

Cours Informatique Industrielle

- ❖ GRAFCET
- ❖ MICROPROCESSEUR 6809

Mr KHATORY 2013

1

Cours Informatique Industrielle

Partie I: GRAFCET

- ✓ Grafcet
- ✓ Matérialisation
- ✓ Programmation

Partie II: Microprocesseur

- ✓ Structure d'un microprocesseur 6809
- ✓ Programmation

2

Cours Informatique Industrielle

Partie I: GRAFCET

✓ Grafcet

- ✓ Matérialisation
- ✓ Programmation

Partie II: Microprocesseur

- ✓ Structure d'un microprocesseur 6809
- ✓ Programmation

3

Un **GRAFCET** ("GRAphe Fonctionnel de Commande Etape-Transition ") est un **mode de représentation** et d'analyse d'un automatisme. C'est un **outil graphique** de description du comportement de la partie commande. Il décrit les interactions informationnelles à travers la frontière d'isolement : partie de commande, partie opérative d'un système isolé.

✓ **GRAFCET** est Inventé en 1977 en France par l' **AFCET**: Association Française pour la **Cy**bernétique **É**conomique et **T**echnique.

✓ **GRAFCET** est Diffusé par l' **ADEPA** (Agence Nationale pour le **DE**veloppement de la **P**roductique **A**ppliquée à l'industrie Norme IEC 1131.3)

✓ Le **GRAFCET** est aussi appelé **DFS** (**D**iagramme **F**onctionnel en **S**équence) ou en anglais, **SFC** (**S**equential **F**unction **C**hart).

4

Pourquoi le GRAFCET ?

Lorsque certaines spécifications sont exprimées en langage courant, il y a un **risque permanent d'incompréhension**. Certains mots sont peu précis, mal définis ou possèdent plusieurs sens.

Le langage courant est mal adapté pour décrire précisément les systèmes séquentiels.



Le **GRAFCET** fut donc créé pour représenter de façon **symbolique et graphique** le fonctionnement d'un automate.

Cela permet une **meilleure compréhension** de l'automatisme par tous les intervenants.

5

Le langage **GRAFCET** doit donc être connu de toutes les personnes concernées par les automatismes, depuis leur conception jusqu' à leur exploitation.

Les Normes de GRAFCET:

1982 : GRAFCET est normalisé en France parution de la norme NF C03-190.

1988 : GRAFCET est normalisé par la CEI/IEC 848.

1993 : Norme IEC 1131-3 permet de valider cinq langages de programmation d'API dont le SFC (inspiré du GRAFCET)

2002 : Norme IEC 60848 est largement modifiée, sa traduction française est publiée en septembre de la même année sous la référence NF EN 60848

Les avantages du GRAFCET :

- ✓il est indépendant de la **matérialisation technologique**;
- ✓il traduit de façon cohérente le **cahier des charges**;
- ✓il est bien adapté aux **systèmes automatisés**.

6

GRAF CET

Le **GRAF CET** est constitué:

↳ d'**éléments graphique** de base:

- ✓ les **étapes**,
- ✓ les **transitions**,
- ✓ les **liaisons orientées** reliant entre elles les étapes et les transitions, structurés en un réseau alterné formant l'ossature (squelette) séquentielle graphique.

↳ D'une **interprétation** traduisant le comportement de la partie commande vis-à-vis de ses entrées et de ses sorties, interprétation caractérisée par:

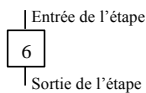
- ✓ les **actions** associées aux étapes,
- ✓ les **réceptivités** associées aux transitions

↳ de **règles d'évolution** définissant formellement le comportement dynamique de la partie commande ainsi décrite.

7

Règles d'écriture du Grafcet

Étape



Une étape correspond à une *situation* du système automatisé dans laquelle le **comportement** de la partie commande est **stable** vis à vis de ses entrées et ses sorties



Etape 2



Etape 3 active



Etape Initial 0

8

Actions associées

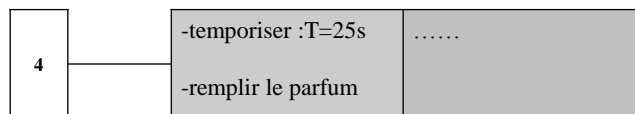
Une ou plusieurs actions élémentaires ou complexes peuvent être associées à une étape. Les actions associées à une étape traduisent ce qui **doit être fait** si l'étape est **active**.

Les actions qui sont les résultats du traitement logique des informations par la partie commande peuvent être :

- ✓ **externes** et elles correspondent aux ordres émis vers la partie opérative ou vers les éléments externes.
- ✓ **Internes** et concernent des fonctions spécifiques de l'automatisme telles que : temporisation, comptage, etc ..

9

Les actions sont précisées dans un ou plusieurs rectangles :



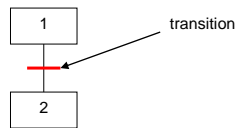
N.B: On utilise toujours dans les actions les **verbes à l'infinitif**: allumer la lampe mais pas allumage de la lampe

10

transition

Une transition indique la **possibilité d'évolution** d'une étape à l'étape suivante. Le passage de l'une à l'autre s'effectue par le **franchissement** d'une transition

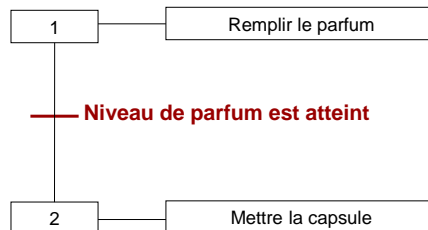
Une transition est soit **validée** soit non validée. Elle est dite validée lorsque toutes les étapes immédiatement précédentes reliées à cette transition sont **achevées**.



11

réceptivité

A chaque transition est associée une **proposition logique** appelée réceptivité qui peut être soit vraie soit fausse.



12

La réceptivité écrite sous forme de proposition logique est une fonction combinatoire:

✓ d'informations externes provenant de la partie opérative ou du poste opérateur,

✓ d'informations internes en rapport avec les fonctions spécifiques de l'automatisme (temporisation, comptage), ou les états actifs ou inactifs des autres étapes.

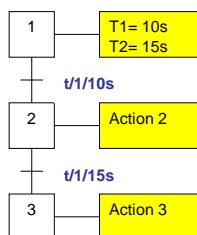
Remarque:

Les notations $a\uparrow$ ou $a\downarrow$ sont employés lorsque la condition de réceptivité est liée au changement d'état variable.

$a\uparrow$: front montant de a, $a\downarrow$: front descendant de a. (Ces notations sont utilisées nécessaire lors d'utilisation des dispositifs électroniques)

13

pour faire intervenir le temps dans une réceptivité, il suffit d'indiquer après le repère t son origine et sa durée.



t/1/10s: passer à l'étape 2 si 10s sont écoulées depuis la dernière activation de l'étape 1.

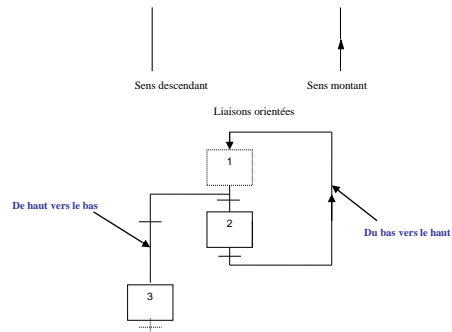
t/1/15s: passer à l'étape 3 si 15s sont écoulées depuis la dernière activation de l'étape 1.

14

Liaisons orientées

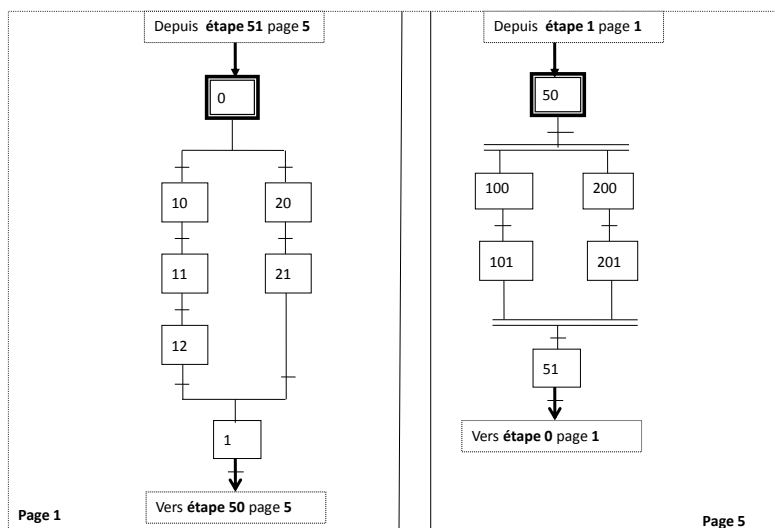
Les liaisons orientées permettent de relier les étapes aux transitions et inversement.

Le sens général du parcours est du haut vers le bas. Lorsque le parcours est différent des flèches sont nécessaires



15

Les renvois de pages



16

Règles d'évolutions

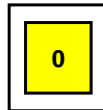
Aux règles d'écriture s'ajoutent les règles d'évolution afin de préciser les conditions pour lesquelles les étapes sont actives ou inactives

Règle 1: Initialisation

La situation initiale caractérise le comportement initial de la partie commande vis à vis de la partie opérative et correspond à l'étape active au début du fonctionnement.

Elle traduit **généralement** un comportement de **repos**

Le symbole est le double carré



17

Règle 2: Franchissement d'une transition

Le franchissement d'une transition s'effectue si:

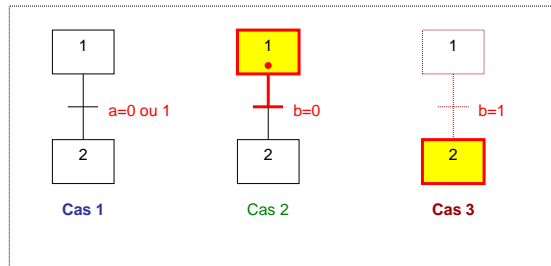
- ✓ l'étape précédente est active
- ✓ la réceptivité associée est vraie

Lorsque ces deux conditions sont réunies, la transition devient **franchissable** et est **obligatoirement franchie**.

18

Règle 3: Évolution des étapes actives

Le franchissement d'une transition entraîne **simultanément l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes.**



Cas 1: La transition 1-2 est non validée, l'étape 2 étant inactive.

Cas 2: L'étape 1 étant active, la transition 1-2 est validée mais ne peut être franchie car la réceptivité n'est pas vraie: $b=0$.

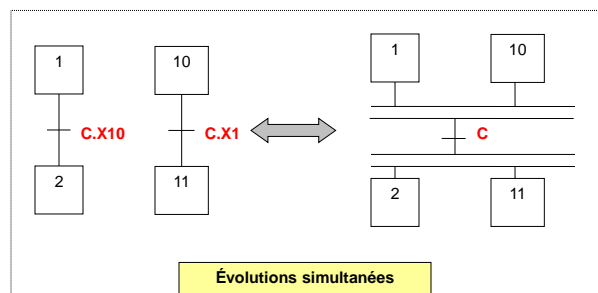
Cas 3: La transition 1-2 est franchie car la réceptivité est vraie: $b=1$. Dans ce cas l'étape 2 est activée et l'étape 1 est désactivée.

19

Règle 4: Évolutions simultanées

Plusieurs transitions simultanément franchissables sont simultanément franchies

Cette règle de franchissement permet notamment de décomposer un grafcet en plusieurs diagrammes indépendants

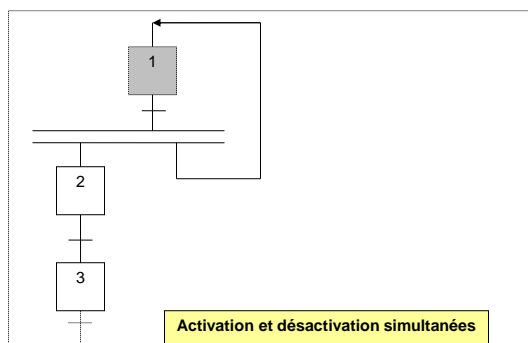


NB: $X_i = 1$ étape i active
 $X_i = 0$ étape i inactive

20

Règle 5 : Activation et désactivation simultanées

Si au cours du fonctionnement de l'automatisme une même étape doit être simultanément activée et désactivée, elle reste activée.

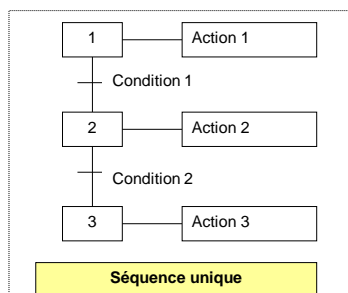


21

Structure de base

Séquence unique

Le début du Grafset est constitué d'une suite d'étapes qui peuvent être activées les unes après les autres. Cette suite d'étapes est appelée une **séquence unique**



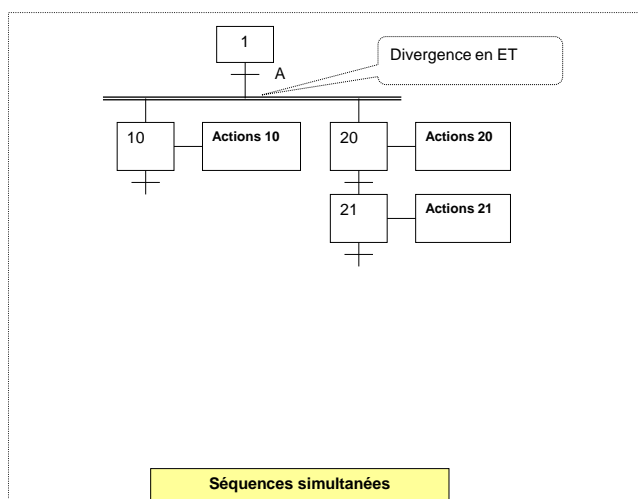
La séquence est dite **active** si **au moins une** des étapes est **active**. Elle est dite inactive si toutes les étapes son inactives

22

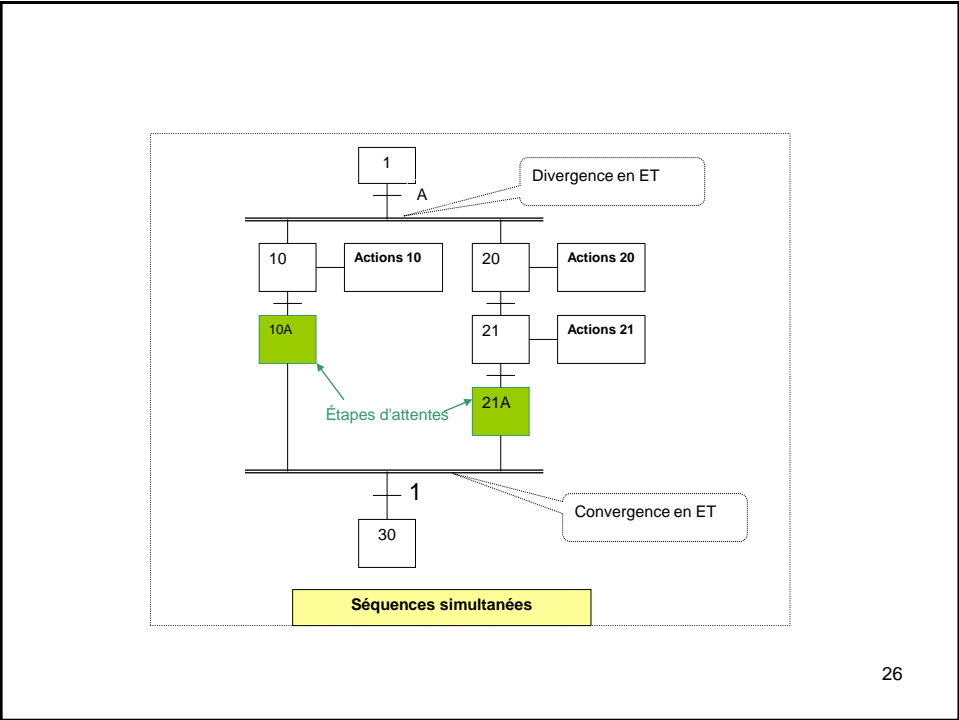
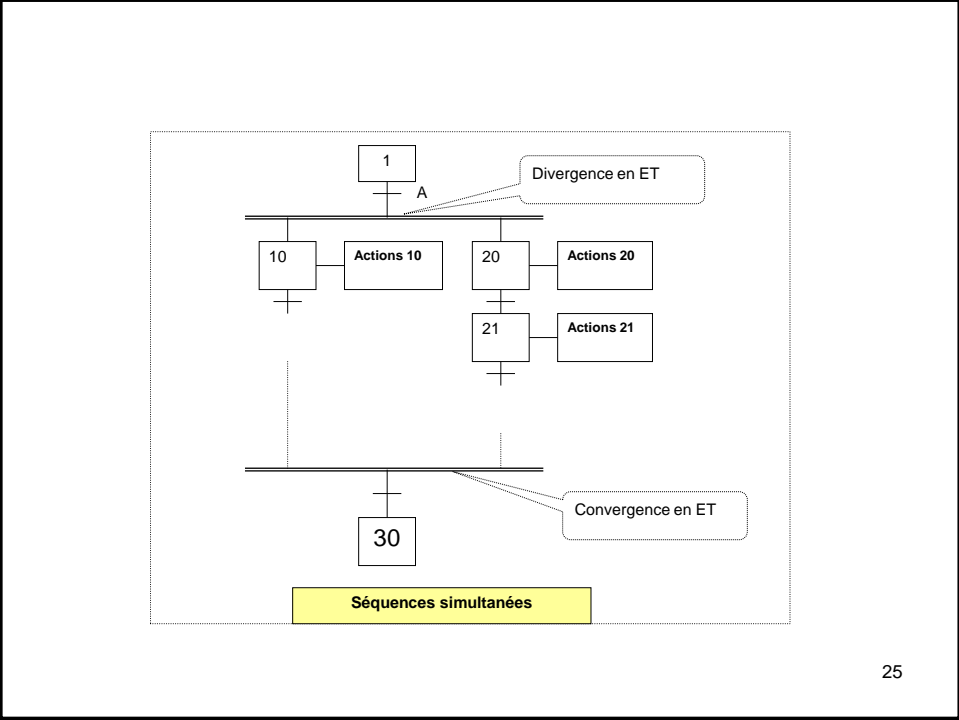
Séquences simultanées

Lorsque le franchissement d'une transition conduit à activer plusieurs séquences en même temps ces séquences sont dites **séquences simultanées**. Après l'activation simultanée de ces séquences, les évolutions des étapes actives dans chacune des séquences deviennent alors indépendantes.

23

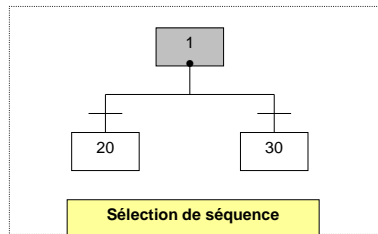


24



Sélection de séquence

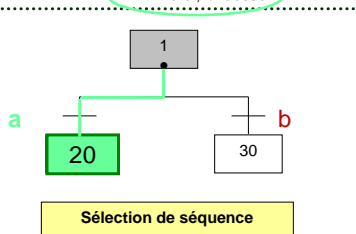
Une sélection ou un **choix d'évolution** entre plusieurs étapes ou séquence se représente, à partir d'une ou plusieurs étapes, par autant de transitions validées qu'il y a d'évolutions possible



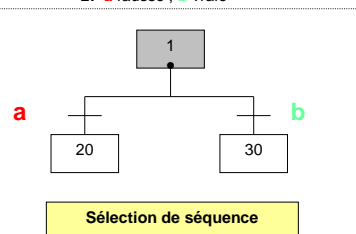
27

Sélection de séquence

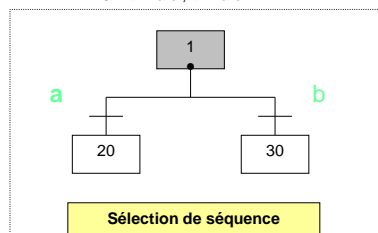
1: a vraie, b fausse



2: a fausse, b vraie



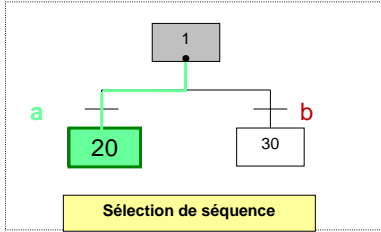
3: a vraie, b vraie



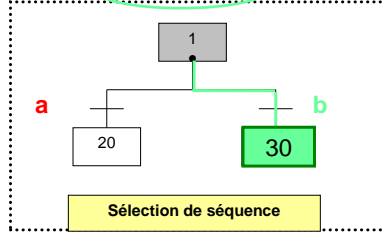
28

Sélection de séquence

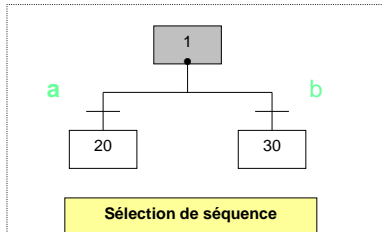
1: a vraie, b fausse



2: a fausse, b vraie



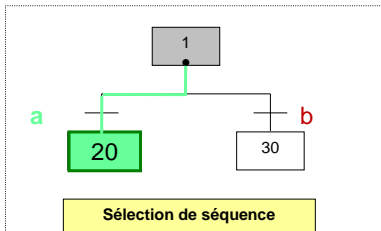
3: a vraie, b vraie



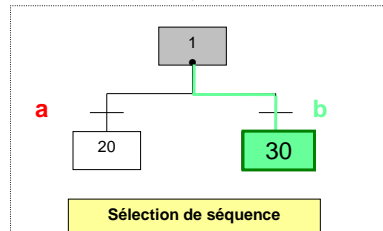
29

Sélection de séquence

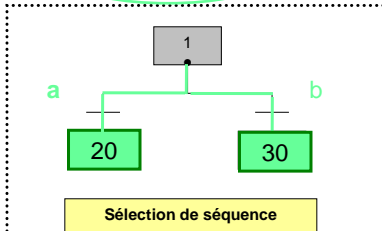
1: a vraie, b fausse



2: a fausse, b vraie

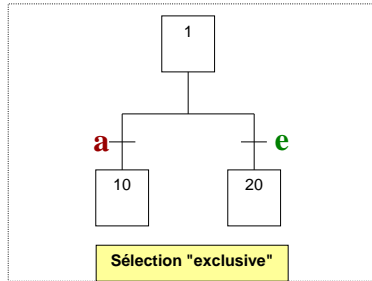


3: a vraie, b vraie



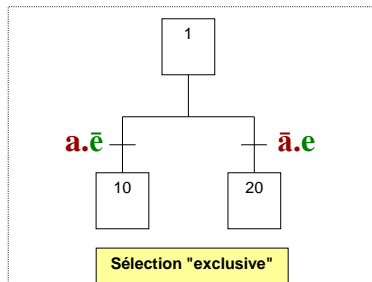
30

Pour obtenir une **sélection exclusive** entre plusieurs étapes: il faut que les réceptivités soient exclusives



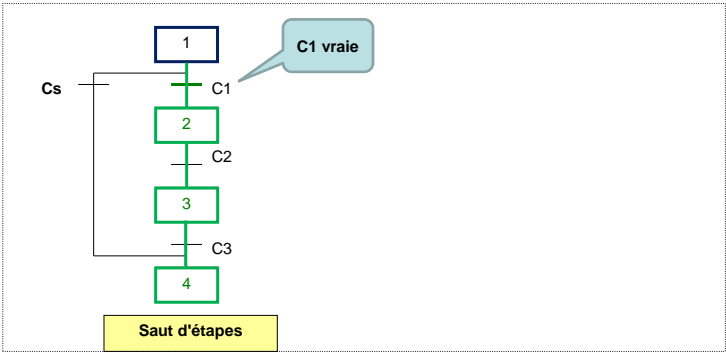
31

Pour obtenir une **sélection exclusive** entre plusieurs étapes: il faut que les réceptivités soient exclusives

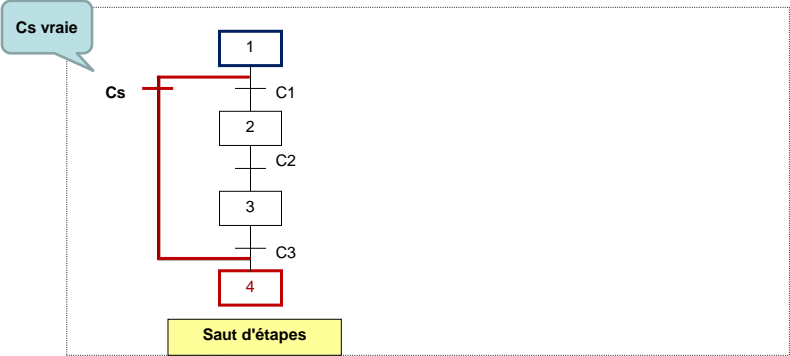


32

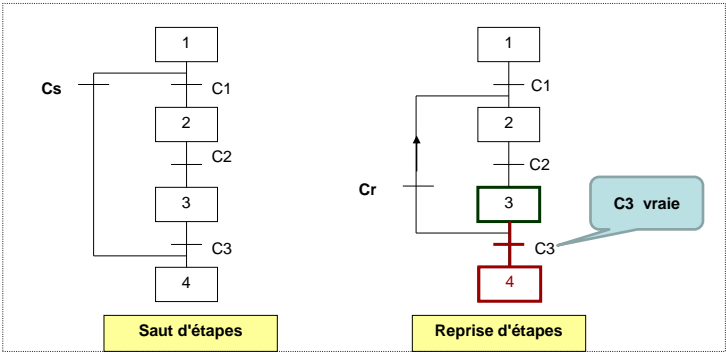
saut d'étapes et reprise d'étapes



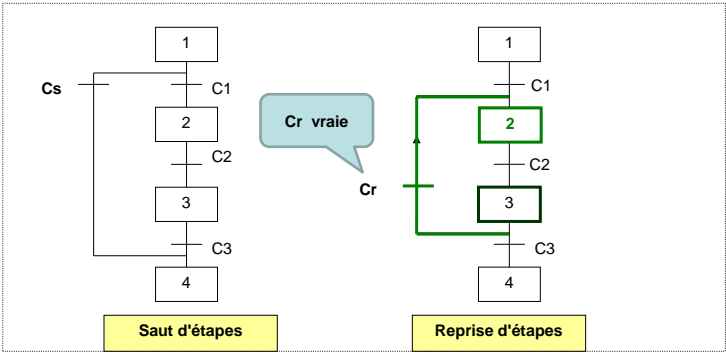
saut d'étapes et reprise d'étapes



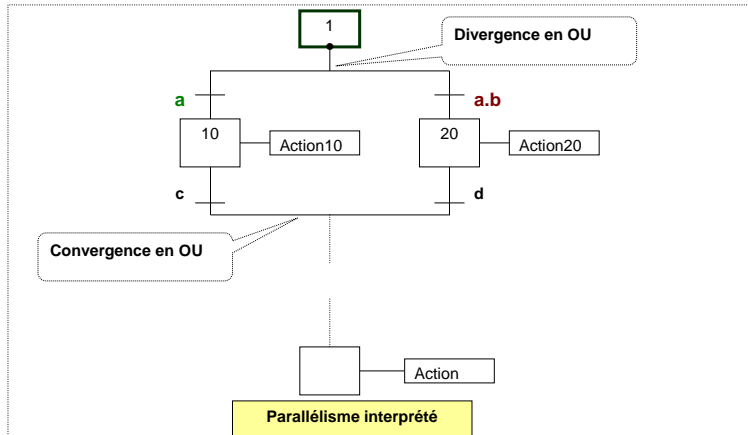
saut d'étapes et reprise d'étapes



saut d'étapes et reprise d'étapes



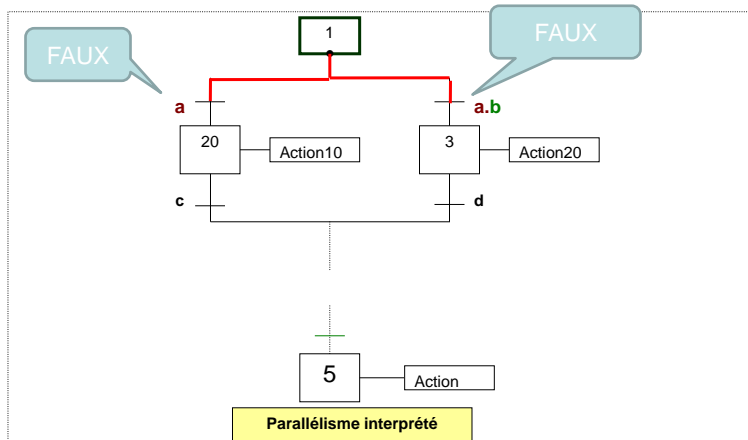
parallélisme interprété



37

parallélisme interprété :

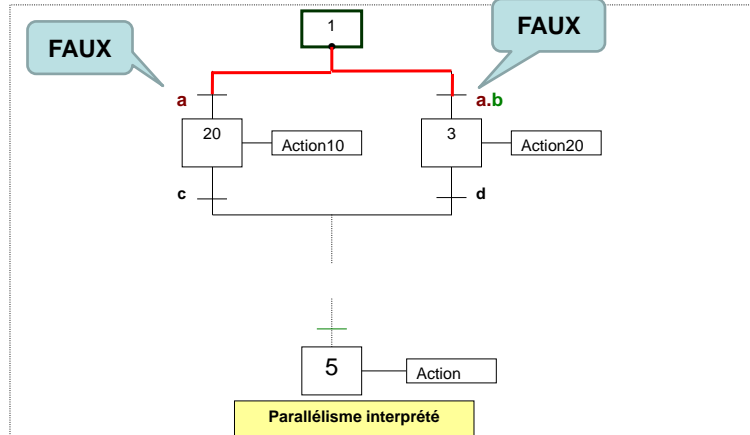
1 : Cas où $b = \text{fausse}$ Et $a = \text{fausse}$



38

parallélisme interprété :

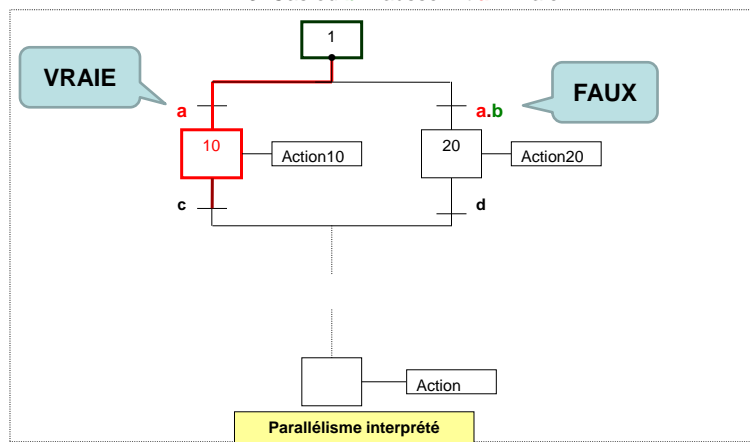
2: Cas où b = vraie Et a = fausse



39

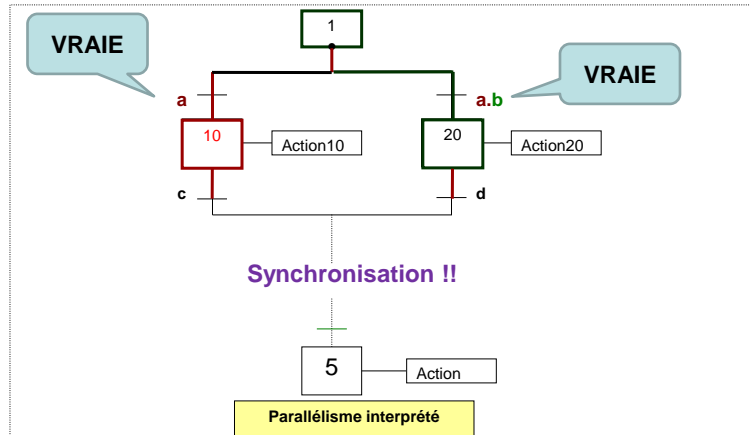
parallélisme interprété

3: Cas où b = fausse Et a = vraie



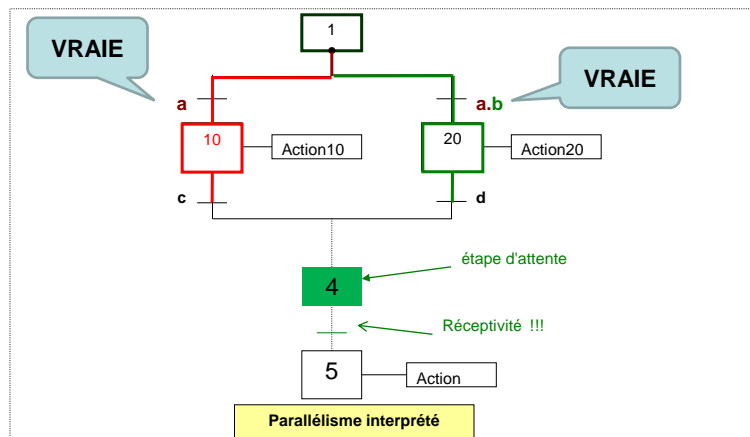
40

parallélisme interprété :

4: Cas où **b**= vraie Et **a** = vraie

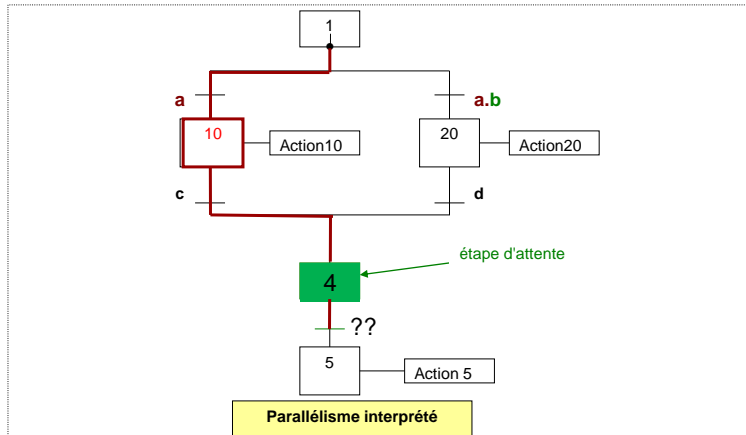
41

parallélisme interprété :

4: Cas où **b**= vraie Et **a** = vraie

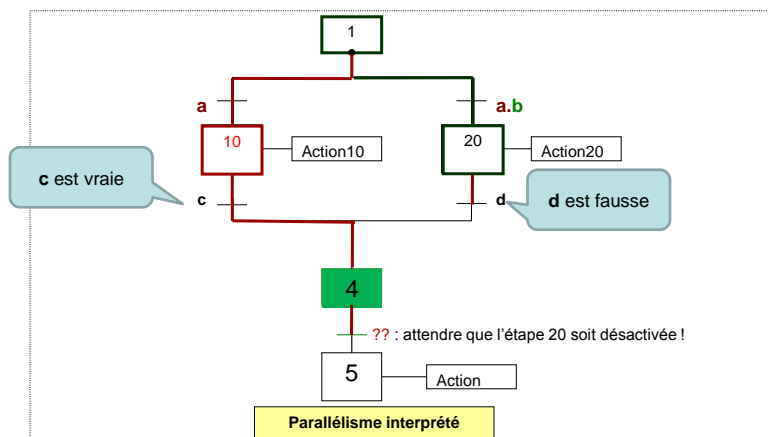
42

parallélisme interprété :

3: Cas où **b**= fausse Et **a** = vraie

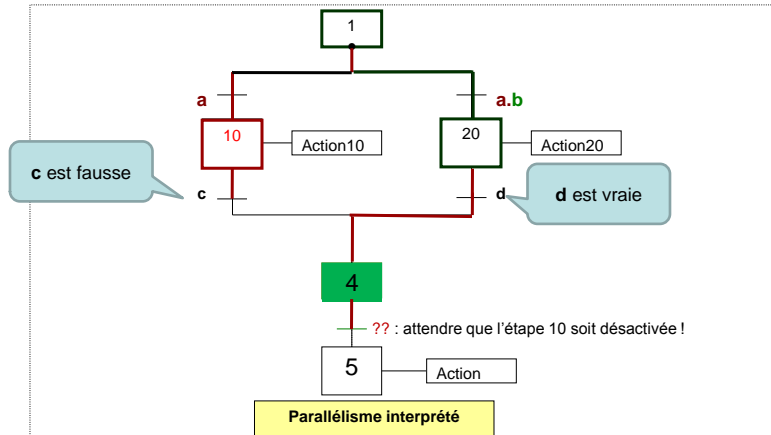
43

parallélisme interprété :

4: Cas où **b**= vraie Et **a** = vraie

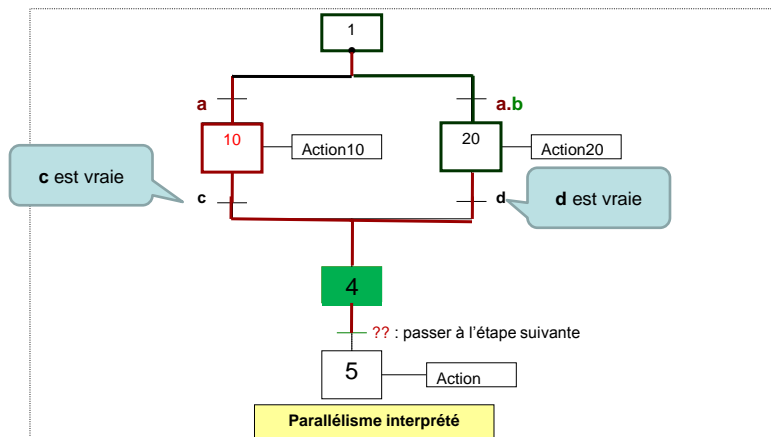
44

parallélisme interprété :

Cas où $b = \text{vraie}$ Et $a = \text{vraie}$ 

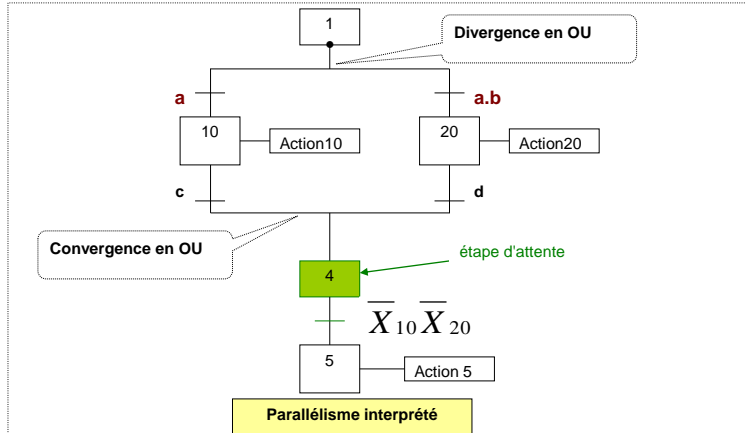
45

parallélisme interprété :

Cas où $b = \text{vraie}$ Et $a = \text{vraie}$ 

46

parallélisme interprété



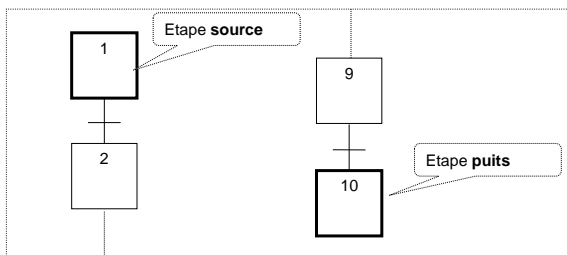
pour la synchronisation l'étape 4 (étape d'attente) est prévue avec la réceptivité : $\bar{X}_{10}\bar{X}_{20}$ (étape 10 et 20 désactivées)

47

Extensions des représentations

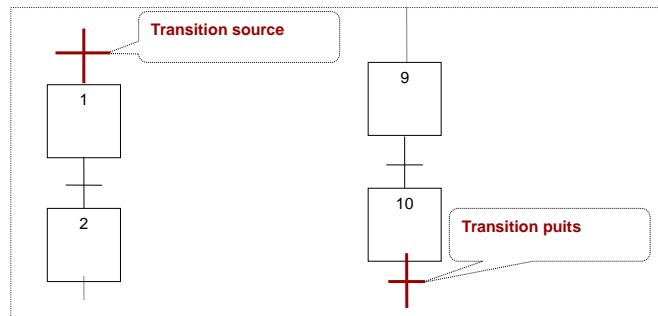
étape source

étape puits



48

transition source et puits



49

Macro-Etape

Une **macro-étape (ME)** est la représentation unique d'un ensemble d'étapes et de transition nommé "Expansion d'étapes", la macro-étape se substitue à une étape du GRAFCET.

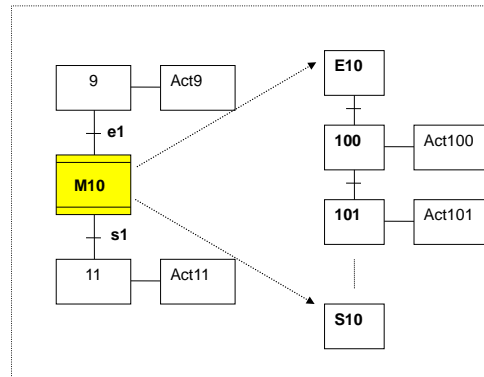
50

Une **macro-étape (ME)** est la représentation unique d'un ensemble d'étapes et de transition nommé "Expansion d'étapes", la macro-étape se substitue à une étape du GRAFCET.

Symbole:



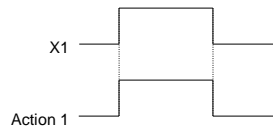
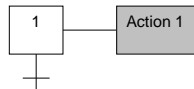
1. L'expansion de **ME** comporte une étape d'entrée et une étape de sortie repérées par E et S.
2. Tout franchissement de la transition amont de la macro-étape active l'étape E d'entrée de son Expansion
3. L'étape de sortie participe à la validation des transitions aval de la macro-étape.
4. La transition suivant la macro-étape n'est validée que lorsque la dernière étape de l'expansion de macro-étape est active.



NB : Il est préférable de ne pas associer d'actions aux étapes d'entrées et de sortie de la macro-étape

Classification des actions

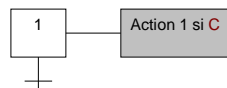
Action continue



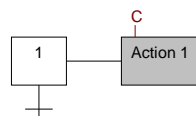
Action 1 = X1

(Exécution de l'action se poursuit tant que l'étape est a)

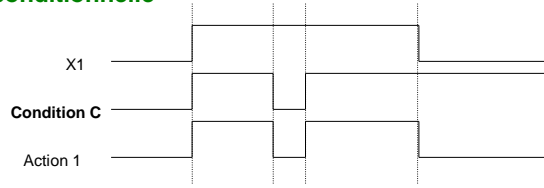
Action conditionnelle



OU

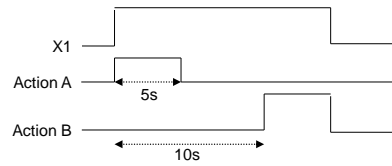
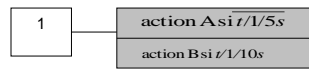


Action conditionnelle



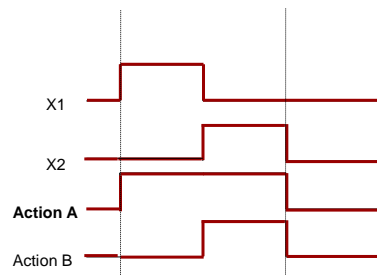
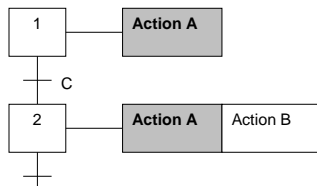
Action 1 = X1 • C
(exemple C:sécurité)

Action temporisée



53

Action maintenue



L'action A est maintenue dans les deux étapes 1 et 2

54

Les principaux grafquets que l'on peut trouver sont :

GRAF CET de surveillance : (de sécurité) ce GRAFCET décrit l'ensemble des procédures de sécurité du système, c'est le GRAFCET hiérarchiquement le plus important. L'arrêt d'urgence et les procédures de mise en route sont décrits dans ce GRAFCET.

GRAF CET de conduite : (ou GRAFCET des Modes de Marches) ce GRAFCET décrit l'ensemble des procédures de Marches (auto, Cycle/Cycle, Manuel,...) et des arrêts normaux.

GRAF CET de maintenance : Précise les procédures d'intervention de l'opérateur et de réglage de la partie opérative.

55

GRAF CET de Production : ce GRAFCET est le niveau de description du fonctionnement normal de l'automatisme. Ce GRAFCET est en général décomposé en plusieurs tâches représentant les différentes fonctions de l'automatisme.

56

FIN

&

MERCI

57

Structuration et hiérarchisation

Principe

Les Systèmes Automatisés de production sont de plus en plus complexes, afin de simplifier l'étude, la mise en oeuvre et la maintenance du système, il est nécessaire de **structurer la partie commande et la partie opérative**.

L'objectif essentiel de la structuration:

- permettre une approche progressive du fonctionnement d'un système automatisé, tant au niveau de l'analyse qu'au niveau de la représentation

Dans l'analyse structurée, le grafset global est décomposé en module, chacun de ces modules correspond à une fonction du système (Sécurité, modes de marche, etc.) ou à une sous partie de la P.O (Poste 1, Poste 2, Poste 3).

58

La structuration est soit Hiérarchique (GRAFCET Maître, GRAFCET Esclave) soit sans hiérarchie (communication entre 2 postes).

L'analyse structurée d'un système permet de décrire celui-ci depuis le niveau le plus général vers des niveaux de plus en plus détaillés

Cette structuration utilise les notions de **Taches** et de **Macro-étape**. Le système est décomposé soit suivant sa topologie (les taches correspondent à des parties opératives indépendantes) soit fonctionnellement (les taches correspondent à des fonctions du système) , soit plus généralement en combinant ces deux formes.

59

Les commandes de **forçage** et **figeage** de grafcet, sont des moyens supplémentaires qui permettent de préciser la hiérarchie des différents grafkets

La hiérarchie des différents grafkets issue de cette structuration est celle qui vient de la description du fonctionnement (succession séquentielle des taches définie par le processus).

60

En fait, il est souvent nécessaire de placer à un niveau hiérarchiquement supérieur des grafjets de gestions.

Les principaux grafjets que l'on peut trouver sont :

GRAFJET de surveillance : (de sécurité) ce GRAFCET décrit l'ensemble des procédures de sécurité du système, c'est le GRAFCET hiérarchiquement le plus important. L'arrêt d'urgence et les procédures de mise en route sont décrits dans ce GRAFCET.

GRAFJET de conduite : (ou GRAFCET des Modes de Marches) ce GRAFCET décrit l'ensemble des procédures de Marches (auto, Cycle/Cycle, Manuel,...) et des arrêts normaux.

GRAFJET de maintenance : Précise les procédures d'intervention de l'opérateur et de réglage de la partie opérative.

61

GRAFJET de Production : ce GRAFCET est le niveau de description du fonctionnement normal de l'automatisme. Ce GRAFCET est en général décomposé en plusieurs tâches représentant les différentes fonctions de l'automatisme.

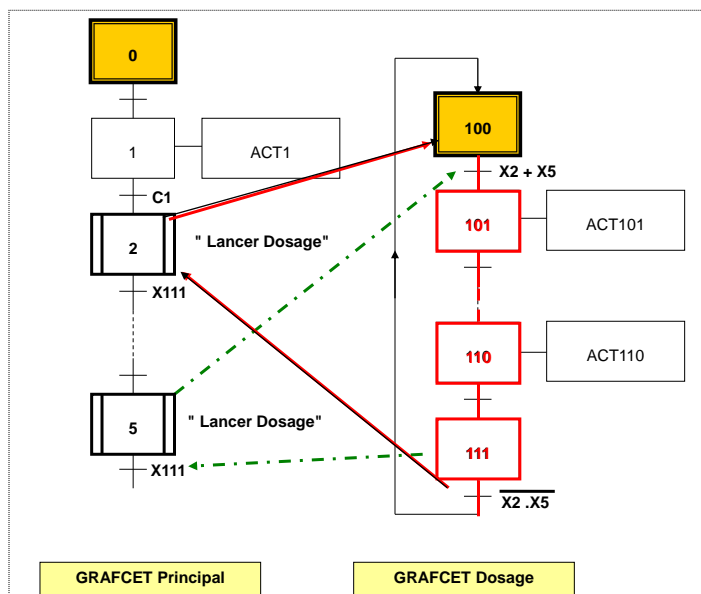
62

Un **sous programme** est un **grafcet indépendant** dont l'exécution et le déroulement sont synchronisés à un grafcet principal

Lorsqu'une **tache** doit être réalisée **plusieurs fois dans un cycle**, la description de cette tache sous la forme d'un grafcet indépendant permet de simplifier l'analyse et de simplifier la programmation.

Ce grafcet **indépendant** est un sous programme lancé par le programme (le grafcet) principal.

63



64

