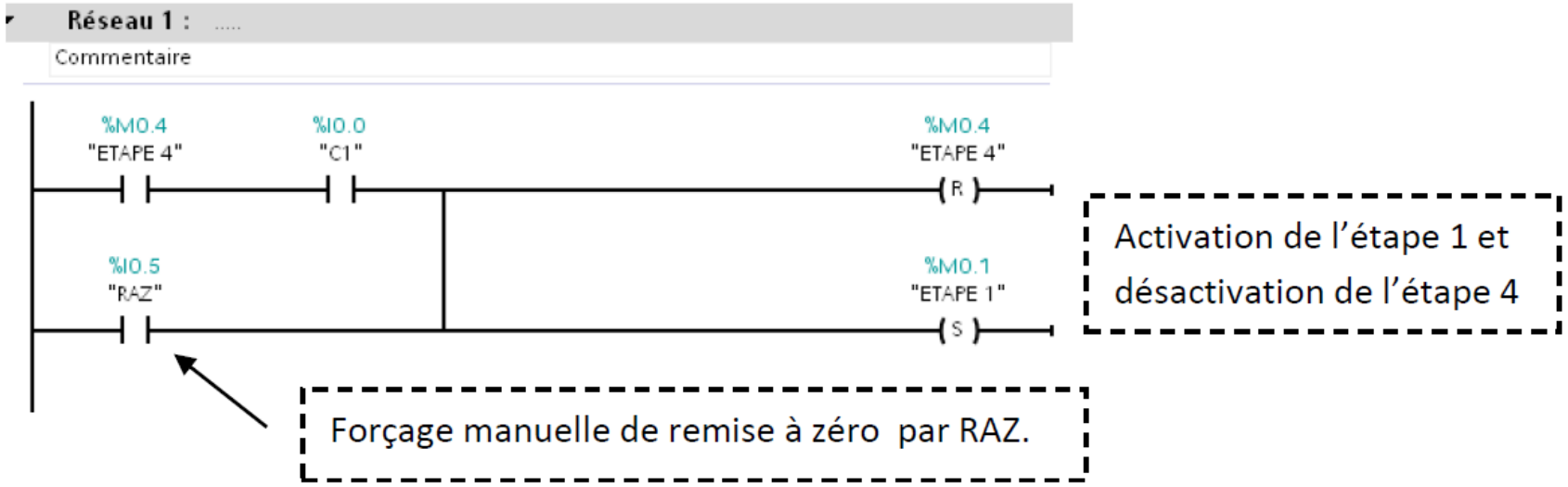
DE : est actionné uniquement à l'étape 3

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

D. Application

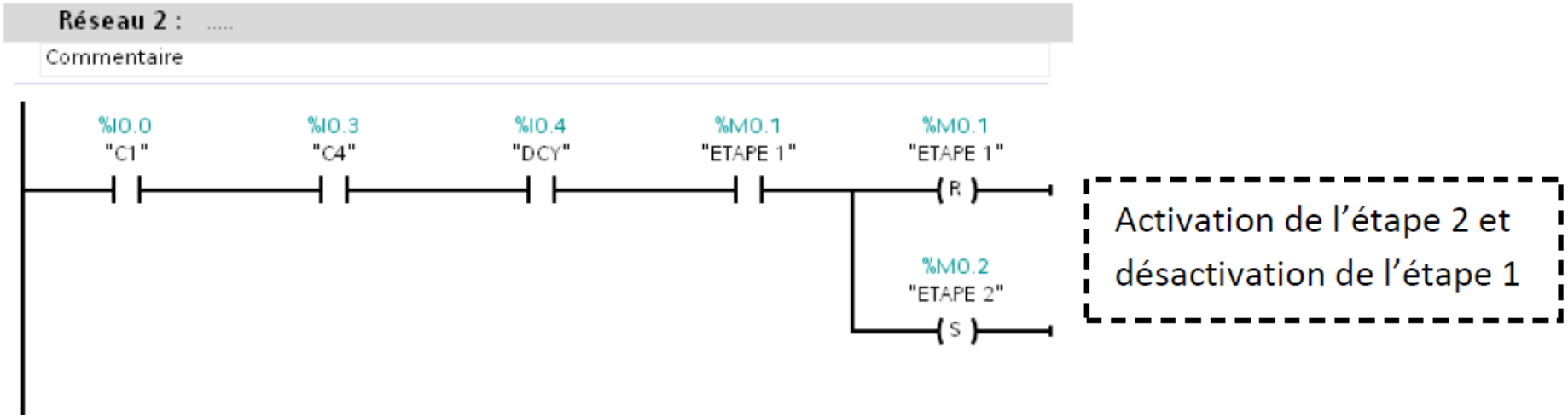
Le programme peut étre simplifier si en utilisant les bobines **Set/ Reset** ou les bascules **SR** ou **RS** et en tenant compte des cinq régles du GRAFCET.



3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

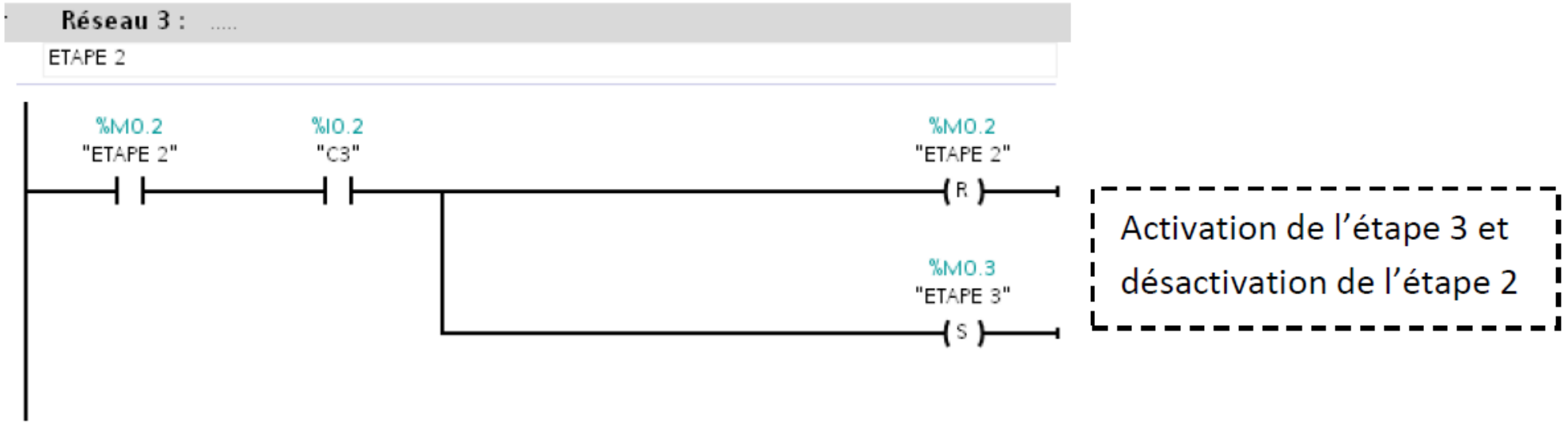
D. Application



3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

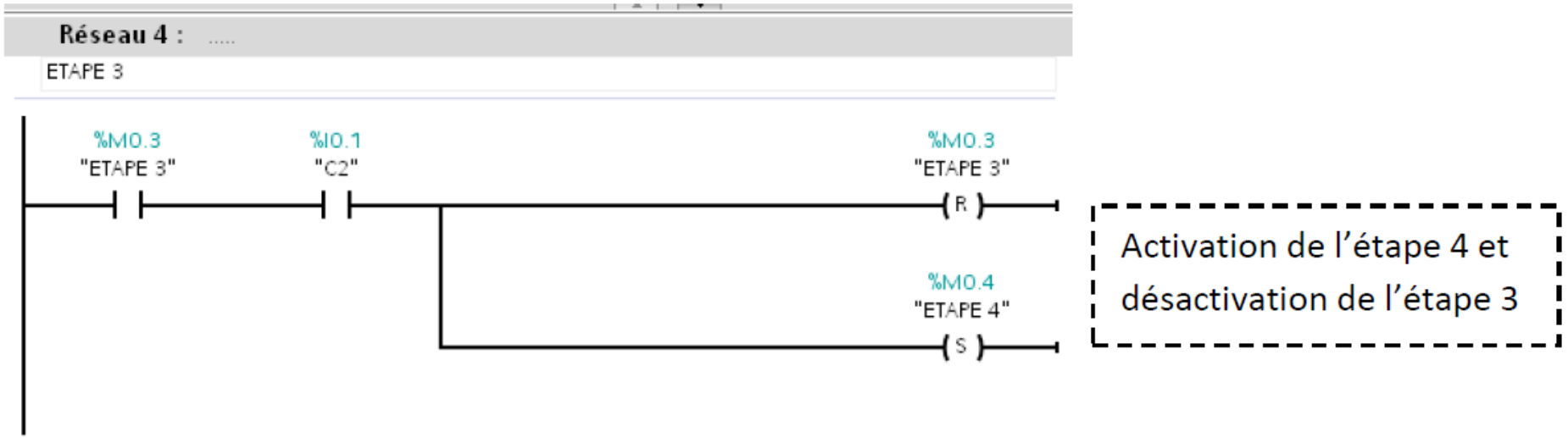
D. Application



3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

D. Application



3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

D. Application

3.Programmation d'un Automate Programmable Industriel

3.2.les bases d'un code LD (LADDER Diagram)

D. Application