

GRAPHE FONCTIONNEL DE COMMANDE DES ETAPES ET TRANSITIONS (GRAFCET)

III.1. INTRODUCTION

L'écriture d'un programme consiste à créer une liste d'instructions permettant l'exécution des opérations nécessaires au fonctionnement du système. Il existe différents types de langage de programmation :

- ✓ **Le langage booléen (Logigramme) ;**
- ✓ **Le langage à contact (Ladder) ;**
- ✓ **Le langage GRAFCET ;**
- ✓ **Le langage LIST ;**
- ✓ **Le langage Littéral structure.**

L'API traduit le langage de programmation en langage compréhensible directement par le microprocesseur. Ce langage est propre à chaque constructeur, il est lié au matériel mis en œuvre.

Chaque instruction du programme est composée :

- de l'opération à effectuer (la nature de l'opération est codée 1 ou 0).
- de la variable sur laquelle l'opération va être effectuée (variable de sortie, variable d'entrée, variable interne,...)
- de la nature de la variable (binaire, numérique, texte, ...)

Chaque instruction est écrite dans une partie de la mémoire appelée adresse ou label. En 1975, un groupe d'universitaires et industriels de la section "Systèmes Logiques" de l'AFCEC (Association Française de Cybernétique Economique et Technique) se sont fixés l'objectif de définir un formalisme adapté à la représentation des évolutions séquentielles d'un système et ayant les caractéristiques suivantes [11] :

- ✓ Simple ;
- ✓ Accepté par tous ;
- ✓ Intelligible à la fois par les concepteurs et les exploitants ;
- ✓ Fournissant potentiellement des facilités de passage à une réalisation, à base matérielle et/ou logicielle de l'automatisme ainsi spécifié.

Dans ce chapitre nous rappellerons la définition et notions de bases, Règles d'établissement du GRAFCET, Transitions et liaisons orientées, Règles d'évolution, Sélection de séquence et séquences simultanées, Organisation des niveaux de représentation, Matérialisation d'un GRAFCET avec des exemples pratiques.

III.2. DÉFINITION

Le GRAFCET (GRAPhe Fonctionnel de Commande par Etapes et Transitions) ou SFC (Sequential Function Chart) est un outil graphique qui décrit les différents comportements de l'évolution d'un automatisme et établit une correspondance à caractère séquentiel et combinatoire entre [9] :

- Les ENTREES, c'est-à-dire les transferts d'informations de la Partie Opérative vers la Partie Commande ;
- Les SORTIES, transferts d'informations de la Partie Commande vers la Partie Opérative.

C'est un outil graphique puissant, directement exploitable, car c'est aussi un langage pour la plupart des API existants sur le marché.

Le GRAFCET comprend : des étapes associées à des actions; des transitions associées à des réceptivités; des liaisons orientées reliant étapes et transitions.

III.3. DESCRIPTION DU GRAFCET

La description du comportement attendu d'un automatisme peut se représenter par un

GRAF CET d'un certain « niveau ». La caractérisation du « niveau » du GRAFCET nécessite de prendre en compte trois dimensions [7] :

Le point de vue, caractérisant le point de vue selon lequel un observateur s'implique dans le fonctionnement du système pour en donner une description. On distingue trois points de vue :

- ✓ Un point de vue système,
- ✓ Un point de vue Partie Opérative,
- ✓ Un point de vue Partie Commande.

La spécifications, caractérisant la nature des spécifications techniques auxquelles doit satisfaire la Partie Commande. On distingue trois groupes de spécifications :

- ✓ Spécifications fonctionnelles,
- ✓ Spécifications technologiques,
- ✓ Spécifications opérationnelles.

La finesse, caractérisant le niveau de détail dans la description du fonctionnement, d'un niveau global (ou macro-représentation) jusqu'au niveau de détail complet où toutes les actions et informations élémentaires sont prises en compte.

III.4. CONCEPTS DE BASE DU GRAFCET

III.4.1. Étape

Une étape symbolise un état ou une partie de l'état du système automatisé. L'étape possède deux états possibles : active représentée par un jeton dans l'étape ou inactive. L'étape i , représentée par un carré repéré numériquement, possède ainsi une variable d'état, appelée variable d'étape X_i . Cette variable est une variable booléenne valant 1 si l'étape est active, 0 sinon. La situation initiale d'un système automatisé est indiquée par une étape dite étape initiale et représentée par un carré double.

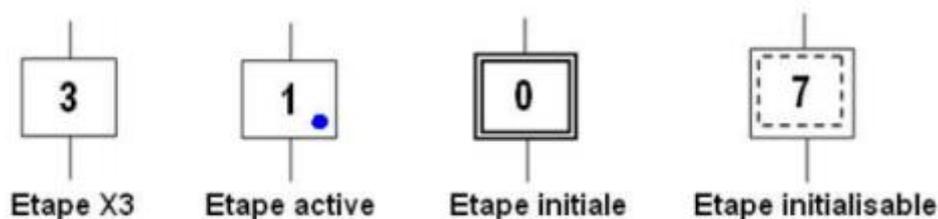


Figure II-1. Différents étapes [5]

Remarque : Dans un grafcet il doit y avoir au moins une étape initiale.

III.4.2. Actions associées aux étapes

A chaque étape est associée une action ou plusieurs, c'est à dire un ordre vers la partie opérative ou vers d'autres grafkets. Mais on peut rencontrer aussi une même action associée à plusieurs étapes ou une étape vide (sans action).

III.4.3. Transition

Une transition indique la possibilité d'évolution qui existe entre deux étapes et donc la succession de deux activités dans la partie opérative. Lors de son franchissement, elle va permettre l'évolution du système. A chaque transition est associée une condition logique appelée réceptivité qui exprime la condition nécessaire pour passer d'une étape à une autre.

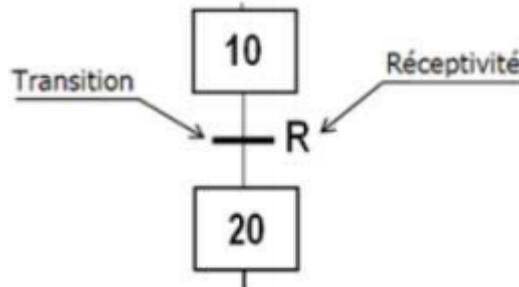


Figure II-3. Transition

La réceptivité qui est une information d'entrée qui est fournie par :

l'opérateur : pupitre de commande, la partie opérative : états des capteurs, du temps, d'un comptage ou toute opération logique, arithmétique...

du graf cets : d'autres graf cets pour la liaison entre graf cets ou de l'état courant des étapes du graf cets (les Xi), d'autres systèmes : dialogue entre systèmes,

Remarque : Si la réceptivité n'est pas précisée, alors cela signifie qu'elle est toujours vraie (=1).

III.4.4. Liaisons orientées

Elles sont de simples traits verticaux qui relient les étapes aux transitions et les transitions aux étapes. Elles sont normalement orientées de haut vers le bas. Une flèche est nécessaire dans le cas contraire.

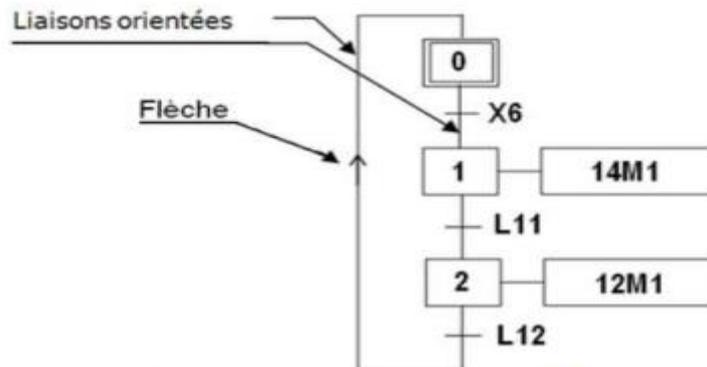


Figure II-4. Liaisons orientées

III.5. CLASSIFICATION DES ACTIONS ASSOCIÉES AUX ÉTAPES

L'action associée à l'étape peut être de 3 types : continue, conditionnelle ou mémorisée.

Les actions peuvent être classées en fonction de leur durée par rapport à celle de l'étape.

III.5.1. Actions continues

L'ordre est émis, de façon continue, tant que l'étape, à laquelle il est associé, est active.



Figure II-5. Actions continues

III.5.2.2. Action retardée (Type D) (Delay)

Le temps intervient dans cet ordre conditionnel comme condition logique. L'indication du temps s'effectue par la notation générale " t / xi / D " dans laquelle " xi " indique l'étape prise comme origine du temps et " d " est la durée du retard. Exemple : " t/x6/ 5s " : prendra la valeur logique 1, 5s après la dernière activation de l'étape 6.

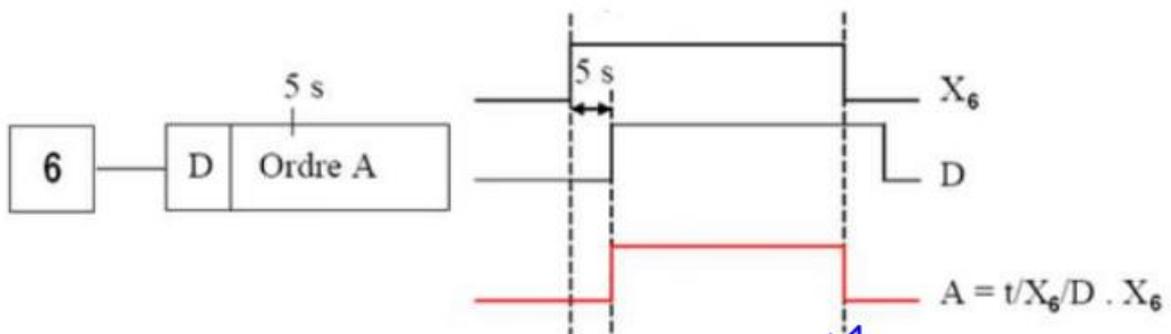


Figure II-7. Action retardée : Type D

III.5.2.3. Action de durée limitée (Type L) (Limited)

L'ordre est émis dès l'activation de l'étape à laquelle il est associé ; mais la durée de cet ordre sera limitée à une valeur spécifiée. L'ordre "A" est limité à 2s après l'activation de l'étape 4.

III.5.3. Action maintenue sur plusieurs étapes

Afin de maintenir la continuité d'une action sur plusieurs étapes, il est possible de répéter l'ordre continu relatif à cette action, dans toutes les étapes concernées ou d'utiliser une description sous forme de séquences simultanées (Les séquences simultanées seront traitées ultérieurement).

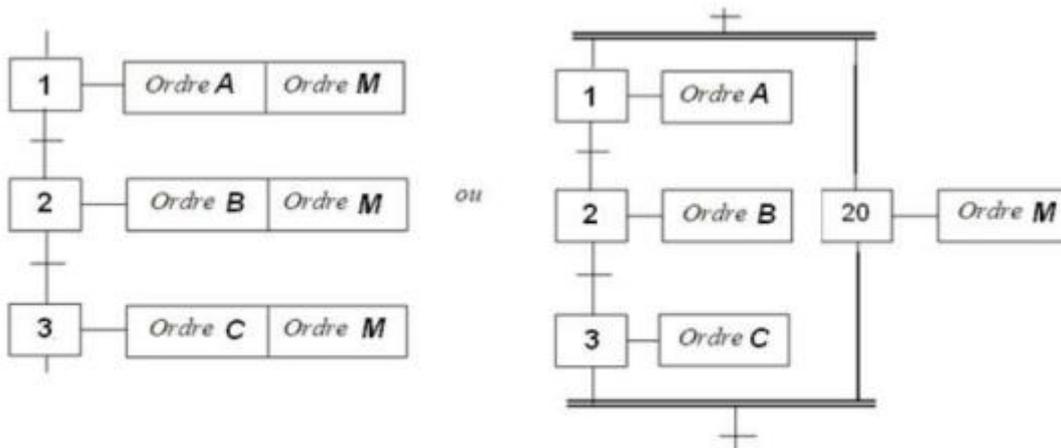


Figure II-9. Action maintenue sur plusieurs étapes

III.5.4. Action mémorisée

Le maintien d'un ordre, sur la durée d'activation de plusieurs étapes consécutives, peut également être obtenu par la mémorisation de l'action, obtenue par l'utilisation d'une fonction auxiliaire appelée fonction mémoire.

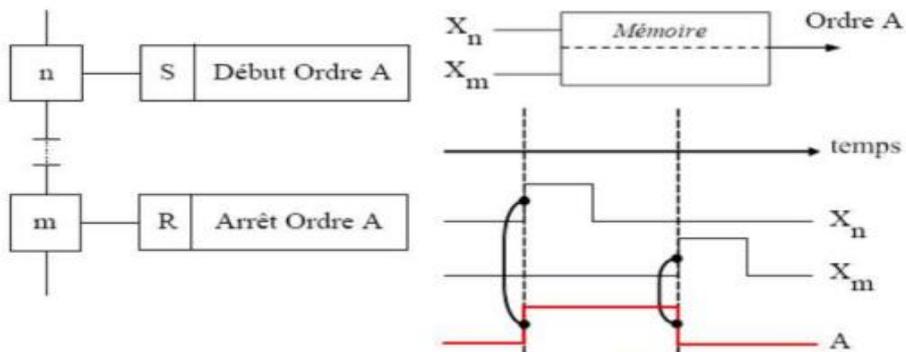


Figure II-10. Action mémorisée

III.6. REGLES D'ÉVOLUTION D'UN GRAFCET

III.6.1. Règle N° 1 : Condition initiale

A l'instant initial, seules les étapes initiales sont actives.

III.6.2. Règle N° 2 : Franchissement d'une transition

Pour qu'une transition soit validée, il faut que toutes ses étapes amont (immédiatement précédentes reliées à cette transition) soient actives. Le franchissement d'une transition se produit lorsque la transition est validée, ET seulement si la réceptivité associée est vraie.

III.6.3. Règle N° 3 : Evolution des étapes actives

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

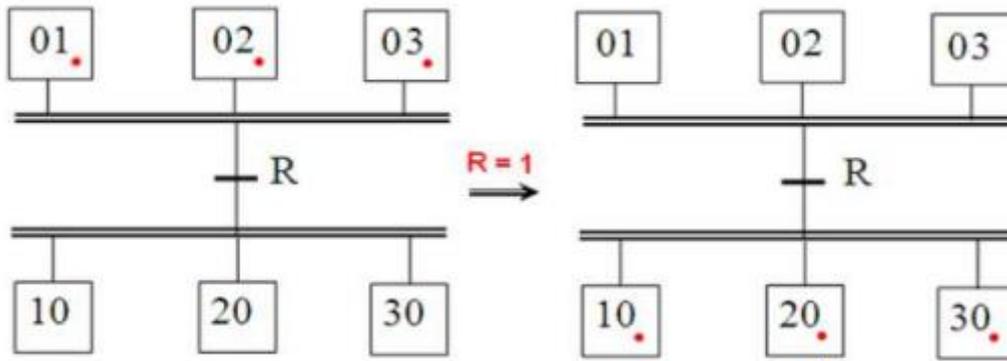


Figure II-11. Evolution des étapes actives

III.6.4. Règle N° 4 : Franchissement simultané

Toutes les transitions simultanément franchissables à un instant donné sont simultanément franchies.

III.6.5. Règle N° 5 : Conflit d'activation

Si une étape doit être simultanément désactivée par le franchissement d'une transition aval, et activée par le franchissement d'une transition amont, alors elle reste active. On évite ainsi des commandes transitoires (néfastes à la partie opérative).

III.7. STRUCTURES DE BASE

III.7.1. Notion de Séquence

Une séquence, dans un Grafcet, est une suite d'étapes à exécuter l'une après l'autre. Autrement dit chaque étape ne possède qu'une seule transition AVAL et une seule transition AMONT.

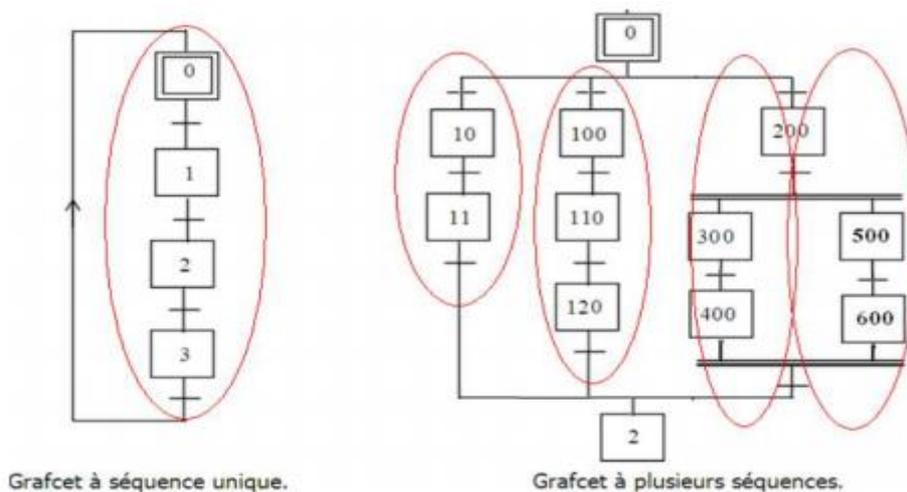


Figure II-12. Notion de Séquence

III.7.2. Saut d'étapes et reprise de séquence

Le saut d'étapes permet de sauter une ou plusieurs étapes lorsque les actions associées sont inutiles à réaliser, La reprise de séquence (ou boucle) permet de reprendre, une ou plusieurs fois, une séquence tant qu'une condition n'est pas obtenue.

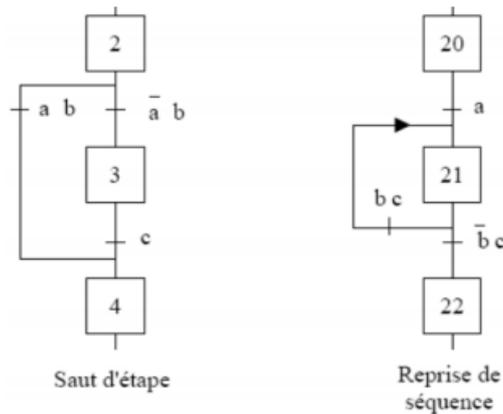


Figure II-13. Saut et reprise

III.7.3. Divergence – Convergence en OU

On dit qu'il y a Aiguillage ou divergence en OU lorsque le grafcet se décompose en deux ou plusieurs séquences selon un choix conditionnel. Comme la divergence en OU on rencontre aussi la convergence en OU. On dit qu'il y a convergence en OU, lorsque deux ou plusieurs séquences du grafcet converge vers une seule séquence.

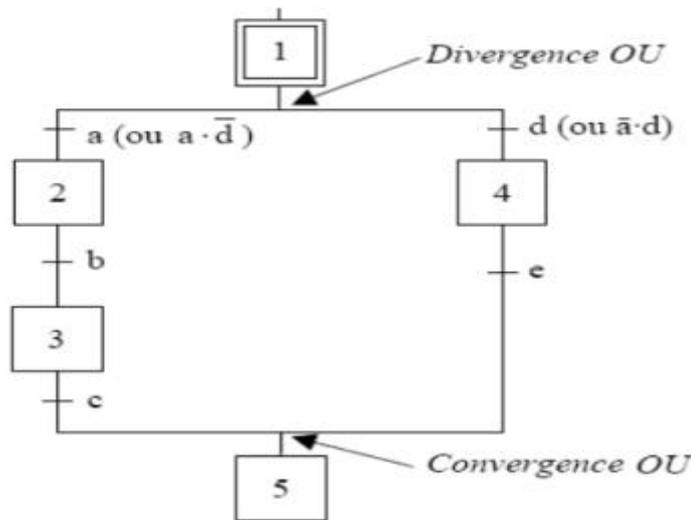


Figure II-14. Divergence – Convergence en OU

Si les deux conditions a et d sont à 1 simultanément, les étapes 2 et 4 vont devenir actives simultanément, situation non voulue par le concepteur. Donc elles doivent être des conditions exclusives.

III.7.4. Divergence – Convergence en ET

Au contraire de l'aiguillage où ne peut se dérouler qu'une seule activité à la fois, On dit qu'on se trouve en présence d'un parallélisme structurel, si plusieurs activités indépendantes pouvant se dérouler en parallèle. Le début d'une divergence en ET et la fin d'une convergence en ET d'un parallélisme structurel sont représentés par deux traits parallèles.

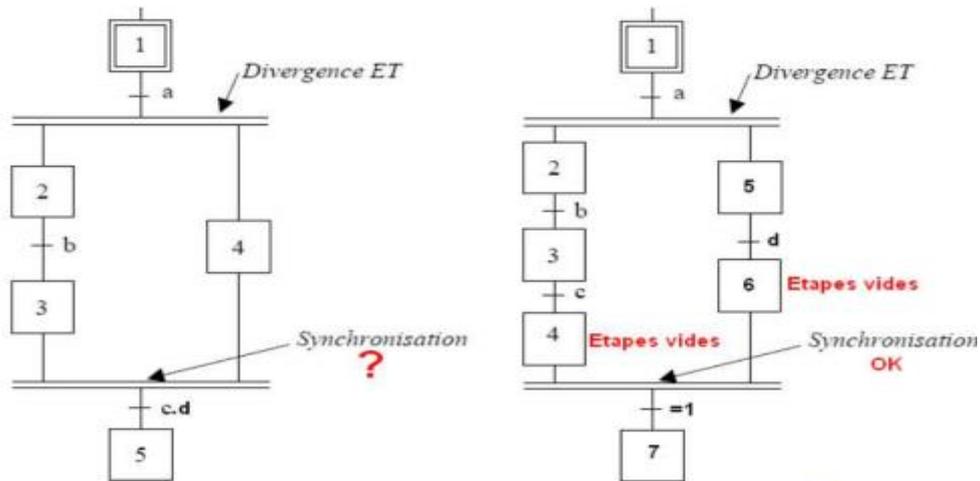


Figure II-15. Divergence – Convergence en ET [3]

La synchronisation permet d’attendre la fin de plusieurs activités se déroulant en parallèle, pour continuer par une seule.

III.8. LIAISON ENTRE GRAFCETS

Une étape dans un grafcet peut servir comme réceptivité à une autre étape d'un autre grafcet. Cette méthode est utilisée aussi pour synchroniser deux grafcets c'est à dire rendre l'évolution de l'un dépendante de l'évolution de l'autre.

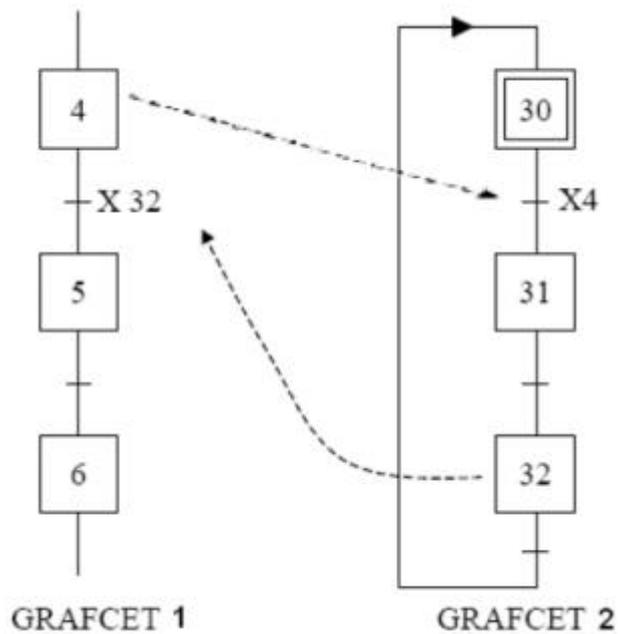


Figure II-16. Liaison entre grafcets [3]