

# Chapitre 2

## Programmation des automates programmable

Quand l'automate programmable industriel n'existait pas, les fonctions étaient réalisées par des câblages. Par exemple, pour réaliser un ET logique avec des interrupteurs, il suffit de les mettre en série. Pour réaliser un OU logique, il faut les mettre en parallèle.

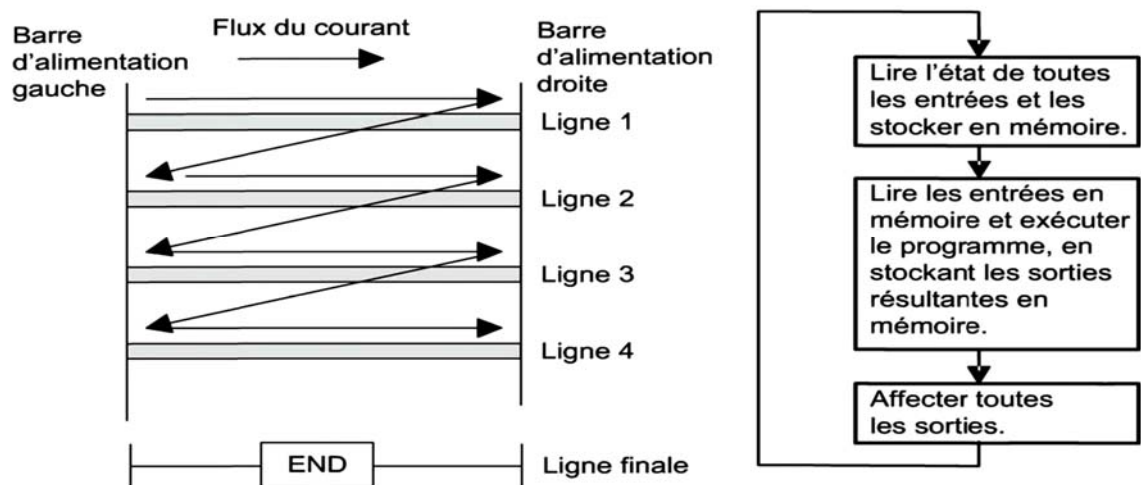
**Ladder Diagram (LD)** ou **Langage Ladder** ou **schéma à contacts** est un langage graphique très populaire auprès des **automaticiens** pour programmer les **automates programmables industriels**. Il ressemble un peu aux **schémas électriques**, et est facilement compréhensible. L'idée initiale du Ladder est la représentation de fonction logique sous la forme de **schémas électriques**.

### II.1 Principe

Un programme Ladder se lit de haut en bas et l'évaluation des valeurs se fait de gauche à droite. Les valeurs correspondent en fait, si on le compare à un **schéma électrique**, à la présence ou non d'un potentiel électrique à chaque nœud de connexion.

En effet, le Ladder est basé sur le principe d'une alimentation en tension représentée par deux traits verticaux reliés horizontalement par des bobines, des contacts et des blocs fonctionnels, d'où le nom 'Ladder' (échelle).

C'est un langage volontairement simple et graphique pour être compréhensible. Cela a permis, dans les années 1990, son utilisation sans formation lourde par les électriciens.



### II.2 Les composants du langage

#### II.2.1 Les entrées (ou contacts)

Il existe deux types de contact :

- Le contact normalement ouvert (NO) (en anglais : NO, normally open) :

Ce contact est fermé lorsque la variable booléenne associée est vraie, sinon, il est ouvert.

Truth Table		Ladder Diagram Symbol
IN	OUT	
reg,bit	W	
0	0	
1	1	

- Le contact normalement fermé (NF) (en anglais : NC, normally closed) :  
Ce contact est ouvert lorsque la variable booléenne associée est vraie, sinon il est fermé.

Truth Table		Ladder Diagram Symbol
IN	OUT	
reg.bit	W	
0	1	
1	0	

## II.2.2 Les sorties (ou bobines)

Il existe, de même que pour les contacts, deux types de bobines :

- la bobine normalement ouverte (NO) (en anglais : NO, normally open) :

Si cette bobine est soumise à un potentiel, c'est-à-dire qu'il existe un circuit fermé reliant cette bobine des deux côtés du potentiel, alors la variable booléenne associée est mémorisée à 'vraie', sinon elle est mémorisée à 'fausse'.

Truth Table		Ladder diagram symbol
IN	OUT	
W	reg.bit	
0	0	
1	1	

- la bobine normalement fermée (NF) (en anglais : NC, normally closed) :

Si cette bobine est soumise à un potentiel, c'est-à-dire qu'il existe un circuit fermé reliant cette bobine des deux côtés du potentiel, alors la variable booléenne associée (X ici) est mémorisée à 'fausse', sinon elle est mémorisée à 'vraie'.

Truth Table		Ladder Diagram Symbol
IN	OUT	
W	reg.bit	
0	1	
1	0	

- la bobine set Lorsque la variable d'entrée est 0, aucune action n'est effectuée, mais lorsque la variable d'entrée est 1, le bit de variable de sortie reg, est mis à 1.

Truth Table		Ladder Diagram Symbol
IN	OUT	
W	reg.bit	
0	no change	
1	Set	

- la bobine reset Lorsque la variable d'entrée est 0, aucune action n'est prise, mais lorsque la variable d'entrée est 1, le bit de variable de sortie reg, est réinitialisé (reset).

Truth Table		Ladder Diagram Symbol
IN	OUT	
W	reg.bit	
0	no change	
1	Reset	

### II.2.3 Fonction logique

Comme dit précédemment, les fonctions logiques sont dérivées de leurs réalisations électriques. Donc chaque fonction logique ([AND](#), [OR](#), [XOR](#), [NAND](#), [NOR](#), [NOT](#)) a une représentation qui correspond à son équivalent électrique.

- fonction logique NOT

Truth Table		Ladder Diagram symbol	Schematic Symbol
IN	OUT		
W	W		
0	1		
1	0		

- fonction logique AND

Truth Table			Ladder Diagram Symbol	Schematic Symbol
IN1	IN2	OUT		
W	reg.bit	W		
0	0	0		
0	1	0		
1	0	0		
1	1	1		

- fonction logique NAND

Truth Table			Ladder Diagram Symbol	Schematic Symbol
IN1	IN2	OUT		
W	reg.bit	W		
0	0	1		
0	1	1		
1	0	1		
1	1	0		

- Fonction Logique OR

Truth Table			Ladder diagram symbol	Schematic symbol
IN1	IN2	OUT		
W	reg.bit	W		
0	0	0		
0	1	1		
1	0	1		
1	1	1		

- **Fonction logique NOR**

Truth Table			Ladder Diagram Symbol	Schematic Symbol
IN1	IN2	OUT		
W	reg.bit	W		
0	0	1		
0	1	0		
1	0	0		
1	1	0		

- **fonction logique XOR**

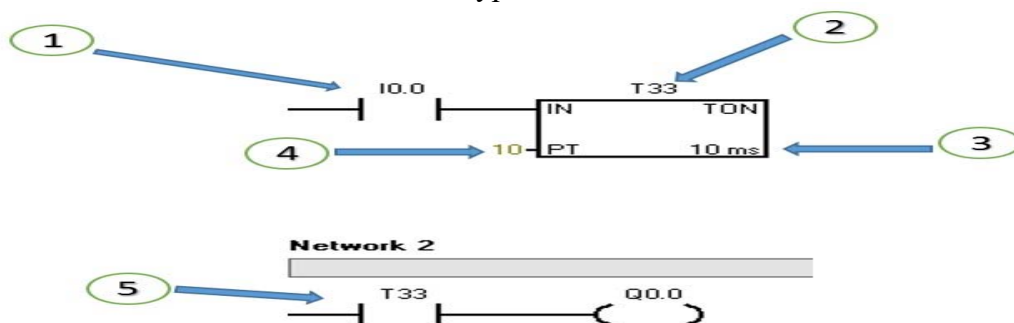
Truth Table			Ladder Diagram Symbol	Schematic Symbol
IN1	IN2	OUT		
W	reg.bit	W		
0	0	0		
0	1	1		
1	0	1		
1	1	0		

## II.2.4 Timer

Les Timers peuvent être utilisés dans une large gamme d'applications où une fonction de temporisation est requise en fonction d'un signal d'entrée.

Tous les événements basés sur le temps que nous pouvons créer à l'aide de timer.

Dans Siemens S7-200, il existe trois types de timers TON, TOFF et TONR.



- **TON**

Comme vous pouvez le voir dans l'image ci-dessus, nous allons diviser Timer ON en différents points dont nous allons discuter point par point.

1. C'est ce que nous avons appelé le bit Timer Enable. Cela signifie que lorsqu'un capteur connectés à l'entrée I0.0 = 1, le Timer démarre le comptage du temps et si ce bit est désactivé, le Timer réinitialise sa valeur à zéro.

2. mémoire de données des Timers où Les Timers sauvegardes en temps sont appelées T Memory.

3. Il existe 3 types de résolution de temporisateurs dans le S7200. 1 ms, 10 ms et 100 ms.

- 1 compte 1 ms = 32767 comptes = 32767 ms = 32,767 s
- 1 compte 10 ms = 32767 comptes = 327670 ms = 327,67 s
- 1 comptage 100 ms = 32767 comptages = 3276700 ms = 3276,7 s

4. PT signifie temps pré-réglé.

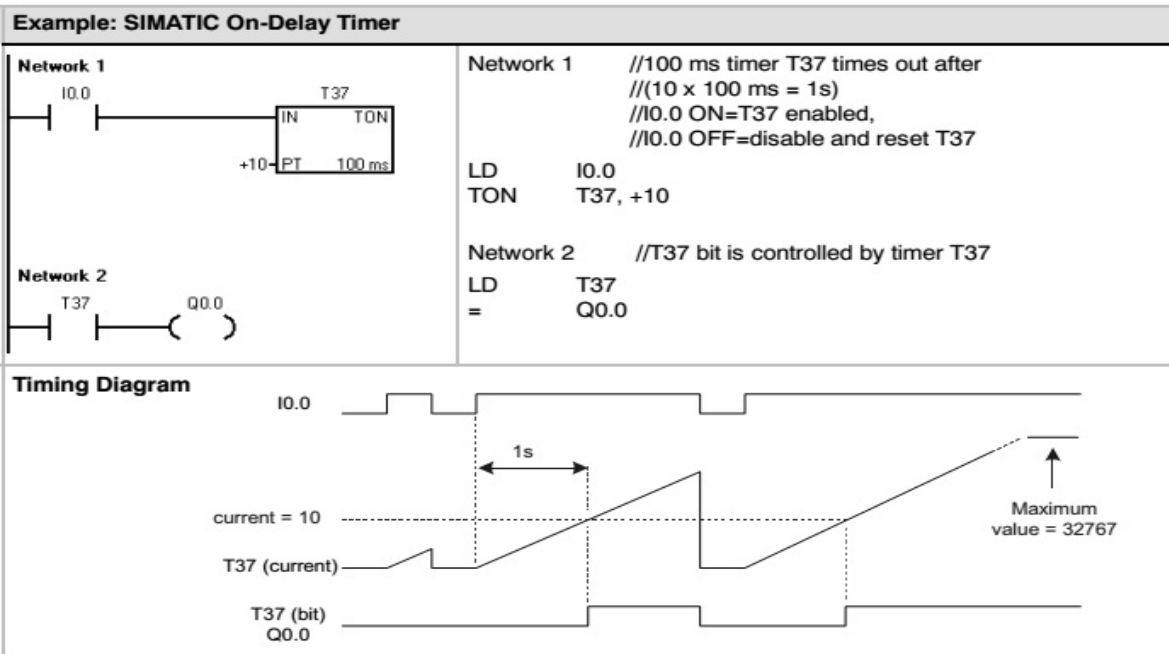
## Numéro de minuterie et résolution là

Timer Type	Resolution	Maximum Value	Timer Number
TONR (retentive)	1 ms	32.767 s (0.546 min.)	T0, T64
	10 ms	327.67 s (5.46 min.)	T1 to T4, T65 to T68
	100 ms	3276.7 s (54.6 min.)	T5 to T31, T69 to T95
TON, TOF (non-retentive)	1 ms	32.767 s (0.546 min.)	T32, T96
	10 ms	327.67 s (5.46 min.)	T33 to T36, T97 to T100
	100 ms	3276.7 s (54.6 min.)	T37 to T63, T101 to T255

For a timer with a resolution of 1 ms, the timer bit and the current value are updated asynchronous to the scan cycle. For scans greater than 1 ms, the timer bit and the current value are updated multiple times throughout the scan.

For a timer with a resolution of 10 ms, the timer bit and the current value are updated at the beginning of each scan cycle. The timer bit and current value remain constant throughout the scan, and the time intervals that accumulate during the scan are added to the current value at the start of each scan.

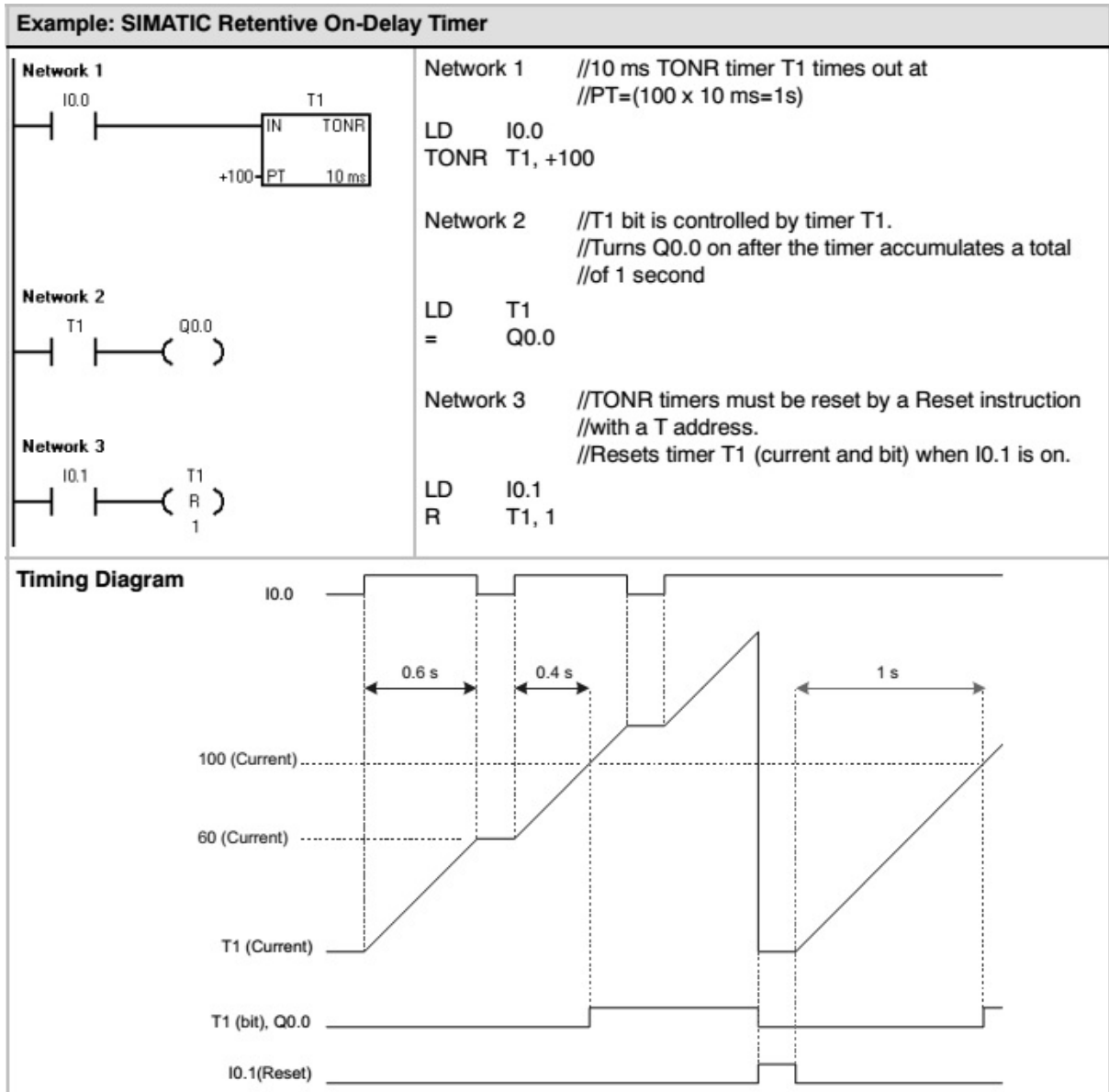
For a timer with a resolution of 100 ms, the timer bit and current value are updated when the instruction is executed; therefore, ensure that your program executes the instruction for a 100-ms timer only once per scan cycle in order for the timer to maintain the correct timing.



Type	Current >= Preset	State of the Enabling Input (IN)	Power Cycle/First Scan
TON	Timer bit on Current continues counting to 32,767	ON: Current value counts time OFF: Timer bit off, current value = 0	Timer bit off Current value = 0
TONR	Timer bit on Current continues counting to 32,767	ON: Current value counts time OFF: Timer bit and current value maintain last state	Timer bit off Current value can be maintained <sup>1</sup>
TOF	Timer bit off Current = Preset, stops counting	ON: Timer bit on, current value = 0 OFF: Timer counts after on-to-off transition	Timer bit off Current value = 0

- TONR

Le timer TONR est utilisée pour capturer plusieurs intervalles de temps. Compte le temps qui s'écoule lorsque l'entrée de validation est activée. Lorsque la valeur en cours (Txxx) est supérieure ou égale au temps prédéfini (PT),le bit de temporisation est activé. Lorsque l'entrée de validation est désactivée, la valeur en cours de la temporisation TONR est conservée. Vous pouvez donc vous servir de la temporisation TONR afin de cumuler plusieurs périodes de temps de l'entrée IN. L'opération R (Mettre à 0) permet d'effacer la valeur en cours de la temporisation TONR.

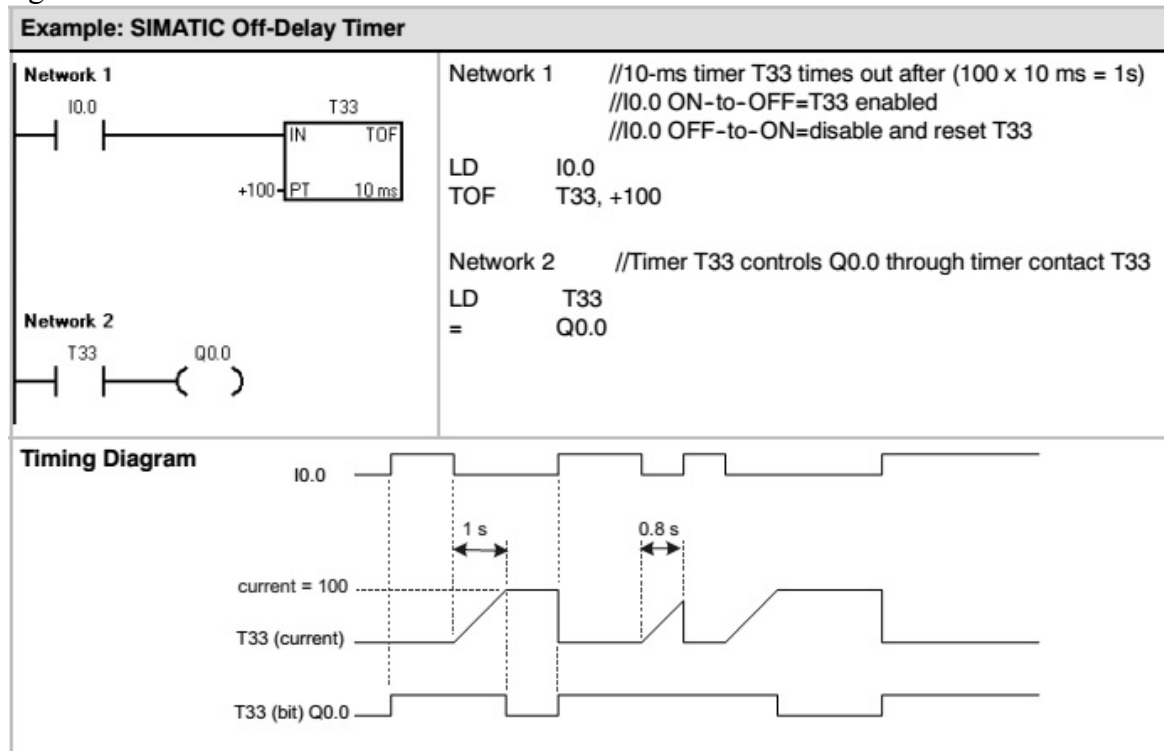


- **TOFF**

TOFF Timer fonctionne un peu différemment de TON et TONR Timer. Nous pourrions comprendre le fonctionnement de TOFF Timer à l'aide de la figure ci-dessous.

Comme vous pouvez le voir dans la figure ci-dessous, lorsque nous activons le bit I0.0 de timer, la valeur actuelle de timer est «0» et le bit du drapeau de timer est activé.

Maintenant, lorsque nous relâchons le bit I0.0 d'activation du timer, le timer commence à compter le temps en fonction de sa résolution. Lorsque le timer atteint ce que nous définissons la valeur à PT (temps prédéfini), le timer arrête le comptage et conserve sa valeur où qu'elle se trouve et le bit du drapeau du timer passe de HAUT à BAS comme vous pouvez le voir dans la figure ci-dessous.



**Remarque**

Vous ne pouvez pas utiliser les mêmes numéros de temporisations pour TOF et TON. Il est, par exemple, impossible d'avoir TON T32 et TOF T32.

Vous pouvez utiliser une temporisation TON pour mesurer un intervalle de temps unique.

**II.2.5 compteur S7-200**

Le compteur **incrémental (CTU)** incrémente en partant de la valeur en cours à chaque front montant de l'entrée d'incrément. Il est remis à zéro lorsque l'entrée de remise à zéro est activée ou que l'opération "Mettre à 0" est exécutée. Le compteur s'arrête lorsqu'il atteint la valeur maximale de 32 767.

Le compteur **incrémental/décémental (CTUD)** incrémente en partant de la valeur en cours à chaque front montant de l'entrée d'incrément et décrémente à chaque front montant de l'entrée de décrémentation. Il est remis à zéro lorsque l'entrée de remise à zéro est activée ou que l'opération "Mettre à 0" est exécutée. Lorsqu'il atteint la valeur maximale de 32 767, le front montant suivant à l'entrée d'incrément fait prendre à la valeur en cours la valeur minimale de -32 768. De façon comparable, lorsque la valeur minimale -32 768 est atteinte, le front montant suivant à l'entrée de décrémentation fait prendre à la valeur en cours la valeur maximale de 32 767.

La valeur en cours des **compteurs incrémental** et incrémental/décrémental correspond à la valeur de comptage en vigueur. Ces compteurs disposent également d'une valeur prédéfinie PV qui est comparée à la valeur en cours à chaque exécution de l'opération de comptage. Lorsque la valeur en cours est supérieure ou égale à la valeur prédéfinie, le bit de compteur (bit C) est activé. Sinon, le bit C est désactivé.

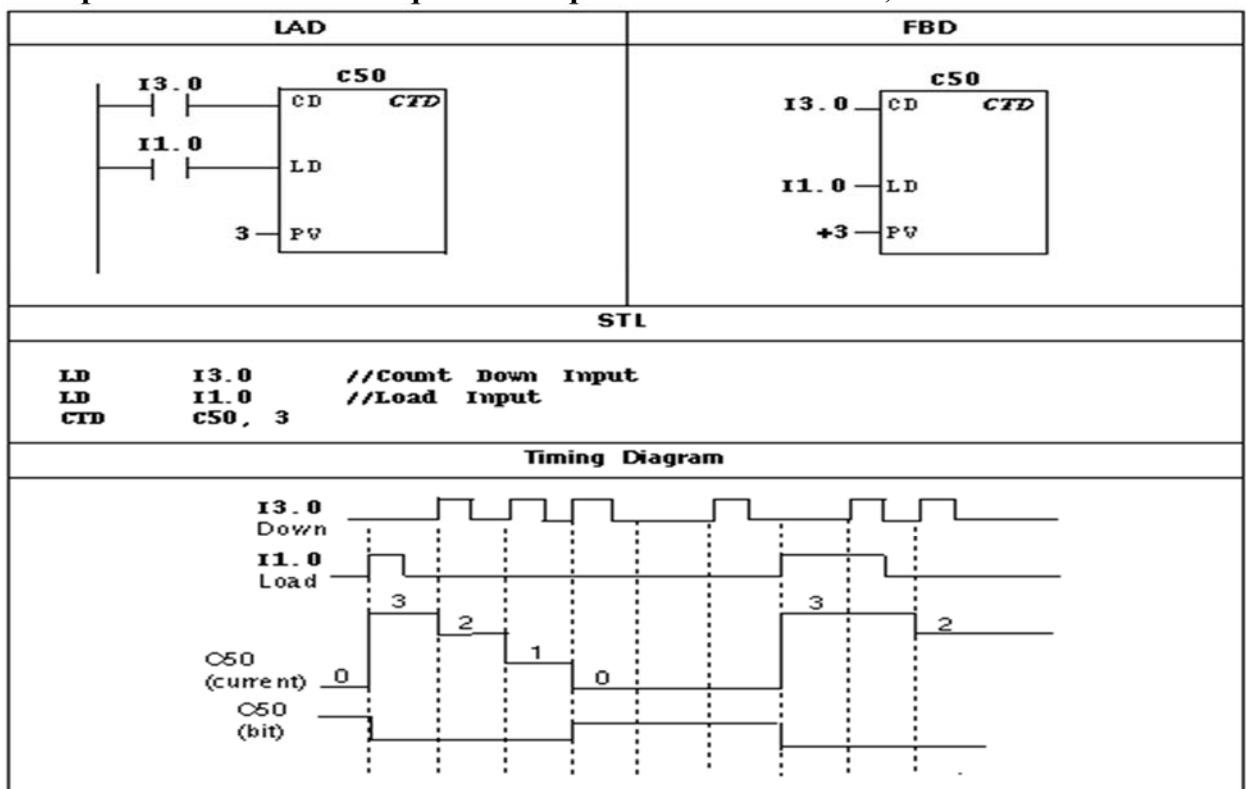
Le compteur **décrémental (CTD)** décrémente en partant de la valeur en cours à chaque front montant de l'entrée de décrémentation. Le compteur remet le bit de compteur à 0 et charge la valeur prédéfinie dans la valeur en cours lorsque l'entrée de chargement est activée. Le compteur s'arrête lorsqu'il atteint zéro et le bit de compteur (C) est alors mis à 1.

La mise à zéro d'un compteur à l'aide de l'opération **R met à zéro** le bit de compteur et la valeur en cours du compteur. Le numéro de compteur permet de désigner à la fois la valeur en cours et le bit C du compteur en question

**Remarque:**

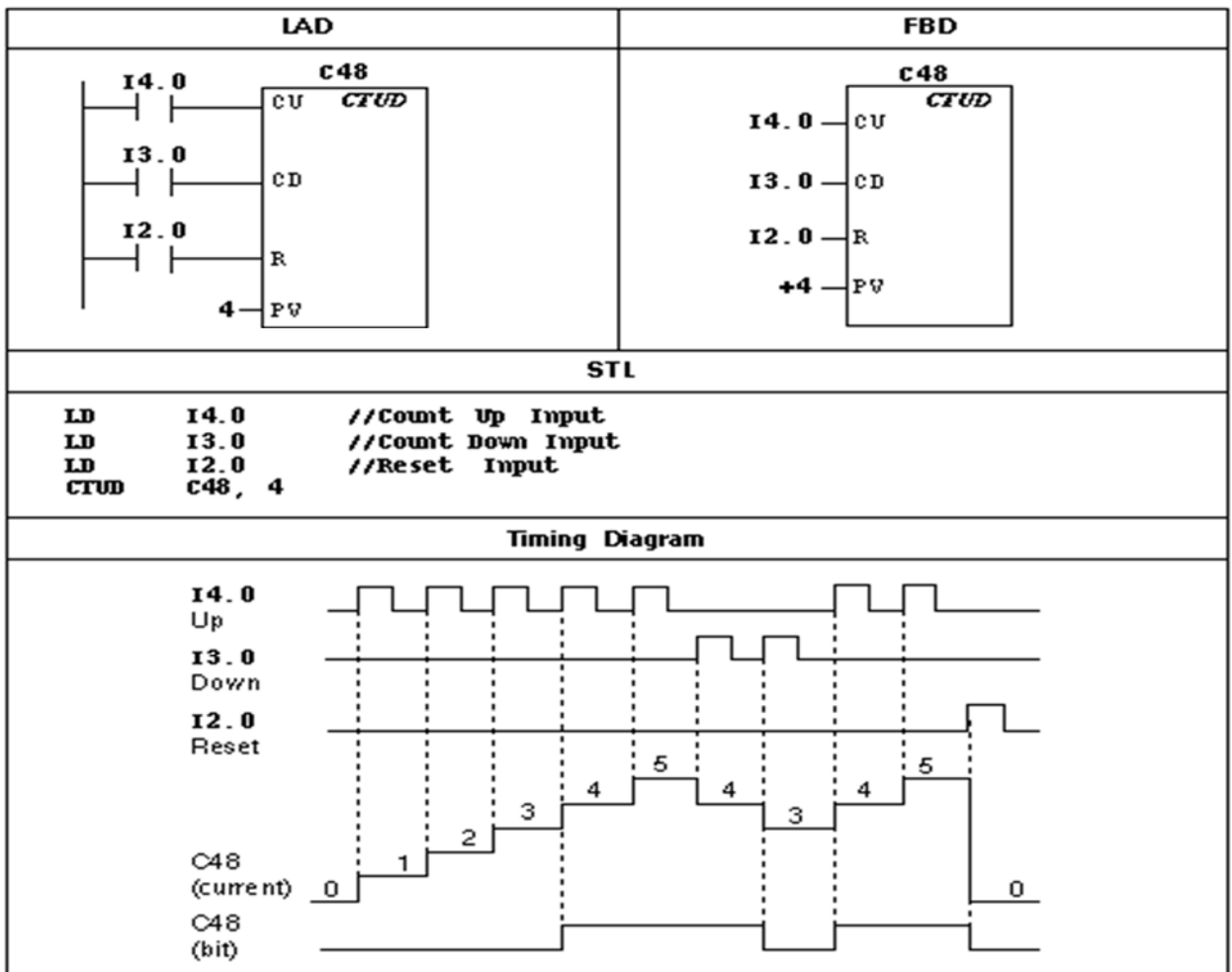
Comme il existe une valeur actuelle pour chaque compteur, n'affectez pas le même numéro de compteur à plus d'un compteur. (Les compteurs ascendants, compteurs ascendants / descendants et compteurs décroissants avec le même numéro accèdent à la même valeur actuelle.)

**Exemple d'instruction de compteur CTD pour SIMATIC CONT, FBD et STL**



**Exemple d'instruction de compteur CTUD pour SIMATIC CONT, FBD et STL**

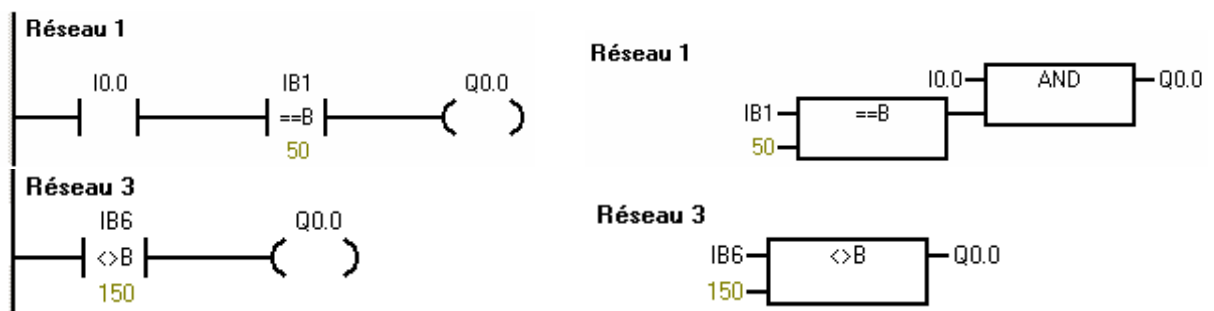




## II.2.6 opération de comparaison LOG :

### 1- Egalité octet et Inégalité octet :

L'opération Comparer octets permet de comparer les valeurs IN1 et IN2.



### 2- Egalité mot et inégalité mot (Word) :

