

# Le GRAFCET

## 1. Actions particulières

- 1.1 Action continue
- 1.2 Action conditionnelle
- 1.3 Action temporisée
- 1.4 Action de comptage d'un temps
- 1.5 Actions fugitives ou ponctuelles
- 1.6 Actions mémorisées

## 2. Réceptivités particulières

- 2.1 Réceptivité fonction du temps
- 2.2 Réceptivité faisant intervenir un changement d'état
- 2.3 Détection de fronts

## 3. Principales définitions concernant le GRAFCET

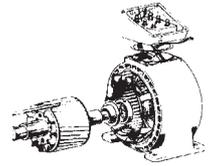
## 4. Représentation des automatismes séquentiels

( Document CETIM)

## 5. Macro-représentation

- 5.1 sous-programme ou reprise de séquence
- 5.2 Classification des sous-programmes
- 5.3 Macro-étape

## 6. Hiérarchisation et forçage



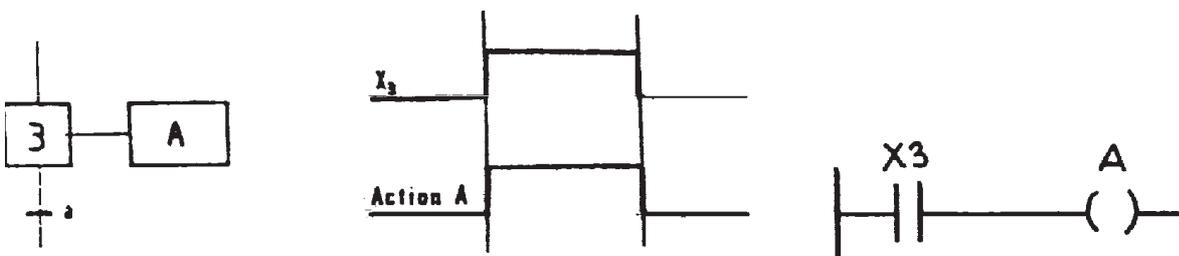
# 1. Actions particulières

Action associée à une étape ( distinction entre la durée de l'action et la durée d'activité de l'étape ).  
 On note

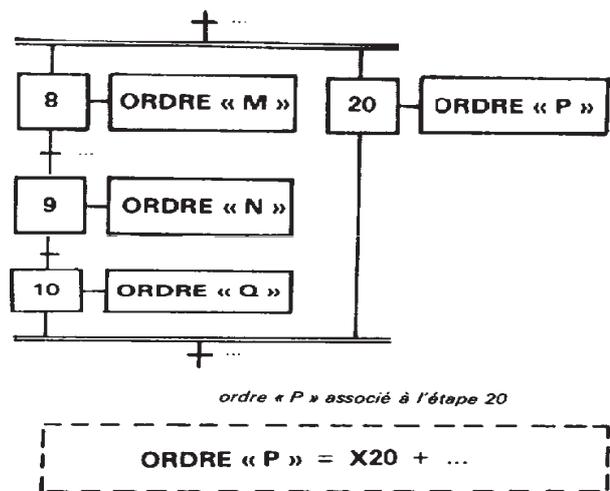
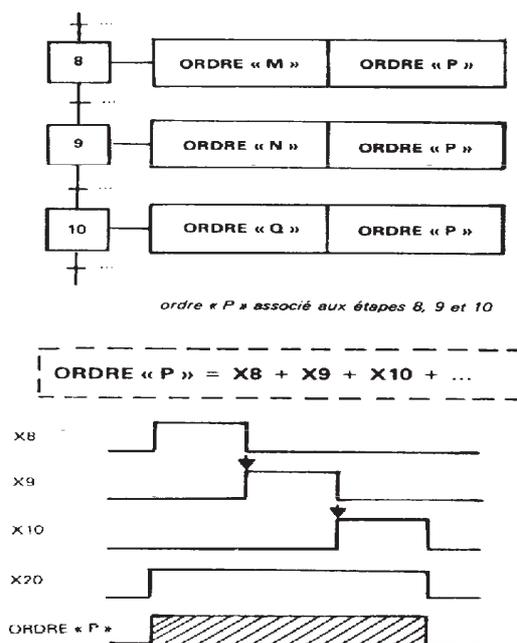
- $X_i$  = Etat actif de l'étape  $i$
- $X_1 = 1$  si l'étape 1 est active
- $X_3 = 1$  si l'étape 3 est inactive

## 1. Action continue

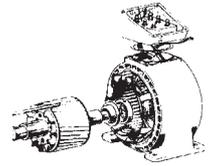
L'action associée à l'étape se continue tant que l'étape à laquelle elle est associée est vraie.



C'est une technologie monostable:  
 Il faut maintenir l'action pendant les étapes intéressées

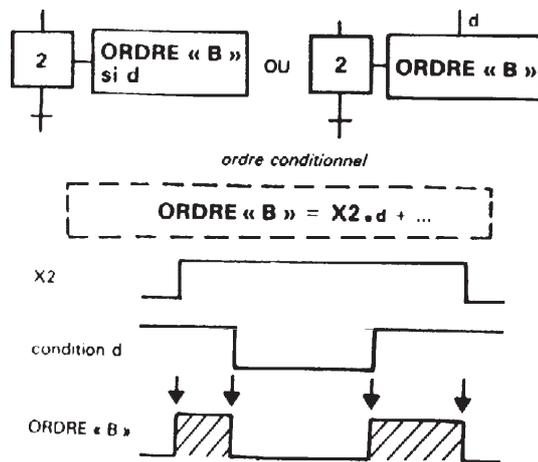


Ces deux GRAFCET sont équivalents

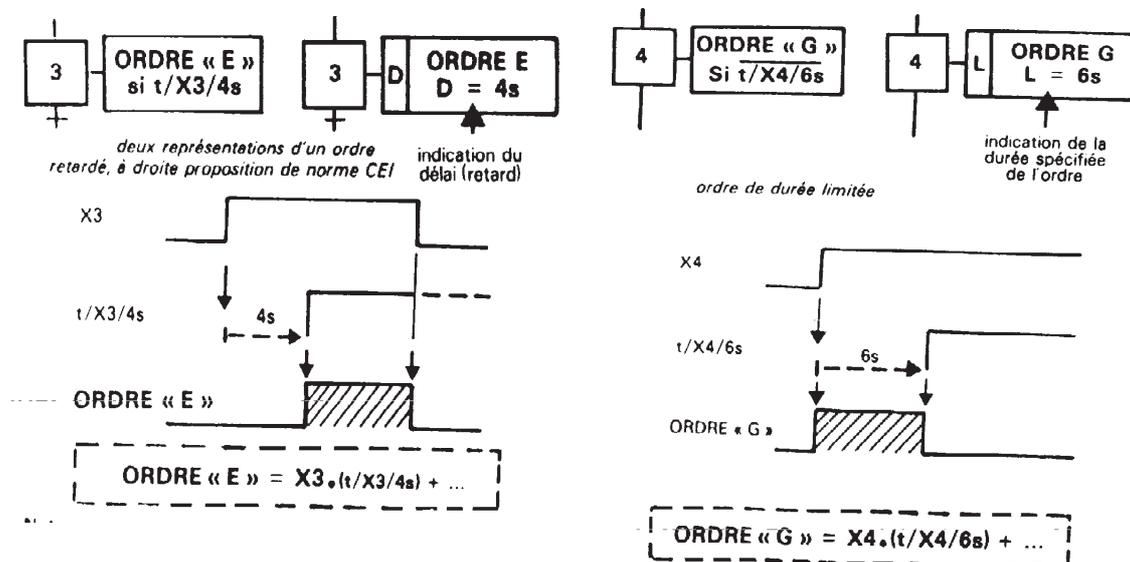


## 1.2 Action conditionnelle

C'est une action continue dont l'exécution est soumise à la réalisation d'une condition logique U.

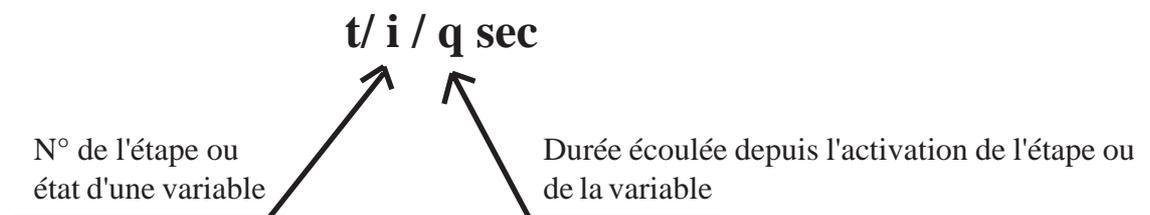


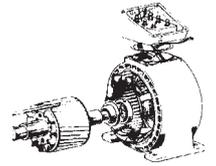
## 1.3 Action temporisée



C'est une action conditionnelle dans laquelle le temps intervient comme condition logique

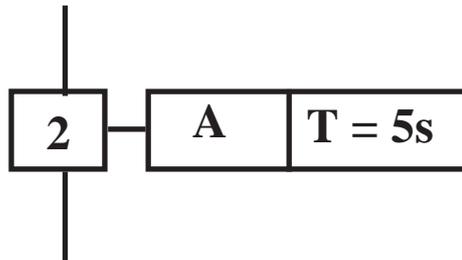
Notation :



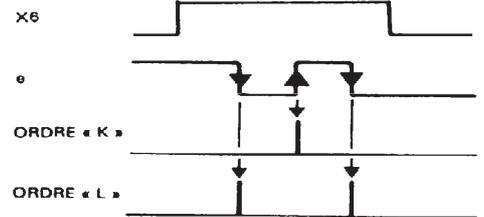
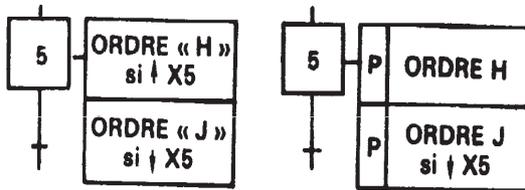


### 1.4 Action de comptage d'un temps

T=q sec en action, par exemple lancement d'une temporisation de 5 secondes



### 1.5 Actions fugitives ou ponctuelles



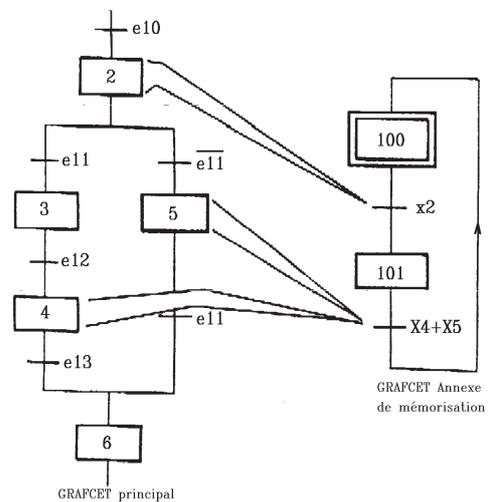
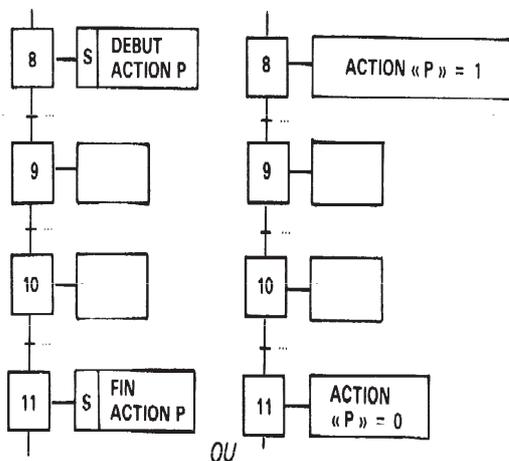
La durée de ces actions est très brève mais non nulle

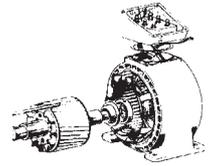
ORDRE « K » = X6 . ↑ e + ...  
 ORDRE « L » = X6 . ↓ e + ...

### 1.6 Action mémorisée

L'action P est active à partir de l'étape 8 jusqu'à l'étape 11

Mémorisation d'étape par un GRAFCET annexe





Définition d'un ordre mémorisé

$$A = X8 + A.\overline{X27}$$

Généralement

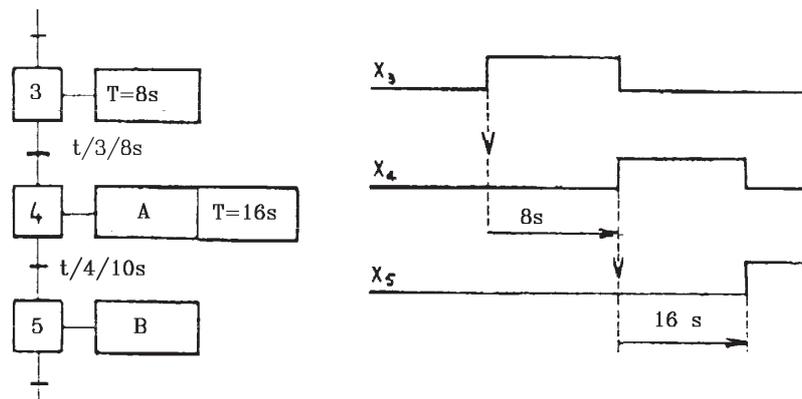
$$A = U(X_i) + A.U(\overline{X_j})$$

Réunion de toutes les étapes pour la mise à 1 de A

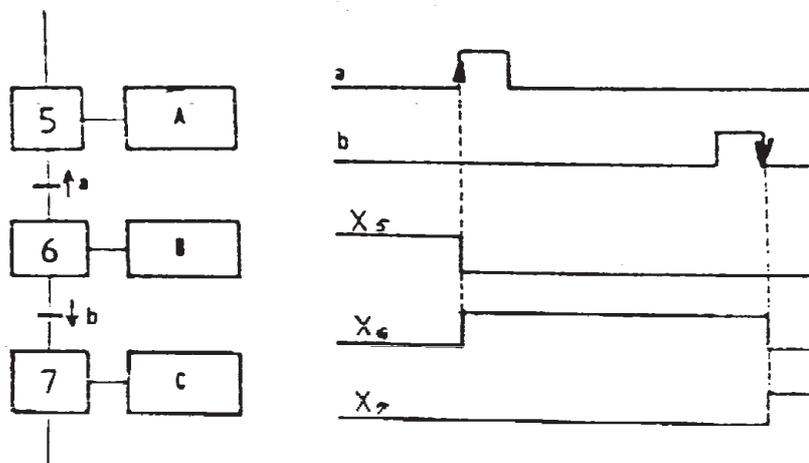
Réunion de toutes les étapes pour la mise à 0 de A

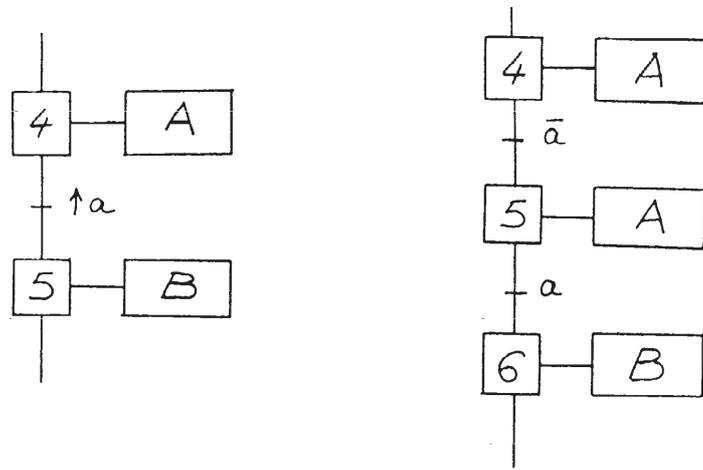
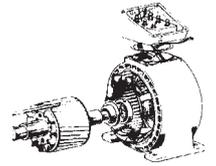
## 2. Les réceptivités

### 2.1 Réceptivité fonction du temps



### 2.2 Réceptivité faisant intervenir un changement d'état





Remarque on peut remplacer cette réceptivité par le GRAFCET de droite:  
 Il faut d'abord vérifier la présence de a puis son absence (  $a = 0$  ), et enfin sa présence  $a=1$ .

Front montant passage de 0 à 1.

Front descendant passage de 1 à 0.

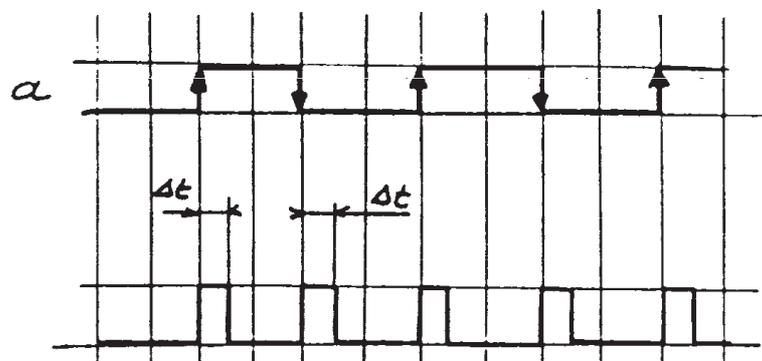
Cette notion de front sera utilisée lorsque l'on désire faire apparaître le changement d'état:

\* Le fait d'appuyer sur un bouton ( ou relâcher ) et non le fait que le bouton soit enfoncé ou coincé.

### 2.3 Détection des fronts.

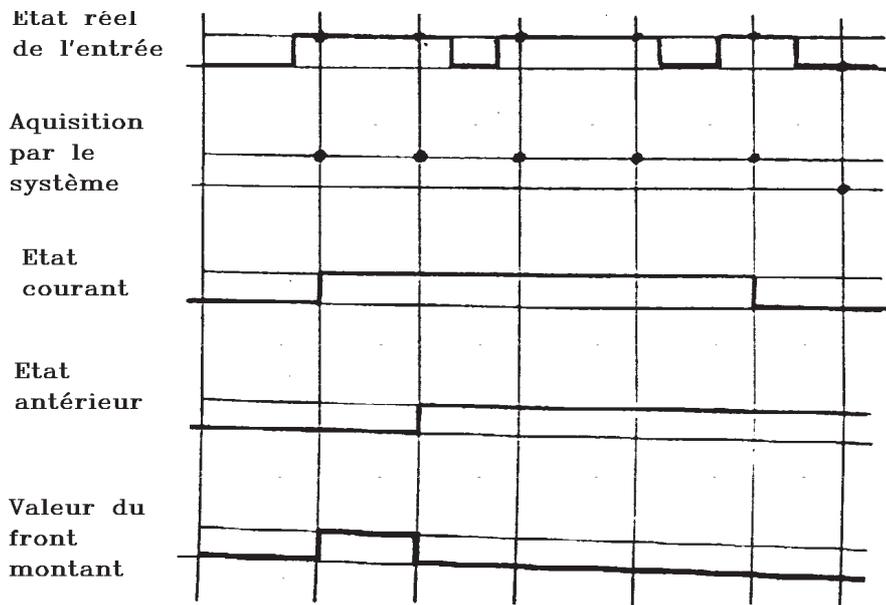
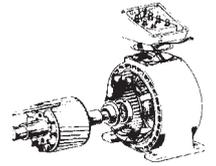
#### - Principe

On provoque une impulsion de largeur  $\Delta t$  à chaque changement d'état de l'entrée a ( détection de front montant et descendant ).



#### - Pour les API

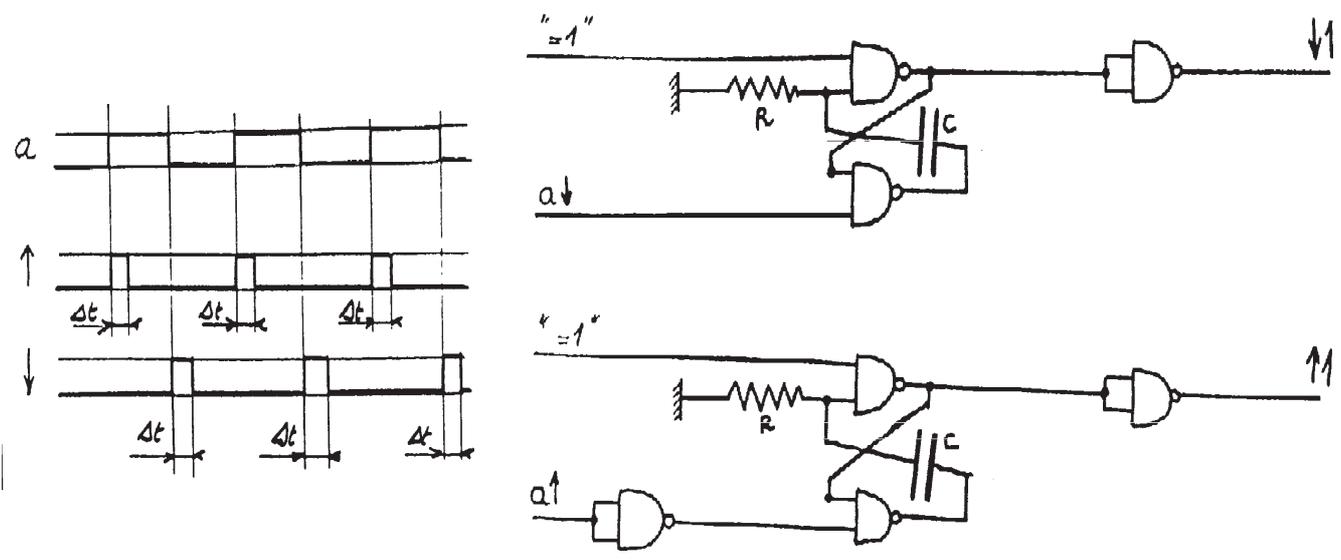
Détection du front montant d'une entrée: C'est le résultat logique d'un test, le front montant est égal à 1 lorsque l'état courant vaut 1 et que l'état antérieur vaut 0. L'acquisition des entrées se fait à des intervalles de temps déterminés.

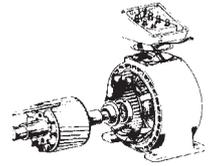


**- Réalisation**

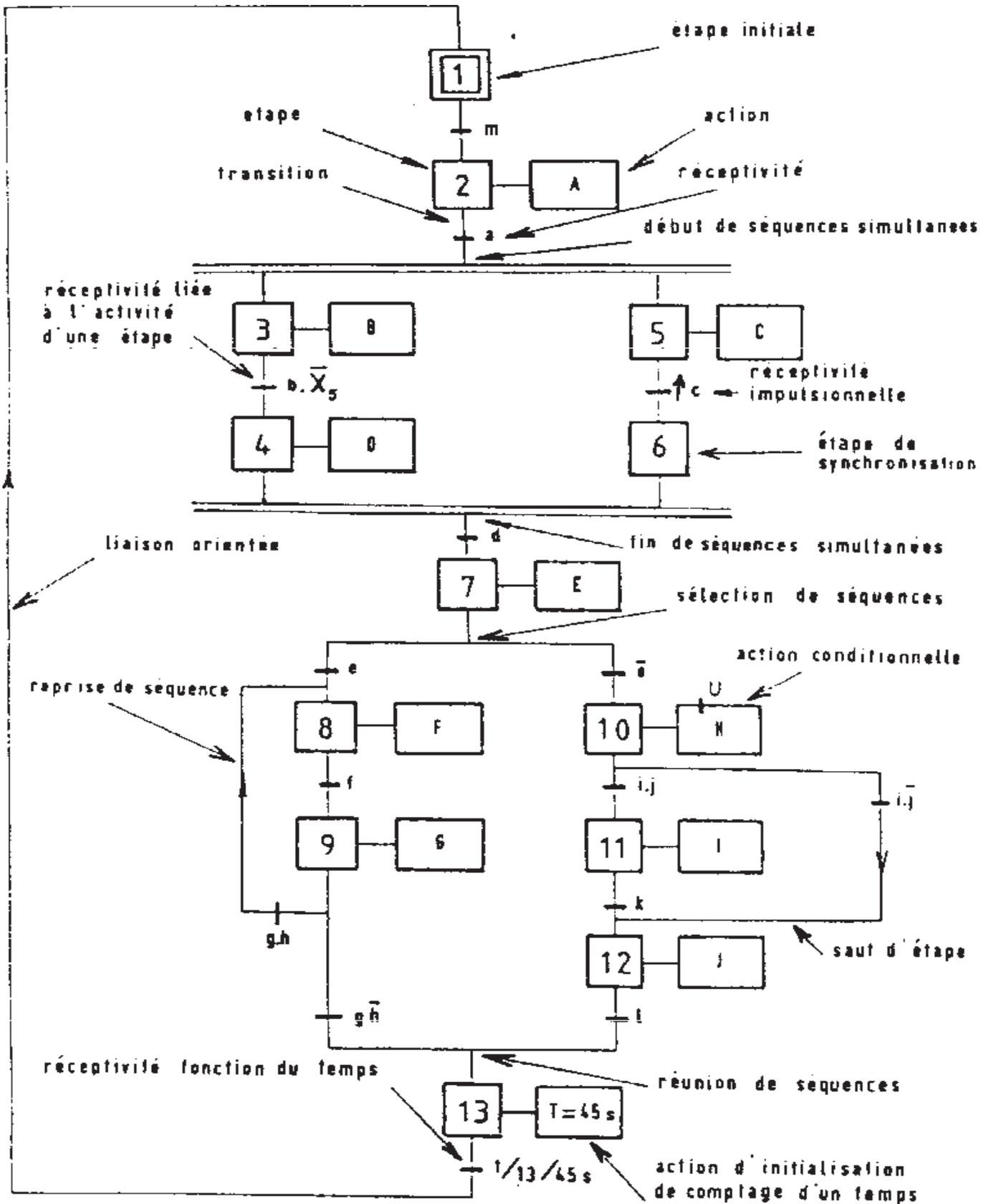
A la détection du changement d'état de la variable d'entrée a, on provoque le lancement d'une impulsion à la valeur logique 1 pendant un temps  $\Delta t$ . C'est un circuit monostable (circuit implanté dans les API permettant de délivrer une information à 1 pendant un temps déterminé.

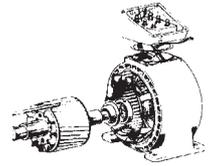
En logique câblée électronique, le circuit séquentiel sera constitué d'une bascule temporisée, une pour chaque front, ou d'un monostable.





### 3. Principales définitions du GRAFCET





## 4. Représentation des automatismes séquentiels

Voir documents

## 5. Macro-représentation

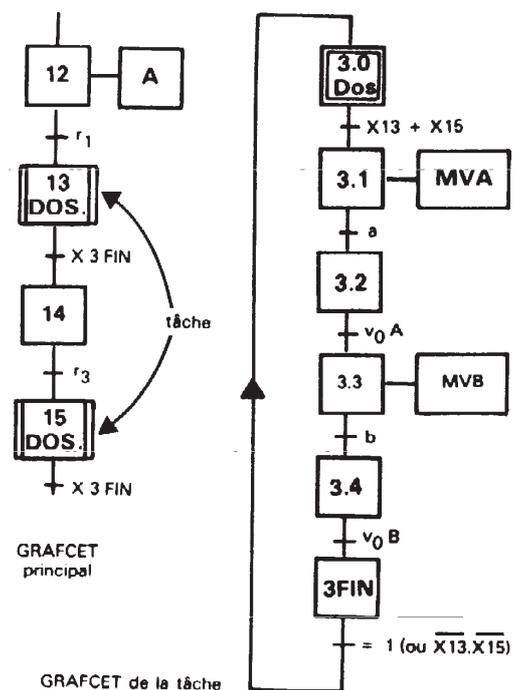
### 5.1 Sous programme, tâche ou reprise de séquence

Si lors d'une analyse, certaines séquences apparaissent de façon répétitive, on peut utiliser la notion de tâche ou reprise de séquence.

Le symbolisme préconisé est représenté par la figure ci dessous. Il apparaît clairement que l'utilisateur doit gérer ses conditions de réceptivités d'entrée et de sortie de la tâche, afin d'éviter tout conflit d'accès.

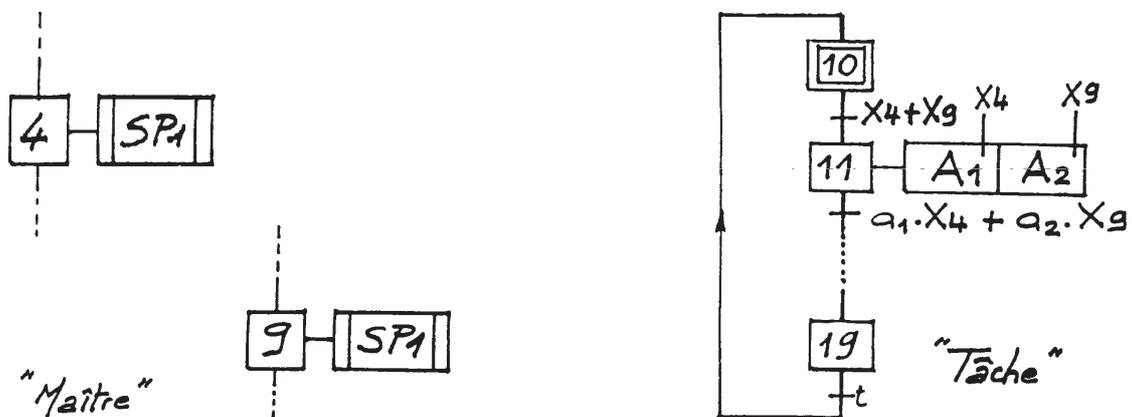
Cette décomposition du GRAFCET permet une approche progressive et structurée des applications industrielles complexes.

Ce sous programme est dit indépendant, car il exécute toujours la même tâche quelque soit l'étape d'appel.

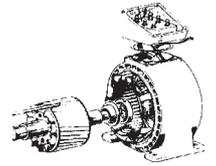


### 5.2 Classification des sous-programmes

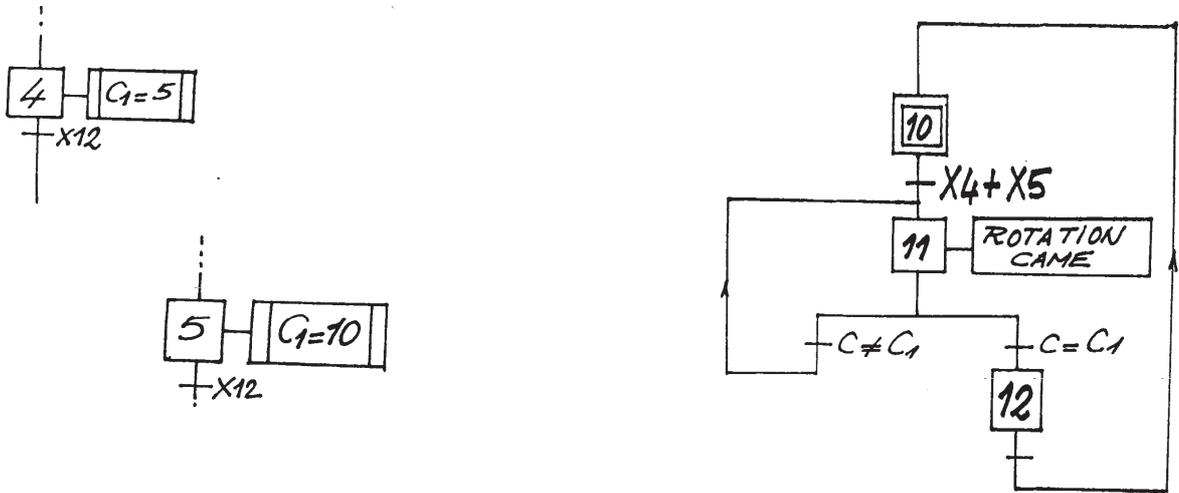
- Indépendant : cf. 5.1
- Sous programme multiplexé:



Exécution d'un s/p en fonction de l'évolution du GRAFCET maître



- Sous programme paramètre



Utilisation de paramètres fixés par le programme principal

### 5.3 Macro-étape

#### 5.31 Définitions

- Une macro-étape est la représentation unique d'un ensemble d'étapes et de transitions nommée expansion de la macro-étape.

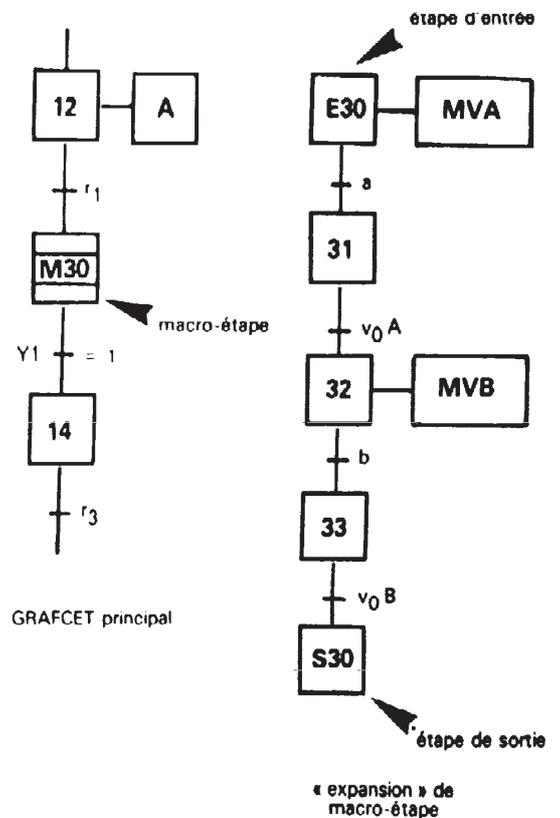
- Une Expansion de macro-étape comporte une étape d'entrée et une étape de sortie

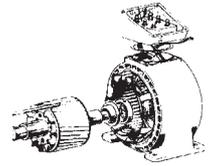
- Le franchissement d'une transition en amont de la macro-étape active l'étape d'entrée de son expansion.

- L'étape de sortie participe à la validation des transitions de sortie de la macro-étape

- Une expansion est liée à une seule macro-étape (cette restriction a pour but d'éviter des conflits d'accès), exemple 1.

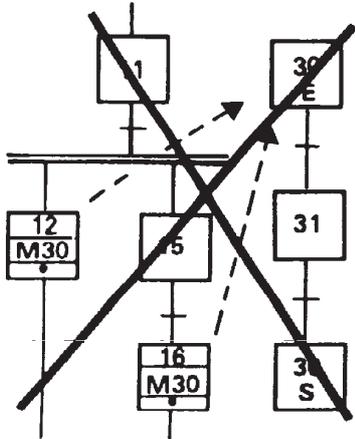
- Une expansion de macro-étape peut elle même comporter d'autre macro-étapes, certaines configurations sont à éviter exemple 2.



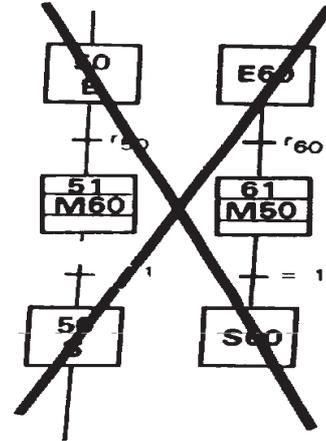


### 5.32 Exemples

N°1



N°2



## 6. Hiérarchisation et forçage

### 6.1 Introduction

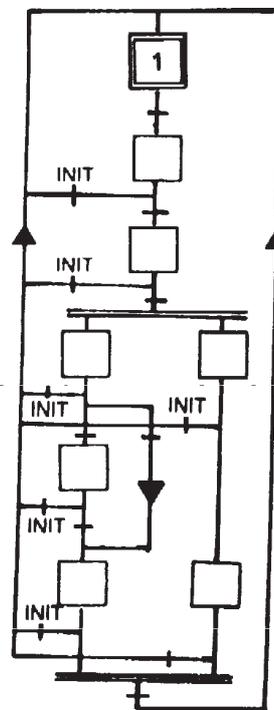
Il s'avère souvent nécessaire de pouvoir "forcer" une situation d'une partie commande isolée, c'est à dire de provoquer une évolution vers une situation définie quelle que soit la situation en cours. c'est le cas lorsque la partie commande est défaillante ou dans le cas d'une situation bloquée. La consigne de forçage de situation à souvent pour objectif de "ré-initialiser" la partie commande. En général un bouton poussoir INI sur le pupitre.

Autres cas :

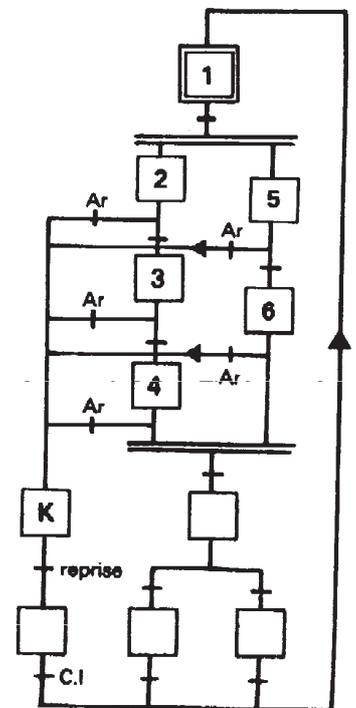
- Arrêt d'urgence
- Coupure d'énergie (PO)
- Arrêts avec reprise

De tels comportements ( évolution d'une situation quelconque vers une situation déterminée, lors de l'apparition d'un événement ) peuvent être décrit au moyen de transitions.

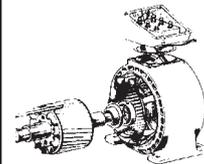
De telles représentations sont très lourdes, voire inextricables.



ex 1 : « ré-initialisation »



ex 2 : arrêt avec reprise



## 6.2 Représentation hiérarchisée de la partie commande

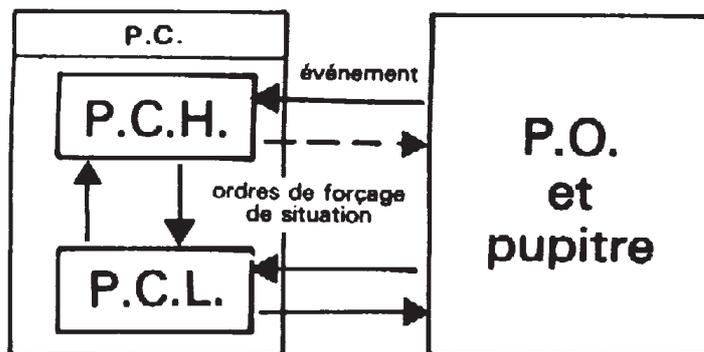
Représentation la plus simple deux niveaux

- 1) Niveau local définissant la partie locale du processus ( P C locale notée PCL ) traitant les informations issues de la PO et du pupitre pour élaborer les ordres.
- 2) Un niveau défini comme hiérarchiquement supérieur au niveau local (P C Hiérarchique notée PCH). Cette partie peut recevoir des informations de la PO, du pupitre, est de la PCL, et émettre des ordres, notamment vers la PCL.

Représentation à deux niveaux

Nous pouvons alors convenir de définir un ordre spécifique de forçage de la situation de PCL par PCH sous la formulation fonctionnelle suivante.

Mettre la partie commande de niveau inférieur (PCL) dans une situation prédéterminée.



## 6.3 Ordre de forçage hiérarchisé

- Notation

**F/PCL:0**

L'ordre de forçage F/PCL : 0 Provoque simultanément

- L'activation de l'étape 0 du GRAFCET de la partie commande du processus PCL
- La désactivation de toutes les autres étapes de PCL

On peut forcer simultanément la partie commande de plusieurs parties commandes d'un niveau inférieur, indépendante.

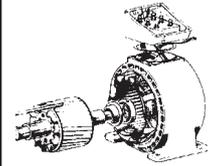
**F/PCL1: ( 5,8) / Gestion : (4)**

Cas particulier :

Forçage de la situation vide (), c'est à dire la désactivation de toutes les étapes de la partie commande. Cet ordre de "désactivation" d'une partie commande PCL par une partie commande PCH, hiérarchiquement supérieure est alors par exemple :

**F/PCL2 : ( )**

Il faut alors gérer la réactivation du PLC2.



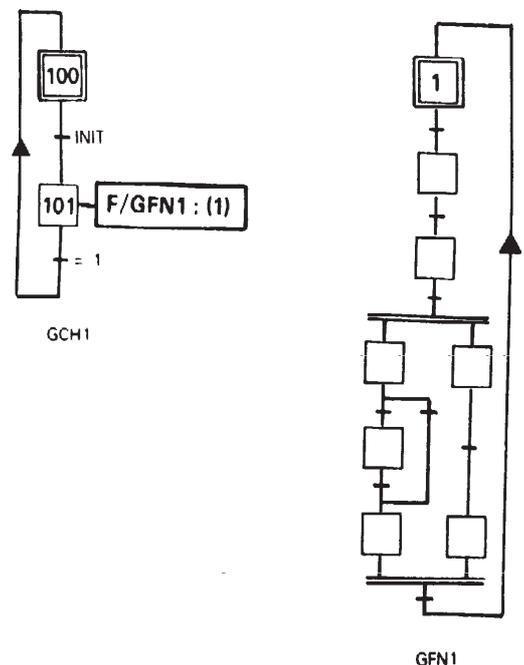
## 6.4 Description hiérarchisée par GRAFCET

A la représentation hiérarchisée de la partie commande peut être associée une description par GRAFCET comportant deux graphes séparés notés GCH ( GRAFCET de commande hiérarchisé ) associé à PCH et GFN ( GRAFCET de fonctionnement normal ) associé à PCL.

- Le comportement de PCL vis à vis de la partie opérative ( y compris le pupitre ) est décrit par le graphe de fonctionnement normal GFN.
- Le graphe de conduite hiérarchisé GCH décrit le comportement de la PCH vis à vis de sa partie opérative POH, constitué de l'ensemble PO (dont le pupitre) + PCL.

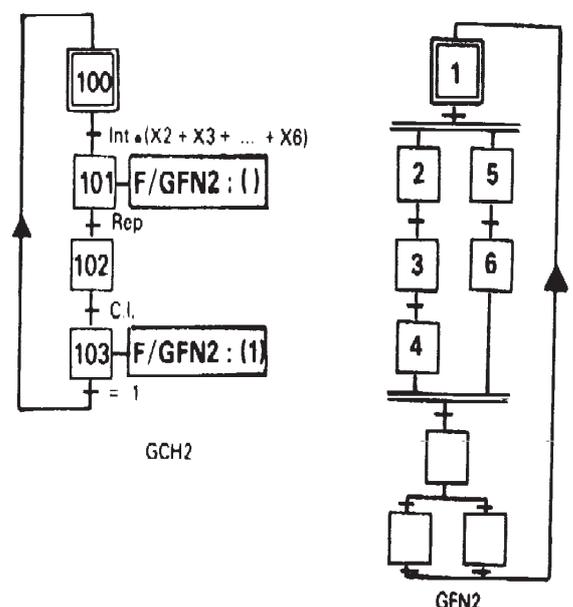
### Exemple 1

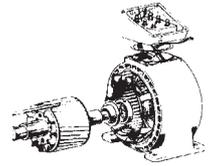
Lors de l'appui sur le B.P. INIT, on force l'étape 1 de GFN1 à 1 et toutes les autres étapes à 0. C'est un ordre ponctuel ( transition =1) qui permet de poursuivre l'évolution de GFN1 car tant que l'ordre de figeage est à 1 le GRAFCET GFN1 ne peut évoluer.



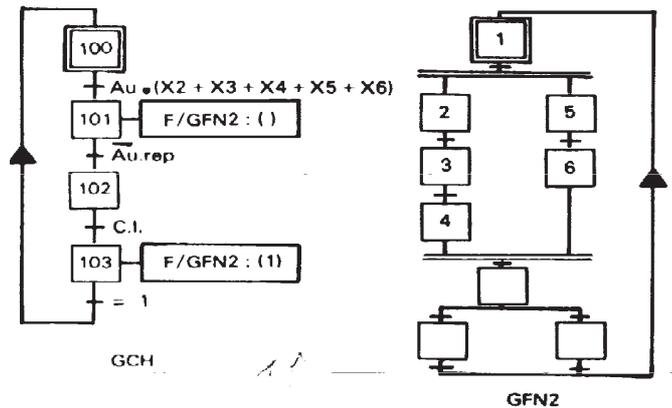
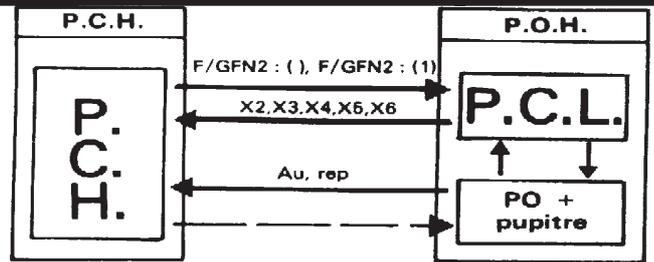
### Exemple 2

Lors de l'appui sur le B.P. INIT, on force toutes les étapes de GFN2 à 0 tant que le BP Rep n'est pas actionné (101) lorsque la transition Rep est franchie on active l'étape 102, GFN2 reste alors sans étapes actives. Lorsque les conditions initiales sont vraies on force l'étape 1 du GFN2, à 1. C'est un ordre ponctuel ( transition =1) qui permet de poursuivre l'évolution de GFN2 car tant que l'ordre de figeage est à 1 le GRAFCET GFN2 ne peut évoluer.

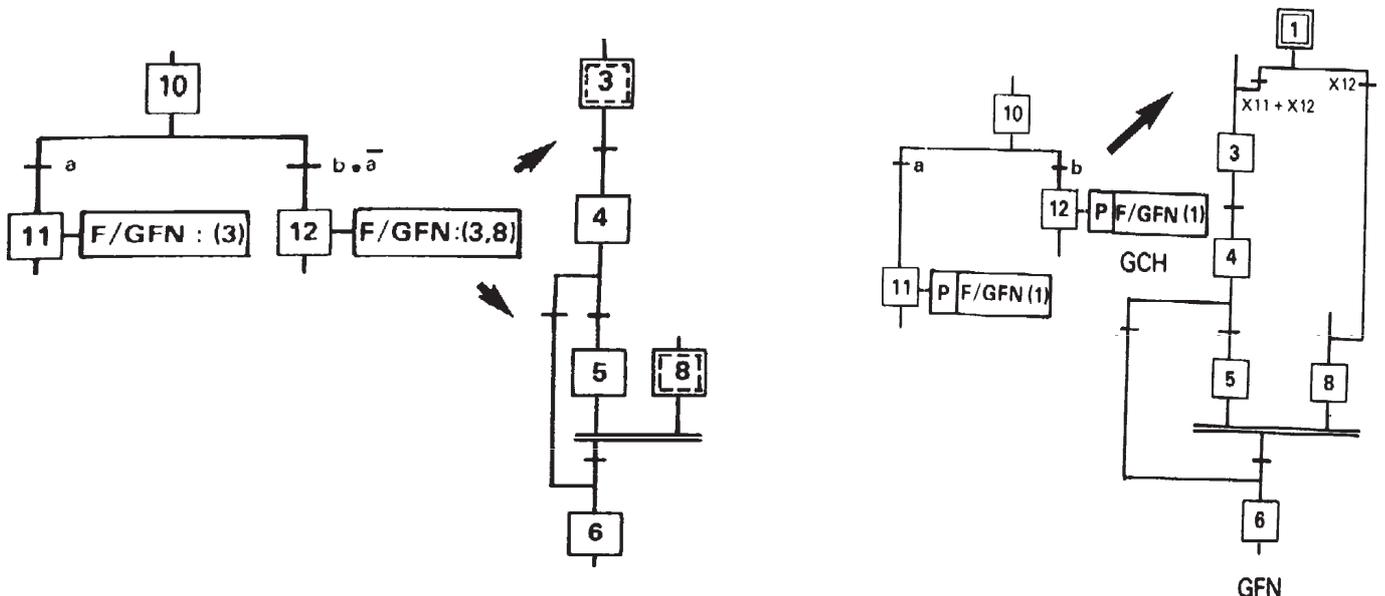




Ce principe de description allège très fortement le graphisme mais il faut préciser la structure de la partie commande. Le GRAFCET est insuffisant, il doit être complété par un schéma d'organisation de la partie commande.



Dans beaucoup d'application la situation à forcer sur une partie commande peut dépendre de conditions. C'est le cas pour l'exemple ci-dessous.

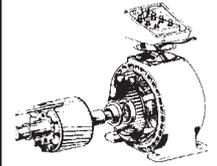


- Si la condition a est vraie on force la situation 3 graphe GFN décrivant PCL.

Ce qui correspond à un paramétrage de la situation initiale

On peut utiliser alors des pointillés et des commentaires pour préciser les étapes initialisables.

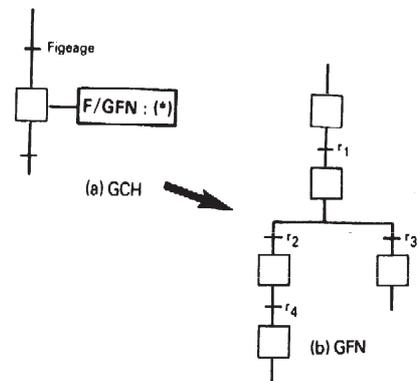
On peut également utiliser une représentation équivalente ne faisant intervenir qu'une situation de forçage. Pour cette représentation il s'est avéré nécessaire de rendre ponctuel les ordres de forçage, avec des ordres continus toute évolution de GFN aurait été impossible.



## 6.5 Un forçage particulier le figeage

Il est souvent nécessaire de spécifier le figeage d'une partie commande dans la situation en cours. Une telle spécification est souvent associée modes de marches et d'arrêts. Ce figeage dans la situation actuelle correspond à un forçage particulier pour lequel:

- Les étapes actives sont maintenues à 1
- Les étapes inactives sont maintenues à 0



## 6.6 Conclusion sur les ordres de forçage et de figeage

- Un ordre de forçage ne peut être émis que par une partie hiérarchique supérieure.
- La situation forcée peut être quelconque, y compris la situation vide () ou la situation actuelle (\*).
- L'ordre de forçage peut présenter tous les caractères des ordres, notamment l'ordre de forçage continu bloque toute évolution de la partie commande "esclave" tant qu'il est émis de la partie commande "maître".
- L'ordre de forçage agit sur l'ensemble de toutes les étapes du graphe de la partie commande esclave, TOUTES les étapes sont forcées, soit à l'activation, soit à la désactivation.
- Les étapes forcées peuvent être repérées par des pointillés

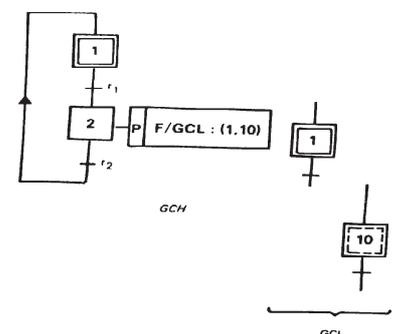
## 6.7 Principaux cas de forçages

### - Forçage d'une situation

Si l'étape (2) de CGH devient active, on force la situation (1,10) de GCL, l'ordre ponctuel, en rendant le forçage fugitif, permet à CGL de pouvoir aussitôt évoluer à partir de la situation 1,10.

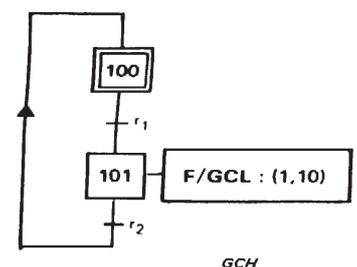
Attention le CGH doit aussi pouvoir continuer à évoluer!

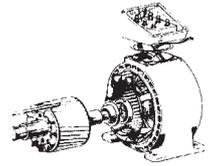
Plusieurs graphes peuvent avoir des numéros d'étape identique: il est alors impératif d'identifier le ou les graphes forcés.



### - Forçage et blocage d'une situation

La situation est forcée tant que l'étape 101 est active. L'ordre continu assure le maintien du forçage et donc un blocage de GCL dans la situation 1,10: GCL ne pourra évoluer que lorsque l'étape 101 de GCH sera désactivée.





Attention :  
Ce blocage peut survenir à la suite d'une erreur de conception

### - Figeage d'un graphe

Tant que l'étape 2 est active, CGL est figé dans la situation actuelle (quelle qu'elle soit). Il ne pourra reprendre son évolution normale qu'après la désactivation de l'étape 2

Désactivation d'un graphe

Le forçage à la situation vide de GCL provoque sa désactivation. GCL est désactivé, plus aucune transition n'est validée.

### - Désactivation d'un graphe

Le forçage à la situation vide de CGL provoque sa désactivation. CGL est inactivé, donc plus aucune transition n'est validée.

Cet ordre devra normalement être suivi, dans une étape ultérieure, par un ordre de forçage de situation de GCL.

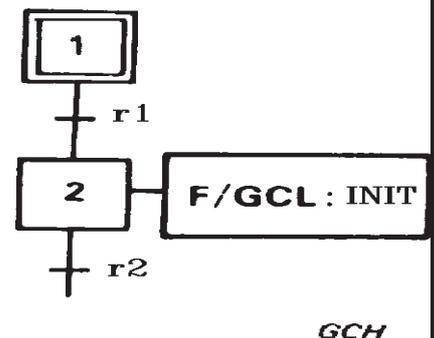
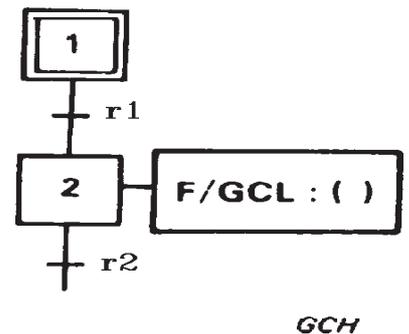
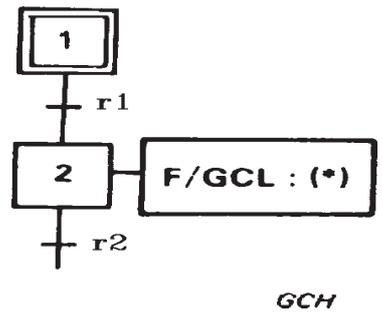
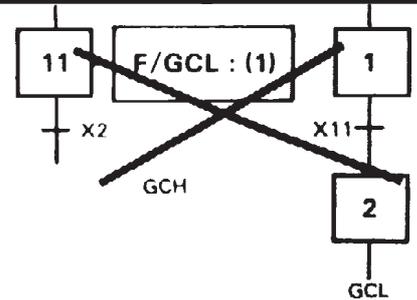
### - Initialisation d'un graphe

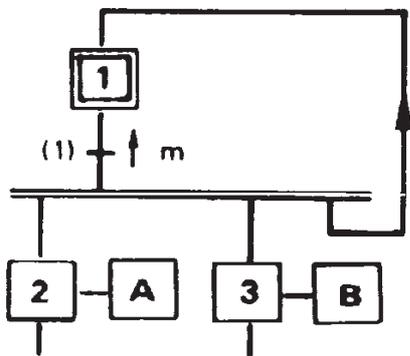
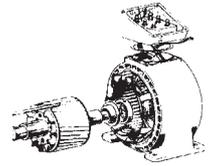
Le graphe CGL est forcé dans sa situation initiale, qu'il est donc inutile de décrire.

## 6.8 Transition source

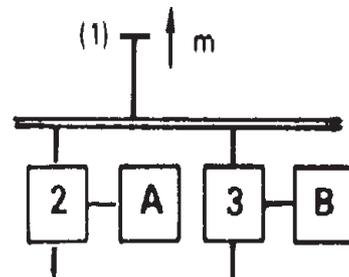
La transition source correspond à une simplification de représentation dans le cas particulier d'une transition toujours validée.

Pour alléger le graphisme et simplifier la description il est possible de ne pas représenter l'étape toujours active validant la transition. La transition est toujours validée, elle est donc franchie à chaque fois que la réceptivité associée à la transition est vraie.

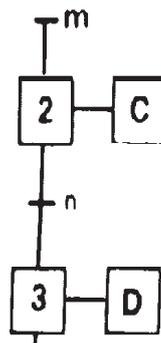




La transition (1) est toujours validée car l'étape 1 reste active lors de son franchissement de par la règle 5 du GRAFCET



Cette représentation doit être utilisée avec une extrême précaution. En particulier il est recommandé d'associer à ces transitions des réceptivités de type front, une réceptivité de type variable peut entraîner de gros problèmes.



**attention !**

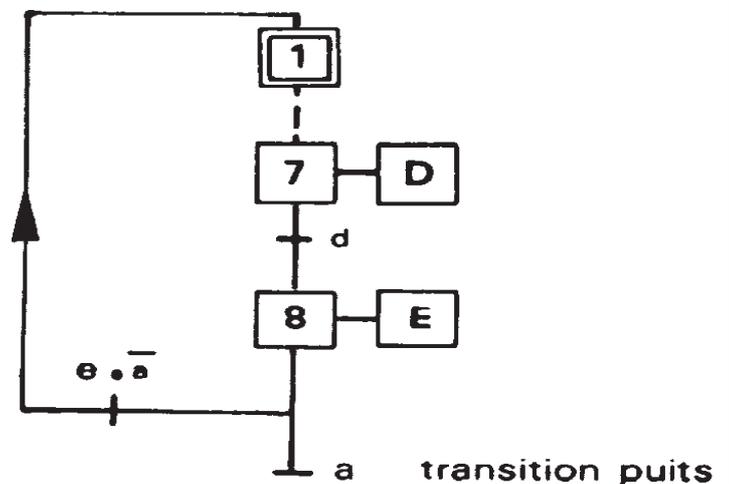
L'étape 2 reste activée tant que l'information m est vraie (règle 5) : on peut obtenir une succession de situations intempêtes telles (2) (2,3) (2,3,4) ...

## 6.9 Transition puits

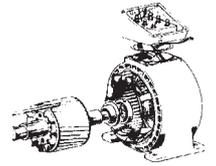
Une transition puits correspond à une transition non suivie d'étape. Ce type de transition exprime une rupture dans l'alternance étape - transition et peut être utilisée pour :

- Représenter la mise à l'état inactif d'une partie commande indépendante
- Pour désactiver une sous-séquence, ...

Il faut noter que pour cet exemple, une fois la transition puits franchie la partie commande n'est plus active ( situation vide). Le fonctionnement ne pourra reprendre qu'après le forçage d'une de ces étapes.



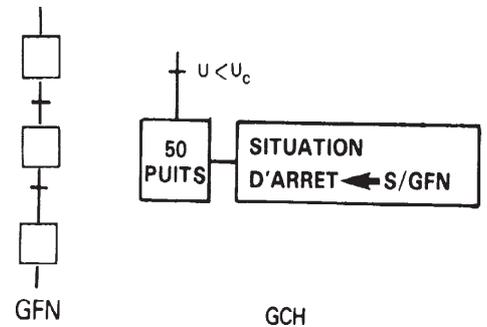
## 6.10 Etape puits



Ce type d'étape est utilisé pour la mémorisation de la situation en cours, à la détection d'une coupure d'énergie sur la PC ( tension < à une tension de seuil UC).

L'action associée à cette étape permet de mémoriser la situation en cours dans une mémoire non volatile identifiée par la variable SITUATION D'ARRET.

Notons que l'étape 50 ne restera active que pendant un temps déterminé par la descente du signal d'alimentation ( suffisant pour permettre l'action de mémorisation ) ce qui justifie l'utilisation d'une étape puits.



### 6.11 Reprise sur coupure d'énergie

Une possibilité pour décrire une reprise d'énergie après une coupure peut être décrite par la figure ci dessous.

Elle suppose l'existence d'une mémoire non volatile contenant l'état de la variable SITUATION D'ARRET.

Lors de la coupure d'énergie de la PC le modèle suppose que le concept de situation n'a plus aucune signification ( donc plus d'étape active).

La transition source de GCH permet de prendre en compte le retour en énergie. Si la situation d'arrêt est non vide on force GFN dans une situation déterminée sinon (cas de première initialisation) on force GFN dans la situation initiale 1.

