

initiation au

Grafcet

Mémento à l'usage des élèves de
**lycées professionnels
et technologiques**

S. Moreno

*Professeur certifié de Génie mécanique
spécialité «automatique et régulation»*

E. Peulot

*Professeur technique honoraire
agrégé de Génie mécanique
spécialité «automatique»*



Éditions Casteilla - 25, rue Monge - 75005 Paris

Sommaire

Introduction	7
Étape – situation – transition – réceptivité – action	8
Initiation à la lecture d'un GRAFCET	9
Les trois parties d'un système automatisé de production (SAP)	10
Les cinq règles d'évolution	11
Compléments sur les transitions	18
Compléments sur les réceptivités	19
Exemples d'évolutions types	21
Liaisons orientées	24
Structures de base du GRAFCET	26
Niveaux de description d'un automatisme	34
Actions associées à l'étape	39
Actions continues	41
Actions mémorisées	43
Structuration des GRAFCET partiels	48
Structuration par sous-programme	50
Structuration par encapsulation	53
Structuration par macro-étape	54
Structuration par forçage de GRAFCET partiel	57
Forçage d'un GRAFCET partiel dans la situation courante (figeage)	61
Modes de marche et d'arrêt	65
Applications particulières des règles d'évolution	70
Postulats sur les durées	75
Réalisation de la partie commande	77

Initiation à la programmation du GRAFCET	79
Programmation en langage GRAFCET graphique (PL7-2 Schneider)	79
Programmation en langage à contacts	80
Standard de la norme CEI 1131-3	82
Principaux symboles graphique du langage « <i>Ladder Diagram</i> » PL7 PRO	85
Implantation du GRAFCET sur mots en langage à contacts	86
Le GRAFCET et les séquenceurs	87
Séquenceur Parker Pneumatic	87
Séquenceur Joucomatic	90
Séquenceur Atlas Copco	91
Séquenceur Festo	93
Principaux opérateurs logiques combinatoires et séquentiels	94
Index	95

Avant-propos

Le GRAFCET créé dans les années 1975-1977 continue à faire ses preuves. Il constitue l'outil d'analyse par excellence des automatismes dits séquentiels.

Son enseignement, à titre d'initiation ou sous forme plus élaborée, est prévu dans la plupart des programmes des classes techniques depuis les lycées professionnels jusqu'aux écoles d'ingénieurs, en passant par les lycées technologiques, les sections de techniciens supérieurs et les instituts universitaires de technologie. Les formations permanentes réservées aux techniciens en activité et concernés par les automatismes lui consacrent également une place de choix.

Cet ouvrage se propose de fournir à l'élève de BEP, Bac Pro et Bac STI, de façon concise et simplifiée, les éléments à connaître pour analyser un automate puis décrire son fonctionnement à l'aide d'un GRAFCET.

Ce livret constitue à la fois une base et une bonne ouverture vers des ouvrages plus complets sur le GRAFCET et le GEMMA (des mêmes auteurs), ouvrages destinés aux étudiants et aux techniciens désirant se spécialiser en automatique.

Introduction

Le GRAFCET est un outil graphique de description des automatismes séquentiels indépendant des moyens de réalisation. Il s'agit donc d'un outil d'analyse (voir exemple page 9). Mais par le biais des logiciels de programmation pour API orientés GRAFCET, le GRAFCET lui-même est devenu un outil de synthèse, c'est-à-dire un outil de réalisation notamment sous sa forme internationale dite « Sequential Function Chart (SFC) ».

Signification du mot GRAFCET

Le mot GRAFCET est un acronyme, c'est-à-dire un nom formé avec les principales initiales de l'expression « GRAphe Fonctionnel de Commande Étape-Transition ».

Créateurs du GRAFCET

Les premiers travaux ont été conduits par une Commission d'études patronnée par l'AFCEC en 1975.

L'ADEPA a ensuite, en 1977, formalisé les concepts de base du GRAFCET en vue de sa normalisation.

Normes et documentations successives

- NF C 03-190 (1982) première norme relative au GRAFCET, rééditée en septembre 1995 sans modifications notables.
- UTE C 03-191 (1993), documentation introduisant la notion de forçage, de macro-étape, d'étapes et de transitions puits et sources, d'actions mémorisées par Set et Reset.
- CEI 61131-3 (1993) norme internationale validant les 5 langages de programmation d'API dont le SFC (Sequential Function Chart) inspiré du GRAFCET.
- CEI 60848 (2000) norme internationale introduisant la notion d'étape encapsulante initiale ou non et d'étape encapsulée, modifiant la notation et les symboles des actions mémorisées, des actions à l'activation et à la désactivation, de l'action de franchissement et les actions sur événement et simplifiant l'écriture des forçages.
- NF EN 60848 (août 2002) norme européenne traduction française de la norme CEI 60848.
- CEI 61131-3 concerne le langage SFC et conserve les notations relatives aux actions liées au temps (L, D, etc.).

ADEPA : Agence nationale pour le DEveloppement de la Production Automatisée.

AFCEC : Association Française pour la Cybernétique Économique et Technique devenue depuis l'Association Française des Sciences et Technologies de l'Information.

API : Automate Programmable Industriel.

CEI : Commission Électrotechnique Internationale.

EN : *European Norm*.

NF : Norme Française.

UTE : Union Technique de l'Électricité.

Remarque concernant la norme NF EN 60848 (voir pages 49 et 64)

Étape – Situation – Transition – Réceptivité – Action

Définitions élémentaires

Étape

Situation atteinte par le GRAFCET et qu'il conserve tant qu'aucun ordre de franchissement de la transition suivante ne lui est signifié. Une étape ne peut avoir que deux états : soit elle est active, soit elle est inactive. Une étape se représente par un carré (ou un rectangle) repéré par un numéro placé dans la partie haute du cadre.

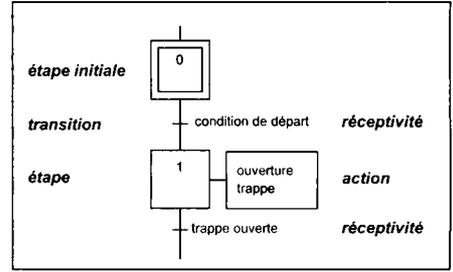


Figure 1

Symboles de base du GRAFCET

Situation

La situation est définie par l'ensemble des étapes actives à un instant donné.

Elle se note $S = \{n\}$ où n est l'étape active à l'instant considéré et $S = \{m,n,p\}$ si les étapes m, n, p sont actives simultanément à l'instant considéré.

Exemples :

$S = \{5\}$ signifie que dans cette situation seule l'étape 5 est active.

$S = \{4,8\}$ signifie que dans cette situation seules les étapes 4 et 8 sont actives.

Étape initiale

Étape particulière activée et la seule activée parmi toutes les étapes d'un GRAFCET au moment de son initialisation. Sans activation de l'étape initiale, activation qui autorise le franchissement de la transition suivante, l'évolution du GRACET est impossible.

L'étape initiale se représente en doublant le cadre de l'étape.

La situation correspondante est dite « situation initiale ».

Un GRAFCET peut comporter plusieurs étapes initiales.

Transition

État intermédiaire entre deux étapes successives. Une transition est franchissable ou elle ne l'est pas. Elle est franchissable et obligatoirement franchie si l'étape précédente est active (transition validée) et si la condition de transition (réceptivité) est vraie.

La transition est figurée par un tiret perpendiculaire au trait de liaison entre la ou les étapes amont et la ou les étapes aval.

Réceptivité

Condition de transition associée à la transition.

La réceptivité ne peut avoir que deux états : elle est soit vraie (état 1) soit fausse (état 0).

L'expression logique de la réceptivité s'écrit à droite de la barre de transition.

Action

Ordre à exécuter lorsque l'étape à laquelle il est associé est active. Le nom ou le symbole de l'action s'écrit à l'intérieur d'un rectangle plus ou moins important relié à l'étape par un tiret.

Plusieurs actions peuvent être associées à la même étape.

Initiation à la lecture d'un GRAFCET

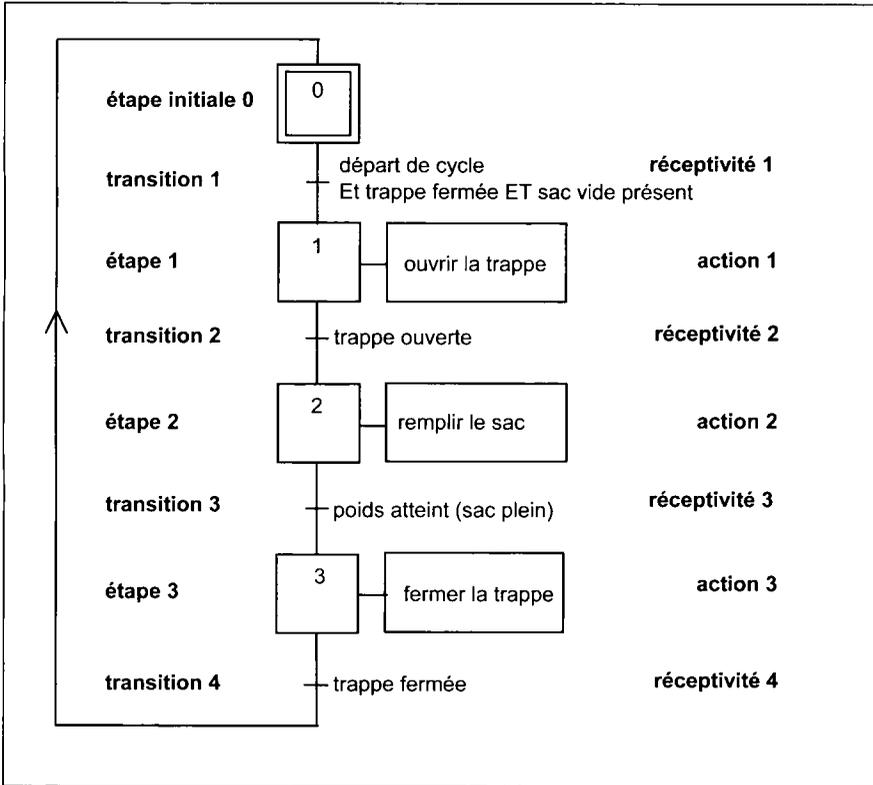


Figure 2
GRAFCET
d'apprentissage à
la lecture

Commentaire de la figure 2

La situation initiale $S = \{0\}$ est obtenue par activation de l'étape 0 et désactivation des autres étapes. Le GRAFCET est dit « initialisé ». La transition 1 est validée.

Si les conditions de départ sont toutes vérifiées, la réceptivité 1 devient vraie et la transition 1 est franchie.

La situation $S = \{1\}$ est obtenue : étape 1 activée, étape 0 désactivée.

L'ordre d'exécution de l'action 1 est émis : la trappe s'ouvre.

La transition 2 est également validée.

Si l'action 1 est terminée (trappe ouverte), la réceptivité 2 devient vraie et la transition 2 est franchie.

La situation $S = \{2\}$ est obtenue : étape 2 activée, étape 1 désactivée.

L'ordre d'exécution de l'action 2 est émis : le sac se remplit.

La transition 3 est validée.

Si l'action 2 est terminée (sac plein → poids atteint), la réceptivité 3 devient vraie et la transition 3 est franchie.

La situation $S = \{3\}$ est obtenue : étape 3 activée, étape 2 désactivée.

Et ainsi de suite jusqu'au retour en situation initiale et la réactivation de l'étape initiale 0.

Les trois parties d'un système automatisé de production (SAP)

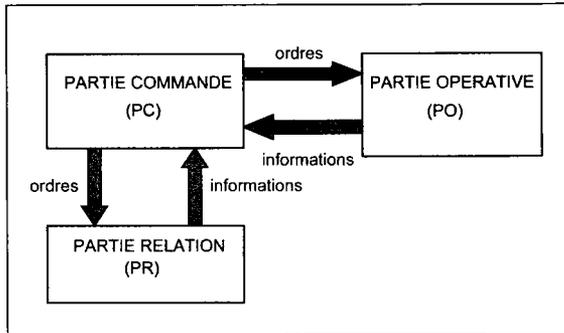


Figure 3
Les trois parties d'un SAP

Un SAP peut être décomposé en trois parties distinctes :

La partie opérative (PO) regroupant :

- les mécanismes (vis, engrenages, leviers, etc.),
- les effecteurs (vannes, pinces, tables tournantes, etc.),
- les actionneurs (vérins, moteurs, électro-aimants, résistances de chauffage, etc.),
- les préactionneurs (distributeurs, contacteurs, relais électromagnétiques, etc.),
- les capteurs-machines (fins de course, détecteurs inductifs, détecteurs photo-électriques, etc.).

La partie relation (PR) regroupant :

- les capteurs-opérateurs (boutons-poussoirs, « coup de poing » d'arrêt d'urgence, interrupteurs, commutateurs, etc.),
- les composants de signalisation (voyants, gyrophares, affichage numérique, Klaxons, sirènes, etc.),
- les composants de visualisation (moniteurs de contrôle, terminaux, etc.),
- les composants de communication (cartes et modems pour réseau interne ou externe).

La partie commande (PC) regroupant :

tous les constituants et les composants permettant le traitement des informations (signaux d'entrées) en provenance de la PO et de la PR. La PC après traitement émet des ordres (signaux de sorties) destinés aux préactionneurs de la PO et aux composants de signalisation, de visualisation et de communication de la PR.

• **Constituants**

Séquenceurs pneumatiques, automates programmables industriels (API), micro-ordinateurs, cartes à microprocesseur, calculateurs de commande numérique (CN), etc.

• **Composants**

Modules opérateurs logiques, relais électromagnétiques, interfaces pneumo-électriques, etc.

Les cinq règles d'évolution

Ces règles sont essentielles pour pouvoir lire et écrire un GRAFCET.

Elles sont une des composantes du « modèle GRAFET » définies par la norme NF C 03-190.

Règle 1 relative à la situation initiale du GRAFCET

La situation initiale du GRAFCET caractérise le comportement initial de la PC vis-à-vis de la PO et correspond à l'ensemble des étapes actives autorisant le début du fonctionnement et l'exécution du premier cycle.

Ces étapes particulières dites étapes initiales ont pour symbole un double cadre.

La situation initiale est atteinte :

- soit lors de la mise en énergie de la PC,
- soit naturellement en fin de cycle,
- soit à la suite d'une demande de retour volontaire dans l'état initial par un « forçage de la situation initiale ».

Il est sous-entendu que la mise en situation initiale entraîne la désactivation de toutes les étapes non initiales.

La situation initiale doit avoir un comportement passif vis-à-vis de la PO. Celle-ci doit rester dans une position de repos particulière choisie par le concepteur comme « position de référence ». En clair, cela revient à dire qu'aucune action externe non conditionnelle ne doit être associée aux étapes initiales.

Il est possible mais non conseillé d'y associer des ordres externes s'ils sont conditionnels.

Par contre, il n'est pas interdit de profiter de la mise en situation initiale pour émettre des ordres internes à la PC comme, par exemple, la remise à zéro (RAZ) des compteurs.

L'initialisation peut être définie comme le passage de la PC d'une situation vide $S = \{ \}$ où aucune étape n'est active à une situation où au moins une étape est active (étape initiale).

L'évolution à partir de la situation vide impose implicitement un forçage inconditionnel de situation par un ordre émis par une structure de contrôle hiérarchiquement supérieure et indépendante du GRAFCET.

Cet ordre peut être émis :

- soit par un opérateur humain (impulsion sur un bouton poussoir),
- soit par une autre partie commande (forçage de situation initiale),
- soit par une procédure particulière liée à la mise en énergie de la PC (cas des API).

Remarque :

La mise en énergie n'entraîne en aucun cas l'initialisation de la PC et c'est en exécutant une procédure appropriée et programmée qu'un API, par exemple, initialisera les GRAFCET qui y sont implantés.

Exemples 1 et 2

Commentaire de l'exemple 1

La situation initiale correspond à la situation $S = \{0\}$.

Si l'étape 0 est active, la transition $0 \rightarrow 1$ est validée.

Si les actions A et B sont terminées (PO en référence) et si le signal de départ de cycle est donné, l'expression logique de la réceptivité correspondante est vraie.

La transition $0 \rightarrow 1$ est franchie et la PC décrite par ce GRAFCET commence son évolution.

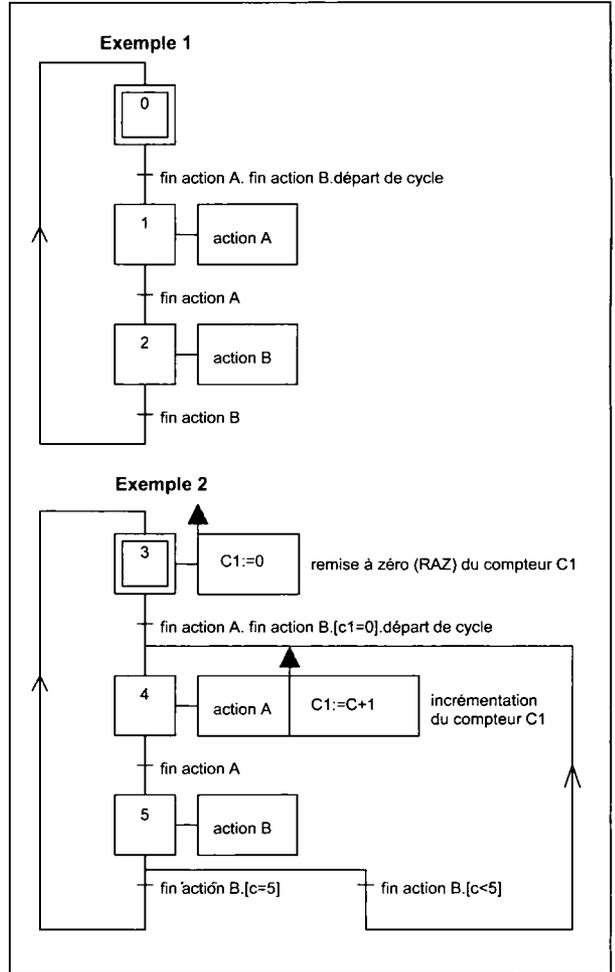


Figure 4

Exemples 1 et 2 d'initialisation d'un GRAFCET

Commentaire de l'exemple 2

La situation initiale correspond à la situation $S = \{3\}$.

Si l'étape 3 est active, la transition $3 \rightarrow 4$ est validée.

Le compteur C1 est mis à zéro (action interne à la PC).

Si les actions A et B sont terminées (PO en référence), le compteur C1 mis à zéro et l'ordre de départ de cycle donné, l'expression logique de la réceptivité correspondante est vraie.

La transition $3 \rightarrow 4$ est franchie et la PC décrite par ce GRAFCET commence son évolution.

L'action A et l'action B sont exécutées 5 fois. Ensuite le GRAFCET est réinitialisé.

Exemple 3

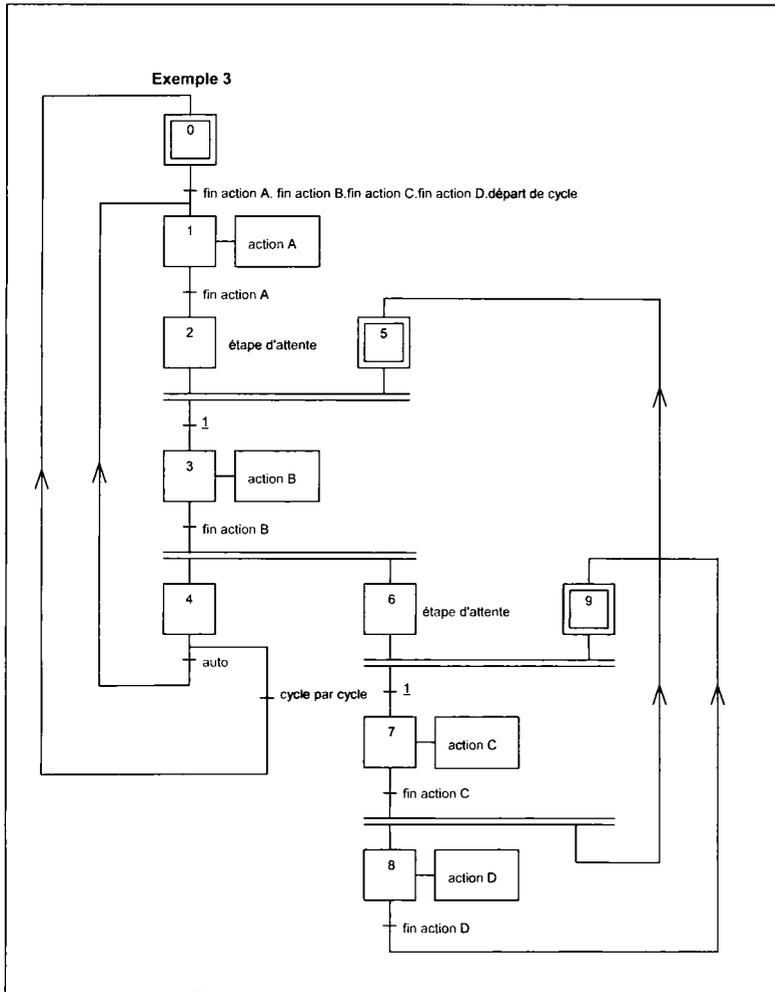


Figure 5
Exemple 3 d'illustration de l'initialisation d'un GRAFCET

Commentaire de l'exemple 3

La situation initiale correspond à la situation $S = \{0,5,9\}$.

Sans l'étape initiale 5, l'évolution du GRAFCET reste bloquée, lors du premier cycle, dans la situation $S = \{2\}$.

En effet, la transition $2,5 \rightarrow 3$ n'est validée que si les étapes 2 et 5 sont actives.

La même remarque est à faire au sujet de l'étape initiale 9 et de la transition $6,9 \rightarrow 7$.

Règle 2 : relative au franchissement d'une transition

L'évolution de la situation du GRAFCET correspondant au franchissement d'une transition ne peut se produire que :

- si cette transition est validée,
- et si la condition de transition ou réceptivité associée est vraie.

Cette condition d'évolution est nommée **fonction transition**.

Lorsque la fonction transition est vraie, la transition devient franchissable et est obligatoirement franchie.

Remarque :

Une transition est validée lorsque toutes les étapes immédiatement précédentes sont actives (voir figure 8).

Règle 3 : relative à l'évolution de la situation de la PC

Le franchissement d'une transition entraîne simultanément l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes (évolution synchrone) (voir figures 7 et 8).

Règle 4 : relative aux évolutions simultanées

Plusieurs transitions simultanément franchissables sont simultanément franchies.

Cette règle autorise la décomposition d'un GRAFCET unique en plusieurs GRAFCET coordonnés entre eux en faisant intervenir dans les conditions de transition les états actifs des étapes (fig. 6).

La notation de l'état actif des étapes est X_i où i est le numéro-repère de l'étape considérée.

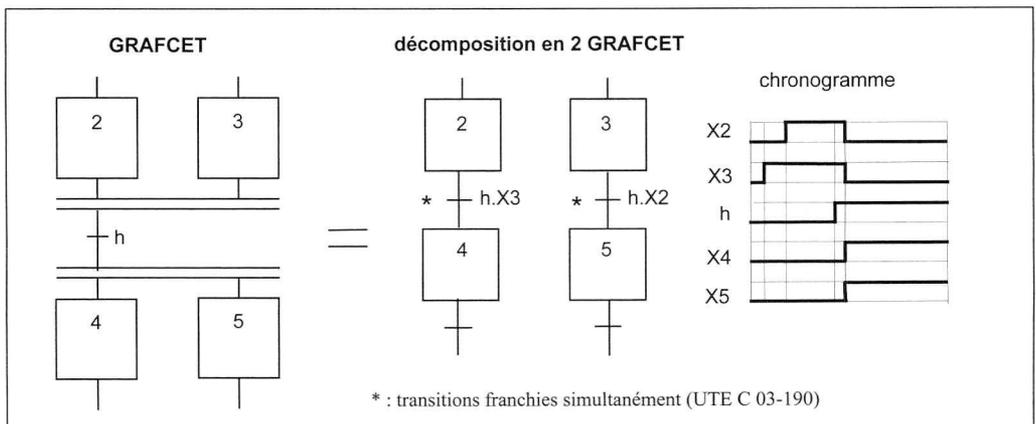


Figure 6

Illustration de la règle d'évolution 4

Exemple 1 des règles d'évolution 2 et 3

GRAF CET	état logique de la réceptivité $r(4 \rightarrow 5)$	commentaires
	0 ou 1	transition (1) non validée car l'étape 4 est inactive
	0	transition (1) validée car l'étape 4 est active mais non franchissable car la réceptivité n'est pas vraie
	1	transition (1) franchissable et obligatoirement franchie car elle est validée et la réceptivité est vraie
	0 ou 1	transition (1) franchie avec, simultanément, activation de l'étape 5 et désactivation de l'étape 4 transition (2) maintenant validée

Figure 7
Exemple 1 des règles d'évolution 2 et 3

Remarque :

Le point placé en bas du cadre d'une étape indique que celle-ci est active au moment considéré.

Exemple 2 des règles d'évolution 2 et 3

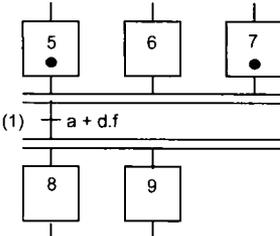
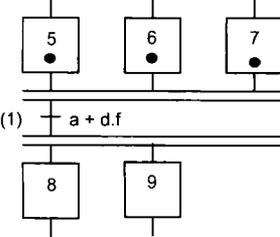
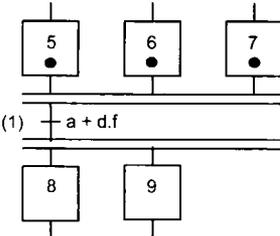
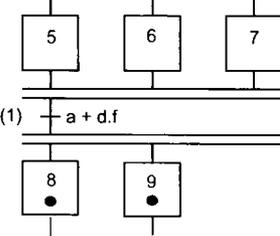
GRAF CET	état logique de la réceptivité $r(5,6,7 \rightarrow 8,9)$	commentaires
	0 ou 1	transition (1) non validée car l'étape 6 est encore inactive
	0	transition (1) validée mais non franchissable car la réceptivité n'est pas vraie
	1	transition (1) franchissable et obligatoirement franchie car elle est validée et la réceptivité est vraie
	0 ou 1	transition (1) franchie avec, simultanément, activation des étapes 8 et 9 et désactivation des étapes 5, 6 et 7

Figure 8
Exemple 2 des règles d'évolution 2 et 3

Règle 5 : relative à l'activation et à la désactivation simultanée d'une même étape.

Si au cours d'une évolution, une même étape se trouve être à la fois activée et désactivée, elle reste activée.

En clair, cela revient à dire que l'activation est prioritaire sur la désactivation en cas d'application simultanée de ces ordres à la même étape.

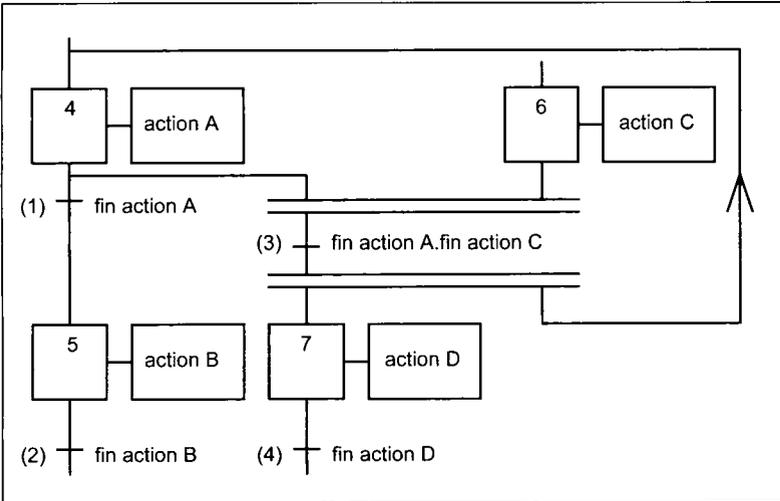


Figure 9

Exemple illustrant la règle d'évolution 5

Commentaire

Le GRAFCET se trouvant dans la situation $S = \{4,6\}$, les actions A et C sont en cours d'exécution.

En fin d'exécution de A et C, les transitions (1) et (3) sont franchies.

Le GRAFCET occupe la situation $S = \{5,7,4\}$.

En effet l'étape 4 désactivée par le franchissement de la transition (1) est également activée par le franchissement de la transition (3).

D'après la règle d'évolution 5, l'étape 4 à la fois désactivée et activée reste activée.

D'autres exemples d'application de cette règle 5 sont vus aux pages 70 à 74 traitant des cas particuliers d'application des règles d'évolution.

Compléments sur les transitions

La transition correspond à une possibilité d'évolution entre une ou plusieurs étapes.

Cette évolution se traduit par le franchissement de la transition.

Le franchissement de la transition provoque un changement de situation du GRAFCET donc de la PC.

Une transition est soit validée soit non validée.

La transition est validée lorsque toutes les étapes immédiatement précédentes sont actives.

La transition est franchie et obligatoirement franchie lorsqu'elle est validée et que la condition de transition ou réceptivité associée est vraie (à l'état 1).

Les conditions nécessaires au franchissement et les conséquences du franchissement sont définies par les règles d'évolution 2, 3 et 4 du modèle GRAFCET normalisé.

La transition entre étapes se représente par un tiret perpendiculaire à la ligne de liaison entre les étapes amont et aval à cette transition.

Pour faciliter la discussion du GRAFCET, les transitions peuvent être repérées, entre parenthèses, à gauche du tiret (UTE C 03-191).

Dans le même ordre d'idée, les étapes actives à un moment donné peuvent être marquées par un point (voir figures 7 et 8).

Une notation plus significative peut être utilisée dans le texte pour repérer les transitions.

Application aux exemples de la figure 9 :

- la transition entre l'étape 1 et l'étape 2 sera notée $t(1 \rightarrow 2)$,
- la transition entre les étapes 3 et 4,5 sera notée $t(3 \rightarrow 4,5)$,
- la transition entre les étapes 6,7 et 8 sera notée $t(6,7 \rightarrow 8)$,
- la transition entre les étapes 9,10 et 11,12 sera notée $t(9,10 \rightarrow 11,12)$.

La condition de franchissement d'une transition est appelée « **fonction transition** ». Elle a pour expression logique, par exemple, pour la transition entre l'étape 1 et l'étape 2 (fig.10) :

$$ft(1 \rightarrow 2) = X1.(r(1 \rightarrow 2))$$

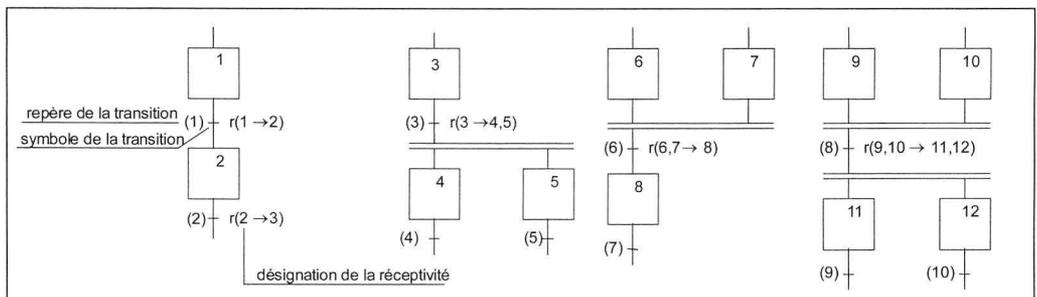


Figure 10

Exemples de transitions et de réceptivités avec leurs notations

Compléments sur les réceptivités

À chaque transition est associée une proposition logique appelée **réceptivité** ou condition de transition.

Les conditions de transitions peuvent être représentées soit par des indications littérales, soit par des expressions logiques soit par des symboles graphiques (schéma à contacts, logigramme, etc.).

La condition de transition peut être vraie (état 1) ou fausse (état 0).

Parmi toutes les informations disponibles à un instant donné la condition de transition regroupe dans une expression logique uniquement celles qui sont nécessaires au franchissement de la transition c'est-à-dire celles auxquelles la partie commande est réceptive dans la situation considérée.

La condition de transition regroupe des informations d'origines diverses émises :

- **par l'opérateur à partir du pupitre de commande (PR) où sont disposés :**
 - des boutons-poussoirs,
 - des interrupteurs,
 - des sélecteurs de mode de marche,
 - un bouton « coup de poing » d'arrêt d'urgence,
 - des claviers de terminaux, etc.
- **par la machine ou par l'installation (PO) où sont positionnés :**
 - des capteurs fin de course (interrupteurs de position),
 - des détecteurs de présence (détecteurs de proximité),
 - des détecteurs de phénomènes physiques (pression, température, etc.).
- **par des opérateurs internes ou externes à la PC tels que :**
 - temporisateurs émettant un signal binaire (0 ou 1) noté t /étape de lancement/durée,
 - temporisateurs avec test sur la durée écoulée (valeur booléenne du prédicat notée $[t > 10^\circ\text{C}]$),
 - compteurs avec test du contenu (valeur booléenne du prédicat notée $[ci = n]$ ou $[ci < n]$ ou $[ci \geq n]$ où n est la valeur de la présélection de leur contenu),
 - décompteurs avec test du contenu (valeur booléenne du prédicat notée $[di = 0]$ ou $[d > 0]$ où 0 est la valeur du contenu du décompteur lors de son retour à zéro),
 - tests divers (valeurs booléennes des prédicats).
- **par la PC fournissant des informations sur sa propre situation :**
 - état actif ou inactif des étapes (notation X_i),
 - variables images d'opérateurs internes (bits internes),
 - résultats de calculs internes (bits internes),
 - résultats de conversions analogiques-numériques ou vice versa (mot ou nombre),
 - par une condition toujours vraie (condition de transition toujours égale à 1 notée $\underline{1}$, anciennement notée =1).

La nature de l'information est à préciser.

L'évolution peut en effet dépendre :

- soit du niveau maintenu du signal : état 1 ou 0,
- soit du changement d'état du signal :
 - passage de l'état 0 vers 1 (front montant),
 - passage de l'état 1 vers l'état 0 (front descendant).

Dans le cas du niveau maintenu, on dit que l'évolution est liée à **une condition**.

Dans le cas du changement d'état, on dit que l'évolution est liée à **un événement**.

La condition de transition est inscrite sous forme littérale ou sous forme symbolique, de préférence à droite du symbole de la transition.

Notations associées aux conditions

Les relations logiques intervenant dans les réceptivités utilisent les symboles de base de l'algèbre binaire représentatifs des opérateurs NON, ET et OU :

- opérateur logique NON ou Complémentation noté « $\bar{\quad}$ » ou « / » (barre inclinée ou slash des claviers) pour faciliter la reproduction typographique et l'affichage sur les écrans vidéo des consoles d'API ou des micro-ordinateurs,
- opérateur logique ET noté « . »,
- opérateur logique OU noté « + ».

Exemples

1/ : $a + \bar{c}.d$ ou $a + /c.d$ 2/ : $a.d + \bar{b}.c$ ou $a.d + /b.c$

3/ : $(a + \bar{d})(c + \bar{b}.e)$ ou $(a + /d).(c + /b.e)$

Notations associées aux événements

front montant : \uparrow , front descendant : \downarrow

Exemples

$\uparrow a$ = front montant de la variable a, $\downarrow a$ = front descendant de la variable a.

Représentation graphique des fronts montants et descendants

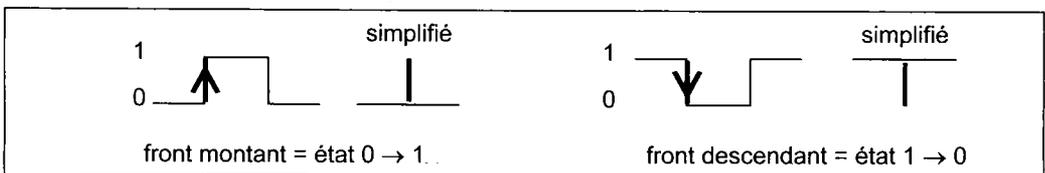


Figure 11

Représentation graphique des fronts montants et descendants

Remarque :

Une évolution peut dépendre également d'une combinaison de conditions et d'événements.

Pour la discussion, la réceptivité peut être notée $r(i \rightarrow j)$ où i et j sont respectivement les étapes amont et aval (voir fig.9) ou être repérée par le numéro de la transition écrit entre parenthèses (UTE C 03-191) (voir fig. 9).

À titre d'exemple la notation $r(9,10 \rightarrow 11,12)$ quoique plus lourde est toutefois plus explicite que $r(8)$.

Exemples d'évolutions types

Exemple 1 : évolution liée à des conditions

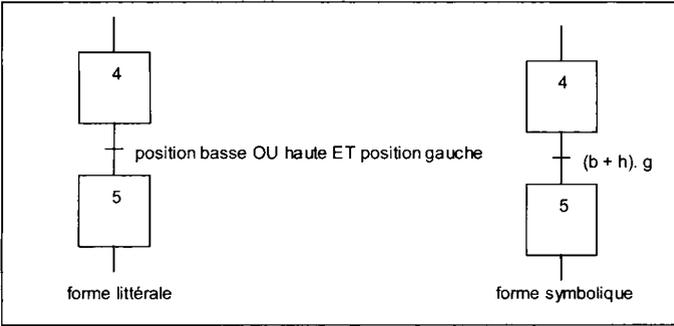


Figure 12
Évolutions liées à une condition.

Commentaire

La transition $4 \rightarrow 5$ est franchie si l'étape 4 est active ET si la variable b OU la variable h ET la variable g sont à l'état 1.

La fonction transition correspondante est :

$$ft(4 \rightarrow 5) = X4.(b + h).g$$

Exemple 2 : évolution liée à des événements (avec durée d'évolution non nulle)

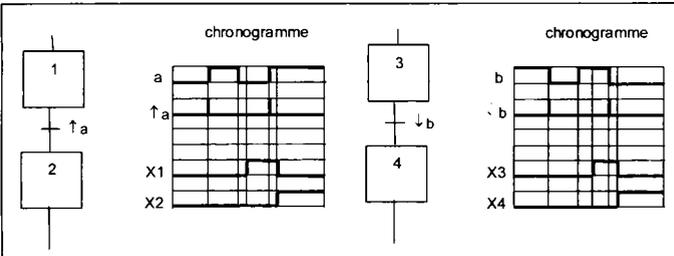


Figure 13
Évolutions liées à des événements

Commentaire

La transition $1 \rightarrow 2$ est franchie si l'étape 1 est active ET si la variable a évolue de l'état 0 vers l'état 1. La fonction transition correspondante est :

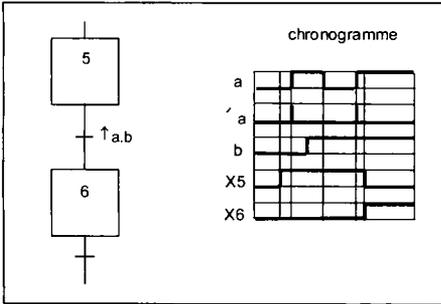
$$ft(1 \rightarrow 2) = X1.\uparrow a.$$

La transition $3 \rightarrow 4$ est franchie si l'étape 3 est active ET si la variable b évolue de l'état 1 vers l'état 0. La fonction transition correspondante est :

$$ft(3 \rightarrow 4) = X3.\downarrow b.$$

Noter que les chronogrammes des exemples prennent en compte le postulat sur la durée de franchissement d'une transition où celle-ci est considérée comme très petite mais non nulle.

Exemple 3 : évolution liée à une combinaison de conditions et d'événements



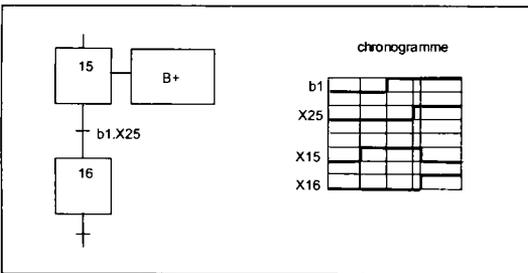
Commentaire

La transition 5 → 6 est franchie si l'étape 5 est active ET si la variable a évolue de 0 vers 1 ET si la variable b est à l'état 1. La fonction transition correspondante est :
 $ft(5 \rightarrow 6) = X5.\uparrow a.b.$

Figure 14

Évolution liée à une combinaison de conditions et d'événements

Exemple 4 : évolution liée à une condition interne



Commentaire

La transition 15 → 16 est franchie si l'étape 15 est active ET si b1 est à l'état 1 ET si l'étape 25 d'un autre GRAFCET est active. La fonction transition correspondante est :
 $ft(15 \rightarrow 16) = X15.b1.X25$

Figure 15

Évolution liée à une condition interne

Exemple 5 : évolution liée au temps

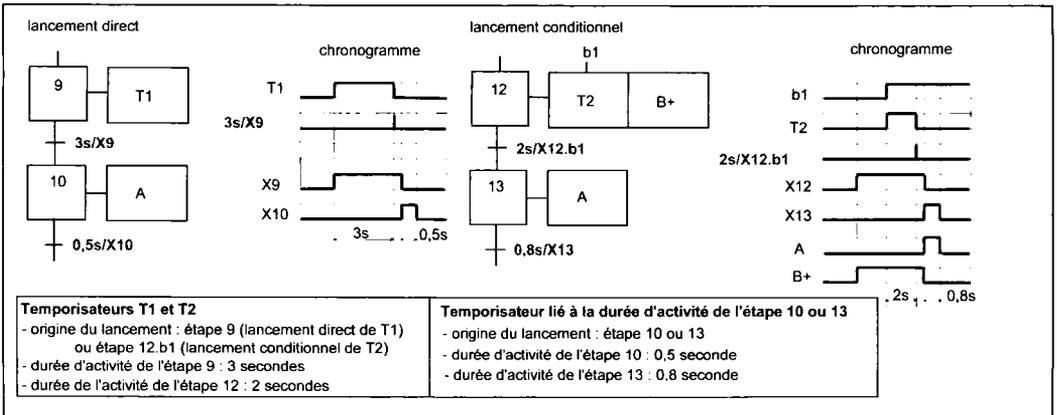


Figure 16

Évolutions liées au temps (voir remarque page 42)

Commentaire

Fonctions transitions : $ft(9 \rightarrow 10) = 3s/X9$; $ft(10 \rightarrow 11) = 0,5s/X10$;

$ft(12 \rightarrow 13) = 2s/X12.b1$; $ft(13 \rightarrow 14) = 0,8s/X13$

Les temporisateurs T1 et T2 peuvent être ou non symbolisés dans le GRAFCET suivant qu'ils font appel ou non à des fonctions opératives de commande (FOC), une minuterie par exemple.

Exemple 6 : évolution liée à un comptage

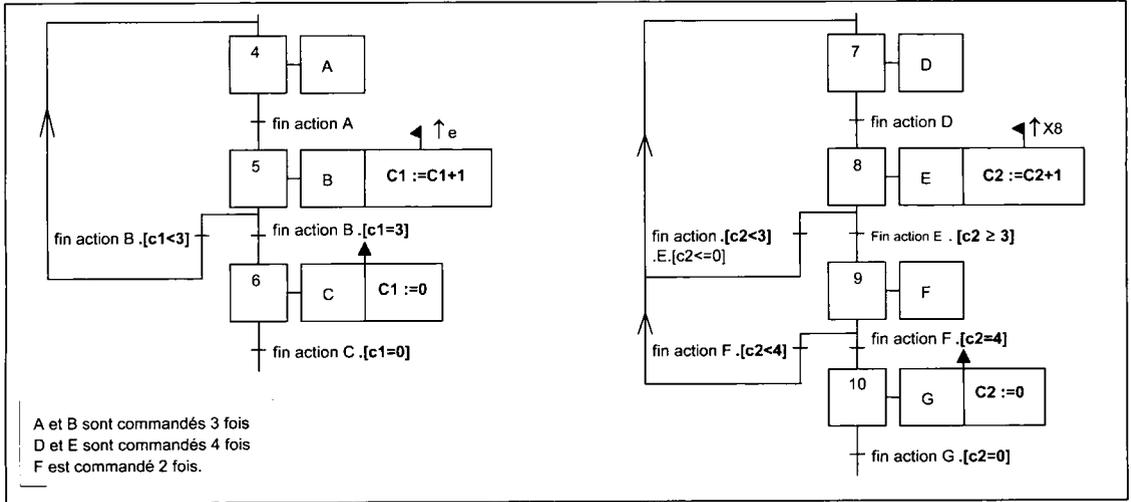


Figure 17

Évolution liée à un comptage

Commentaire

Fonctions transitions :

$ft(4 \rightarrow 5) = X4.fin\ action\ A$; $ft(5 \rightarrow 6) = X5.fin\ action\ B.[c1 = 3]$

$ft(5 \rightarrow 4) = X5.fin\ action\ B.(c1 < 3)$; $ft(6 \rightarrow 7) = X6.fin\ action\ C.[c1 = 0]$

$ft(7 \rightarrow 8) = X7.fin\ action\ D$; $ft(8 \rightarrow 9) = X8.fin\ action\ E.[c2 \geq 3]$

$ft(8 \rightarrow 7) = X8.fin\ action\ E.(c2 < 3)$; $ft(9 \rightarrow 7) = X9.fin\ action\ F.[c2 < 4]$

$ft(9 \rightarrow 10) = X9.fin\ action\ F.(c2 = 4)$; $ft(10 \rightarrow 11) = X10.fin\ action\ G.[c2 = 0]$

Exemple 7 : évolution liée à une condition de transition toujours vraie

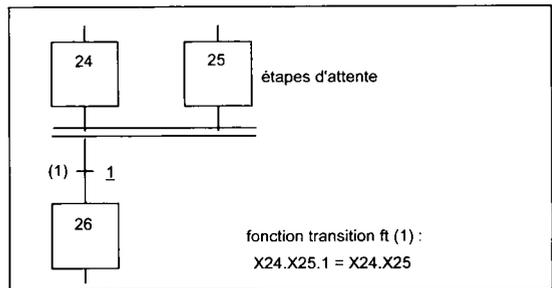
Une condition de transition toujours vraie se note « 1 » (1 souligné).

(évolution dite « toujours fugace »)

Dans la norme précédente, elle se notait « =1 »

Figure 18

Évolution liée à une condition de transition toujours vraie



Remarque :

La situation $S = (24,25)$ est une situation instable. En effet dès que ces deux étapes sont actives, la transition $24,25 \rightarrow 26$ est obligatoirement franchie et franchie instantanément. L'évolution est dite toujours fugace (NF EN 60848).

Pour en savoir plus à propos de la stabilité et de l'instabilité du GRAFCET, voir le paragraphe, page 75, consacré à ce sujet.

Liaisons orientées

Les liaisons orientées relient les étapes aux transitions et les transitions aux étapes. Elles indiquent les voies suivant lesquelles se font les évolutions.

Elles se représentent par des lignes verticales et horizontales, les lignes obliques étant impérativement déconseillées.

Par convention, les évolutions s'effectuent toujours dans le sens du haut vers le bas.

Des flèches doivent être utilisées pour marquer l'orientation :

- lorsque la convention n'est pas respectée : sens du bas vers le haut,
- lorsqu'on veut éviter des erreurs d'interprétation sur le sens exact des évolutions le long des lignes horizontales.

Le tracé des liaisons ne doit donner lieu à aucune ambiguïté au niveau de la lecture (fig. 19).

Le point n'étant pas admis pour marquer une connexion, il faut :

- décaler les traits de liaisons verticaux aboutissant à un trait horizontal et vice versa pour montrer qu'il y a connexion,
- tracer des liaisons en croix pour montrer qu'il n'y a pas de connexion.

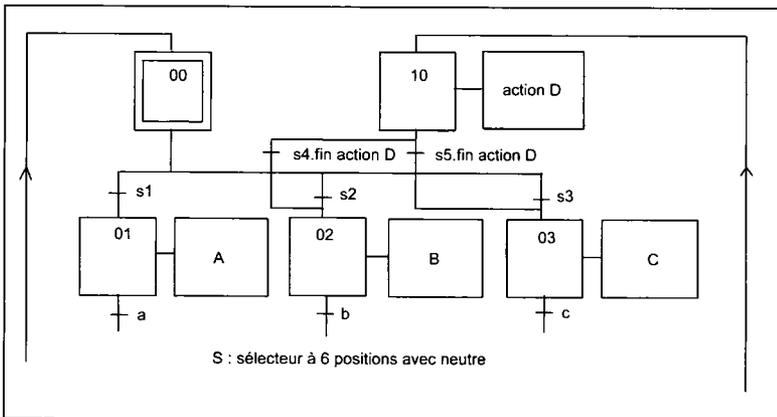


Figure 19

Tracé des liaisons avec et sans connexions

Commentaire

Compte tenu des conventions d'écriture précisées ci-dessus :

- à partir de l'étape 0 active,
 - si s1 = 1, évolution vers l'étape 1
 - si s2 = 1, évolution vers l'étape 2
 - si s3 = 1, évolution vers l'étape 3
- à partir de l'étape 10 active :
 - si s4.fin action D, évolution vers l'étape 2
 - si s5.fin action D, évolution vers l'étape 3.

Le double trait horizontal (ISO 5807 et NF Z 67-010) sert à regrouper plusieurs liaisons liées à la même transition, cette transition conditionnant l'évolution entre deux situations mettant en jeu plus de deux étapes (fig. 20).

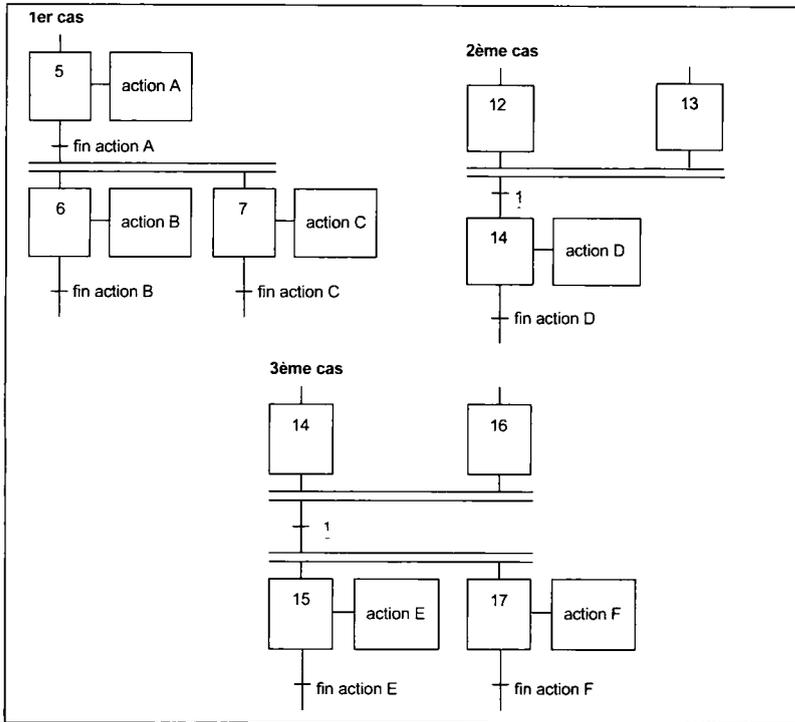


Figure 20
Représentation des liaisons dans le cas d'une transition liée à plus de 2 étapes

Les trois cas de la figure 20 :

- 1^{er} cas : transition d'une étape vers plusieurs étapes,
- 2^e cas : transition de plusieurs étapes vers une seule étape,
- 3^e cas : transition de plusieurs étapes vers plusieurs étapes.

Commentaire

Fonctions transitions :

- 1^{er} cas : $ft(5 \rightarrow 6,7) = X5.fin\ action\ A$
- 2^e cas : $ft(12,13 \rightarrow 14) = X12.X13$
- 3^e cas : $ft(14,16 \rightarrow 15,17) = X14.X16$

À titre mnémotechnique, on peut dire :

- qu'à des traits simples horizontaux correspondent des divergences et des convergences en OU exclusif ou inclusif suivant les conditions de transitions associées aux transitions,
- qu'à des traits doubles horizontaux correspondent des divergences et des convergences en ET.

Remarque :

Les étapes 12, 13 et 14,16 sont des étapes d'attente prévues en cas de fin non simultanée des séquences précédentes.

Structures de base du GRAFCET

Il s'agit de l'organisation des séquences à l'intérieur du GRAFCET.

Séquence unique

La séquence unique est composée d'une suite d'étapes alignées verticalement sans divergence ni convergence.

Chaque étape n'est suivie que par une seule transition et chaque transition n'est suivie que par une seule étape.

La séquence unique peut correspondre soit à une branche de GRAFCET soit à un GRAFCET complet.

La séquence est dite active si au moins une étape est active. Elle est dite inactive si aucune étape n'est active. Dans ce cas elle correspond à une situation vide notée $S = \{ \}$.

Un GRAFCET comportant une ou plusieurs séquences uniques et dont une seule étape peut être active à la fois est dit « GRAFCET à étape active unique ».

Commentaire

Le GRAFCET à séquence unique de la figure 21, compte tenu de ses réceptivités, est un GRAFCET à étape active unique, c'est-à-dire que seules les situations suivantes sont possibles : $S = \{0\}$ ou $\{1\}$ ou $\{2\}$ ou $\{3\}$.

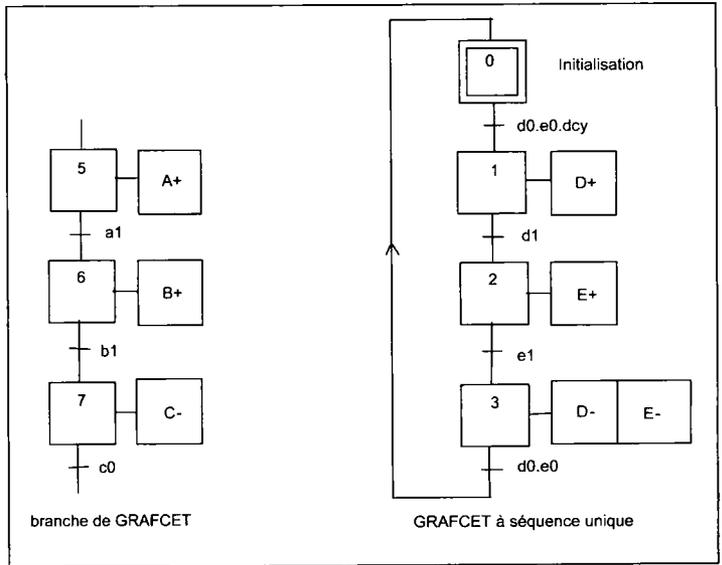


Figure 21
Exemples de séquences

La partition du GRAFCET envisagée par la NF EN 60848 distingue le GRAFCET connexe (structure où il existe toujours une suite d'alternances d'étapes et de transitions) et le GRAFCET partiel constitué d'un ou plusieurs GRAFCET connexes.

Notations relatives aux GRAFCET partiels :

G^* : GRAFCET partiel de repère *.

XG^* : variable caractérisant l'état actif ou inactif du GRAFCET partiel de repère *.

$G\#\{\dots,\dots\}$: situation d'un GRAFCET partiel de repère #.

$G\#\{*\}$: situation courante du GRAFCET partiel de repère #.

$G\#\{ \}$: situation vide du GRAFCET partiel de repère #.

$G\#\{INIT\}$: situation initiale du GRAFCET partiel de repère #.

Séquences simultanées

Si le franchissement d'une transition conduit à activer plusieurs étapes en même temps, ces étapes sont à l'origine de séquences dont les évolutions seront à la fois simultanées et indépendantes.

Ce type de structure est appelé « **parallélisme structural** ».

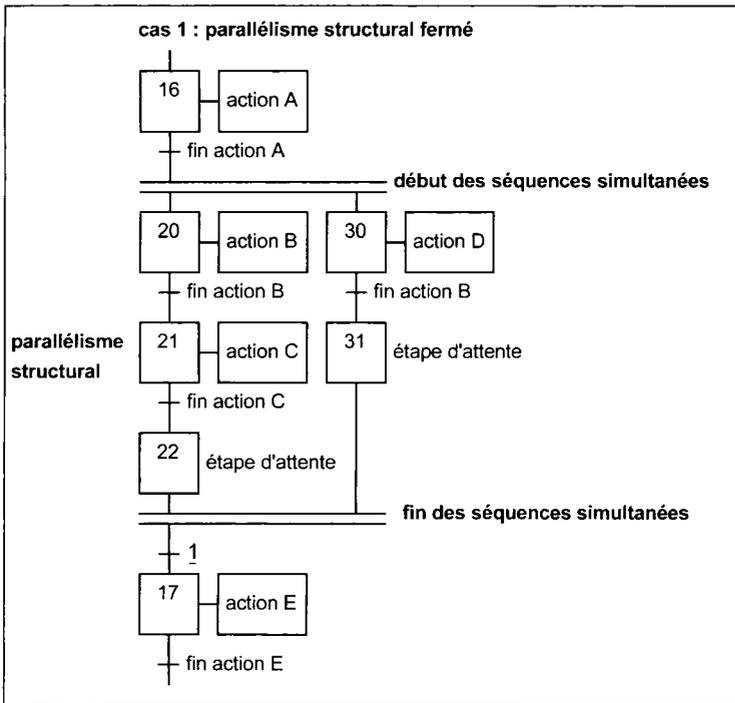
Le début et la fin d'un parallélisme structural sont figurés symboliquement par deux traits parallèles horizontaux (symbole n° 9.2.2.5 repris de la norme ISO 5807).

Lorsque l'évolution de chacune des séquences est terminée, la transition suivante est validée et franchie lorsque la réceptivité associée est vraie.

Avant la publication de la norme NF C 03-190 sur le GRAFCET, on parlait de **divergence et de convergence en ET** pour désigner respectivement l'entrée et la sortie de ce type de structure.

Exemples de séquences simultanées

• Cas 1 : parallélisme structural fermé



Commentaire

Le système étant dans la situation $S = \{16\}$, si la condition de transition « fin action A » devient vraie, le système évolue vers la situation $S = \{20,30\}$ en effectuant simultanément l'activation des étapes 20 et 30 et la désactivation de l'étape 16.

Les séquences 20 à 22 et 30 à 31 évoluent alors indépendamment l'une de l'autre.

Figure 22

Parallélisme structural fermé

Lorsque les étapes 22 et 31 sont actives ($S = \{22,31\}$), la transition 22, 31 → 17 est franchie (1 : condition de transition toujours vraie) ce qui entraîne simultanément l'activation de l'étape 17 et la désactivation des étapes 22 et 31.

Les étapes 22 et 31 mémorisent la fin de chaque séquence : ce sont des étapes d'attente.

- Cas 2 : parallélisme structural ouvert

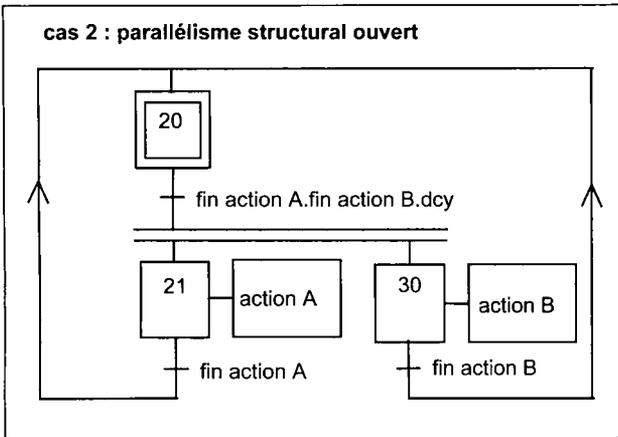


Figure 23

Parallélisme structural ouvert

Commentaire

Le GRAFCET étant dans la situation $S = \{20\}$, si la condition de transition « fin action A.fin action B.dcy » devient vraie, la situation du GRAFCET évolue vers la situation $S = \{21,30\}$.

Si la réceptivité « fin action A » devient vraie avant la réceptivité « fin action B », la situation du GRAFCET évolue de $S = \{21,30\}$ vers la situation $S = \{20,30\}$.

Si la réceptivité « fin action B » devient alors vraie, la situation du GRAFCET évolue de $S = \{20,30\}$ vers la situation $S = \{20\}$.

L'étape 20 activée et réactivée reste activée.

Le même raisonnement est applicable si, étant dans la situation $S = \{21,30\}$, l'action B est terminée la première.

Si étant dans la situation $S = \{21,30\}$, les actions A et B se terminent ensemble, l'évolution est comparable à celle de la sortie du parallélisme structural du cas 1. Le GRAFCET se retrouve en situation $S = \{20\}$.

Remarque :

Le cas 2 montre qu'il est possible d'avoir un parallélisme structural ouvert, c'est-à-dire que le tracé du GRAFCET peut ne comporter qu'une seule fois le symbole « 2 traits ».

Mais il faut souligner que dans beaucoup de cas, il peut s'agir d'erreurs d'écriture du GRAFCET.

Il est donc important de s'assurer que toutes les situations accessibles sont conformes au cahier des charges de l'automatisme.

Sélection de séquences

La sélection de séquences permet de choisir entre plusieurs évolutions possibles à partir d'une ou de plusieurs étapes figurées par autant de transitions.

Généralement, les réceptivités associées à toutes ces transitions sont exclusives.

Séquences exclusives à partir d'une seule étape

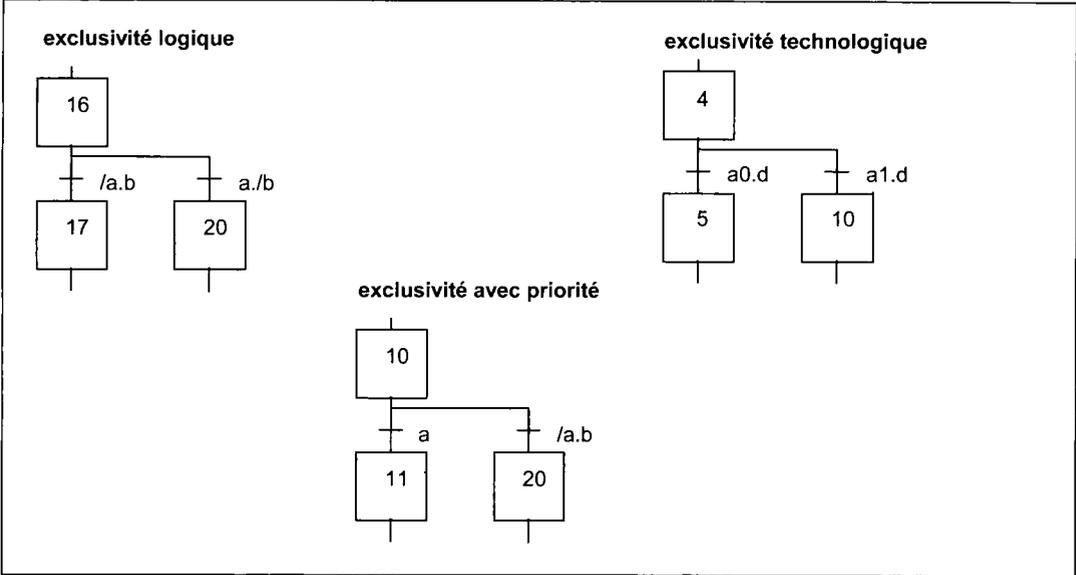


Figure 24

Séquences exclusives à partir d'une seule étape

- **Exclusivité logique**

Les réceptivités $/a.b$ et $a./b$ sont logiquement exclusives c'est-à-dire qu'elles ne peuvent être simultanément à l'état 1.

- **Exclusivité technologique**

Les réceptivités $a0.d$ et $a1.d$ sont technologiquement exclusives par l'intermédiaire des capteurs fin de course $a0$ (fin de course retour) et $a1$ (fin de course aller) du vérin A. Il est en principe impossible, du fait de leur position sur la machine, qu'ils puissent être actionnés en même temps.

Par sécurité, on peut s'assurer que le capteur opposé n'est plus actionné.

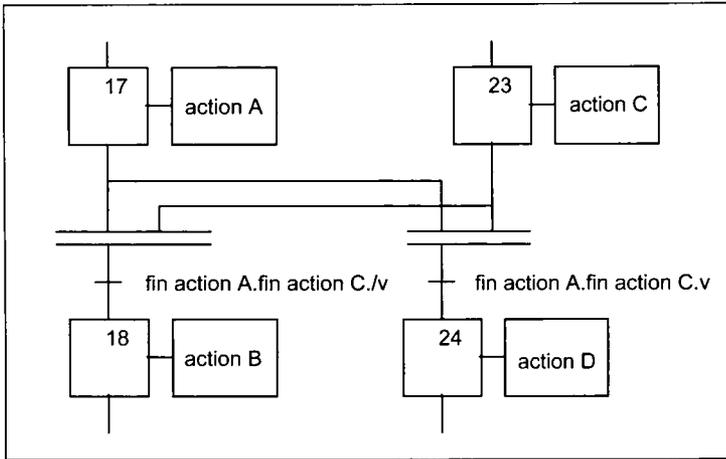
Les expressions logiques des réceptivités deviennent alors pour cet exemple :

$a0./a1.d$ et $a1./a0.d$.

- **Exclusivité logique avec priorité**

Les réceptivités a et $/a.b$ sont logiquement exclusives avec priorité à la transition $10 \rightarrow 11$ sur la transition $10 \rightarrow 20$ si a et b sont mis en même temps à l'état 1.

Séquences exclusives à partir de plusieurs étapes



Commentaire

Évolutions possibles :
 $S = \{17,23\}$ vers $S = \{18\}$ ou $S = \{17,23\}$ vers $S = \{24\}$ suivant l'état de la variable v .
 Les étapes 17 et 23 peuvent aussi être des étapes d'attente, auquel cas aucune action ne devra leur être associée.

Figure 25

Séquences exclusives à partir de plusieurs étapes

Saut d'étapes

Ce type d'évolution permet de sauter une ou plusieurs étapes suivant l'état de la variable de sélection.

Noter que la branche de court-circuit ne contient pas d'étape.

La variable de sélection est dite aussi « **variable d'aiguillage** ».

Commentaire

Si la condition de transition fin action E.m est vraie, alors saut de l'étape 12 vers l'étape 15.

La sélection est assurée par la variable m .

Les actions F et G ne seront pas exécutées tant que la variable m restera à l'état 1.

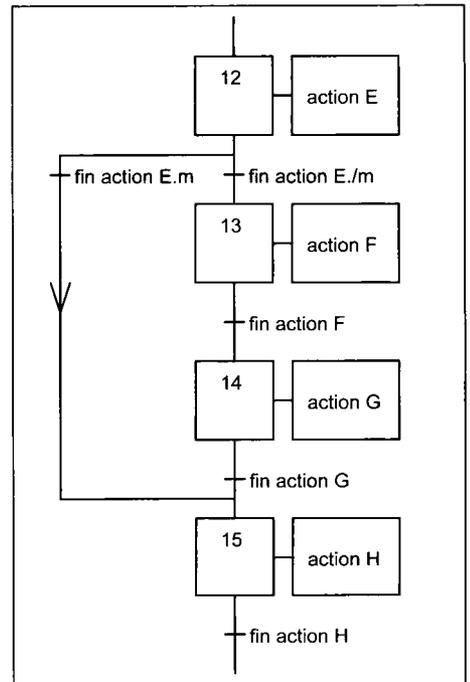


Figure 26

Saut d'étapes

Reprise de séquence

Ce type d'évolution permet de reprendre une ou plusieurs fois la même séquence tant que l'état de la variable d'aiguillage est maintenu.

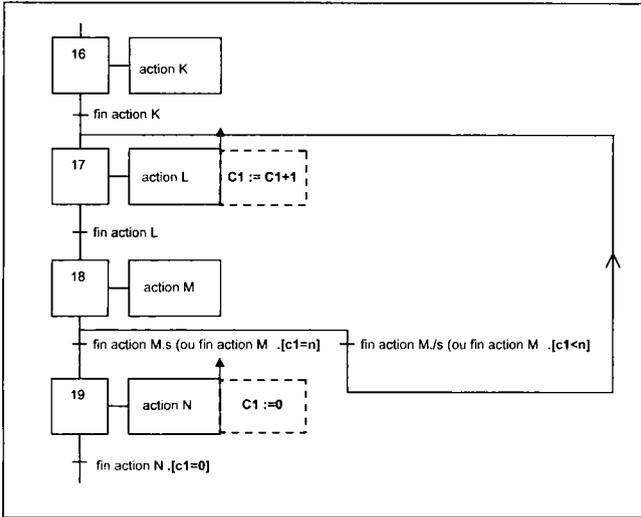


Figure 27

Reprise de séquence

Commentaire

Si la variable d'aiguillage se conserve l'état 0, alors reprise de la séquence 17-18 et exécution des actions successives L et M.

Noter que la branche de retour ne contient pas d'étape.

La variable d'aiguillage est souvent liée au test du contenu d'un compteur (ajouté ici en pointillés) (prédicats : $[c1=n]$, $[c1<n]$, $[c1=0]$).

Ici, si $[c1 < n]$, nouvelle exécution de la séquence 17-18 et exécution des actions L et M.

L'incrémement du compteur est associée à l'étape 17 : $C1 := C1+1$.

La remise à zéro (RAZ) du compteur est associée à l'étape 19 : $C1 := 0$.

Sélection alternative par bit drapeau

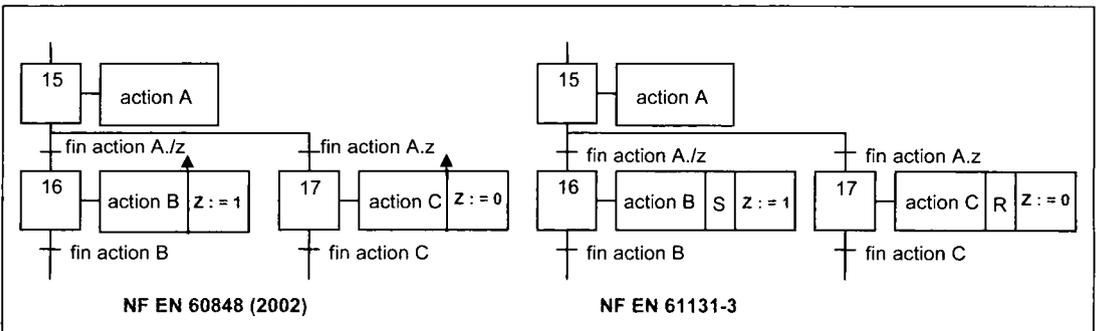


Figure 28

Sélection alternative par bit drapeau

Commentaires

Le bit Z associé à l'étape 16 et à l'étape 17 joue le rôle de drapeau (flag) ou plus simplement d'aiguilleur automatique.

Si on se trouve dans la situation $S = \{15\}$ avec fin action $A./z=1$ donc avec $z=0$, on évolue vers la situation $S = \{16\}$. L'ordre $Z=1$ associé à l'étape 16 met la variable z à l'état 1.

Au prochain passage dans la situation $S = \{15\}$ comme $z=1$, le GRAFCET évolue vers la situation $S = \{17\}$. L'ordre $Z=0$ associé à l'étape 15 remet la variable z à l'état 0.

Ainsi à chaque passage dans la situation $S = \{15\}$, on se trouve orienté alternativement soit vers l'étape 16 soit vers l'étape 17.

Ce type d'aiguillage dit par « bit drapeau » est très utilisé en informatique. On peut facilement exploiter cette procédure dans les programmes des API.

Sélection par étapes de synchronisation

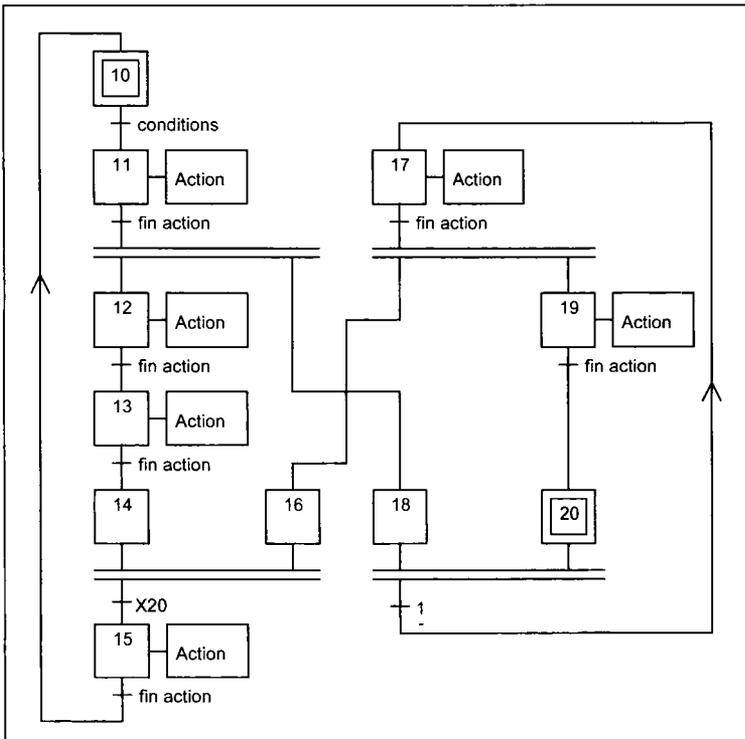


Figure 29
Étapes de synchronisation

Commentaire

Les étapes 16 et 18 autorisent respectivement le franchissement des transitions $14,16 \rightarrow 15$ et $18,20 \rightarrow 17$.

Les étapes 16 et 18 sont dites « étapes de synchronisation ». L'étape 14 est une « étape d'attente ».

Parallélisme interprété

Les divergences des exemples précédents sont toutes exclusives ce qui assure une évolution absolument déterminée lorsque la condition de transition change d'état.

Il existe pourtant des cas où on peut tolérer d'avoir des séquences inclusives. Si on adopte ce genre d'évolution, il faut impérativement réfléchir à ses conséquences. On parle alors de « **parallélisme interprété** » par opposition au parallélisme structural dont l'évolution est sans ambiguïté.

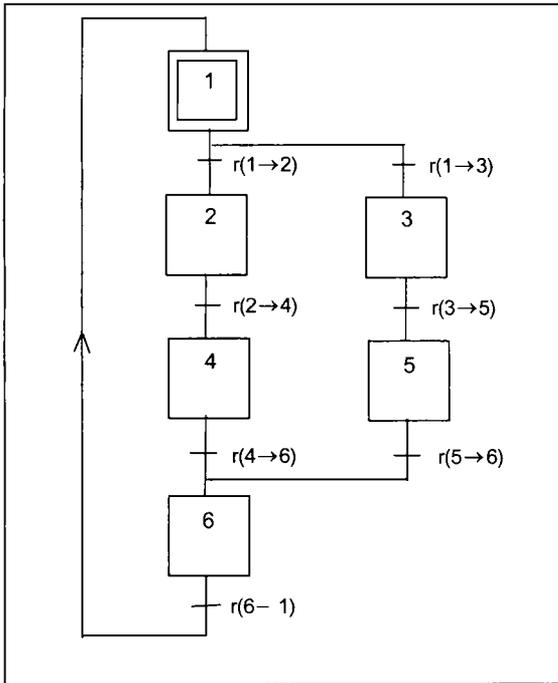


Figure 30
Parallélisme interprété

Trois cas sont à considérer :

- **1^{er} cas**

Les réceptivités $r(1 \rightarrow 2)$ et $r(1 \rightarrow 3)$ sont incompatibles, dans ce cas les évolutions sont **exclusives**.

- **2^e cas**

Les réceptivités $r(1 \rightarrow 2)$ et $r(1 \rightarrow 3)$ sont identiques, c'est-à-dire toujours vraies simultanément. Dans ce cas les évolutions sont **simultanées**.

- **3^e cas**

Les réceptivités $r(1 \rightarrow 2)$ et $r(1 \rightarrow 3)$ sont **compatibles**. Trois types d'évolutions sont alors possibles :

- si $r(1 \rightarrow 2)$ est vraie seule, évolution de $S = \{1\}$ vers $S = \{2\}$,
- si $r(1 \rightarrow 3)$ est vraie seule, évolution de $S = \{1\}$ vers $S = \{3\}$,
- si $r(1 \rightarrow 2)$ et $r(1 \rightarrow 3)$ sont vraies en même temps, évolution de $S = \{1\}$ vers $S = \{2,3\}$.

Conclusion

- Le premier cas signifie que le concepteur a voulu spécifier des évolutions exclusives. Il doit donc s'assurer de l'incompatibilité technologique ou logique des conditions de transitions.
- Le deuxième cas correspond à des évolutions simultanées et il est préférable alors d'adopter une représentation avec divergence en ET ou, ce qui est plus logique, de fusionner les étapes 2 et 3 et les étapes 4 et 5 afin d'obtenir un GRAFCET à séquence unique (à condition que le séquençage des actions ne soit pas modifié).
- Le troisième cas correspond typiquement au parallélisme interprété. Les réceptivités en sortie des séquences 2-4 et 3-5 devront comporter des verrouillages logiques pour que leur exécution soit complète.
- Ce dernier cas conduit souvent à des GRAFCET au graphisme simplifié. Malgré tout, ce type de structure est, si possible, à éviter et même absolument déconseillé par certains concepteurs.

Niveaux de description d'un automatisme

La description d'un automatisme se fait par approches successives selon le principe de « l'analyse descendante ».

L'automatisme est analysé selon trois points de vue :

- un point de vue processus,
- un point de vue PO, PC,
- un point de vue réalisateur.

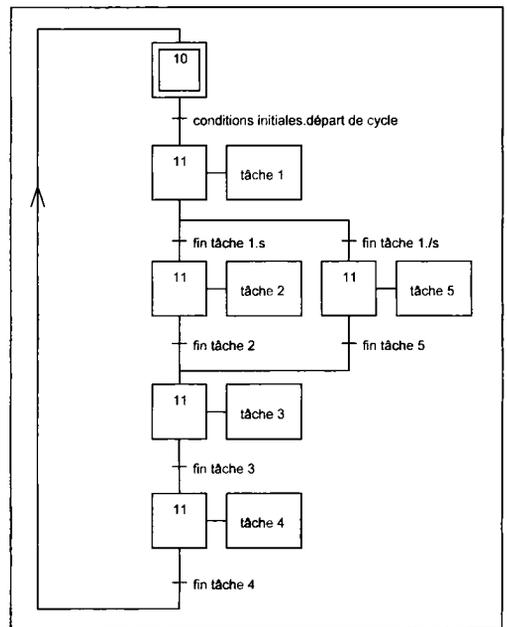
Point de vue processus

Le comportement de la partie commande (PC) vis-à-vis de la partie opérative (PO) est décrit indépendamment de tout choix technologique.

Cette description est celle du processus. Elle nécessite un inventaire préalable des fonctions principales et des fonctions secondaires puis des tâches qui en découlent. Ensuite intervient l'étude de la coordination des tâches entre elles afin de vérifier, d'une part, la faisabilité du projet, d'autre part, d'estimer les cadences de production pouvant être atteintes.

À ce niveau, le GRAFCET correspond à un Graphe de Coordination des tâches (fig.31)

Figure 31
GRAFCET de coordination des tâches



Point de vue PO, PC

La PC est décrite en tenant compte :

- des choix technologiques faits pour les mécanismes
- des options prises pour les blocs-opératifs (association actionneur-préactionneur) pour réaliser les tâches inventoriées au niveau précédent,
- des capteurs-machines fournissant les informations nécessaires au contrôle des évolutions du système automatisé de production (SAP).

À ce niveau, on peut structurer la description sous forme d'un ensemble de GRAFCET comprenant, par exemple :

- des GRAFCET de gestion des tâches,
- des GRAFCET sous-programmes,
- des GRAFCET particuliers,
- un GRAFCET de Production assurant la coordination des GRAFCET ci-dessus.

Dans un second temps ou parallèlement, cette description est complétée par une étude des modes de marche et d'arrêt conduite à l'aide du Guide d'Étude des Modes de Marche et d'Arrêt (GEMMA). Elle permet de tenir compte des contraintes imposées par le respect des normes et/ou des recommandations concernant la sécurité des personnes et par le souci de protéger le matériel.

Cette étude débouche sur un ensemble de GRAFCET partiels hiérarchisés dit « **GRAFCET global** » (UTE C 03-191)(fig. 32). On y trouve dans le cas le plus simple, un GRAFCET de sécurité contrôlant l'énergie sur la PO et gérant les procédures de sécurité et un GRAFCET de conduite gérant les modes de marche et d'arrêt. Ces deux GRAFCET supervisent le GRAFCET de production regroupant le GRAFCET de coordination des tâches, les GRAFCET des tâches et/ou des sous-programmes et les GRAFCET particuliers (GRAFCET global de production).

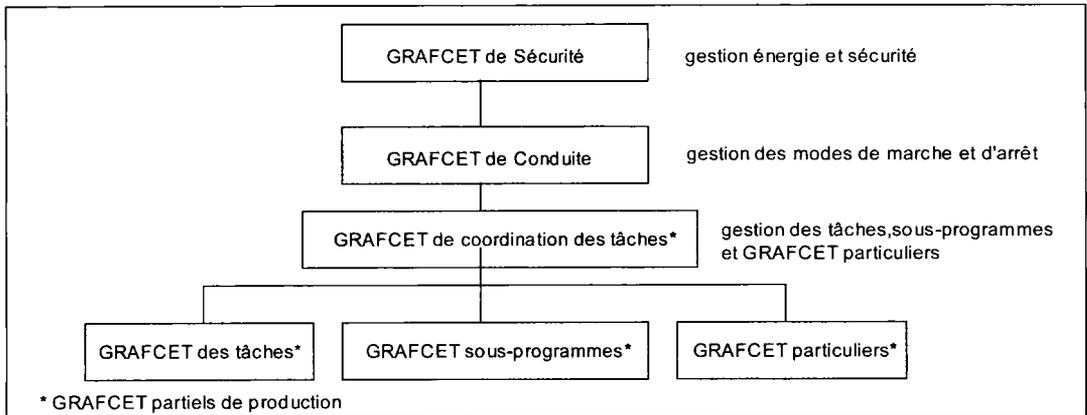


Figure 32
Hiérarchie entre les GRAFCET partiels du GRAFCET global

Le complément d'informations fourni par cette étude permet de définir progressivement la partie relation (PR) dont notamment le pupitre de commande. Il faut :

- choisir les capteurs-opérateurs,
- choisir les composants de visualisation et de signalisation,
- étudier leur implantation sur la face avant du pupitre de commande.

Point de vue réalisateur (CEI 61131-3)

Il s'agit de décrire l'automatisme dans sa version définitive (réalisation), après avoir :

- fait un choix technologique du constituant et/ou des composants de la PC (emploi conseillé des *technoguides* de l'ADEPA),
- levé les options prises au sujet des préactionneurs quant à leur type et à leur mode de commande (monostable ou bistable),
- reporté ou non sur des composants externes à la PC certains traitements réalisables par celle-ci (fonctions opératives de commande : mémorisation, temporisation, comptage, etc.),
- défini les caractéristiques et les performances du constituant retenu pour réaliser la PC :
 - nombre d'entrées-sorties,
 - capacité mémoire,
 - capacité en étapes,
 - traitement analogique-numérique,
 - puissance de calcul,
 - vitesse d'acquisition en comptage et en conversion analogique-numérique (AN) ou en conversion numérique-analogique (NA),
 - langages disponibles, etc. si par exemple il s'agit d'un API.

Le constituant retenu pour la PC conditionne la technique de réalisation : choix entre solution câblée, solution programmée ou solution mixte.

Il faut tenir également compte du choix définitif :

- de la technologie des capteurs,
- de la technologie de commande des entrées des préactionneurs,
- des interfaces permettant de combiner des technologies différentes : interfaces pneumo-électriques, par exemple.

Tous ces choix influencent la description définitive dans sa structure et ses notations.

La structure des GRAFCET devra être adaptée en fonction du choix du constituant de la PC :

- au langage disponible, orienté GRAFCET ou non, si un API est retenu,
- aux particularités du séquenceur notamment pour les GRAFCET comportant une boucle à deux étapes (voir page 85).

Afin d'alléger l'écriture et le vocabulaire, s'il n'y a aucune ambiguïté avec le reste du texte, la désignation des descriptions selon l'un des trois points de vue analysés ci-dessus peut être résumée de la manière suivante :

- point de vue processus = niveau 1,
- point de vue PO, PC = niveau 2,
- point de vue réalisateur = niveau 3 (utilisation éventuelle du langage SFC).

Analyse descendante interne à chacun des points de vue

La démarche se fait selon un « axe finesse » allant du SAP global jusqu'aux détails. Des « spécifications techniques » sont également à prendre en compte.

Au niveau 1 (point de vue processus) sont considérées principalement les « spécifications fonctionnelles » concernant les fonctions ou tâches à effectuer par le SAP et leur coordination.

Au niveau 2 (point de vue PO, PC) sont considérées principalement les « spécifications technologiques » concernant le choix technologique des constituants et composants de la PO, de la PR et de la PC.

Au niveau 3 (point de vue réalisateur) sont considérées principalement les « spécifications opérationnelles » concernant le comportement de la PC dans le contexte de production :

- production normale,
- modes de marche et d'arrêt,
- procédures de sécurité,
- etc.

La figure 33 montre suivant quels axes procède l'analyse descendante d'un SAP

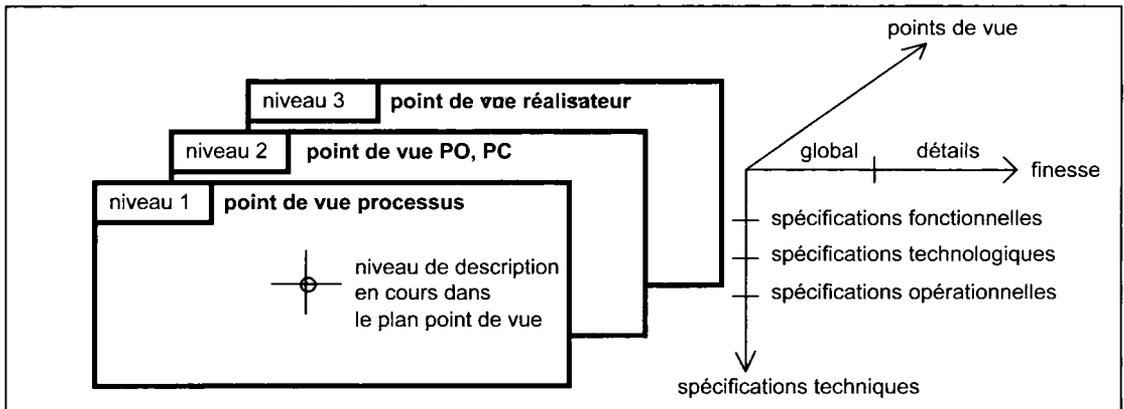


Figure 33

Les points de vue et les axes de progression de l'analyse descendante d'un SAP

Pour la conception de SAP sûrs, conformes aux normes nationales, européennes et internationales sur la sécurité, l'analyse ne peut se limiter uniquement à la prise en compte de ces trois points de vue.

La sûreté, qui sous-entend sécurité et maintien de la disponibilité, doit faire l'objet d'une étude et d'une recherche de solutions à tous les stades de la conception du SAP (*).

(*) Voir à ce sujet l'ouvrage intitulé « LE GEMMA », publié aux éditions Casteilla.

Les spécialistes semblent d'accord actuellement sur un certain nombre de points de vue à considérer. Il s'agit, dans l'ordre où ils sont étudiés :

- du point de vue produit,
- du point de vue production,
- du point de vue procédé,
- du point de vue processus (GRAFCET de niveau 1),
- du point de vue PO,
- du point de vue PO, PC (GRAFCET de niveau 2),
- du point de vue réalisation (GRAFCET de niveau 3),
- du point de vue mise au point,
- du point de vue exploitation,
- du point de vue maintenance.

Chaque point de vue, lui-même, doit être détaillé en examinant dans l'ordre :

- le cahier des charges du point de vue,
- les contraintes initiales à lever,
- les problèmes soulevés par le respect des normes de sécurité,
- les moyens propres à assurer la sécurité,
- les contraintes induites par la levée des contraintes initiales,
- la rédaction du dossier correspondant.

Organisation de travail : de la conception à l'exploitation d'un SAP

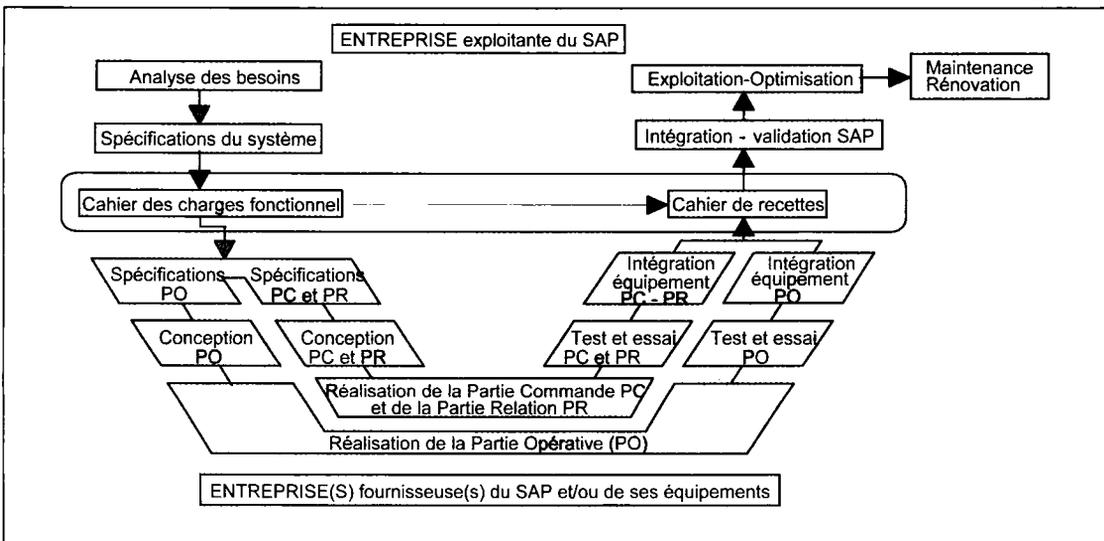


Figure 34
De la conception à l'exploitation d'un SAP

Actions associées à l'étape

Une ou plusieurs actions peuvent être associées à une étape. Les actions traduisent ce qui doit être fait chaque fois que l'étape à laquelle elles sont associées est active moyennant ou non certaines conditions supplémentaires.

Niveaux de description

Dans le premier niveau de description (point de vue processus), les actions correspondent aux fonctions à assurer ou aux tâches à exécuter par l'automatisme. Elles sont donc décrites de façon littérale par un nom (ou substantif) ou par un verbe à l'infinitif (fig. 35) ou encore par un symbole (à expliciter par ailleurs) à l'intérieur d'un ou plusieurs rectangles reliés à l'étape.

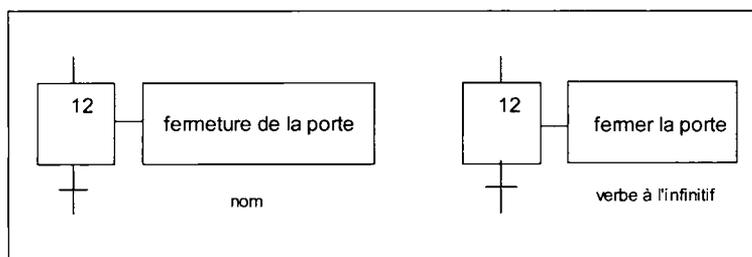


Figure 35
*Désignation littérale
de l'action*

Dans le niveau suivant de description (point de vue PC), le choix technologique des composants de la PO et de la PR permet une définition plus précise des ordres à émettre à partir de la PC.

Principaux préactionneurs de la PO et composants de la PR auxquels ces ordres sont destinés :

- bobines de contacteurs,
- pilotages de distributeurs,
- bobines de relais électromagnétiques,
- voyants, etc.

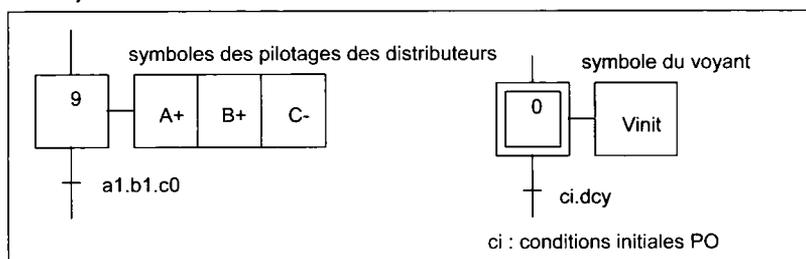


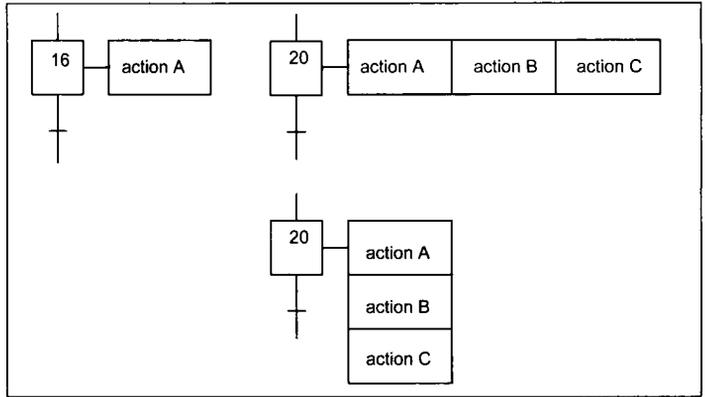
Figure 36
*Repérage symbolique des
actions*

La notation, à ce niveau, utilise une écriture symbolique (fig. 36).

Au niveau 3 (point de vue réalisateur), les symboles sont accompagnés des adresses retenues dans le cas d'une PC réalisée avec un API.

La figure 37 donne les diverses représentations possibles des actions associées à l'étape proposées **par les normes**.

Figure 37
Représentation normalisée des actions



Le terme **action** présente l'avantage de recouvrir dans le langage courant :

- ce qui doit être fait : **besoin ou effet sur la PO**
- comment on le fait : **ordre émis par la PC.**

Ce terme peut donc s'appliquer à tous les niveaux de description et assurer ainsi la cohérence du langage propre au GRAFCET. Ceci permet, sans ambiguïté, d'un niveau à l'autre, d'enchaîner le raisonnement sur les actions associées aux étapes.

• **Description niveau 1 :**

(1) Besoin ou fonction ou tâche ou action à réaliser.

• **Description niveau 2 :**

(2) Effet recherché de l'action sur la PO après choix technique du mécanisme.

(3) Effet réel sur la PO après choix technologique de l'actionneur.

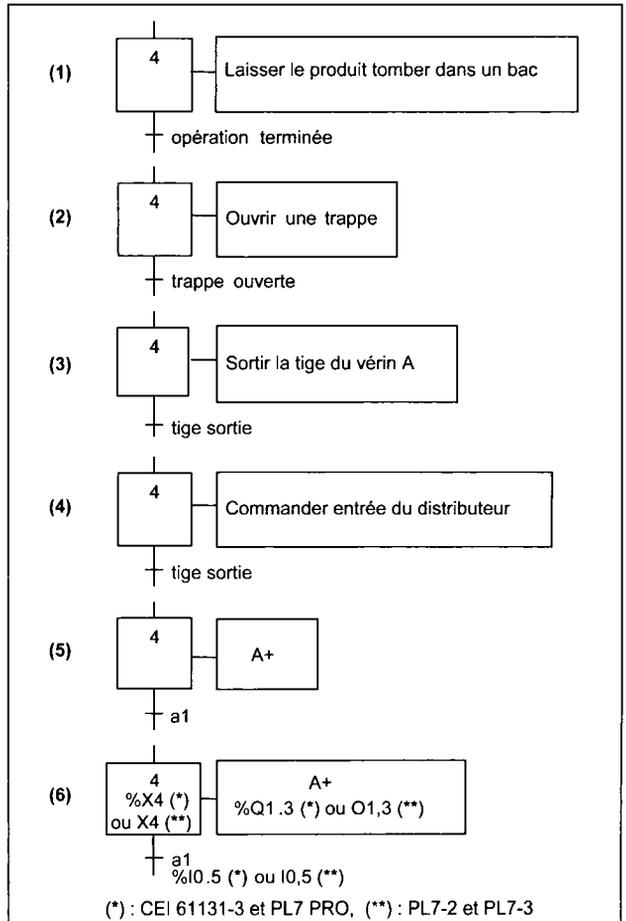
(4) Ordre issu de la PC assurant l'effet recherché par commande de l'entrée du préactionneur adopté.

(5) Écriture symbolique de l'ordre émis par la PC sur l'entrée du préactionneur.

• **Description niveau 3 :**

(6) Écriture symbolique de l'entrée du préactionneur et de l'adresse retenue après choix technologique du constituant de la PC, ici un API TSX Schneider, langages CEI 61131-3, PL7 PRO et PL7-2, PL7-3.

Figure 38
Enchaînement du raisonnement sur les actions en fonction du niveau de description



Noter, en fonction du niveau, l'évolution parallèle de l'écriture de la réceptivité associée à la transition 4 → 5.

Classification des actions ou ordres

Le critère retenu pour les actions est la durée d'activation de l'étape à laquelle celles-ci sont associées.

On distingue principalement les modes continus (assignation sur état) et les modes mémorisés (affectation sur événement) et plus particulièrement :

- les actions continues ou à niveau (symbole N pour *Not memorized*),
- les actions continues avec condition d'assignation (symbole C pour *Conditioned*),
- les actions continues avec condition d'assignation dépendante du temps (symbole D pour *Delayed (ou retardée)* ou (symbole L pour *Limited* ou à durée limitée),
- les actions impulsionnelles ou ponctuelles (symbole P pour *Pulsed*),
- les actions maintenues ou mémorisées (symbole S pour *Set* et R pour *Reset*).

Les symboles recommandés (CEI 61131-3 2003) sont placés dans une case à gauche de celle de l'action.

Action continue ou à niveau (symbole N)

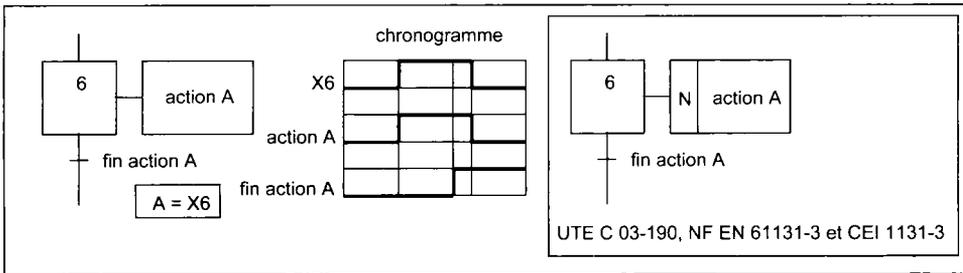


Figure 39
Action continue

L'exécution de l'action se poursuit tant que l'étape à laquelle elle est associée reste active.

Les actions continues avec condition d'assignation (symbole C pour *Conditioned*)

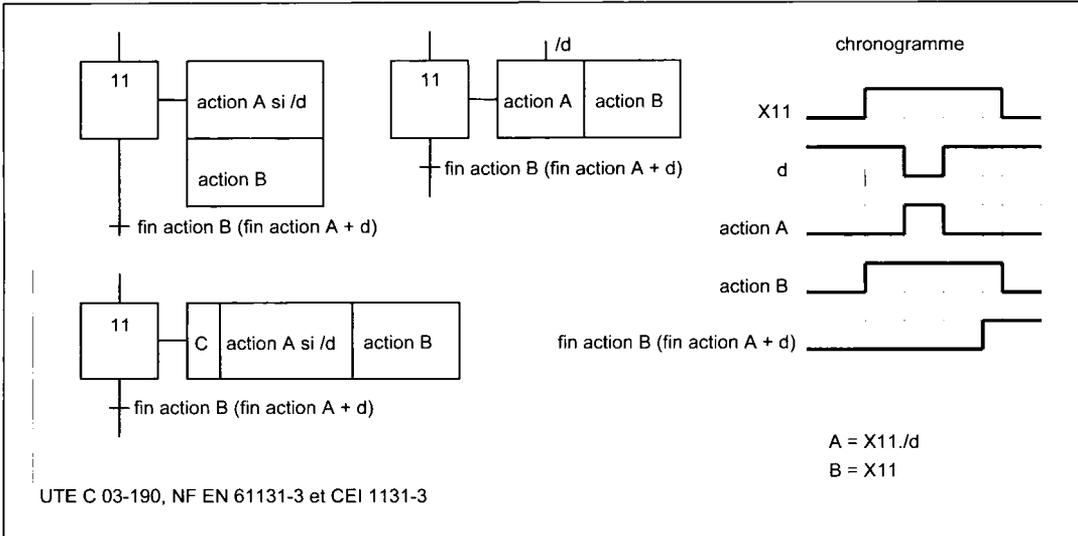


Figure 40
Actions continues avec condition d'assignation

L'exécution de l'action ne devient effective que lorsque l'étape, à laquelle elle est associée, devient active et que la condition d'assignation associée à l'action devient vraie, c'est-à-dire égale à 1.

Cette condition peut être écrite à l'intérieur du cadre réservé à l'action ou à l'extérieur de ce cadre, à côté d'un tiret situé à la partie supérieure. Cette dernière notation est uniquement utilisée pour une raison pratique lorsque la place fait défaut à l'intérieur du cadre.

Il faut noter que la situation de la PC n'est pas modifiée lorsque la condition associée à l'action devient vraie. On dit que l'automatisme n'est pas « réceptif » mais est seulement « sensible » à cette condition.

Les actions continues avec condition d'assignation dépendante du temps

Il s'agit de cas particuliers de l'action conditionnelle où le temps intervient comme condition associée à l'action.

La notation générale durée/étape de lancement est applicable. À titre de rappel 5s/X15 signifie que ce signal sera émis 5 secondes après le début de l'activation de l'étape 15.

Dans l'exemple (fig. 41), l'action A est une action à durée limitée par rapport à la durée d'activité de l'étape 17 tandis que l'action B est une action retardée par rapport au début d'activité de l'étape 17.

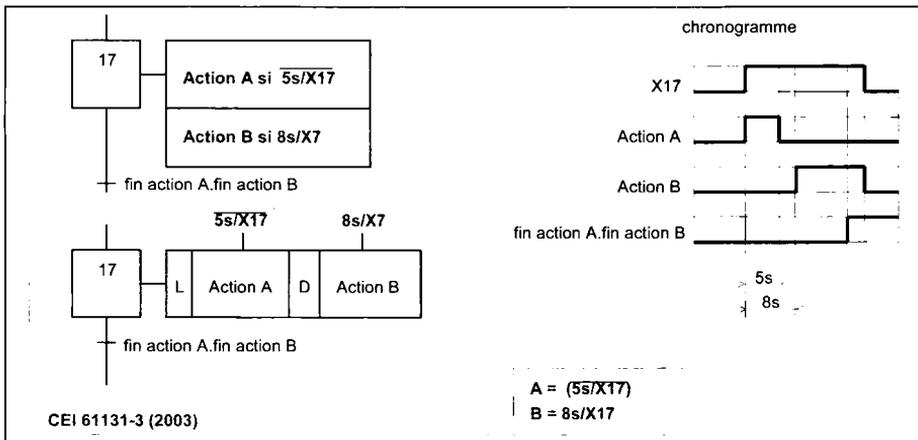


Figure 41
Action retardée et action à durée limitée

Remarque :

La variable de temporisation, d'après les normes CEI 60848 (2003) et NF EN 60848 (2002) s'écrit maintenant durée/ X_i où i est le numéro de l'étape de lancement au lieu de « t/étape/durée ».

Actions mémorisées

Il s'agit d'actions ou d'ordres dont l'effet doit se poursuivre durant l'activité d'étapes consécutives.

La norme NF EN 60848 propose les nouvelles représentations suivantes :

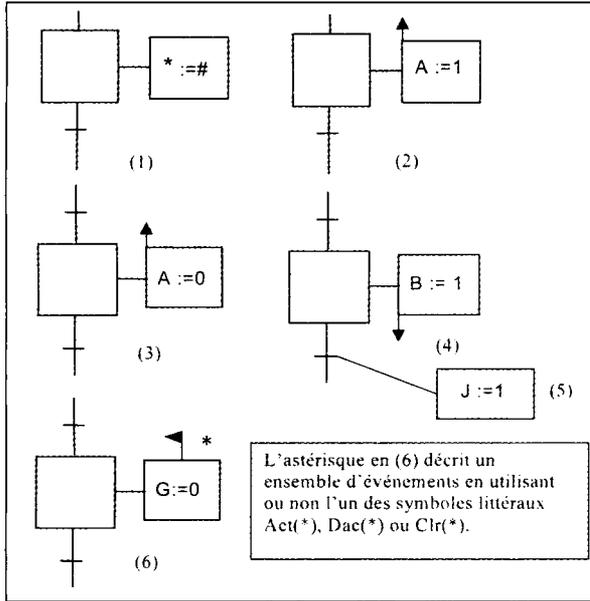


Figure 42
Ensemble des actions mémorisées

(1) Affectation de la valeur # à une valeur * (symbole de base).

(2) Action à l'activation avec affectation de la valeur 1 à la variable booléenne A.

(3) Action à l'activation avec affectation de la valeur 0 à la variable booléenne A.

(4) Action à la désactivation avec affectation de la valeur 1 à la variable booléenne B.

(5) Action au franchissement d'une transition.

(6) Action sur événement avec description littérale éventuelle d'événements internes :

activation d'une étape :

Act (repère de l'étape)

désactivation d'une étape :

Dac (repère de l'étape)

franchissement d'une transition :

Clr (repère de la transition).

Action mémorisée sur événement (symbole P)

Ce type d'action est associé à un événement (front montant ou front descendant) dont la durée est naturellement limitée mais toujours considérée de durée suffisante pour que l'action en question soit exécutée, par exemple :

- l'incréméntation d'un compteur ou la décrémentation d'un décompteur,
- la validation de la présélection d'un temporisateur, d'un compteur, d'un décompteur, etc.

L'événement seul ou combiné avec une condition peut trouver sa source dans le changement d'état d'une variable interne ou externe :

$\uparrow X_i$ ou $\downarrow X_i$ ou $\uparrow e$ ou $\downarrow e$

Remarque :

L'action, commandée par le front montant de l'étape à laquelle elle est associée, n'est exécutée qu'une seule fois si l'étape est simultanément activée et désactivée. En effet, d'après la règle 5, cette étape reste activée et par conséquent elle ne peut pas émettre un nouveau front montant.

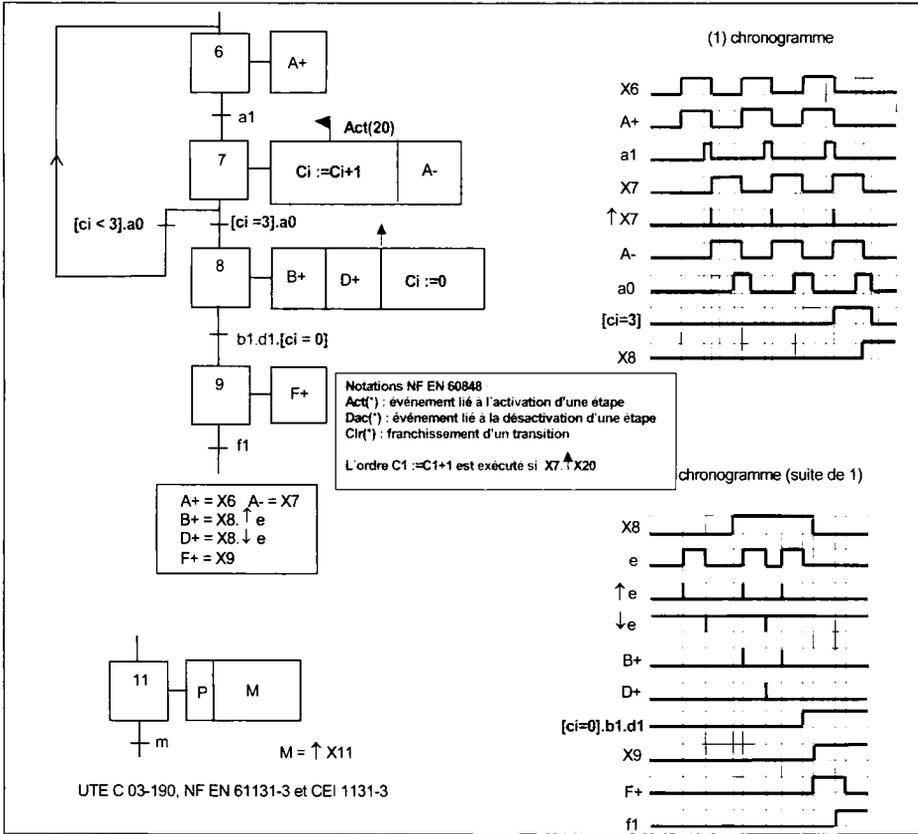
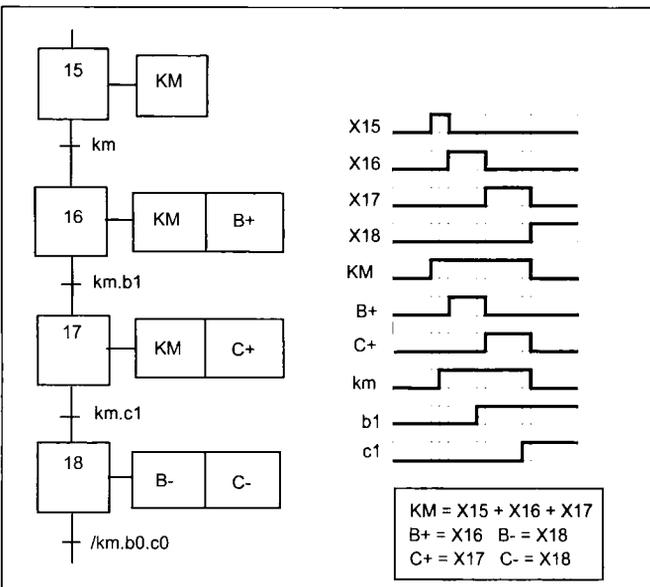


Figure 43
Actions mémorisées sur événement

D'autres moyens peuvent être exploités pour mémoriser des actions.



- par association à des étapes consécutives Xa à Xi d'un GRAFCET.

Figure 44
Mémorisation par association à des étapes consécutives Xa à Xi d'un GRAFCET.

- en associant l'action à une étape d'un parallélisme structural

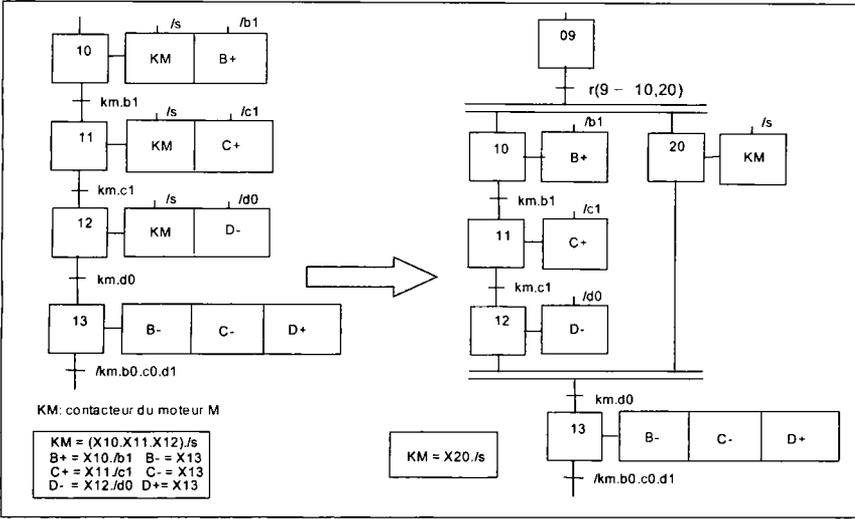
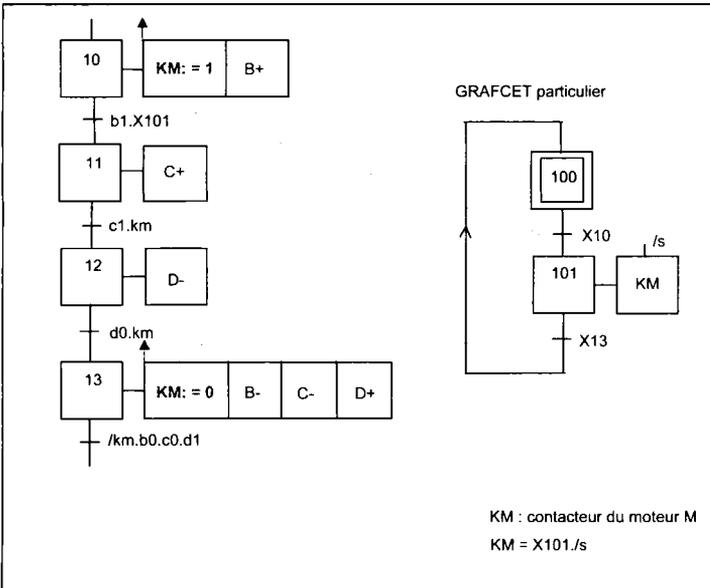


Figure 45
Mémorisation par association à une étape dans un parallélisme structural

Commentaire

La mémorisation de KM par association à une étape d'un parallélisme structural a pour seul avantage de simplifier l'expression logique de cette sortie.

- en mémorisant l'action par l'intermédiaire d'un GRAFCET particulier à cette action



Commentaire

L'écriture S pour *Set* et R pour *Reset* ne fait que souligner la mise à 1 de KM par l'étape 10 et sa mise à 0 par l'étape 13. Le GRAFCET particulier est un moyen parmi d'autres pour exploiter cette forme d'écriture.

Figure 46

Mémorisation par lancement d'un GRAFCET particulier

• en exploitant les instructions SET et RESET propres à la PC (API)

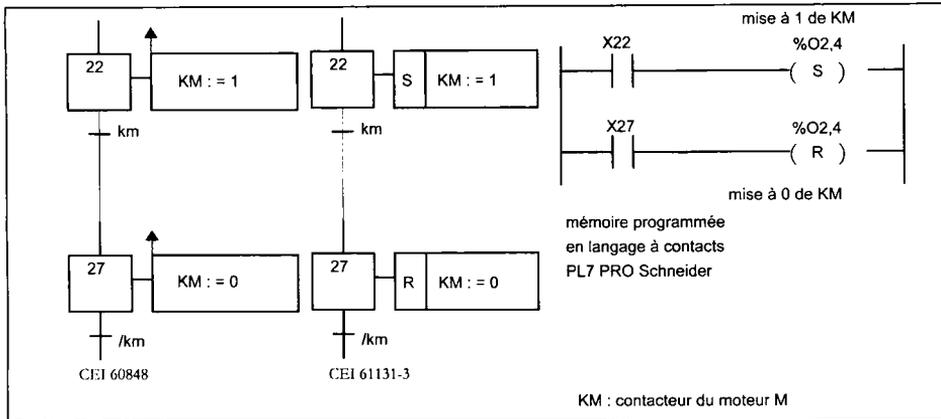


Figure 47

Mémorisation par des instructions propres à la PC

Commentaire

La mémoire est réalisée en langage à contacts en utilisant les instructions Set (mise à 1) et Reset (mise à 0).

• par mémorisation par un opérateur mémoire extérieur (ou intérieur) à la PC (fonction opérative associée (UTE C 03-191))

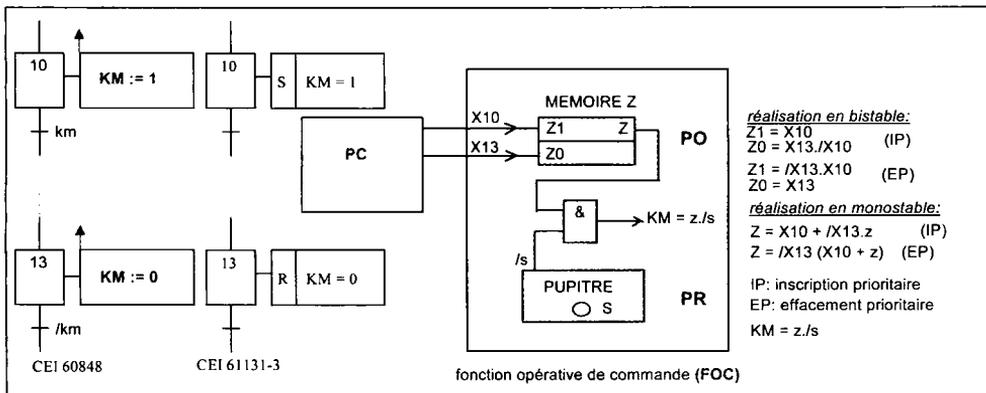


Figure 48

Mémorisation par un opérateur extérieur à la PC

La fonction opérative associée réalisée avec un opérateur mémoire prend ici le nom particulier de **fonction opérative de commande (FOC)**.

Il faut rappeler que l'opérateur mémoire peut être réalisé, par exemple :

- soit avec un composant bistable faisant office de mémoire : relais électromagnétique bistable, distributeur à double pilotage, etc.
- soit par câblage de la fonction mémoire, avec des composants monostables : relais électromagnétiques monostables auto-alimentés, modules logiques pneumatiques ou électroniques montés en rétroaction, etc.

Dans la documentation UTE C 03-190 et les normes EN 61131-3 et CEI 61131-3, la case d'action est divisée en trois sections a, b, c.

La section a contient une lettre symbole ou une combinaison de lettres symboles encore appelées identificateurs.

La section b contient le symbole ou la description de l'action.

Exemple : KM ou marche moteur M.

La section c indique le numéro de référence du signal de fin d'exécution correspondant.

La section b doit être au moins deux fois plus grande que chacune des sections a et c.

Les sections a et c ne sont spécifiées que si nécessaires.

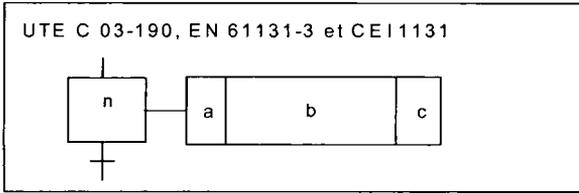
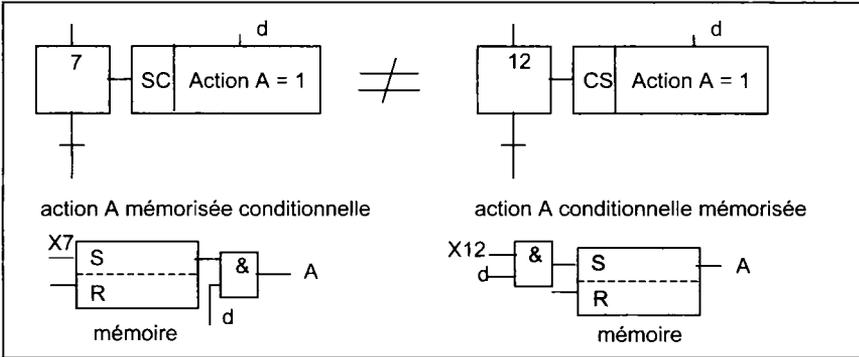


Figure 49
Sections de la case d'action

Plusieurs lettres symboles peuvent être associées à la même action. Dans ce cas faire très attention à l'ordre d'écriture. À titre d'exemple, la figure 48 montre la différence entre une action mémorisée conditionnelle (SC) et une action conditionnelle mémorisée (CS).



La société Schneider a incorporé dans le langage PL7-3 de ses API les notations suivantes concernant les actions particulières :

- actions conditionnelles :
(S (C) repère d'étape)
- actions impulsionnelles activées par le front montant de l'étape :
(S (A) repère étape),
- actions impulsionnelles activées par le front descendant de l'étape :
(S (D) repère étape),
- actions conditionnelles temporisées liées au temps d'activité de l'étape :
(S (C) repère étape + durée).

Structuration des GRAFCET partiels

Le concept de « **GRAFCET sous-programme** » (ignoré par la norme 60848) ainsi que le concept de « **macro-étape associée à son expansion** » font partie de ce qu'il est convenu d'appeler des « **macro-représentations** ».

Avec une macro-représentation, on se donne le moyen de reporter à plus tard (analyse descendante) ou sur une autre page (écran vidéo) la description détaillée de certaines séquences du GRAFCET.

Le nouveau concept d'encapsulation peut être utilisé pour structurer hiérarchiquement un GRAFCET global.

Exemple d'un cycle d'usinage comportant le perçage de multiples trous

La séquence :

- mise en rotation de la broche,
- descente d'approche en vitesse rapide,
- descente perçage en vitesse lente,
- remontée de la broche en vitesse rapide,
- arrêt de la broche en rotation,

est à répéter autant de fois qu'il y a de trous à percer. Cette séquence peut être organisée en sous-programme. Dans le cas d'une description par GRAFCET, le GRAFCET sous-programme sera appelé à chaque perçage par le GRAFCET principal (fig. 50).

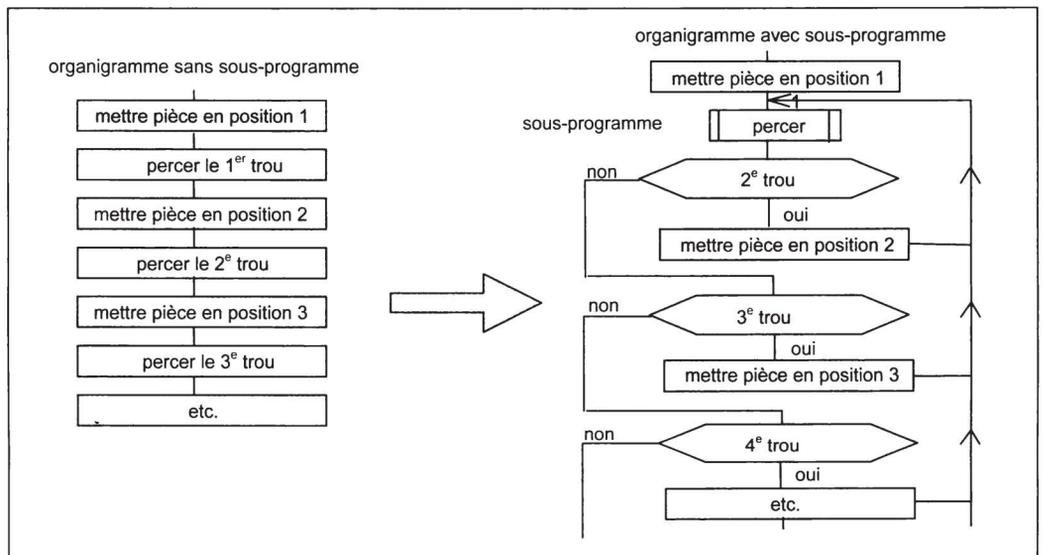


Figure 51

Organigramme sans et avec sous-programme

Symboles de la macro-représentation de lancement d'un GRAFCET sous-programme (fig. 51)

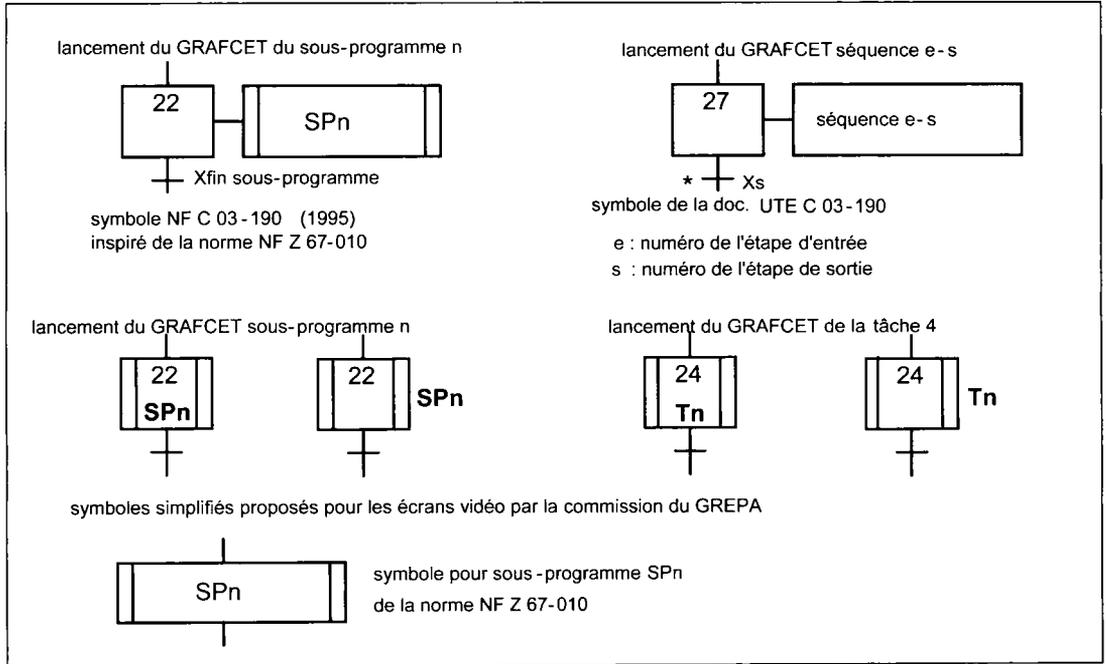


Figure 52

Symboles associés au lancement de sous-programmes

Remarques des auteurs concernant la NF EN 60848 de 2002

On peut se poser la question sur l'intérêt de modifier régulièrement les normes concernant le GRAFCET avec ses allers-retours sur les concepts, les symboles et les termes.

Est-ce vraiment nécessaire ?

Tout ce qui a été écrit, publié et enseigné jusque-là devient pour partie obsolète. Pauvres auteurs, enseignants et étudiants !

Comment comprendre les documentations établies par les bureaux d'étude avant la parution des nouvelles normes ?

Qu'apportent les nouveaux termes tels que « prédicat », « assignation », « affectation », « variable d'étape », « évolution fugace », GRAFCET connexe et GRAFCET partiel, le 1 souligné « 1 » au lieu de « =1 » ? (suite et fin, page 64)

Structuration par sous-programme

Cette structure est caractérisée :

- par son **étape d'entrée** jouant le rôle d'étape initiale indispensable pour le lancement du sous-programme,
- par son **étape de sortie** concrétisant la fin de l'exécution du sous-programme.

Aucune action extérieure ne doit être associée à ces deux étapes.

Coordination entre le GRAFCET principal et le GRAFCET sous-programme

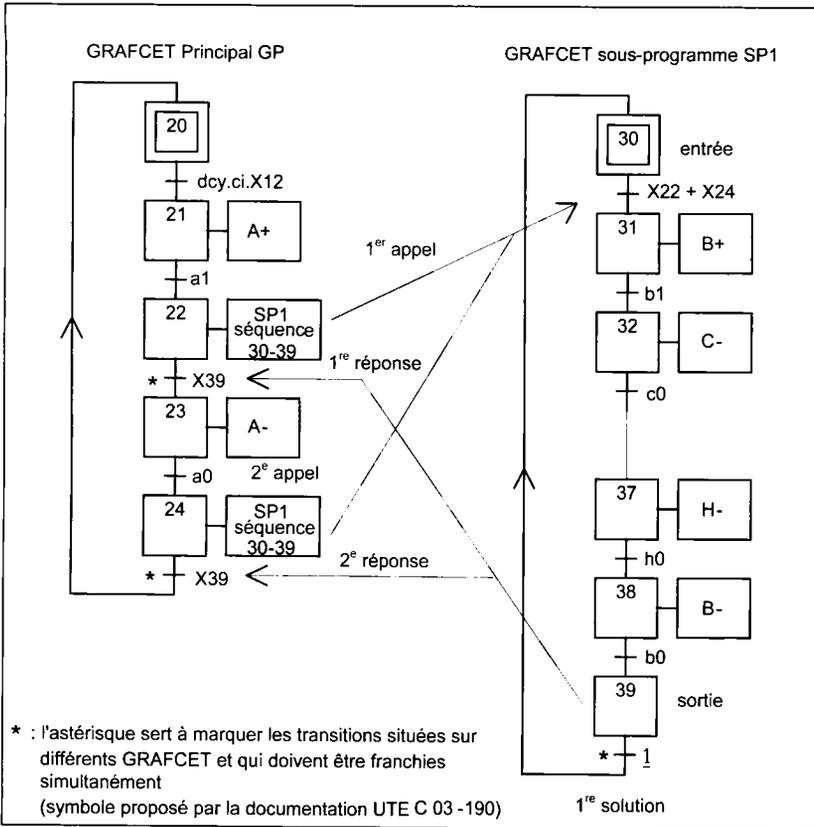


Figure 53

Coordination GRAFCET principale - GRAFCET sous-programme

Commentaire

Le GRAFCET principal (GP) comporte les étapes de lancement 22 et 24 du GRAFCET sous-programme SP1 (macro-représentation).

Le GRAFCET SP1 comporte une étape initiale d'attente ou d'entrée (étape 30) et une étape de sortie ou de retour au GRAFCET GP (étape 39).

L'activation de l'étape d'appel 22 autorise le franchissement de la transition 30 → 31 validée par l'étape 30. Il y a donc bien lancement du GRAFCET SP1.

Ensuite le GRAFCET SP1 s'exécute jusqu'à l'activation de l'étape 39 de sortie ou de retour. L'activation de l'étape 39 autorise le franchissement de la transition 22 → 23, le GRAFCET GP reprend le cours de ses évolutions. Simultanément la transition 39 → 30 est franchie (condition de transition toujours vraie (1)). Le GRAFCET SP1 se réinitialise. Noter le repérage par des astérisques des transitions franchies simultanément.

La relance du GRAFCET SP1 par l'étape 24 se fera suivant le même enchaînement : lancement du GRAFCET SP1, exécution, retour au GRAFCET GP et réinitialisation du GRAFCET SP1.

Cette coordination entre GP et SP1 fonctionne suivant le principe dit **appel-réponse** :

- un **ordre de lancement ou d'appel** est émis par le GRAFCET principal en direction du GRAFCET sous-programme : conditions X22 et X24,
- une **réponse** est fournie au GRAFCET principal en fin d'exécution du GRAFCET sous-programme : condition X39.

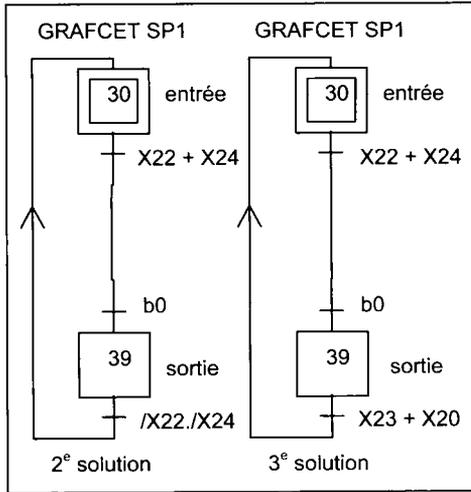


Figure 54
Solutions comparées de retour
au GRAFCET principal

Noter que si le GRAFCET sous-programme n'est pas réinitialisé, son lancement est impossible. Il n'est donc pas nécessaire d'introduire la condition X30 dans les réceptivités du GRAFCET principal.

La réinitialisation du GRAFCET sous-programme peut être obtenue de deux manières :

- soit par une condition de transition toujours vraie (1) suivant solution UTE C 03-190 (1^{re} solution, **fig. 53**),
- soit par le contrôle de la désactivation des étapes de lancement (2^e solution, **fig. 54**) : condition /X22./X24,
- soit par le contrôle de l'activation des étapes faisant suite aux étapes de lancement (3^e solution, **fig. 53 et 54**) : condition X23+X20.

La solution 2 permet, avant de réinitialiser le GRAFCET sous-programme, de vérifier si la relance du GRAFCET principal est effective. Ceci est obtenu en contrôlant la désactivation de la ou des étapes de lancement du sous-programme. Sur le plan pratique, la condition de transition de sortie est obtenue en prenant le complément de la condition de transition d'entrée.

Un GRAFCET sous-programme peut lui-même contenir une macro-représentation de lancement d'un autre GRAFCET sous-programme (imbrications des sous-programmes) (fig. 55) :

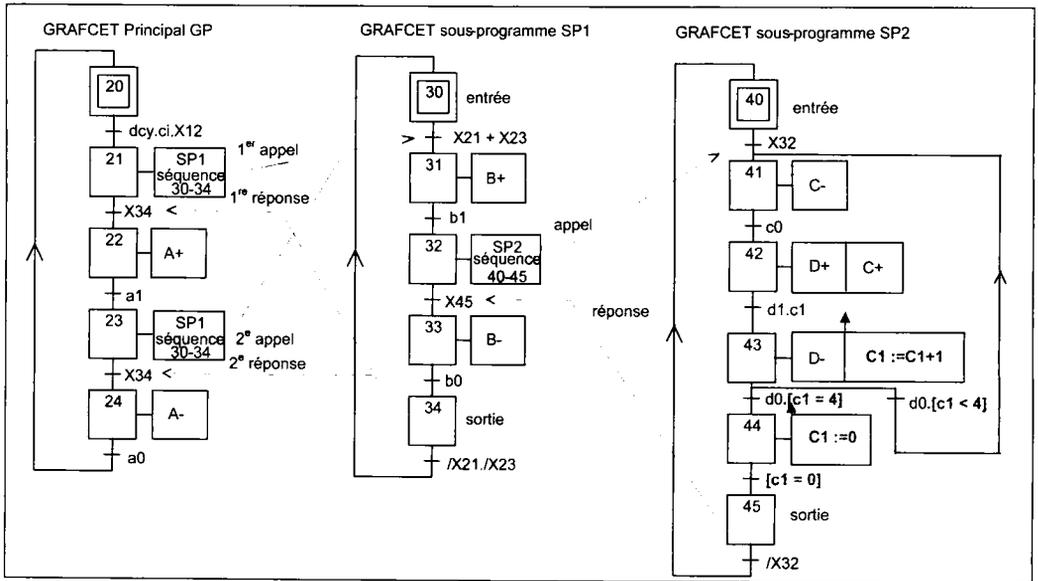


Figure 55
Imbrication de GRAFCET sous-programmes

Remarque :

La notation proposée par la documentation UTE C 03-190 suppose le GRAFCET sous-programme déjà décrit par ailleurs puisqu'il faut indiquer le numéro de l'étape d'entrée e et le numéro de l'étape de sortie s . Cette condition n'est pas remplie lorsqu'on pratique l'analyse descendante.

De ce point de vue, la notation proposée par la norme NF C 03-190 est plus souple. Elle permet de remettre à plus tard l'analyse de la séquence à décrire par le GRAFCET sous-programme.

GRAFCET de tâche

La tâche correspond à la réalisation d'une fonction bien déterminée qui pourra n'être exécutée qu'une seule fois dans le cycle de production par opposition à un sous-programme qui sera toujours exécuté plusieurs fois.

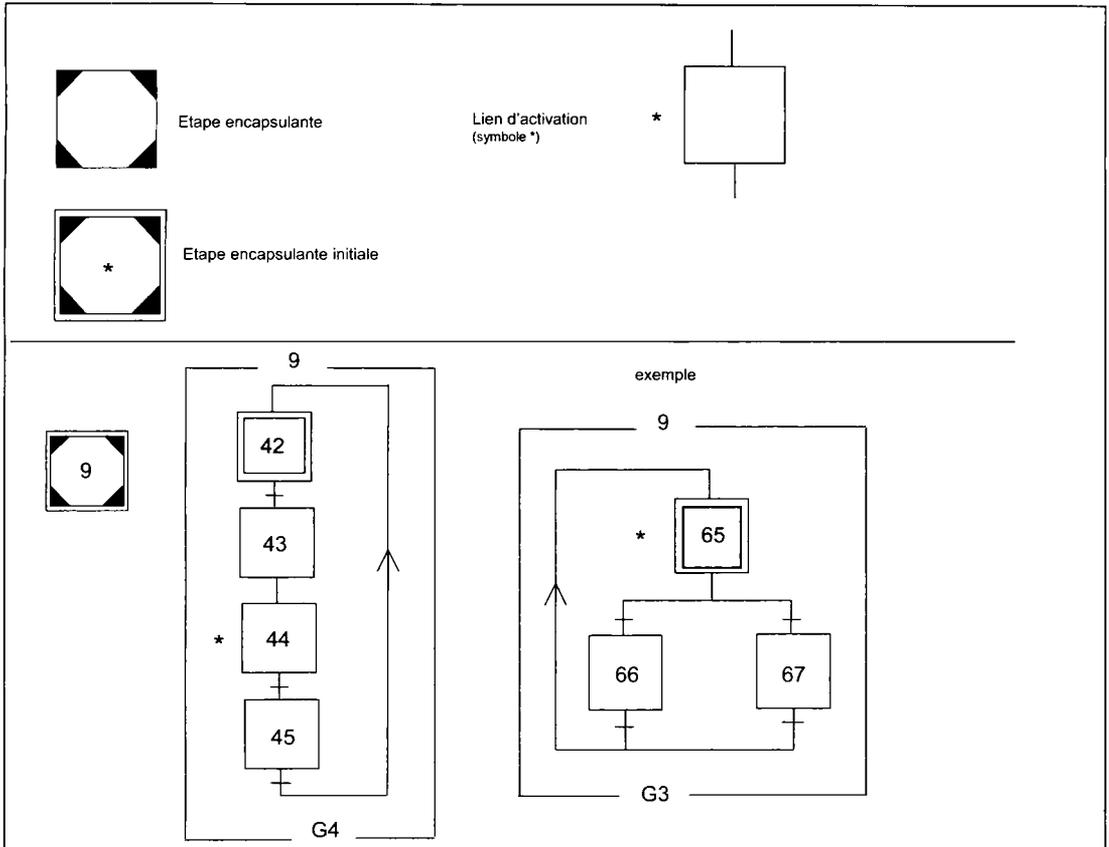
Exemple de tâches :

- dosage d'un produit,
- lancement des moteurs,
- marche de préparation (état F2 du GEMMA),
- marche de clôture (état F3 du GEMMA), etc.

Il n'y a aucune différence de structure et de principe de lancement entre un GRAFCET de tâche et un GRAFCET sous-programme.

Structuration par encapsulation

Il y a encapsulation d'un ensemble d'étapes, dites encapsulées, par une étape, dite encapsulante, si et seulement si, lorsque cette étape encapsulante est active, l'une, au moins, des étapes encapsulées est active. L'encapsulation peut être utilisée pour hiérarchiser un GRAFCET global.



Figures 56

Symboles pour étapes encapsulantes et encapsulées

L'étape encapsulante est nécessairement une étape initiale car elle encapsule l'étape initiale 42. L'encapsulation G4 contient les étapes 42, 43, 44 et 45. L'étape 42 participe à la situation initiale, par contre l'étape 44 est activée lorsqu'à chaque évolution du GRAFCET l'étape 9 est active.

De même l'encapsulation G3 contient l'étape initiale 65, elle-même activée à chaque évolution du GRAFCET.

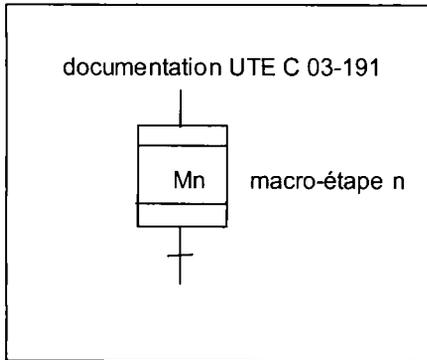
Structuration par macro-étape

Définition de la macro-étape

La macro-étape est une représentation unique d'un ensemble d'étapes et de transitions appelée « **expansion de la macro-étape** ».

La macro-étape se substitue à une étape normale du GRAFCET.

L'expansion peut toujours s'insérer en lieu et place de la macro-étape.



La macro-étape est symbolisée par un rectangle divisé en trois parties par deux traits horizontaux (fig. 57).

La macro-étape est repérée à l'intérieur (ou à l'extérieur) de la case centrale du rectangle par un identificateur numérique ou alphanumérique. Les autres cases pourront être utilisées pour un repérage complémentaire, par exemple le numéro des étapes d'entrée et de sortie lorsque l'expansion associée aura été définie.

Figure 57

Symbole de la macro-étape

La macro-étape n'a pas d'existence physique et ne peut être assimilée à une étape sinon les règles d'évolution du GRAFCET ne seraient pas respectées.

En effet comme on associe, le plus souvent, à la transition aval une condition de transition toujours vraie, la transition devrait être franchie dès que la macro-étape est activée. Or ce n'est justement pas le cas puisque l'expansion n'a pas été exécutée (voir ci-dessous).

Il s'ensuit également qu'aucune action ne peut être associée à une macro-étape.

Structure de l'expansion de la macro-étape

Cette structure obéit à certaines règles de construction, à savoir :

- l'expansion a toujours une étape d'entrée E (ou IN) et une étape de sortie S (ou OUT),
- le franchissement de la transition amont de la macro-étape déclenche l'activation de l'étape d'entrée de l'expansion,
- l'activation de l'étape de sortie de l'expansion déclenche le franchissement de la transition aval de la macro-étape, c'est pourquoi la condition de transition généralement associée à cette transition est **toujours vraie**,
- l'étape d'entrée est une étape normale qui, une fois activée, assure le début d'évolution de l'expansion. Elle ne peut jamais être une étape initiale. On peut **éventuellement** lui associer une action,
- l'étape de sortie sert à concrétiser la fin de l'exécution de l'expansion. On doit éviter de lui associer une action externe à cause de l'instabilité liée à la réceptivité \perp du GRAFCET de gestion (on peut **éventuellement** lui associer une action si la réceptivité est la fin de l'action),

- l'expansion peut comporter des étapes initiales pour pouvoir, notamment, franchir l'entrée ou la sortie d'un parallélisme structural interne,
- une expansion de macro-étape peut elle-même contenir des macro-étapes,
- une expansion peut comporter tous les types de structures propres au GRAFCET : convergences en OU exclusif ou non et convergences en ET, divergences en OU et en ET,
- l'expansion possède trois états :
 - état repos : aucune étape active,
 - état de fonctionnement : évolution en cours,
 - état final : étape de sortie active.

Conditions d'évolutions de l'expansion de la macro-étape

L'exemple envisagé (fig. 58) ne comporte que des expansions sans étapes initiales. Si une expansion n'était pas dans ce cas, elle se trouverait dans une situation non vide.

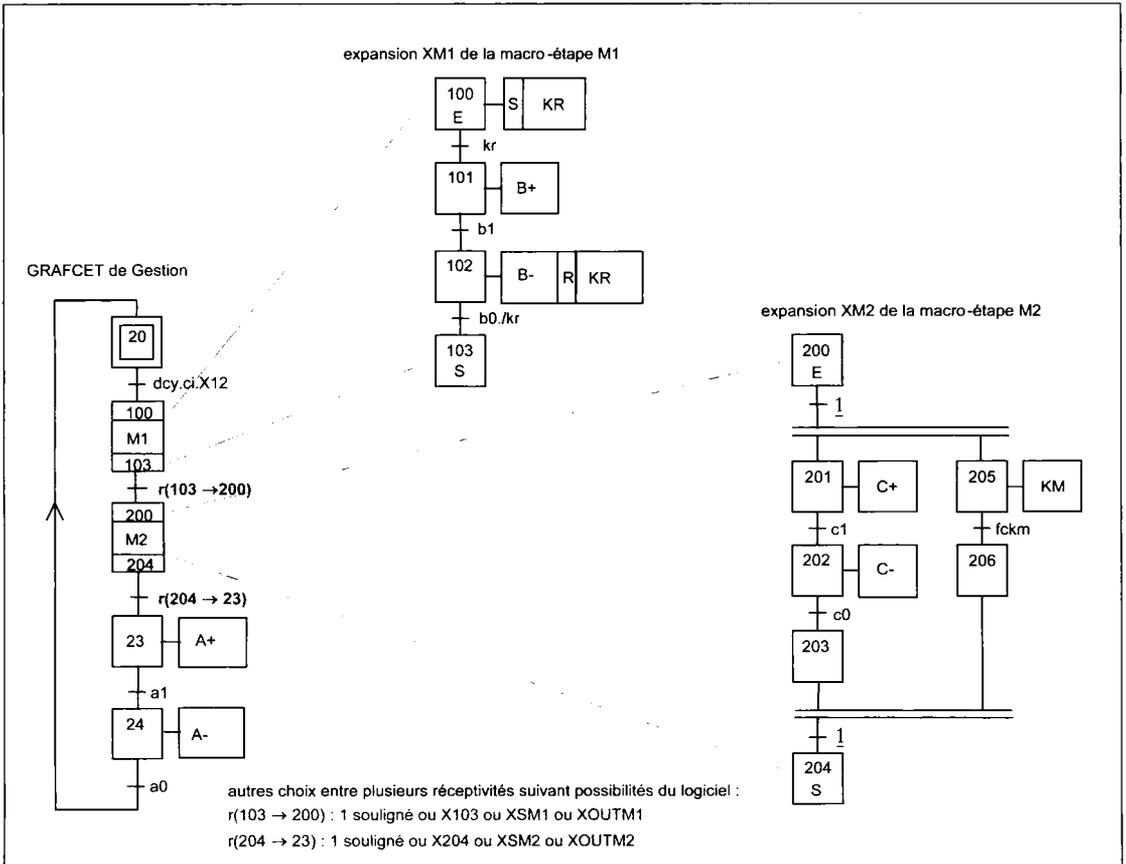


Figure 58
Macro-étapes et leurs expansions

Commentaire

L'expansion XM1 en attente est au repos (situation vide). Le franchissement de la transition amont de la macro-étape M1 active l'étape 100 de l'expansion XM1 et désactive l'étape 20 du GRAFCET principal. La macro-étape est active. L'expansion XM1 évolue alors de façon autonome.

En fin d'évolution de l'expansion XM1, l'activation de l'étape 103 valide la transition aval de la macro-étape M1. Le franchissement de cette transition active la macro-étape M2 et désactive l'étape 103 de l'expansion XM1. L'expansion XM1 se trouve de nouveau en situation vide. La macro-étape M1 est désactivée.

Le même processus est à mettre au compte de la macro-étape M2. Son expansion XM2 est lancée et évolue jusqu'à l'étape 204. La transition M2 → 23 est franchie et l'expansion XM2 se retrouve en situation vide.

Intérêt et limites des macro-représentations

L'analyse descendante d'un système par niveaux successifs permet d'affiner progressivement la connaissance du SAP, objet de l'étude.

Les macro-représentations permettent de ne pas s'encombrer de détails inutiles dès le début de l'analyse et de se consacrer uniquement à l'analyse des tâches et de leur coordination.

Le découpage en tâches ou en sous-programmes permet de mieux structurer l'ensemble de la description par GRAFCET.

La remise en cause d'une tâche ou d'un sous-programme est sans effet sur la macro-représentation à laquelle chacune ou chacun est associé. Ceci est important lorsqu'on doit faire, ultérieurement, des modifications en vue d'améliorer le fonctionnement du système.

De plus, dans le cas d'une réalisation programmée, l'emploi de sous-programmes pour les tâches ou les séquences répétitives permet un gain appréciable de place en mémoire.

Les macro-étapes et leurs expansions associées sont d'un intérêt limité lorsqu'on dessine manuellement les GRAFCET et qu'on ne débouche pas sur une réalisation sachant exploiter ce type de macro-représentation. Par contre un langage d'API orienté macro-étape met à profit cette notion en répartissant chacune des expansions sur des pages différentes en mémoire. Celles-ci sont éditées ou appelées sur l'écran vidéo de la console par "un zoom" déclenché en pointant la macro-étape concernée.

Remarque :

Une séquence ou une tâche devant être exécutées n fois dans le même cycle nécessite n macro-étapes distinctes associées à autant d'expansions de même structure mais composées d'étapes différentes.

En conclusion, le concept de GRAFCET sous-programme ou de tâche présente l'avantage d'être indépendant du langage de l'API. Il est donc implantable sur tous les types d'automates programmables industriels. De plus, par rapport au concept de macro-étape, il apporte un gain en espace mémoire important dans le cas de séquences ou de tâches répétitives.

Structuration par forçage de GRAFCET partiel

Définition du forçage de situation

L'ordre de forçage de situation émis par un GRAFCET partiel permet d'imposer une situation à un autre GRAFCET partiel.

La situation du GRAFCET forcé évolue alors sans franchissement de transition de sa situation courante à la situation forcée.

L'ordre de forçage ne peut être émis que par une étape d'un GRAFCET partiel hiérarchiquement supérieur pour modifier la situation d'un GRAFCET partiel hiérarchiquement inférieur.

Il est donc indispensable de structurer la description de manière à établir une hiérarchie entre les GRAFCET.

Exemple de hiérarchie simple couramment adoptée

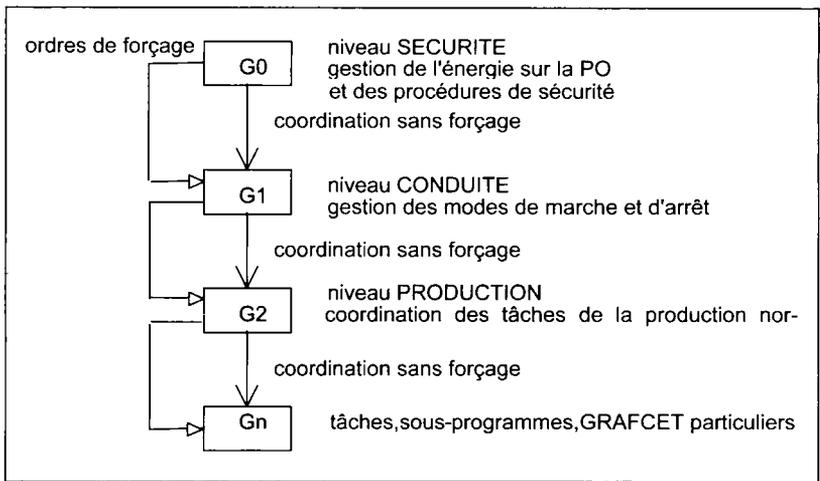


Figure 59
Hiérarchie du GRAFCET global

Situation courante forcée dans une situation précisée non vide

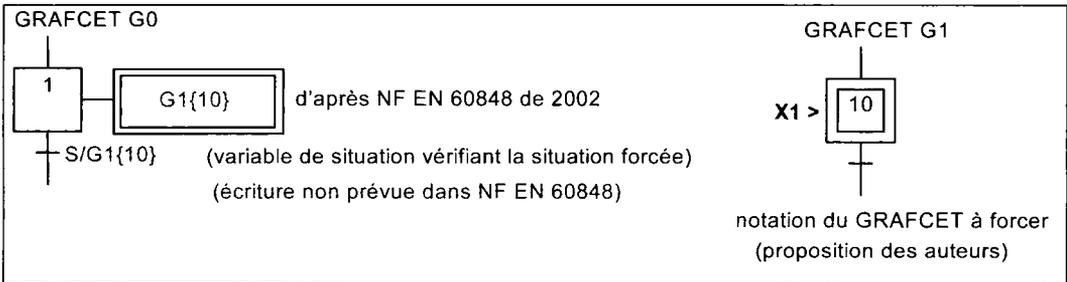


Figure 60
Forçage d'une situation non vide

Commentaire

L'activation de l'étape 1 du GRAFCET G0 force le GRAFCET G1 sur la situation non vide $S = \{10\}$ de ce GRAFCET. L'étape 10 est activée (forcée à 1), les autres étapes du GRAFCET G1 sont désactivées (forcées à 0).

Notation du forçage d'une situation non vide :

- de l'action de forçage : **GRAFCET à forcer**{situation forcée}
- du GRAFCET forcé : **étape de forçage** > accolé à la (ou les) étape(s) à forcer à 1

Situation courante forcée dans une situation vide

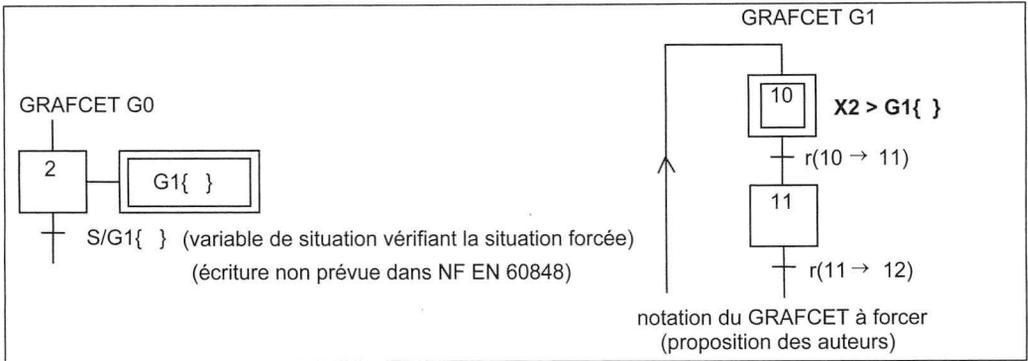


Figure 61

Forçage d'une situation vide

Commentaire

L'activation de l'étape 2 du GRAFCET G0 force le GRAFCET G1 sur la situation vide $S = \{ \}$ de ce GRAFCET. Toutes les étapes du GRAFCET G1 sont désactivées y compris les étapes initiales.

Notation du forçage d'une situation vide :

- de l'action de forçage : **GRAFCET à forcer** { }
- du GRAFCET forcé : **ordre de forçage** > **GRAFCET à forcer** { }

! Les étapes encapsulées permettent de se libérer de l'écriture du forçage dans une situation vide.

Ordre de forçage

Règles d'évolution par forçage

- Un GRAFCET ne peut être forcé que **par un ordre émis par un GRAFCET** hiérarchiquement supérieur.
- Un GRAFCET inférieur ne peut être forcé que dans une seule situation à la fois à partir d'un ou plusieurs GRAFCET supérieurs.
- L'ordre de forçage est prioritaire sur les autres conditions assurant l'évolution du GRAFCET forcé.
- Lorsqu'il s'agit d'un forçage de situation non vide, l'ordre de forçage provoque simultanément l'activation des étapes correspondant à la situation imposée et la désactivation des autres étapes du GRAFCET forcé.

Exemple 1 :

$G3\{30\}$ est un ordre de forçage du GRAFCET G3 dans la situation imposée $S = \{30\}$. Dès que l'étape, à laquelle cet ordre est associé, devient active, cet ordre est obligatoirement émis. Il provoque simultanément l'activation de l'étape 30 et la désactivation des autres étapes du GRAFCET G3.

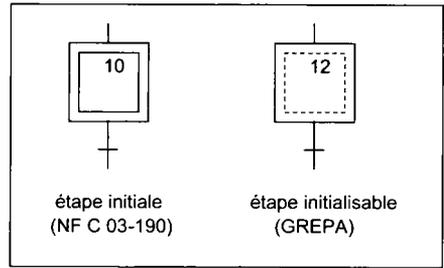
- Lorsqu'il s'agit d'un forçage de situation vide, l'émission de l'ordre de forçage provoque simultanément la désactivation de toutes les étapes du GRAFCET désigné.

Remarque :

Dans certains cas, on peut envisager de simplifier l'écriture des forçages (hors normes) :

- si la situation forcée ne prête à aucune confusion
exemple : $\{10,20\}$ force la situation $S = \{10,20\}$,
- si le forçage s'applique à plusieurs GRAFCET
exemple : $G1,G2\{10,20\}$ force G1 et G2 dans les situations respectives $S(G1) = \{10\}$ et $S(G2) = \{20\}$,
- si le forçage entraîne la réinitialisation d'un ou plusieurs GRAFCET
exemples : $G1\{INIT\}$ force G1 dans sa situation initiale.
 $G1,G5\{INIT\}$ force G1, G2, G3, G4, G5 dans leurs situations initiales.

Figure 62
Symboles des étapes particulières



Les étapes correspondant à la situation forcée et non vide, si elles ne sont pas déjà des étapes initiales, sont par définition des étapes initialisables.

On les distinguera des étapes initiales en représentant le cadre intérieur avec des pointillés ou des tirets courts (fig. 62).

On peut rappeler, par une flèche annotée placée à gauche de l'étape initiale ou initialisable, l'origine du forçage à 1 de cette étape, étant sous-entendu que les autres étapes sont forcées à 0 (fig. 63).

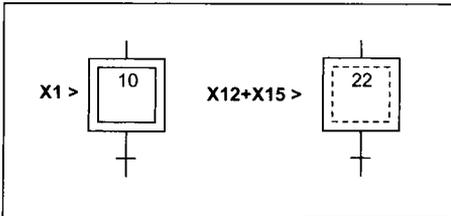


Figure 63
Repérage de l'origine de l'ordre de forçage non prévu par la NF EN60848

Exemple 2 :

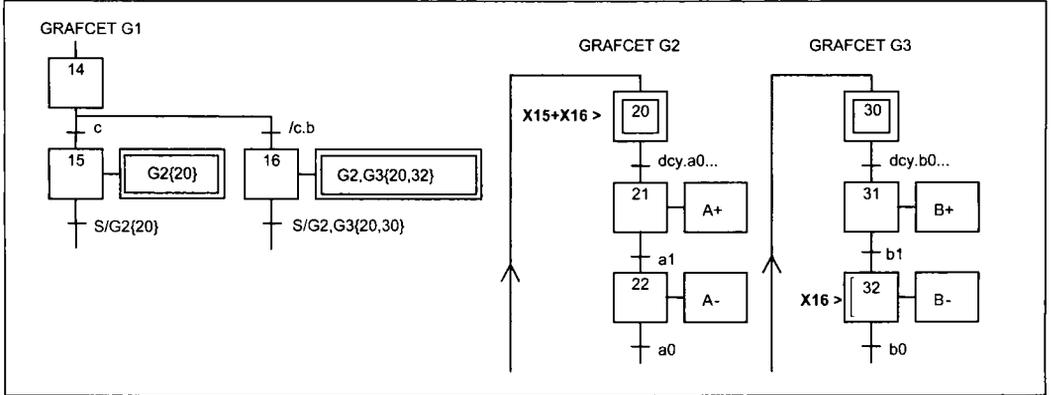


Figure 64

Ordres de forçages et GRAFCET forcés

Commentaire (fig. 64)

Suivant la situation du GRAFCET G1, le système aura deux comportements distincts :
 si $S = \{15\}$: initialisation du GRAFCET G2,
 si $S = \{16\}$: initialisation du GRAFCET G2 et forçage du GRAFCET G3
 dans la situation $S = \{32\}$

– Un ordre de forçage peut être conditionnel.

Exemple 3 :

Commentaire

Le forçage du GRAFCET G2 dans la situation $S = \{20\}$ sera effectif si l'étape 12 est active et si le GRAFCET G0 est dans la situation $S = \{1\}$.

Activation de l'étape 20 : $A(20) = X12.X1 + \dots$

Désactivation des autres étapes : $D(21) = X12.X1 + \dots$,
 $D(22) = X12.X1 + \dots$, etc.

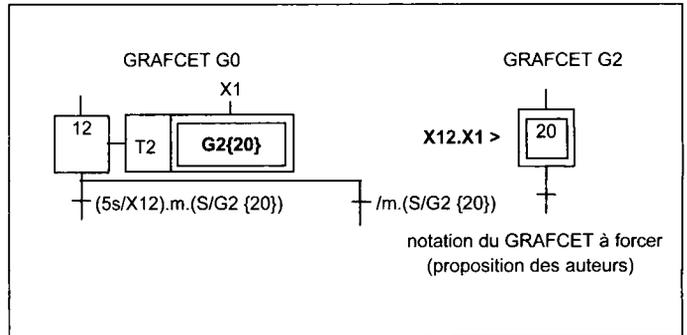


Figure 65

Forçage conditionnel

Variable de situation

L'exécution du forçage de situation doit être vérifiée. La variable de test associée est appelée « **variable de situation** ».

☛ Notation de la variable de situation : **S/GRAFCET forcé{situation forcée}**.

Cette notation qui n'est qu'une simplification d'écriture doit être explicitée par ailleurs.

Exemple :

$S/G2\{20\} = X20./X21./X22./X23$ si le GRAFCET G2 contient les étapes 20 à 23.

Forçage d'un GRAFCET partiel dans la situation courante (figeage)

Il ne s'agit plus ici, à proprement parler, de forcer une situation mais au contraire de bloquer l'évolution du GRAFCET dans une situation choisie à l'avance : situation courante ou situation prédéterminée.

Situation figée dans la situation courante

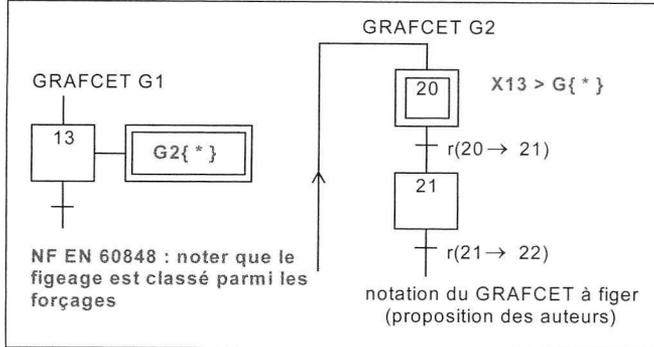


Figure 66
Ordre de figeage dans la situation courante

Commentaire

L'activation de l'étape 13 du GRAFCET G1 fige la situation du GRAFCET G2 dans sa situation courante. Les étapes actives restent activées, les étapes inactives restent désactivées.

Notation du figeage dans la situation courante :

- de l'action de figeage : **GRAFCET à figer{ * }**
- du GRAFCET à figer : **ordre de figeage > GRAFCET à figer{ * }**

Situation évoluant vers une situation de figeage prédéterminée

Commentaire

La situation du GRAFCET G2 continue d'évoluer jusqu'à la situation $S = \{22\}$ où elle se figera. Il s'agit en fait d'un figeage conditionné par la situation de figeage à atteindre. Aucune action ne doit être associée à cette étape.

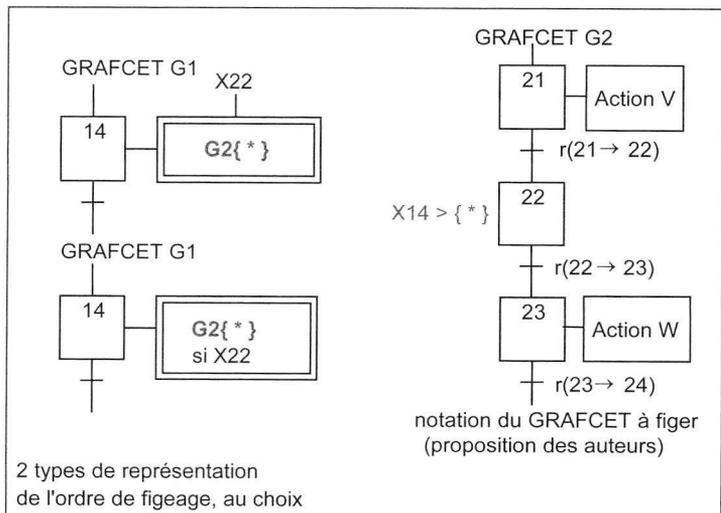


Figure 67
Ordre de figeage dans une situation déterminée

Notation du figeage dans une situation prédéterminée

- de l'action de figeage : **GRAF CET à figer { * } si situation de figeage atteinte**
- du GRAFCET à figer : **ordre de figeage > { * }** accolé à l'étape où se fera le figeage.

Problèmes entraînés par le figeage de situation

Cas du figeage dans la situation courante

Le figeage du GRAFCET de production normale dans sa situation actuelle (situation courante) sur ordre de l'opérateur ou à la suite d'un incident présente l'avantage de permettre la reprise de la production sans procédure compliquée. Il suffit de quitter l'étape émettant le signal de figeage. La désactivation de cette étape annule l'ordre de figeage du GRAFCET de production normale et la production reprend là où elle s'était arrêtée.

Le figeage obtenu, les ordres associés aux étapes actives des GRAFCET de tâches continuent d'être émis. Si ces ordres sont destinés à des entrées de préactionneurs électriques, que va-t-il se passer ?

Les bobines des contacteurs et des distributeurs vont rester sous tension pendant toute la période de figeage. Les moteurs, par exemple, vont continuer à tourner ce qui n'est généralement pas souhaitable sur le plan sécurité.

Comment résoudre ce problème ?

Il faut en même temps que l'ordre de figeage du GRAFCET de production, émettre un ordre de forçage à l'état 0 des sorties de la PO.

Le forçage à 0 des sorties la PO, en fonction de la sécurité souhaitée, peut entraîner :

- soit l'arrêt sur place des actionneurs,
- soit l'arrêt après terminaison des mouvements en cours (course des mécanismes actionnés par des vérins ou des moteurs, etc.).

Une des solutions consiste d'une part à annuler les ordres émis par les GRAFCET de tâches et d'autre part à choisir des actionneurs et des préactionneurs de technologie appropriée :

- moteurs-freins, vérins à bloqueurs de tige,
- distributeurs à 3 positions à centre fermé,
- distributeurs à 2 positions associés à des bloqueurs pilotables à clapet, etc.

L'annulation des ordres de sortie peut être obtenue :

- soit par forçage à 0 des entrées des préactionneurs. Concrètement obtenu, ici, dans le GRAFCET de production (fig. 67) par la condition /X12 associée aux actions,
- soit par un câblage du contact /fig du capteur-opérateur FIG de demande de figeage dans le circuit d'alimentation des entrées des préactionneurs ou des cartes de sorties de l'API (sécurité de premier niveau) (fig. 68),
- soit en exploitant un bit système approprié (%S9 pour les API Schneider),
- soit en combinant les procédés afin d'obtenir une redondance améliorant la sécurité.

Exemple (avec GRAFCET de Conduite simplifié)

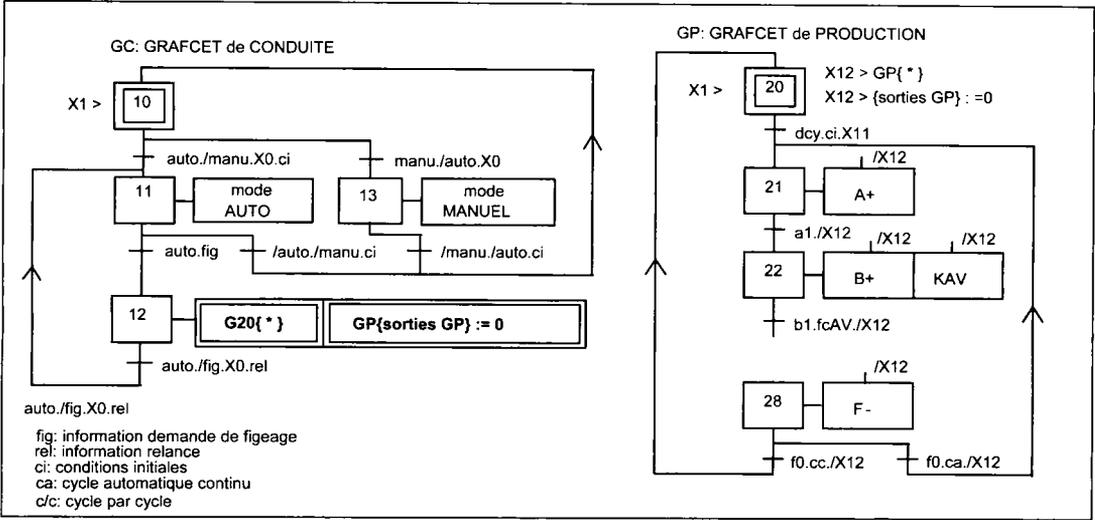


Figure 68

Exemple de figeage assorti d'un forçage à 0 des sorties

Commentaire

Le figeage du GRAFCET de Production de l'exemple est obtenu par la mise à zéro des réceptivités si l'étape 12 est active. Les vérins A, B,....., F terminent leurs courses, le moteur s'arrête immédiatement.

Il est sous-entendu qu'un choix approprié des actionneurs et des préactionneurs a été fait.

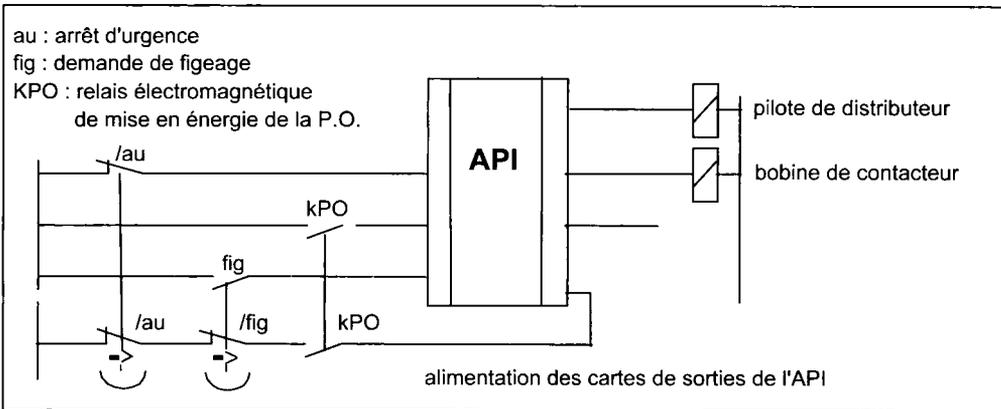


Figure 69

Solution câblée du forçage à 0 des sorties

Commentaire

Dans cette solution, l'action sur le « coup de poing FIG » ou sur le « coup de poing AU » entraîne, dans les deux cas, la coupure de l'alimentation des cartes de sorties.
Par suite, les entrées des préactionneurs ne sont plus alimentées.

Cas du figeage de situation avec mise en situation vide

Il est possible de figer le GRAFCET de production normale et les GRAFCET de tâches qu'il supervise en procédant de cette manière :

- sur ordre de figeage, mémorisation de la situation courante,
- puis mise des GRAFCET de production dans une situation vide.

■ notation (écriture extrapolée de la documentation UTE C 03-191) :

Sauv Gn{ * }

Les ordres associés aux étapes de ces GRAFCET sont ainsi systématiquement annulés.
Pour relancer l'automatisme, il faut forcer les GRAFCET concernés sur la situation sauvegardée.

■ notation (écriture extrapolée de la documentation UTE C 03-191) :

Récup Gn{ * }

Cette procédure suppose qu'on dispose des moyens appropriés pour la réaliser. Le logiciel de programmation assistée AUTOMGEN de la société IRAI ainsi que le langage PL7-3 V5 de Schneider possèdent des instructions autorisant ce type de procédure.

Exemples avec notations AUTOMGEN

- **G<G1>** : <15> mémorise l'état du GRAFCET G1 dans une série de bits (un par étape) à partir du bit 15, ces bits étant obligatoirement consécutifs,
- **F<G1>** : < > force le GRAFCET G1 dans une situation vide,
- **F<G1>** : <15> force le GRAFCET G1 sur la situation sauvegardée en récupérant l'état des étapes dans la série de bits dont le premier est à l'adresse 15.

Remarques des auteurs concernant la NF EN 60848 (suite et fin)

Pourquoi la NF EN 60848 ignore-t-elle la structure de sous-programme et sa variante concernant les tâches, structure qui a fait et fait toujours ses preuves en informatique et que la norme d'origine, la C 03-190 de 1995 proposait aux automaticiens ?

Pourquoi ignore-t-elle aussi la variable de situation (proposée par le GREPA) qui permet de vérifier que l'ordre de forçage a bien été exécuté ?

Est-ce pure coïncidence, si la structure par encapsulation et la structure par macro-étape conduisent les concepteurs à utiliser les logiciels Automgen de la société IRAI et ceux de chez SCHNEIDER ?

Modes de marche et d'arrêt

La mise et le maintien en production d'un système automatisé de production (SAP) nécessite un certain nombre de procédures appelées modes de marches et modes d'arrêts.

Le Guide d'Étude des Modes de Marches et d'Arrêts (GEMMA) distingue les familles suivantes (fig. 69) :

Procédures de fonctionnement (F) :

- Production normale (F1),
- Marche de préparation (F2) (mise en condition de produire du SAP),
- Marche de clôture (F3) (remise en état de marche du SAP),
- Marche de vérification dans le désordre (F4) (marche manuelle),
- Marche de vérification dans l'ordre (F5) (marche étape par étape, par exemple),
- Marche de tests (F6).

Procédures d'arrêt et de remise en route (A) :

- Arrêt dans état initial (A1) (initialisation de la PC, mise en référence de la PO),
- Arrêt demandé en fin de cycle (A2),
- Arrêt demandé dans un état déterminé (A3),
- Arrêt obtenu (A4),
- Préparation pour remise en route après défaillance (A5),
- Mise PO dans état initial (A6) (mise en référence de la PO),
- Mise PO dans un état déterminé (A7).

Procédures en défaillance de la PO (D) :

- Marche ou arrêt en vue d'assurer la sécurité (D1) (procédure d'arrêt d'urgence),
- Diagnostic et/ou traitement de défaillance (D2),
- Production tout de même (D3).

Procédures relatives à la PC (PZ).

La grille du GEMMA propose tous ces modes parmi lesquels un choix doit être fait compte tenu des souhaits du client et du cahier des charges du SAP.

L'évolution entre les modes retenus doit être conditionnée de manière que le SAP ne puisse se trouver dans deux modes en même temps (respect de l'unicité de mode).

La méthode des « boucles fonctionnelles séparées » facilite la recherche de ces conditions (fig. 71) (*). Le Graphe Descriptif des Modes de Marches et d'Arrêts (GDMMA) regroupe les boucles étudiées séparément (fig. 72) (*).

Et c'est à partir du GDMMA que sont établis le GRAFCET de sécurité (fig. 73) et le GRAFCET de Conduite (fig. 75) (*). En est déduite également la partie câblée de la sécurité (fig. 74).

(*). Consulter l'ouvrage « Le GEMMA » au catalogue des éditions Casteilla.

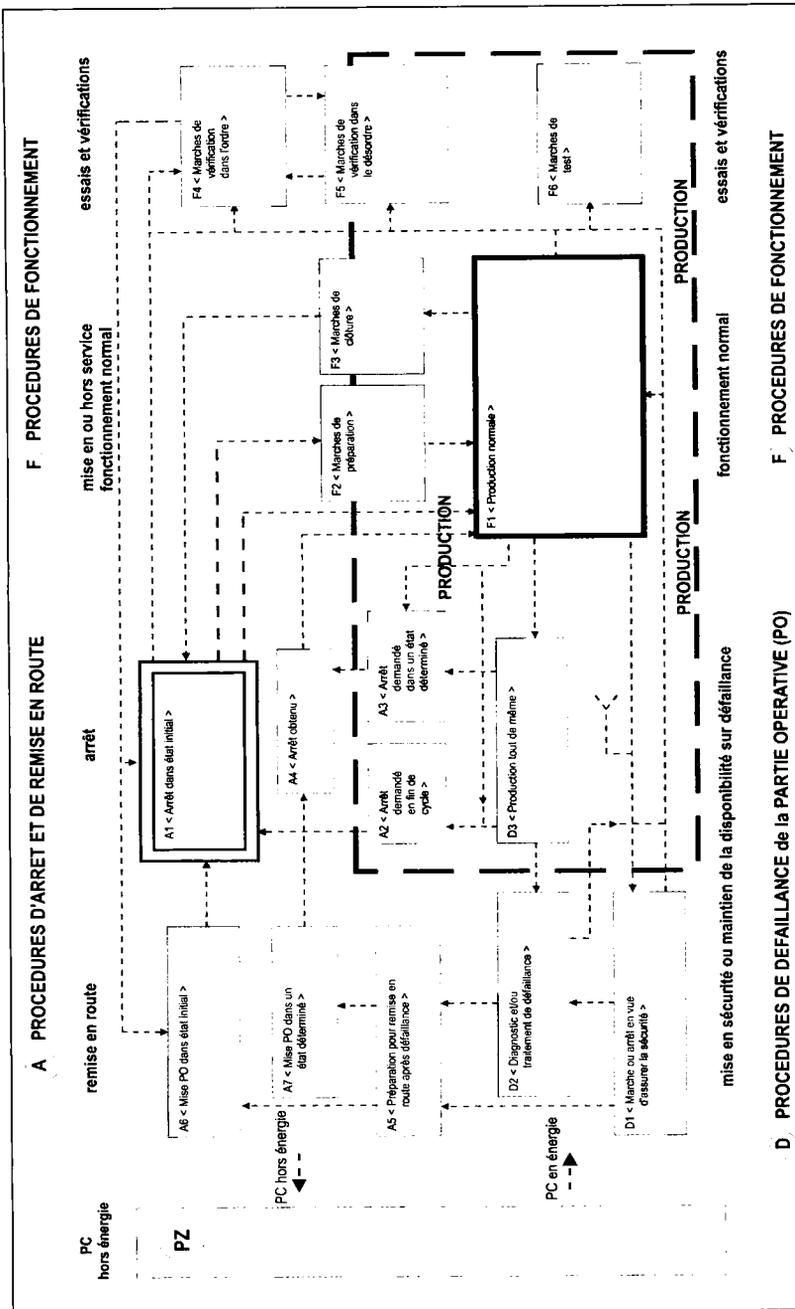


Figure 70
Grille du Guide d'Études des Modes de Marches et d'Arrêts (GEMMA)

Exemple de boucle fonctionnelle séparée

A1 < arrêt dans état initial >

si conditions « ci.EPO.auto.val » vraies
alors aller en F1.

F1 < production normale >

lancement par dcy,
arrêt du cycle par figeage (fig=1)
si conditions « auto.EPO.fig » vraies
alors aller en A3,
si conditions « ci.EPO./modes.val » vraies
alors retour en A1.

A3 < arrêt demandé dans un état déterminé >

figeage des tâches associées à F1,
mise PO hors énergie,
si conditions « auto.fig./EPO » vraies
alors aller en A4.

A4 < arrêt obtenu >

tâches associées à la production normale figées,
remise en énergie PO dès retour souhaité en F1 ou en A6.
(EPO : énergie sur la PO) si conditions « auto.EPO./fig » vraies alors aller en F1,
si conditions « EPO.réf./fig » vraies alors aller en A6.

A6 < mise PO dans état initial >

mise en référence PO si BPref=1,
si conditions « EPO.ci./modes.val » vraies alors aller en A1.

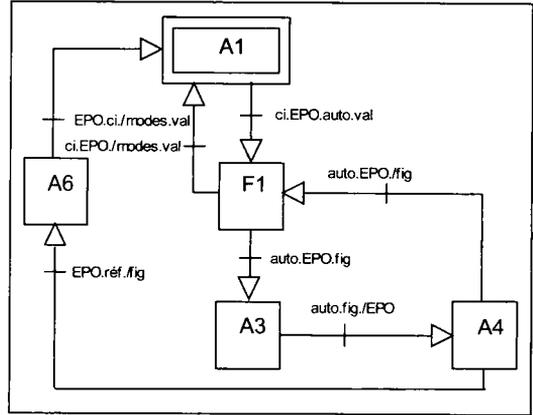
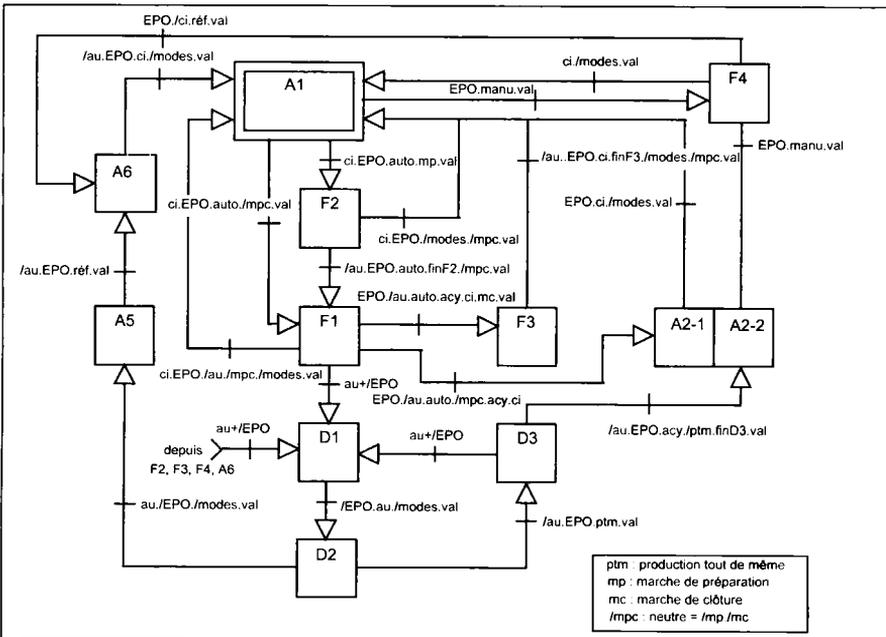


Figure 71

Boucle « production normale avec mise PO dans état initial »



Ce graphe appelé GDMMA est obtenu en regroupant les boucles analysées séparément.

Figure 72

Exemple de graphe descriptif des modes de marches et d'arrêts (GDMMA)

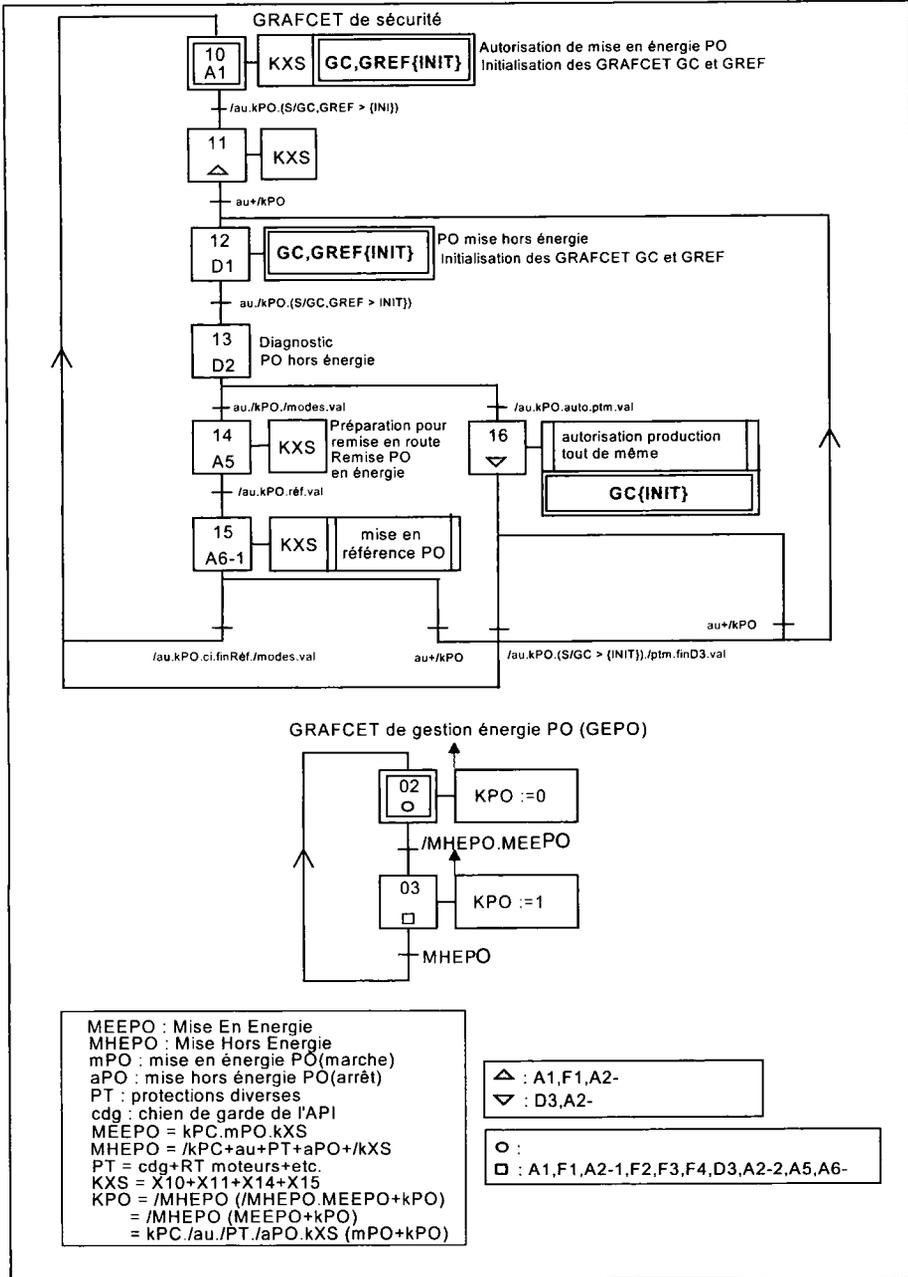


Figure 73

GRAFCET de sécurité (GS) et de gestion de l'énergie (GEPO) déduits du GDMMA, figure 72

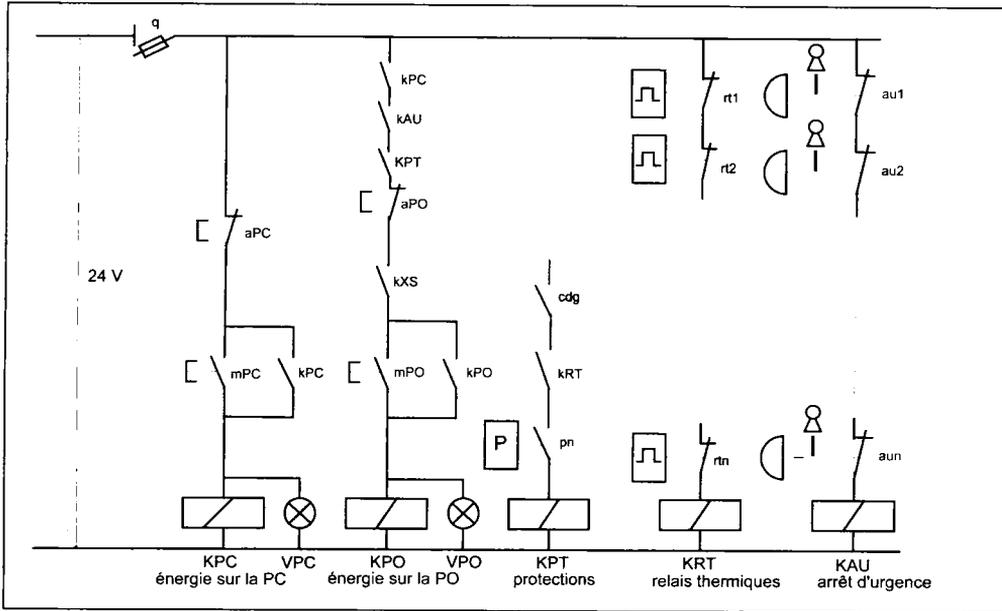


Figure 74

Réalisation câblée de la gestion de la mise en énergie de la PC et de la PO déduite du GDMMA, figure 72

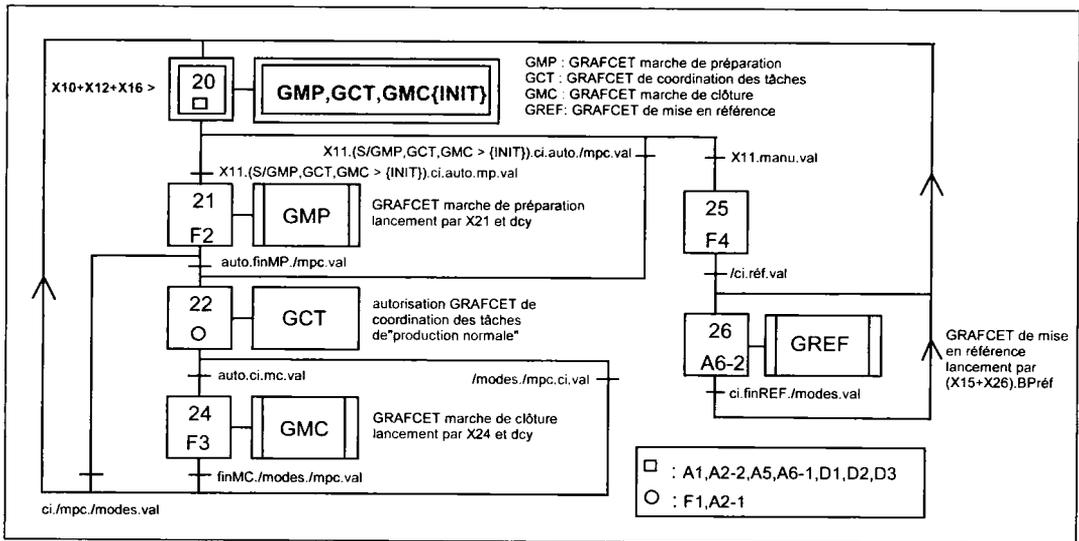


Figure 75

GRAF CET de conduite (GC) déduit du GDMMA, figure 72

Applications particulières des règles d'évolution

Transition et étapes source et puits

Figure 76

Transitions et étapes source et puits

Une transition source est une transition sans étape immédiatement précédente.

Une transition source est toujours validée.

Une transition puits est une transition sans étape immédiatement suivante.

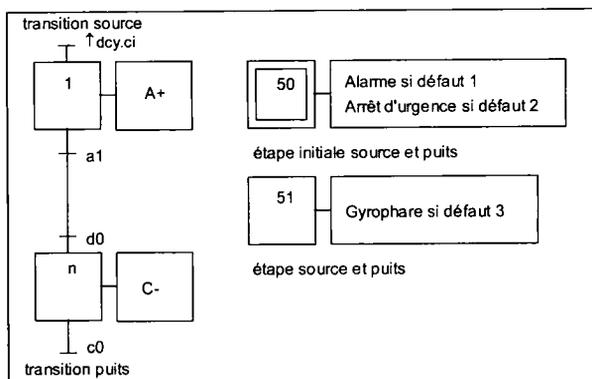
La transition puits conduit le plus souvent à une situation vide.

Une étape peut être à la fois « source » et « puits ». Elle appartient à un GRAFCET

à une seule étape. Une étape « source et puits » peut être initiale (fig. 76).

L'activation d'une étape source ne peut être obtenue que par initialisation ou par forçage à 1.

La désactivation d'une étape puits ne peut être obtenue que par mise en situation vide ou par forçage à 0.



Commentaire

Noter les expressions logiques découlant de ces transitions particulières :

activation étape 1 : $A(1) = \uparrow dcy.ci$, désactivation étape 1 : $D(1) = X1.a1$

désactivation étape n : $D(n) = Xn.c0$

Pour que la désactivation de l'étape 1, lors du franchissement de la transition suivante, soit possible, il faut qu'à cet instant, la réceptivité de la transition source soit fausse. Si elle était vraie, d'après la règle 5, l'étape 1 resterait activée.

La solution la plus sûre consiste à introduire un front montant (ou descendant) dans l'expression de la réceptivité, ici $\uparrow dcy$.

Transition toujours validée

Figure 77

Transition toujours validée

Commentaire

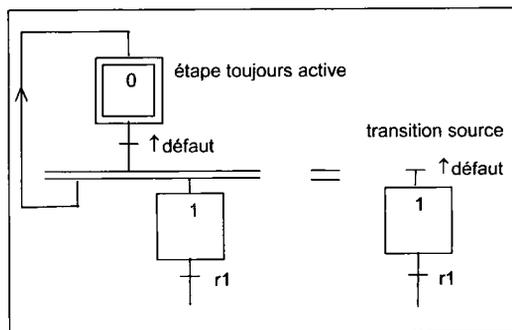
La situation initiale correspond à $S = \{0\}$.

Lorsque la condition de transition « \uparrow défaut » associée à la transition $0 \rightarrow 1$ est vraie, le GRAFCET évolue vers la situation $S = \{0,1\}$.

D'après la règle 5, l'étape 0, à la fois activée et désactivée, reste activée.

Si en cours de fonctionnement, on se trouve dans la situation $S = \{0,n\}$ avec n différent de 1, le franchissement de la transition $0 \rightarrow 1$ mène à la situation $S = \{0,1,n\}$.

On peut supprimer l'étape 0 toujours active. La transition ainsi isolée en tête du GRAFCET est, par définition, une transition source.



Registre à décalage

L'exemple est inspiré d'une chaîne d'enrobage de fromages (fig. 78).

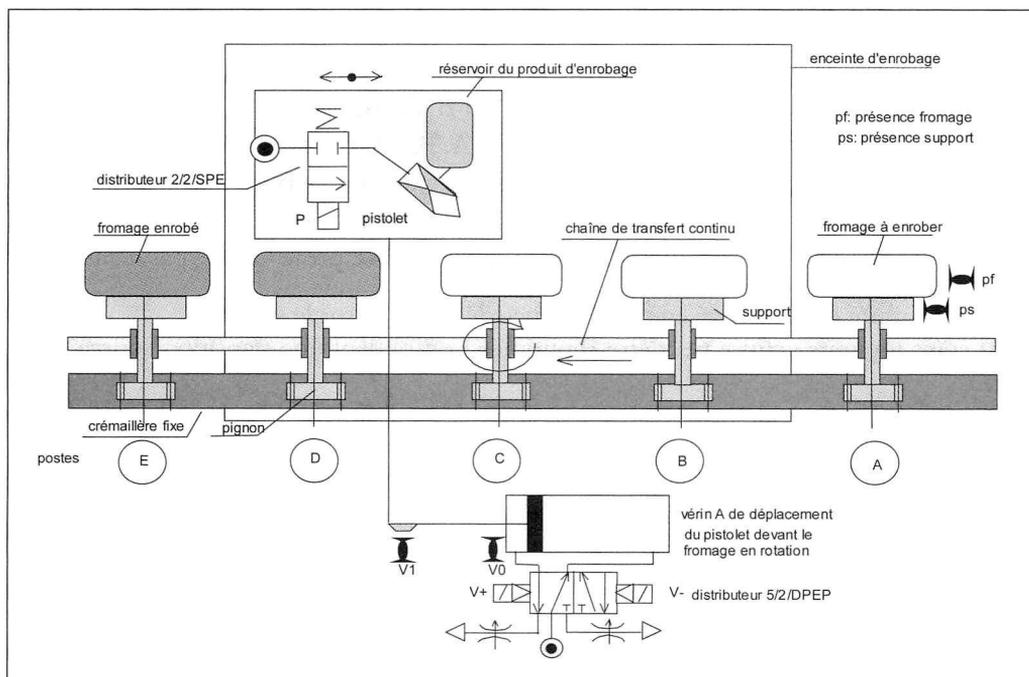


Figure 78

Schéma du poste d'enrobage des fromages

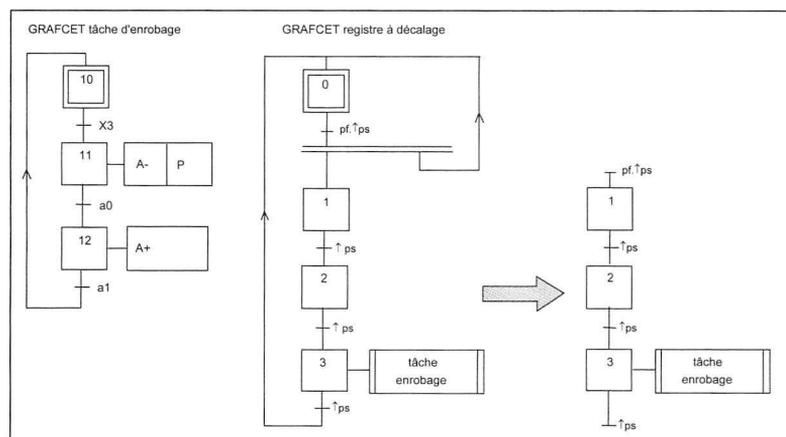


Figure 79

GRAFCET de la tâche d'enrobage et GRAFCET du registre à décalage

La détection directe à l'intérieur de l'enceinte d'enrobage du support vide par une cellule photoélectrique n'est pas conseillée à cause des projections de produit enrobant.

Le registre à décalage permet, ici, de suivre un support portant un fromage jusqu'à son entrée dans l'enceinte d'enrobage. Si le support est vide, le registre n'évolue pas. C'est un moyen indirect de neutraliser le poste d'enrobage.

Étude du GRAFCET du poste d'enrobage des fromages (fig. 79)

- $S = \{0\}$ et pf et $\uparrow ps = 1 \rightarrow S = \{0,1\}$, poste A plein, postes B et C vides, pas d'enrobage
- $S = \{0,1\}$ et pf et $\uparrow ps = 1 \rightarrow S = \{0,1,2\}$, postes A et B pleins, poste C vide, pas d'enrobage
- $S = \{0,1,2\}$ et pf et $\uparrow ps = 1 \rightarrow S = \{0,1,2,3\}$, postes A, B et C pleins, enrobage
- $S = \{0,1,2,3\}$ et $/pf$ et $\uparrow ps = 1 \rightarrow S = \{0,2,3\}$, poste A vide, postes B et C pleins, enrobage
- $S = \{0,2,3\}$ et pf et $\uparrow ps = 1 \rightarrow S = \{0,1,3\}$, postes A et C pleins, poste B vide, enrobage
- $S = \{0,1,2\}$ et pf et $\uparrow ps = 1 \rightarrow S = \{0,1,2\}$, postes A et B pleins, poste C vide, pas d'enrobage.

Le poste de peinture est désactivé en cas d'absence de fromage sur le support.
 L'étape 0 étant toujours activée peut être supprimée : on obtient une transition source avant l'étape 1.

Le registre à décalage se vide progressivement si on cesse d'alimenter la chaîne en fromages.
 Le rebouclage du GRAFCET est inutile : on obtient une transition puits après l'étape 3.

Ici, on applique les règles d'évolution du GRAFCET normalisé notamment la règle 4 : « plusieurs transitions franchissables sont simultanément franchies » et la règle 5 : « une étape à la fois activée et désactivée reste activée ».

Registre à empilage

Des cartons provenant de 2 chaînes de conditionnement arrivent, suivant sélection par $s1$ et $s2$, par les goulottes 1 et/ou 2. Après détection par $p1$ et $p2$, les cartons sont transférés sous le poste de marquage. Une fois marqués les cartons sont évacués, 4 par 4, vers le poste de palettisation.

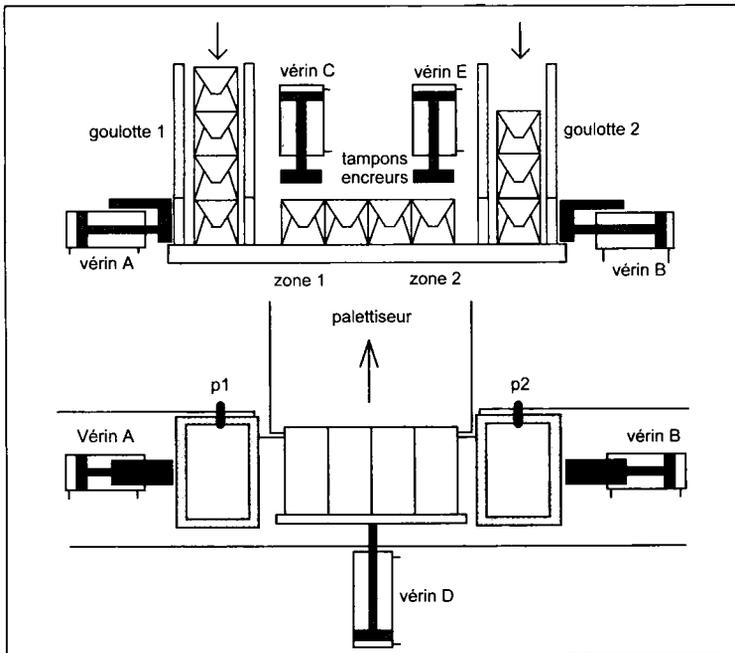


Figure 80
 Poste de marquage de cartons

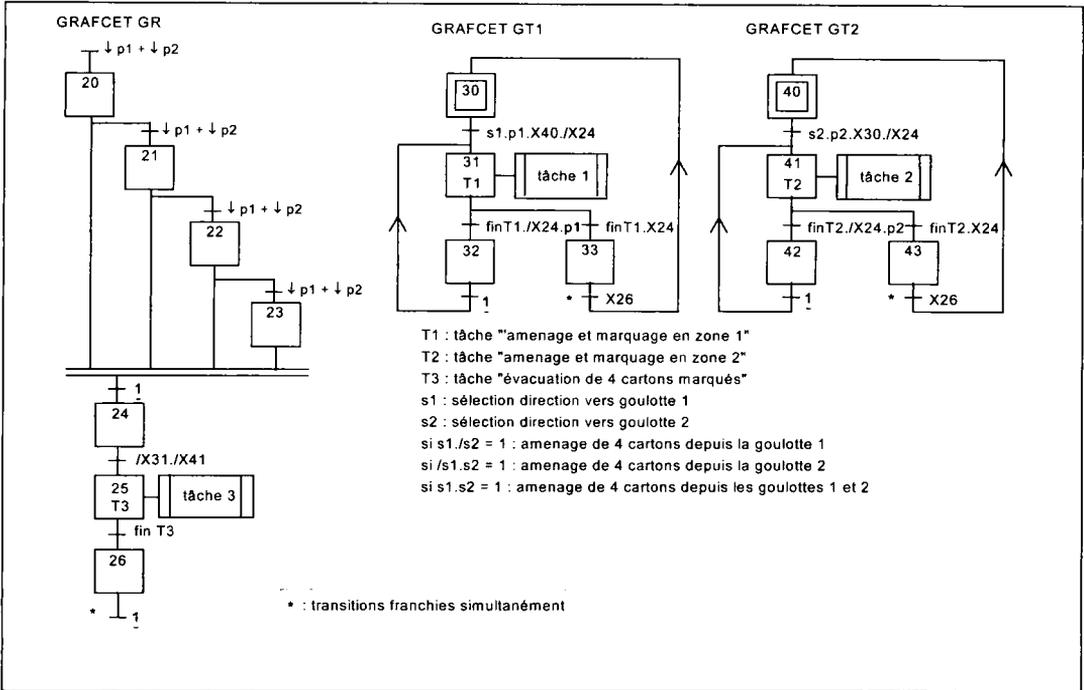


Figure 81
GRAFCET du poste de marquage de cartons

Commentaire

Exécution de la tâche 1 et/ou de la tâche 2,

Situations atteintes :

- après transfert du 1^{er} carton :
 $S(\text{GR}) = \{20\}$,
- après transfert du 2^e carton :
 $S(\text{GR}) = \{20,21\}$,
l'étape 20 activée et réactivée reste donc activée,
- après transfert du 3^e carton :
 $S(\text{GR}) = \{20,21,22\}$,
les étapes 20 et 21 activées et réactivées restent activées,
- après transfert du 4^e carton :
 $S(\text{GR}) = \{20,21,22,23\}$,
les étapes 20, 21 et 22 activées et réactivées restent activées,
- évacuation des 4 cartons marqués.

Les transitions notées « * » sont franchies simultanément.

Registre à déempilage

Après avoir rempli en parallèle une goulotte avec trois objets, ceux-ci sont amenés sur un tapis roulant (information gp). Ensuite les objets sont sortis un par un lorsque le tapis roulant est en marche. L'opération d'évacuation des objets (déempilage) est surveillée par le passage devant un détecteur d'objet (information $\uparrow do$).

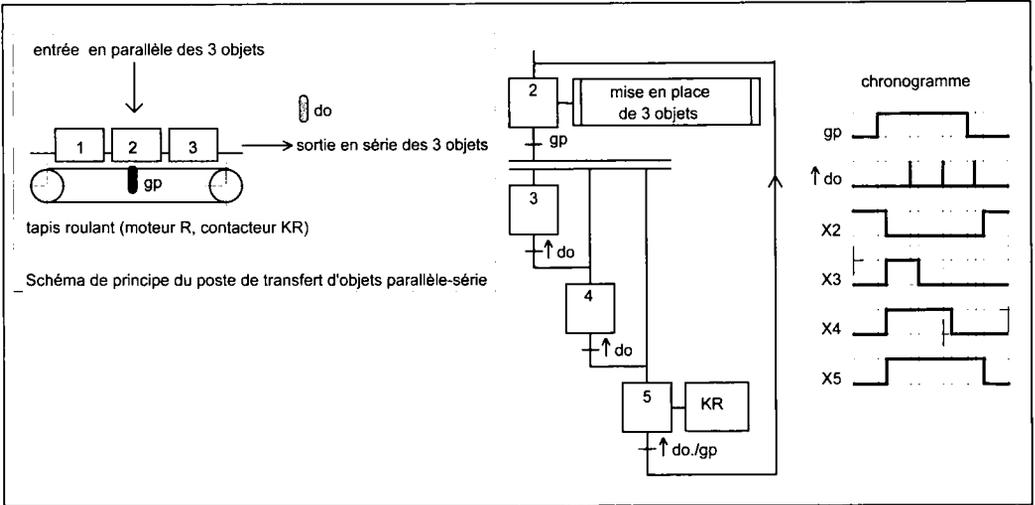


Figure 82
Poste de transfert d'objets parallèle-série

Commentaire

- Si $X2.gp = 1$ alors $S = \{3,4,5\}$, mise en marche du tapis roulant ($KR = 1$).
- Première détection par $\uparrow do$ d'un objet en sortie série :
 $S = \{4,5\}$.
 L'étape 4 activée et désactivée reste activée, il reste 2 objets sur le tapis roulant.
 L'étape 5 activée et réactivée reste activée, l'étape 3 est désactivée,
- Deuxième détection par $\uparrow do$ d'un objet en sortie série :
 $S = \{5\}$.
 L'étape 5 activée et réactivée reste activée, l'étape 4 est désactivée. Il reste un objet sur le tapis roulant.
- Troisième détection par $\uparrow do$ d'un objet en sortie série :
 $S = \{2\}$.
 Le tapis est vide et s'arrête ($KR = 0$).

Postulats sur les durées

Aspect temporel du GRAFCET (documentation UTE C 03-191)

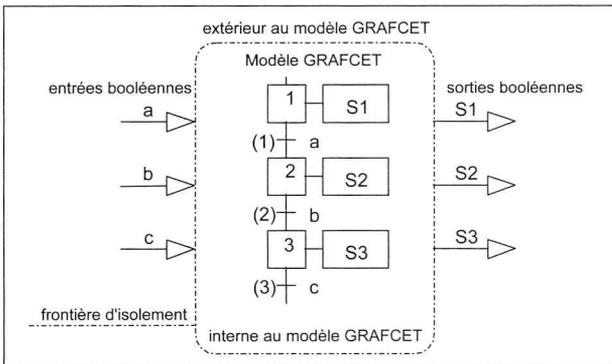
L'isolement d'un système décrit par le GRAFCET établit une frontière de description entre la partie interne et la partie externe au modèle GRAFCET. À chacune des parties est applicable une échelle de temps interne et externe sans commune mesure.

Postulat 1 : À l'échelle de temps externe, tout changement d'état des entrées est pris en compte par le modèle dès son apparition. La totalité des conséquences de cet événement sur le modèle est déterminée à temps nul. Depuis l'extérieur du modèle, les événements d'entrée et les états des sorties qui en résultent sont vus à la même date.

Postulat 2 : À l'échelle de temps interne, la durée séparant l'instant où une transition est franchissable de l'instant où elle est franchie (appelée aussi durée d'évolution) est aussi petite qu'il est nécessaire, mais non nulle. En conséquence la durée d'activité d'une étape ne sera jamais nulle.

À l'échelle du temps interne, il y a synchronisme :

- de l'activation de la ou des étapes aval et de la désactivation de la ou des étapes amont d'une transition franchissable,
- du franchissement de transitions simultanément franchissables (règles d'évolution 2, 3 et 4 du GRAFCET).



Tout GRAFCET est déterministe, c'est-à-dire qu'à toute séquence de variation de ses entrées correspond une séquence unique de variation de ses sorties.

Figure 83
Frontière d'isolement et déterminisme d'un GRAFCET

Application aux situations instables

Une situation $S = \{n\}$ est dite instable (**évolution fugace**) si la condition de transition associée à une transition est déjà vraie (cas notamment d'une réceptivité toujours vraie notée 1) lorsque cette transition est validée (étape n activée).

À l'échelle de temps interne, la durée d'activité de l'étape n est non nulle. Toute action interne au modèle GRAFCET associée à l'étape n pourra être exécutée (voir fig. 83).

Par contre, à l'échelle de temps externe, la durée d'activité de l'étape n peut être considérée comme nulle et en tout cas de durée insuffisante pour commander une action externe au modèle GRAFCET. Cette donnée est à prendre en compte d'un point de vue réalisation.

Application aux transitions et aux étapes (à l'échelle de temps interne)

D'après le postulat 2 :

- **Durée de franchissement d'une transition**
Cette durée peut être considérée aussi petite que l'on veut mais non nulle.
- **Durée d'activation d'une étape**
Cette durée ne sera jamais nulle.

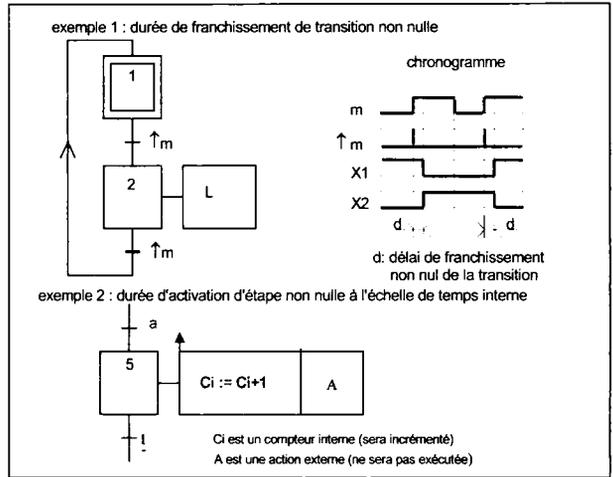


Figure 84
Durées relatives aux évolutions (transition et activité d'étape)

Exemple 1 : la bascule (fig. 84) :

À chaque impulsion sur le bouton poussoir, la lampe L s'allume si elle est éteinte et s'éteint si elle est allumée.

La situation considérée correspond à la situation $S = \{1\}$

La transition $1 \rightarrow 2$ est validée. Lorsque m passe de l'état 0 à l'état 1 (front montant), la transition $1 \rightarrow 2$ franchissable est franchie. Comme la durée de franchissement est non nulle, la transition $2 \rightarrow 1$ n'est pas encore validée.

On se retrouve dans la situation $S = \{2\}$, la transition $2 \rightarrow 1$ n'étant pas franchissable. Il faut une nouvelle impulsion sur m pour revenir à la situation $S = \{1\}$.

Exemple 2 (fig. 84) :

Le postulat fait sur la durée d'activation non nulle de l'étape 5 autorise cette représentation.

L'incréméntation du compteur Ci est obtenue (sortie interne) bien que la transition $5 \rightarrow 6$ soit franchissable dès que l'étape 5 est activée.

Par contre l'action A (sortie externe) ne sera jamais exécutée (échelle de temps interne différente de l'échelle de temps externe).

Réalisation de la partie commande

La conception de la partie commande (PC) des SAP fait appel aux possibilités offertes par les automates programmables industriels (API).

Mais dans certains cas où l'environnement impose ses contraintes (produits explosifs, atmosphère poussiéreuse, humidité, par exemple), si l'automatisme est relativement simple, le séquenceur pneumatique peut être substitué à l'API.

La commande séquentielle des sorties décrite par un GRAFCET est obtenue :

- par programmation avec un API,
- par câblage avec un séquenceur pneumatique.

Architecture d'un automate programmable industriel

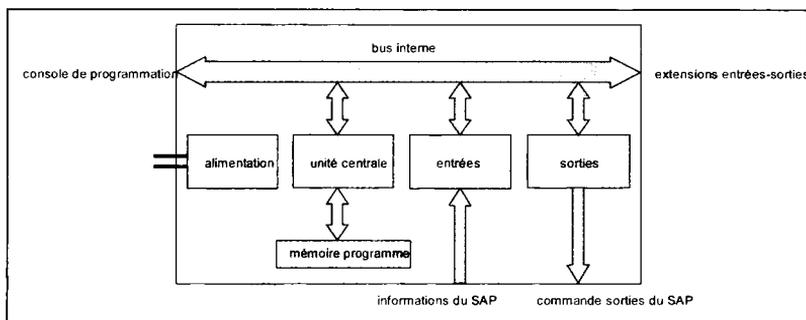


Figure 85

Architecture d'un API

Commentaire

- Le module d'alimentation fournit l'énergie électrique appropriée aux autres modules de l'API.
- L'unité centrale est réalisée autour d'un ou plusieurs processeurs (microprocesseur intégré) et des mémoires mortes ROM à lecture seule (*Read Only Memory*) contenant le programme de gestion interne de l'API (moniteur) et le langage de programmation. Par l'intermédiaire du programme moniteur et du logiciel du langage, elle exécute le programme utilisateur.
- La mémoire programme contient le programme utilisateur écrit par le concepteur. Elle est du type mémoire vive RAM sauvegardée par batterie (*Random Access Memory*) ou du type mémoire morte EEPROM (*Electrically Erasable Programmable ROM*) effaçable électriquement ou du type mémoire morte EPROM (effaçable par exposition aux rayons ultraviolets ou *UV Erasable (aux UV) Programmable ROM*).
- Le ou les modules d'entrées reçoivent les informations fournies par les divers capteurs Tout ou Rien ou TOR du SAP.
- Le ou les modules de sorties envoient les ordres de commandes destinés aux divers préactionneurs du SAP : bobines des distributeurs, des contacteurs, etc.

L'API peut comporter d'autres modules ou cartes spéciales selon le degré de complexité atteint par l'automatisme : cartes d'entrées ou de sorties analogiques, cartes de comptage rapide, cartes de commandes d'axes numériques, cartes de régulation, cartes coupleurs pour le montage en réseau avec d'autres API, ordinateurs, terminaux, superviseurs, variateurs électroniques, etc.

Configuration de programmation d'un automate programmable industriel (API)

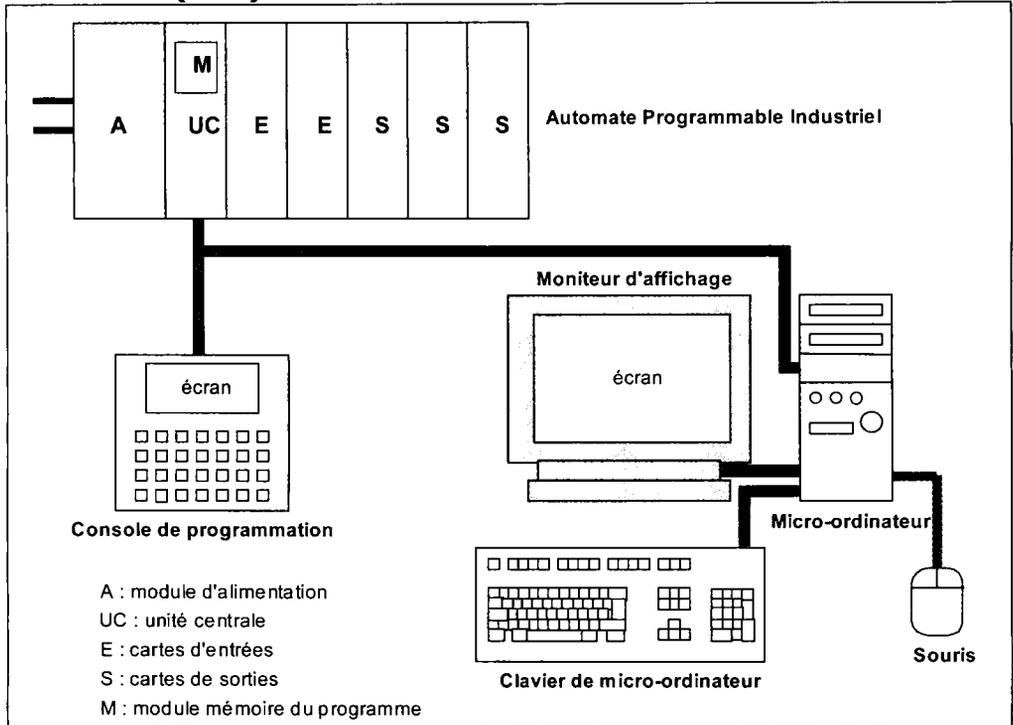


Figure 86

Configuration de programmation d'un API

Les programmes sont entrés dans la mémoire utilisateur par l'intermédiaire d'une console de programmation ou d'un micro-ordinateur émulé en console de programmation à l'aide d'un logiciel spécifique. La programmation peut s'effectuer en ligne (On Line) ou hors ligne (Off Line) avec le micro-ordinateur.

Langages de programmation

Le langage de programmation est un logiciel de conception d'applications. Il permet de transcrire sur l'écran de la console ou du moniteur une suite d'instructions littérales ou graphiques qui seront comprises par l'UC et fourniront une fois traduites, mises en mémoire et traitées les séquences de commande des sorties souhaitées par le concepteur.

On distingue les langages littéraux donc entièrement écrits avec des symboles alphanumériques des langages graphiques qui utilisent à la fois des symboles graphiques et des symboles alphanumériques. Sont dans ce cas les langages graphiques à contacts (*ladder diagram*) et les langages graphiques GRAFCET.

La programmation du GRAFCET, si on ne dispose pas d'un langage littéral ou graphique spécifique, fait appel à des structures d'implantation permettant de concrétiser les concepts du GRAFCET normalisé (fonction mémoire écrite en littéral ou en langage à contacts, utilisation de mots, etc.)*.

(*) Voir « Le GRAFCET, conception et implantation dans les API » aux éditions Casteilla.

Initiation à la programmation du GRAFCET

Programmation en langage GRAFCET graphique (PL7-2 Schneider)

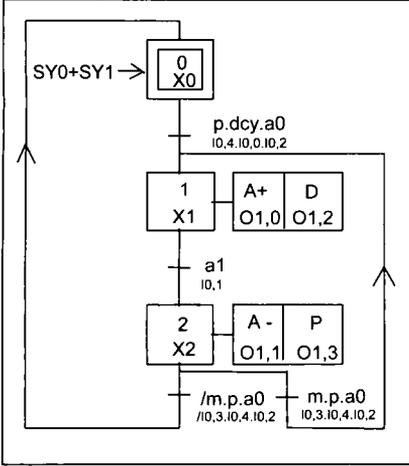


Figure 87
GRAFCET à programmer

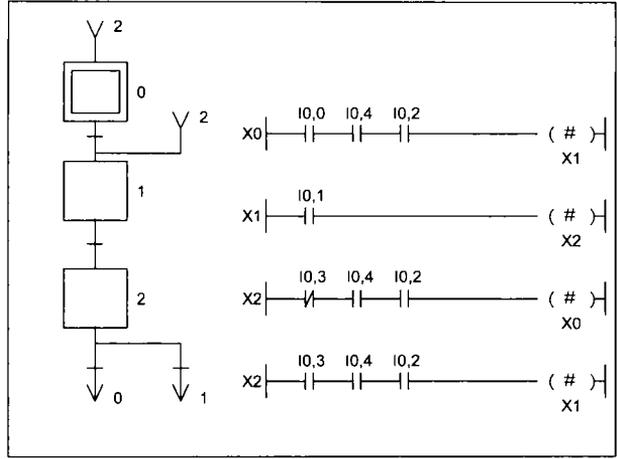


Figure 88
Traitement séquentiel du GRAFCET et des transitions

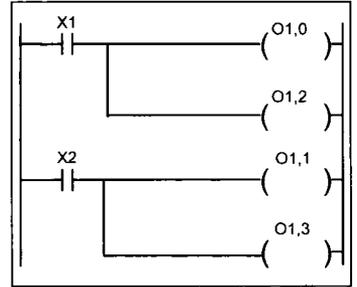
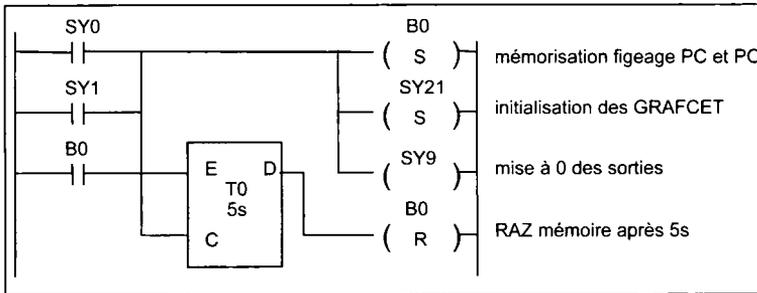


Figure 89
Traitement préliminaire

Figure 90
Traitement postérieur

Le programme est structuré en 3 parties :

- **Le traitement préliminaire** où s'effectue l'initialisation du GRAFCET global (bit SY21), la neutralisation momentanée des sorties (bit SY9) et les forçages de situation (bit SY23).
- **Le traitement séquentiel** où l'on trace un GRAFCET suivant les conventions du langage GRAFCET graphique. Les transitions sont programmées en langage à contacts.
- **Le traitement postérieur** où l'on programme les expressions logiques des sorties, les compteurs, les temporisateurs, etc. en langage à contacts.

L'initialisation du GRAFCET est assurée par les bits système SY0 et SY1.

Programmation en langage à contacts (voir symboles, page 85)

Chaque étape est concrétisée par une mémoire monostable à inscription prioritaire (respect de la règle d'évolution 5) ou par une mémoire bistable obtenue par l'intermédiaire des instructions Set et Reset.

Version monostable

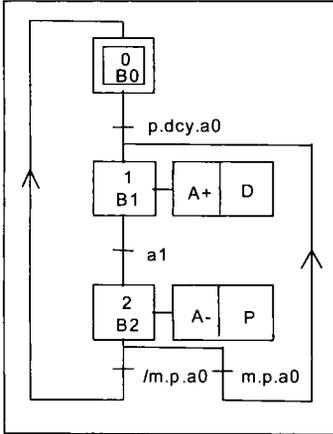


Figure 91
GRAFCET à programmer

La mémoire B0 image de l'étape 0 est initialisée lors du premier cycle par le bit B20 à l'état 0.

Ensuite ce bit est maintenu à l'état 1 dans le réseau 6.

Les fonctions transitions sont calculées et associées aux bits B10 à B13.

Les variables d'entrées et de sorties sont à remplacer par leurs adresses en fonction de l'API et de son langage à contacts.

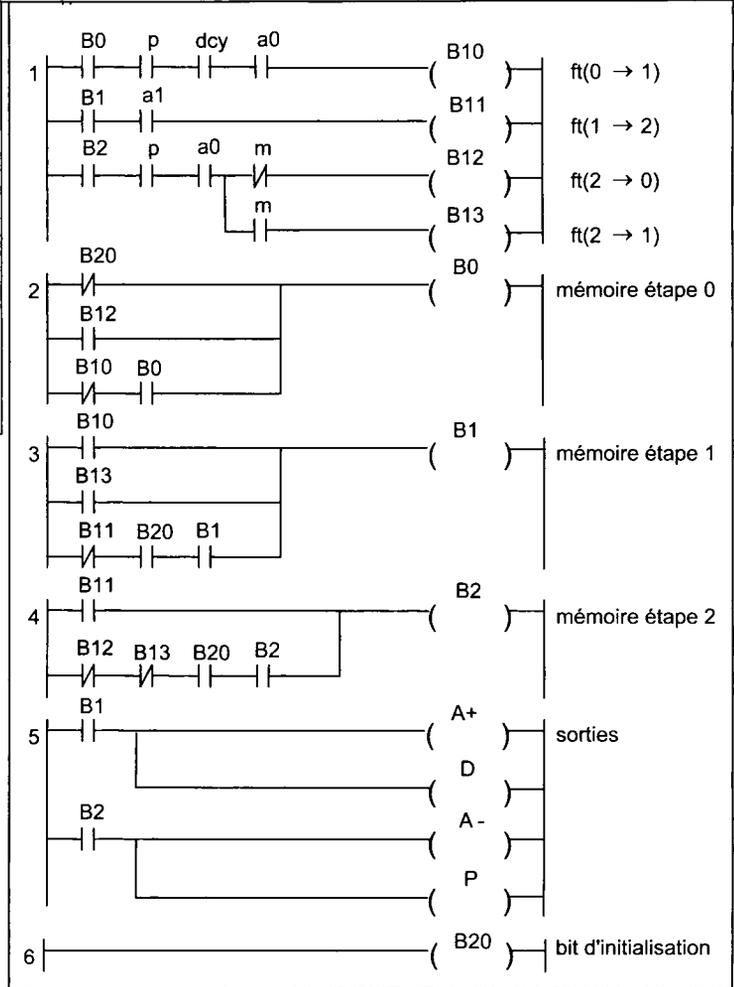
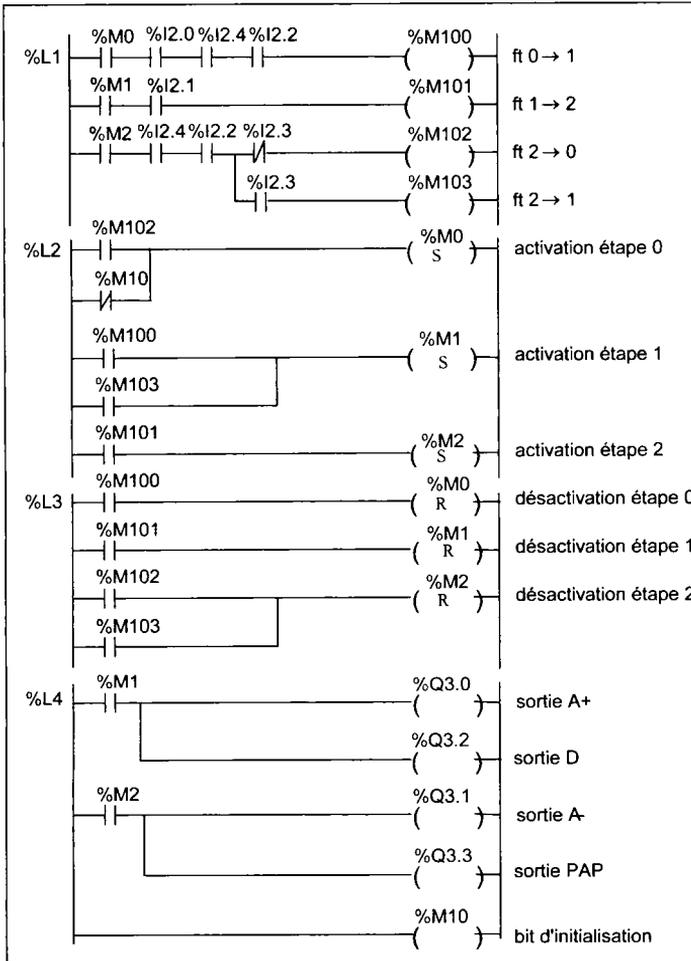
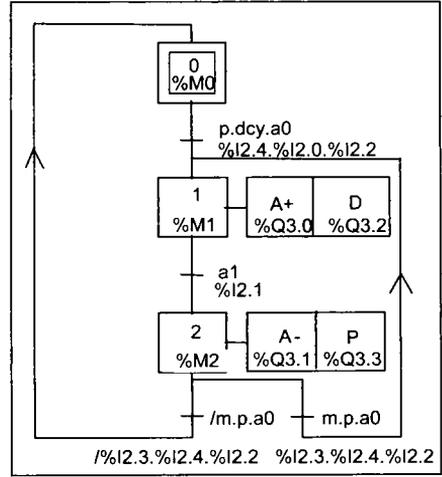


Figure 92
Programme en langage à contacts, version monostable

Figure 93
GRAFSET à programmer



Le langage PL7 Pro Schneider utilise le standard de la norme CEI 61131-3 « Automates programmables - Partie 3 : Langages de programmation ». Ce standard spécifie la syntaxe et la sémantique des éléments logiciels mis en œuvre pour la programmation des API.

L'initialisation est assurée par le bit %M10.

Figure 94
Programme en langage à contacts PL 7 Pro, version bistable

Standard de la norme CEI 61131-3

Ce standard propose la syntaxe et la sémantique des éléments logiciels mis en œuvre pour programmer les automates programmables industriels (API).

Il comporte la description :

- de deux langages textuels dits :
 - *Instruction List (IL)*,
 - *Structured Text (ST)*.
- de deux langages graphiques dits :
 - *Ladder Diagram (LD)*,
 - *Function Block Diagram (FBD)*.
- d'un formalisme graphique dit :
 - *Sequential Function Chart (SFC)*.

Il permet ainsi de structurer l'organisation interne d'une séquence programmée.

Le standard CEI 1131-3 ne définit pas les règles d'interactivité du logiciel fourni par un constructeur se proclamant conforme au standard. Le concepteur du langage, donc, reste libre quant au choix du mode de présentation et de saisie des éléments de programmation.

Ce standard est adopté maintenant par de nombreux constructeurs sur le plan international.

Le tableau ci-dessous présente les principales instructions du langage PL7 Micro Schneider conforme au standard CEI 1131-3 utilisées principalement en langages IL, LD et FBD.

Instructions du logiciel PL7 Micro

DÉSIGNATION	ÉCRITURE
Symboles	
Identification bits et mots	%
Variable interne	M
Valeur immédiate constante	K
Variable système	S
Mot interne M d'un octet B (<i>Byte</i>) (8 bits)	MB
Mot interne M simple W (<i>Word</i>) (16 bits) ou double D (<i>Double</i>) (32 bits)	MW ou MD
Mot flottant F (<i>Floating</i>) (nombre entier ou décimal E ± exposant)	MF
Mot constant K simple W ou double D	KW ou KD
Mot système S simple W ou double D	SW ou SD
Adressage des bits de repère i	
Bit d'entrée I (<i>Input</i>) d'adresse de position de module x et de voie i	%Ix.i
Bit de sortie Q d'adresse de position de module x et de voie i	%Qx.i
Bit interne M	%Mi
Bit système S	%Si
Bit d'étape X	%Xi
Temps T d'activité de l'étape Xi	%Xi.T
Bloc fonction temporisateur de repère i	
Temporisateur TMi	%TMi
Valeur courante V du temporisateur TMi	%TMi.V
Valeur de présélection P du temporisateur TMi	%TMi.P
Bit Q de fin de temporisation	%TMi.Q
Entrée-sortie du bloc fonction Tmi :	
Entrée (<i>Input</i>)	IN
Sortie	Q
Modes retard à l'enclenchement et au déclenchement	TON, TOF
Base de temps	TB

Bloc fonction compteur de repère i

Compteur Ci	%Ci
Valeur courante V du compteur Ci	%Ci.V
Valeur de présélection P du compteur Ci	%Ci.P
Bit D de fin de comptage (<i>Done</i>)	%Ci.D
Bit E du compteur Ci plein (<i>Empty</i>)	%Ci.E
Bit F de débordement (<i>Full</i>) du compteur Ci	%Ci.F
Entrées-sorties du bloc fonction Ci :	
Set (mise à 1), Reset (mise à 0),	S, R
Comptage (<i>Up</i>), décomptage (<i>Down</i>)	CU, CD

Adressage des mots de repère i

Mot d'un octet MBi (<i>Byte</i>)	%MBi
Mot interne M simple W (<i>Word</i>) ou double D (<i>Double</i>)	%MWi ou %MDi
Mot M flottant F (<i>Floating</i>)	%MFi
Mot constant K simple W ou double D	%KW _i ou %KD _i
Mot constant K d'un octet B	%KB _i
Mot constant K flottant F	%KF _i
Mot d'entrée I simple W ou double D d'adresse x,y du module d'entrée	%IW _{x,y} ou %ID _{x,y}
Mot de sortie Q simple W ou double D d'adresse x,y du module de sortie	%QW _{x,y} ou %QD _{x,y}
Mot système S simple W ou double D	%SW _i ou %SD _i

Adressage de bits et de mots de repère i

Bit j du mot variable M simple W	%MWi:Xj
Bit j du mot constant K simple W	%KW _i :Xj

Affectation d'une valeur n à des mots de repère i

Valeur n constante affectée au mot M simple W ou double D	%MWi:=n ou %MDi:=n
Valeur du contenu du mot M simple Wi affectée au mot M simple Wj	%MWi:=%MWj
Valeur du contenu du mot M double Dj affectée au mot M double Di	%MDi:=%MWj
Valeur du contenu du mot constant K simple Wj affectée au mot M simple Wi	%MWi:=%KWj
Valeur du contenu du mot constant K double Dj affectée au mot M double Di	%MDi:=%KDj

Lecture de bits I, Q, M, S de repère i : NOT, RE, FE

Lecture de l'état du bit i	%bit i
Lecture du complément de l'état du bit i	NOT %bit i
Prise en compte du front montant du bit i	RE %bit i
Prise en compte du front descendant du bit i	FE %bit i

Mise à 1 ou à 0 d'un bit de repère i

Mise à un du bit i	SET %bit i
Mise à zéro du bit i	RESET %bit i

Opérations op +, -, *, /, REM sur des mots de repère j, k

Opération arithmétique op entre deux mots simples MWj et MWk affectée au mot simple MWi	%MWi:=%MWj op %MWk
Opération arithmétique op entre deux mots doubles MDj et MDk affectée au mot double MDi	%MDi:=%MDj op %MDk

+ : addition, - : soustraction, * : multiplication, / : division, REM : reste de la division

Opérations logiques op AND, OR, XOR, NOT bit à bit sur mots de repères i, j, k

Opération logique op entre deux mots simples MWj et MWk affectée au mot simple MWi	%MWi:=%MWj op %MWk
Opération logique op entre deux mots doubles MDj et MDk affectée au mot double MDi	%MDi:=%MDj op %MDk
Complémentation du mot simple MWj affectée au mot simple MWi	%MWi:= NOT %MWj
Complémentation du mot double MDj affectée au mot double MDi	%MDi:= NOT %MDj

Opérations op >, <, <=, >=, = <> de comparaison entre mots de repères i, j

Comparaison op entre deux mots simples MWj et MWk	(%MWj op %MWk)
Comparaison entre deux mots doubles MDj et MDk	(%MDj op %MDk)

< : plus petit, > : plus grand, <= : plus petit ou égal, >= : plus grand ou égal, = : égal, <> différent
 Un bit doit être associé au test : si la comparaison est vraie, bit = 1, si la comparaison est fautive, bit = 0

Instructions SHL, SHR, ROL, ROR de décalage sur mots de repères i, j

Décalage de n positions du contenu du mot simple MWj affecté au mot simple MWi	%MWi:= op(%MWj,n)
Décalage de n positions du contenu du mot double MDj affecté au mot double MDi	%MDi:= op(%MDj,n)

SHL : décalage d'un bit à gauche (Shift Left),
 SHR : décalage d'un bit à droite (Shift Right),
 ROL : décalage circulaire d'un bit à gauche (Roll Left),
 ROR : décalage circulaire d'un bit à droite (Roll Right)

Instructions SQRT, ABS sur le mot de repère i

Opération op sur le mot simple Wi	Op:= %MWi
Opération op sur le mot double Di	Op:= %MDi

SQRT : racine carrée, ABS : valeur absolue

Instructions INC, DEC de comptage et de décomptage sur le mot de repère i

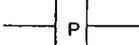
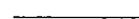
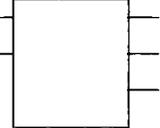
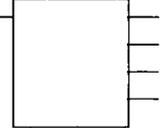
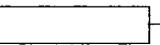
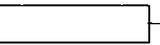
Comptage ou décomptage d'un mot simple Wi	Op %MWi
Comptage ou de décomptage d'un mot double Di	Op %MDi

INC : incrémentation, DEC : décrémentation
 Application : remplacer le symbole op par le symbole de l'opérateur correspondant dans l'écriture des instructions précédentes.
 Exemples : %MW5:= %MW6 + %MW7, %MD10:= %MD11 AND %MD12, %MW7:=%SHL(%MW8,1),
 SQRT:= %MW21, DEC %MW17

Instructions BCD TO INT, INT TO BCD, GRAY, DINT de conversion numérique

Conversion code décimal codé binaire en entier (code binaire pur) d'un mot simple Wj affectée au mot simple Wi	MWi:= BCD TO INT(%MWj)
Conversion code décimal codé binaire en entier (code binaire pur) d'un mot double Dj affectée au mot double Di	MWi:= BCD TO INT(%MWj)
Conversion d'un entier (code binaire pur) en décimal codé binaire d'un mot simple Wj affectée au mot simple Wi	MWi:= INT TO BCD(%MWj)
Conversion d'un entier (code binaire pur) en décimal codé binaire d'un mot double Dj affectée au mot double Di	MWi:= INT TO BCD(%MWj)
Conversion du code Gray en entier (code binaire pur) d'un mot simple Wj affectée au mot simple Wi	MWi:= GRAY TO INT(%MWj)

Principaux symboles graphiques du langage « Ladder Diagram » du logiciel PL7 PRO

Désignation	Graphes	Fonctions
Éléments de test		
Contact à fermeture		Contact passant quand l'objet bit qui le pilote est à l'état 1.
Contact à ouverture		Contact passant quand l'objet bit qui le pilote est à l'état 0.
Contacts à détection de changement d'état		Front montant : détection du passage de 0 à 1 de l'objet bit qui le pilote.
		Front descendant : détection du passage de 1 à 0 de l'objet bit qui le pilote.
Éléments de liaison		
Connexion horizontale		Pour relier en série les éléments de test et d'action.
Connexion verticale		Pour relier en parallèle les éléments de test et d'action.
Éléments d'action		
Bobine directe		L'objet bit associé prend la valeur du résultat de la zone de test.
Bobine inverse		L'objet bit associé prend la valeur inverse du résultat de la zone de test.
Bobine d'enclenchement		L'objet bit associé est mis à 1 lorsque le résultat de la zone de test est à 1.
Bobine de déclenchement		L'objet bit associé est mis à 0 lorsque le résultat de la zone de test est à 1.
Blocs fonctions		
Élément de test		
Temporisateur Compteur Monostable Registre Programmeur		Chacun des blocs fonctions standards utilise des entrées et des sorties permettant de les relier aux autres éléments graphiques. Les entrées et les sorties ainsi que leur nombre diffèrent suivant le bloc fonction considéré.
Blocs opérations		
Élément de test		
Bloc comparaison vertical		Pour comparer le contenu de deux opérands. Suivant le résultat, la sortie correspondante passe à 1.
Bloc comparaison horizontal		Pour comparer le contenu de deux opérands. Suivant le résultat, la sortie correspondante passe à 1.
Élément d'action		
Bloc opération		Pour réaliser les opérations arithmétiques, logiques, de décalage, de comptage, de conversion, etc. Fait appel à l'écriture du langage littéral (voir pages 83 et 84).

Implantation du GRAFCET sur mots en langage à contacts (Ladder Diagram)

Chaque étape est associée au contenu d'un mot, ici le mot W0.

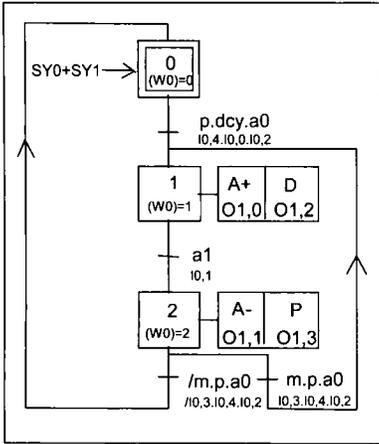


Figure 95
GRAFCET à programmer

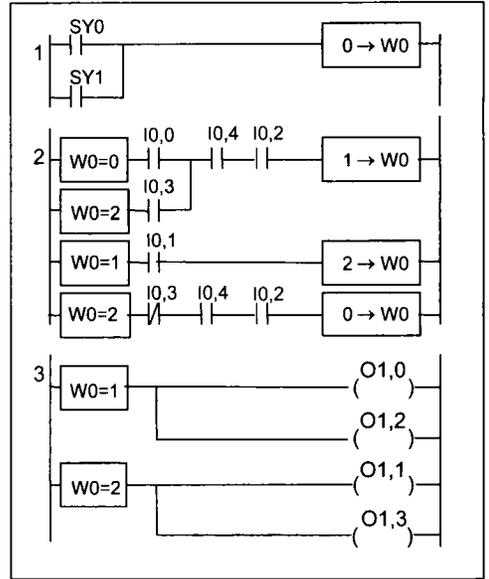


Figure 96
Programme en langage calcul à contacts (PL7-2 et PL7-3)

Figure 97

Programme en langage calcul littéral (PL 7-3)

```
! IF SY0+SY1
    THEN 0 → W0
! IF IO,4.IO,2([W0=0].I,0+[W0=2].IO,3)
    THEN 1 → W0
! IF [W0 = 1].IO,1
    THEN 2 → W0
! IF [W0=2].NOT IO,3.IO,4.IO,2
    THEN 0 → W0
! [W0 = 1] → O1,0
! [W0 = 2] → O1,1
! [W0 = 1] → O1,2
! [W0 = 2] → O1,3
```

L'initialisation est assurée par les bits systèmes SY0 et SY1.

Un mot doit être associé à chaque GRAFCET à étape active unique ou à chaque branche de GRAFCET à étape active unique.

Chaque mot, à la limite, permet de programmer 9999 étapes.

La règle d'évolution 3 (synchronisme de l'activation et de la désactivation des étapes) est parfaitement respectée.

Cette méthode est actuellement utilisée pour programmer les fonctions annexes des commandes numériques NUM 1020, 1040 et 1060.

Le GRAFCET et les séquenceurs

Le séquenceur permet de concrétiser un automatisme analysé par GRAFCET moyennant quelques entorses aux règles d'évolution du GRAFCET normalisé.

Il est réalisé à l'aide de composants dits « modules d'étapes » accolés les uns aux autres. Le nombre de modules d'étapes utilisés dépend du nombre d'étapes du GRAFCET.

Chaque module matérialise une fonction mémoire. Dans la version pneumatique, le module d'étape est composé d'un distributeur associé à un opérateur logique ET pour le traitement de la fonction transition d'activation et d'un opérateur logique OU pour le traitement de la désactivation du module précédent par le module suivant ou par la RAZ générale (fig. 98).

Remarque : Les séquenceurs électriques réalisés à l'aide de relais électromagnétiques bistables ne sont plus utilisés depuis l'avènement des automates programmables industriels (API).

Les séquenceurs peuvent se monter en parallèle pour réaliser des séquences exclusives ou simultanées.

L'effacement de la mémoire du module précédent par la sortie du module suivant n'est pas conforme aux règles d'évolution 2, 3 et 4 du GRAFCET normalisé. Ce n'est pas un inconvénient sauf qu'il est impossible de réaliser des boucles à deux étapes avec certains séquenceurs.

Conception du séquenceur pneumatique PARKER-PNEUMATIC

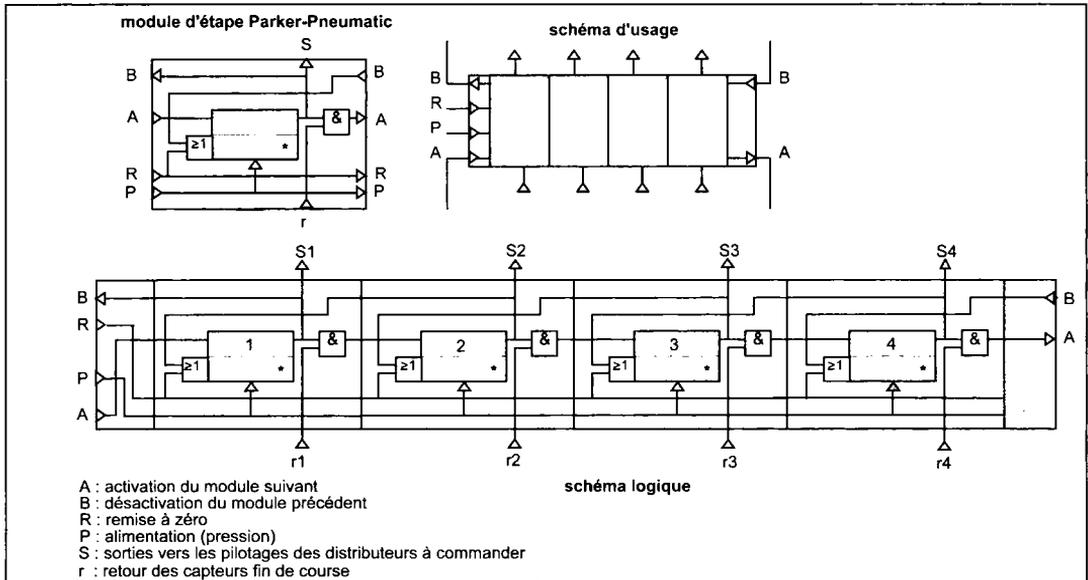


Figure 98 - Séquenceur pneumatique PARKER PNEUMATIC

Commentaire

Le module d'étape est composé d'un opérateur mémoire à effacement prioritaire, d'un opérateur ET pour le calcul de la fonction transition et d'un opérateur OU pour la désexcitation de l'étape précédente.

Le séquenceur est obtenu par accolement des embases des modules d'étape et des modules d'entrée et de sortie pour pouvoir effectuer les liaisons d'extrémités par tubes (fig. 98).

Un module de dérivation s'interposant entre deux modules d'étapes autorise le branchement entre deux modules ou entre deux séquenceurs.

Application

Commentaire

La boucle à deux étapes 1-2 du GRAFCET original réalisable avec deux modules d'étapes est instable. En effet, par construction, le signal X2 intervient à la fois dans la désactivation et l'activation de l'étape 1. Il en est de même du signal X1 vis-à-vis de l'étape 2.

Une étape, donc un module d'étape supplémentaire, doit être installée entre l'étape 1 et l'étape 2 ou comme ici entre l'étape 2 et l'étape 1.

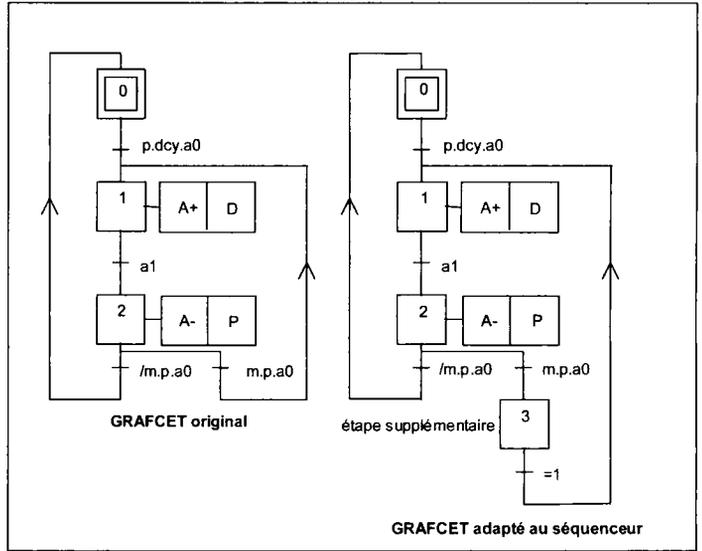


Figure 99

GRAFCET original et GRAFCET adapté séquenceur Parker Pneumatic

Schéma de réalisation avec un séquenceur pneumatique Parker-Pneumatic

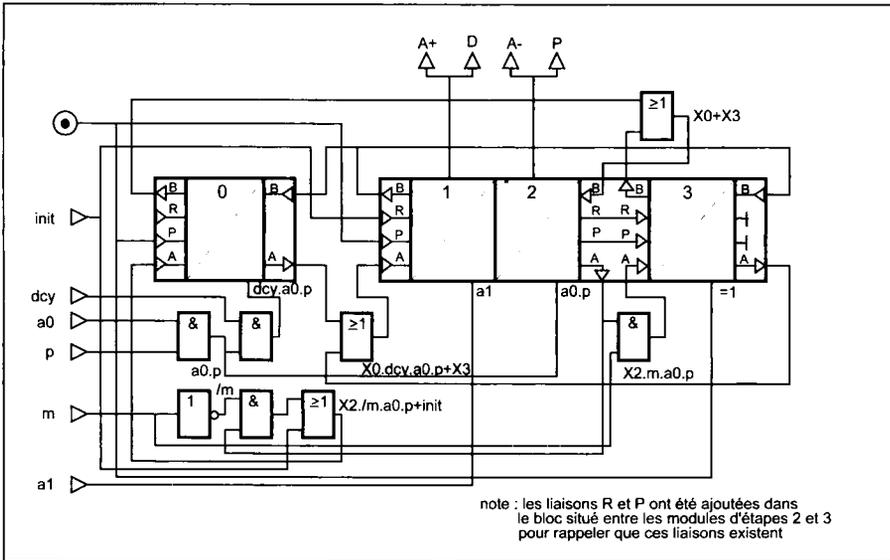


Figure 100

Schéma de réalisation du séquenceur Parker Pneumatic

Traitement logique des réceptivités et des actions

Le traitement logique des réceptivités et des actions (sorties) fait appel à des opérateurs logiques combinatoires OU, ET, IN, NON et OUI (fig. 101).

Noter que les opérateurs logiques combinatoires pneumatiques n'ont que deux entrées. Il s'ensuit que pour effectuer des opérations logiques à plus de deux variables, plusieurs opérateurs sont nécessaires.

Les opérateurs OUI et NON acceptent un signal de pression plus faible en E1 qu'en E2. Ce sont des amplificateurs utilisés également, si besoin, pour régénérer un signal.

L'opérateur NON est aussi utilisé comme capteur fin de course des vérins sous les appellations « capteur à seuil de pression » et « capteur à chute de pression ».

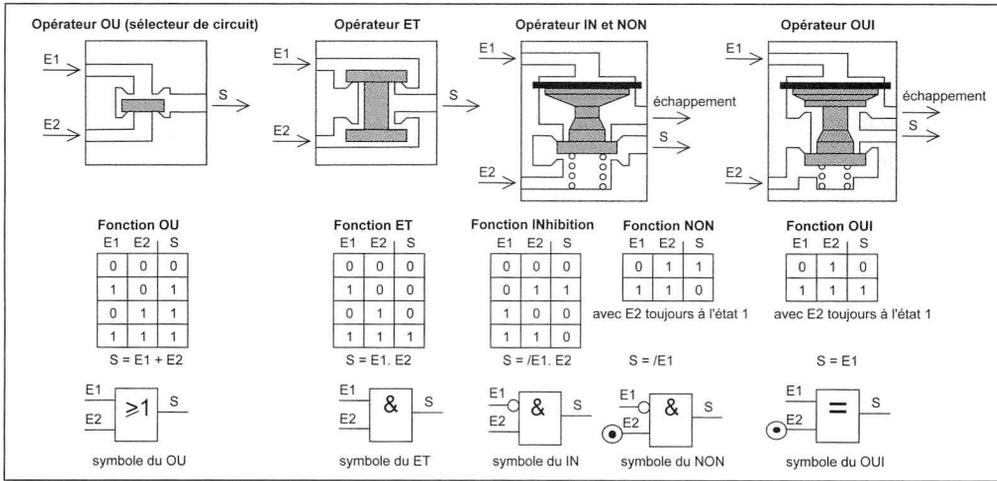


Figure 101 - Opérateurs logiques combinatoires pneumatiques OU, ET, IN, NON et OUI

Pour simplifier la réalisation et surtout réduire le nombre de tubes à raccorder, les opérateurs logiques du système Parker Pneumatic se présentent sous forme de cellules associables.

Des sélecteurs à deux positions permettent de choisir entre le montage en cascade et le montage avec commun d'entrée (fig. 102).

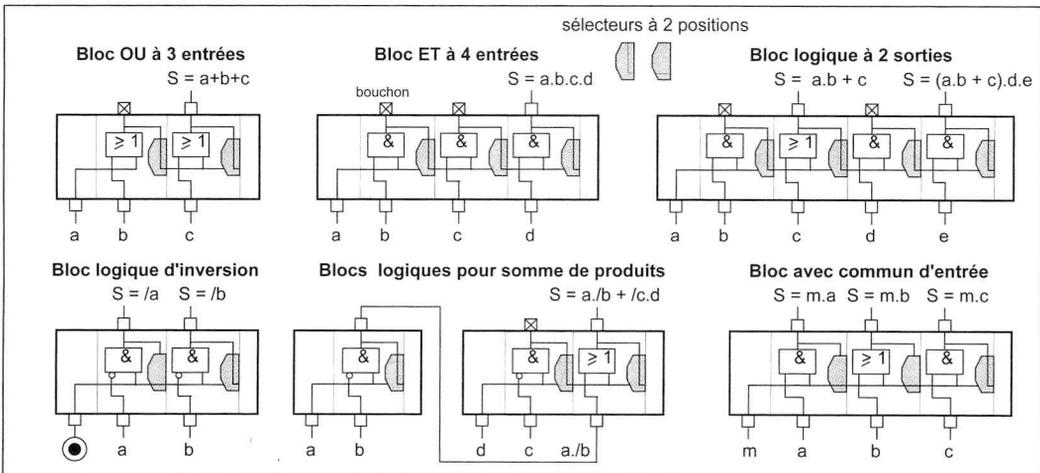


Figure 102 - Système logique Parker Pneumatic

Conception du séquenceur Joucomatic (origine Crouzet)

Composants du registre séquenceur

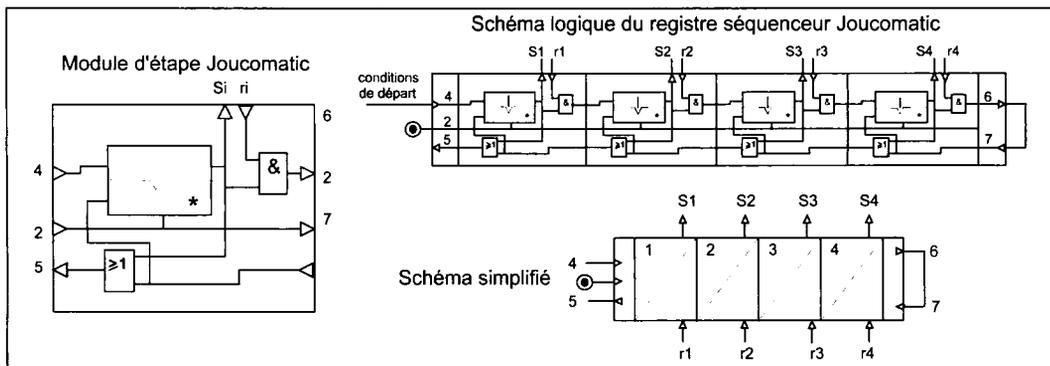


Figure 103 - Réalisation d'un registre séquenceur Joucomatic

Commentaire

Le module d'étape est composé d'un opérateur mémoire à effacement prioritaire, d'un opérateur ET ou IN, au choix, pour le calcul de la fonction transition et d'un opérateur **OU pour la désexcitation de toutes les étapes précédentes**. L'opérateur ET ou IN accessible est monté sur le dessus du module d'étape. Le séquenceur est obtenu par accollement des embases des modules d'étape contenant l'opérateur OU et des modules d'entrée et de sortie pour pouvoir effectuer les liaisons d'extrémités par tubes (fig. 103). Un bloc de dérivation permet de raccorder deux séquenceurs.

Le traitement logique des réceptivités et des actions fait appel à des opérateurs logiques OU, ET, OUI et NON ainsi qu'à des composants périphériques : temporisateurs, impulsurs, etc. se montant sur embases accolables avec sélecteur de liaison série ou parallèle.

Application

Figure 104

GRAF CET original et GRAF CET adapté au séquenceur Joucomatic

Commentaire

Les étapes initiales d'un GRAF CET ne sont pas concrétisables avec ce séquenceur. Elles resteront fictives. La transition du premier module est ainsi toujours validée : c'est « **une transition source** ». Les opérateurs OU étant montés en cascade, le franchissement d'une transition entraîne l'effacement de la mémoire immédiatement précédente ainsi que celles situées en amont. Cet effacement permanent des mémoires est une sécurité prévue pour éviter un dysfonctionnement du séquenceur. **Noter qu'il est possible de réaliser un GRAF CET à 2 étapes mais pas une boucle sur 2 étapes**. La dernière transition n'ayant pas d'étape suivante est par définition « **une transition puits** ».

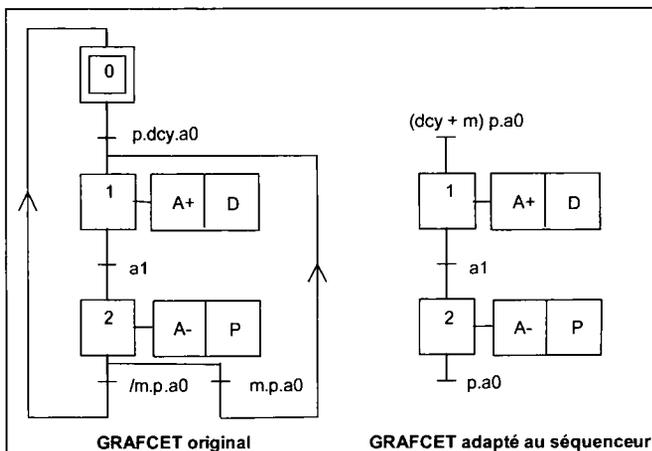


Schéma de réalisation avec un séquenceur Joucomatic

Figure 105

*Schéma de réalisation
du registre séquenceur Joucomatic*

Commentaire

Le séquenceur est composé de deux modules d'étape, l'étape 0 devenant une étape fictive (fig. 106).

Si $p.dcy.a0 = 1$, alors activation du module 1 (étape 1 active).

Les actions A+ et D sont lancées.

En fin d'exécution de A+, si $a1 = 1$, alors activation du module 2 (étape 2 active) puis désactivation de l'étape 1 (étape 1 non active).

Les actions A- et P sont lancées.

En fin d'exécution de A-, $a0 = 1$.

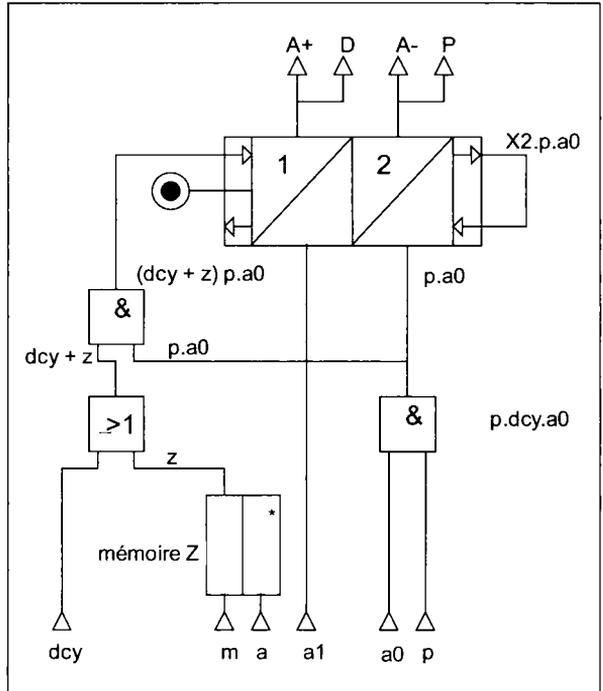
Si $p.a0 = 1$, alors désactivation du module 2 (étape 2 désactivée) et confirmation de la désactivation du module 1.

Si $m = 0$, le séquenceur reste désactivé.

Si $m = 1$, alors activation du module 1 (étape 1 active). Le cycle se poursuit et se relance automatiquement tant que $m = 1$.

Le sélecteur M étant du type bistable doit être remplacé par un bouton-poussoir monostable pour éviter un démarrage intempestif hors contrôle de l'opérateur (directive 93-40, art. R. 233-18 du 11/1/93 du Code du Travail).

Un opérateur mémoire externe Z résout le problème : $Z = /a (m + z)$. Un bouton-poussoir d'arrêt A permet d'effacer la mémoire Z.



Séquenceur VE Step Atlas Copco

Composants du séquenceur

Pour réaliser un séquenceur VE Step, on dispose :

- de modules VE Step matérialisant les étapes sauf la dernière du cycle,
- de modules VE Stop matérialisant la dernière étape du cycle,
- de modules d'entrée et de sortie du séquenceur,
- de modules de dérivation dans le cas de GRAFCET comportant des sélections de séquences.

Des modules demi-pas VE- 1/2 Step ne comportant qu'un opérateur ET sont utilisables pour réduire le nombre de modules VE Step dont le coût est plus élevé.

Les modules VE Step et VE Stop comportent :

- un distributeur 5/2 à double pilotage particulier dont les deux sorties sont à l'échappement à l'état repos,
- un opérateur IN pour interdire la mise à 1 du 5/2 si le capteur fin de course du mouvement à exécuter est toujours actionné (sécurité),
- un opérateur OU pour mettre le 5/2 à l'état 0 par un ordre local ou général.

La position différente des opérateurs OU permet d'initialiser le séquenceur en activant le module VE Stop et en désactivant tous les modules VE Step.

Un témoin de pression de couleur verte indique dans quel état se trouve la sortie.

Le traitement des signaux d'entrée et de sortie fait appel à des cellules logiques montées sur embases.

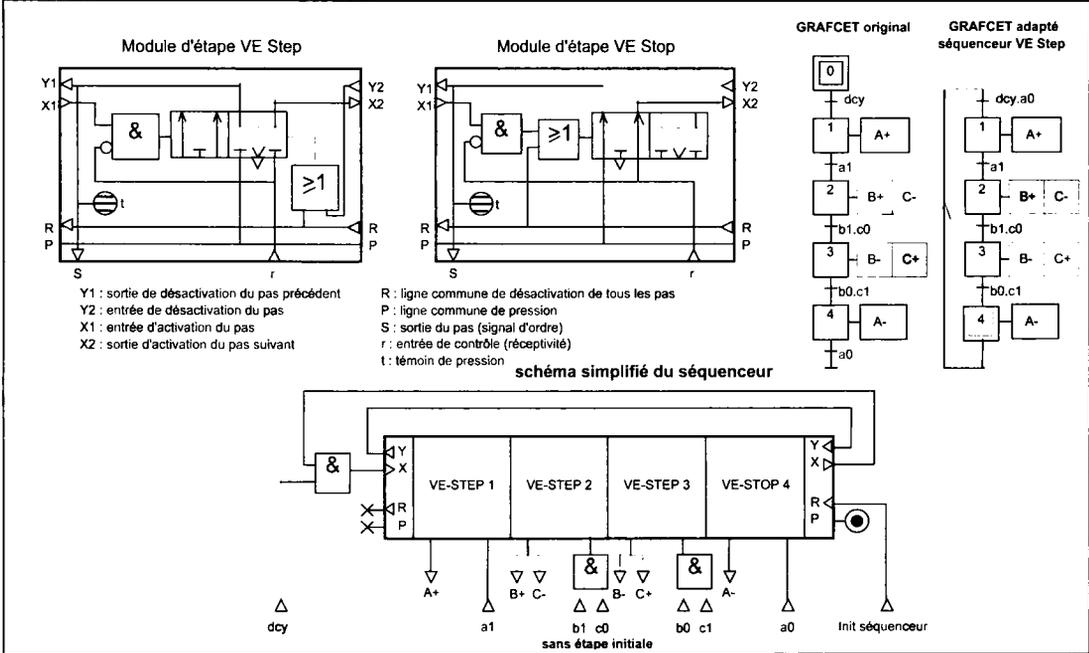


Figure 106
Séquenceur VE STEP Atlas Copco

Application

Commentaire

Les étapes 1 et 2 vont être matérialisées par des modules VE STEP tandis que l'étape 0 le sera avec un module VE STOP.

Les conditions de fonctionnement en cycle par cycle avec dcy et en cycle automatique continu avec m sont regroupées au niveau de la transition 0 → 1.

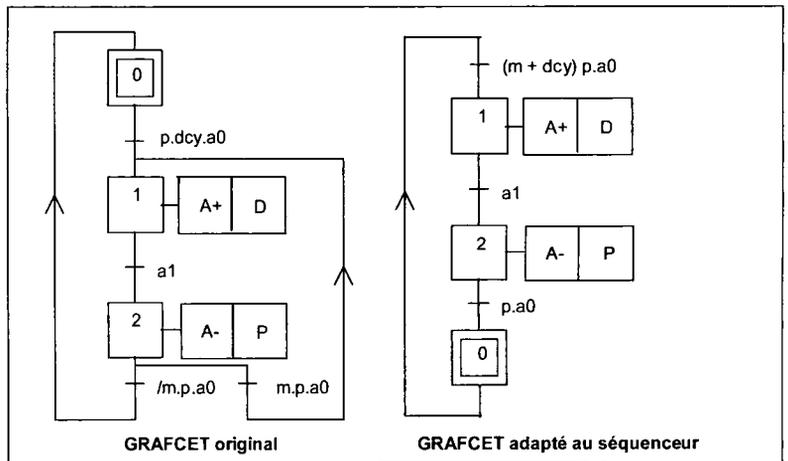


Figure 107
GRAFCET original et GRAFCET adapté au séquenceur VE STEP Atlas Copco

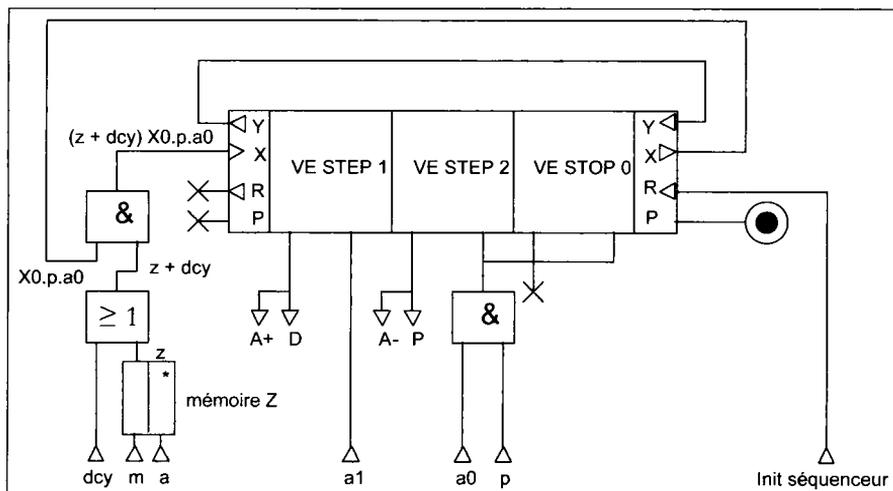


Figure 108
Schéma de réalisation avec un séquenceur VE STEP Atlas Copco

Commentaire

État repos du séquenceur : les trois modules d'étape sont désactivés. L'entrée R du module VE STOP 0 permet d'activer le module 0 et de désactiver les modules 1 et 2 par les voies Y.

Si $p.dcy.a0 = 1$, alors activation du module 1 et désactivation du module 0 par la voie Y.

Les actions A+ et D sont lancées.

En fin d'exécution $a1 = 1$, alors activation du module 2 et désactivation du module 1.

Les actions A- et P sont lancées.

En fin d'exécution, $p.a0 = 1$, alors activation du module 0 et désactivation du module 2.

Si $m = 1$, alors le cycle recommence et se poursuit tant que $m = 1$.

Le sélecteur M étant du type bistable doit être remplacé par un bouton-poussoir monostable pour éviter un démarrage intempestif hors contrôle de l'opérateur (directive 93-40, art. R. 233-18 du 11/1/93 du Code du Travail).

Un opérateur mémoire externe Z résout le problème : $Z = /a (m + z)$. Un bouton-poussoir d'arrêt A permet d'effacer la mémoire Z.

Séquenceur Quickstepper Festo

Le séquenceur Quickstepper est un dispositif pas à pas pneumomécanique de 12 pas.

Le pas suivant ne peut être activé que si le pas précédent est actif et si l'action déclenchée par le pas précédent est terminée. Un seul pas est en pression à la fois, les autres pas sont à l'échappement.

Le séquenceur permet ainsi de matérialiser une séquence unique de GRAFCET de 12 étapes.

Le nombre de pas peut être augmenté en montant plusieurs séquenceurs en série ou en parallèle.

Les pas non utilisés sont court-circuités. Les branchements pneumatiques sont prévus en face arrière du boîtier.

La face avant porte des commandes permettant de passer en mode test pas à pas (bouton Man/step), de bloquer ou non les sorties (sélecteur Output) et de visualiser le pas actif et le signal retour associé à ce pas.

La logique d'entrée et de sortie est traitée à l'aide de cellules logiques ET, OU et IN montées sur socle à canules et regroupables sur un châssis.

Principaux opérateurs logiques combinatoires et séquentiels

(normes NF E 49-600, E 49-602, C 03-103)

Opérateurs logiques combinatoires			Opérateurs logiques séquentiels																	
Désignation	Symbole	Expression logique	Désignation	Symbole	Expression logique															
Opérateur OU (OR) (OU exclusif)		$S = e1 + e2$ <table border="1"> <tr><td>e1</td><td>e2</td><td>S</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	e1	e2	S	0	0	0	0	1	1	1	0	1	1	1	1	Mémoire à inscription prioritaire (IP)		$X = I + /E.x$ I : signal d'inscription E : signal d'effacement
e1	e2	S																		
0	0	0																		
0	1	1																		
1	0	1																		
1	1	1																		
Opérateur ET (AND)		$S = e1.e2$ <table border="1"> <tr><td>e1</td><td>e2</td><td>S</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	e1	e2	S	0	0	0	0	1	0	1	0	0	1	1	1	Mémoire à effacement prioritaire (EP)		$X = /E (I + x)$
e1	e2	S																		
0	0	0																		
0	1	0																		
1	0	0																		
1	1	1																		
Opérateur IN Opérateur NON		$S = e1./e2$ $S = /e2$ si $e1 = 1$ <table border="1"> <tr><td>e1</td><td>e2</td><td>S</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	e1	e2	S	0	0	0	0	1	0	1	0	1	1	1	0	Mémoire à entrées simultanées passives (ESP)		$X = I./E + (I + /E) x$
e1	e2	S																		
0	0	0																		
0	1	0																		
1	0	1																		
1	1	0																		
Opérateur XOR (OU inclusif)		$S = e1./e2 + e1.e2$ <table border="1"> <tr><td>e1</td><td>e2</td><td>S</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	e1	e2	S	0	0	0	0	1	1	1	0	1	1	1	0	Mémoire à entrées simultanées incompatibles (ESI) (mémoire RS)		$Q = S + /R.q$ (IP) $Q = /R (S + q)$ (EP)
e1	e2	S																		
0	0	0																		
0	1	1																		
1	0	1																		
1	1	0																		
Opérateur OUI (amplificateur)		$S = e$ <table border="1"> <tr><td>e</td><td>S</td></tr> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </table>	e	S	0	0	1	1	Mémoire à entrée dynamique (bascule RS)		$Qt+1 = f (^G, Qt, R, S)$ ^G = front montant de G									
e	S																			
0	0																			
1	1																			
Opérateur NON (amplificateur)		$S = /e$ <table border="1"> <tr><td>e</td><td>S</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	e	S	0	1	1	0	Mémoire à entrée dynamique (bascule JK)		$Qt+1 = f (vG, Qt, J, K)$ vG : front descendant de G									
e	S																			
0	1																			
1	0																			
Opérateur NON OU (NOR)		$S = /(e1 + e2)$ $= /e1 ./e2$ <table border="1"> <tr><td>e1</td><td>e2</td><td>S</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	e1	e2	S	0	0	1	0	1	0	1	0	0	1	1	0	Mémoire à entrée dynamique (bascule D)		$Qt+1 = f (^G, Qt, D)$
e1	e2	S																		
0	0	1																		
0	1	0																		
1	0	0																		
1	1	0																		
Opérateur NON ET (NAND)		$S = /(e1.e2)$ $= /e1 + /e2$ <table border="1"> <tr><td>e1</td><td>e2</td><td>S</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	e1	e2	S	0	0	1	0	1	1	1	0	1	1	1	0	Bascule monostable convertisseur d'échelon (en impulsion)		
e1	e2	S																		
0	0	1																		
0	1	1																		
1	0	1																		
1	1	0																		
Symboles complémentaires pour les opérateurs séquentiels			Opérateurs à retard																	
	Se met dans la case appropriée pour indiquer la priorité en cas de mise à l'état 1 simultanées des entrées.		Désignation	Symbole	Nature des retards															
	Se met dans la case appropriée pour indiquer l'état préférentiel lors de remise en énergie.		Opérateurs à retard (exemples)		même retard t de S dans les transitions /X - X et X -/X															
	Indique le maintien, lors de la remise en énergie, de l'état de la sortie obtenu avant mise hors énergie.				retard t égal et réglable pour les deux transitions															
	Indique une entrée dynamique sensible aux impulsions.				retards t1, t2 différents retards t1, t2 différents pour chaque transition															
Exemples			Opérateurs à retard (exemples)																	
* 			Mémoire à effacement prioritaire à effacement préférentiel lors de la mise ou remise en énergie.																	
			Mémoire à effacement prioritaire avec maintien, lors de la remise en énergie, de l'état de la sortie obtenu avant mise hors énergie.																	

Index

A

Action conditionnelle	41-42-47
Actions continues	41
Actions mémorisées	43-44-47
Action à durée limitée	42
Action retardée	42
Action continue	41
Action impulsionnelle	43
Action maintenue	44-47
Action mémorisée	44-47
Action ponctuelle	43
Adressage	82-83
Aiguillage	30-31-32
Affectation	43
Analyse descendante	34-37-48-52-53-58
Arrêt d'urgence	10-19-65
Assembleur	48
Assignment	41
Automate programmable	7-10-56-77-78-81-82-87
Automgen	64

B

Base de temps	83
BCD	84
Bit interne	82
Bit système	62-82
Bloc fonction	82-83
Boucle à deux étapes	87-88-90
Boucle fonctionnelle séparée	65-67
Byte	82-83

C

Capacité mémoire	36
Capteur fin de course	19-29-89-90
Capteur machine	10-34
Capteur opérateur	10-35-62
Chronogramme	21
Composant de signalisation	10-35
Composant de visualisation	10-35
Compteur	11-12-19-31-43
Condition de transition	8-14-18-19-20-23-27-28-30-33-51-54-70-75
Console	20-56-78
Contacteur	10-39-62-77

Coordination entre GRAFCET	35-50
Coordination de tâches	35-37-53-56
Cycle automatique	92
Cycle par cycle	92

D

Décompteur	19-43
Décrémentation	43-84
Distributeur	10-39-46-62-77-87-91
Drapeau	31-32

E

Encapsulation	48
EEPROM	77
Etape d'attente	25-27-30-32-50
Etape encapsulable	48
Etape initiale	8-11-50-54-59
Etape initialisable	59
Evolution asynchrone	14
Evolution synchrone	14
Expansion	48-54-55-56

F

Figeage de situation	61-62-64
Flag	32
Fonction mémoire	46-78-87
Fonction transition	14-21-22-23-25-80-87-88-90
Forçage de situation	7-11-57-58-59-60-62
Franchissement de transition	8-14-17-18-19-21-27-32-51-54-56-57-70-75-76-90
Front descendant	19-20-43-47-70-83
Front montant	19-20-43-47-70-76-83
Fugace	23

G

GEMMA	35-52-65
GRAFCET	7
GRAFCET connexe	26
GRAFCET de Sécurité	65-68
GRAFCET de Conduite	63-65-69
GRAFCET partiel	26
GRAFCET de Production	35-62-63-64

I

Incrémentation	31-43-76-84
Initialisation de la PC	8-11-12-13-60-79-81-86

L

Ladder diagram	78-82-85-86
Langage à contact	79-80-85-86
Langage GRAFCET graphique	79
Langage logigramme	19
Liaisons orientées	8-18-24-25

M

Macro-représentation	48-49-50-52-56
Macro-étape	7-48-54-55-56
Mémoire	46-56-77-78-80
Mécanisme	10-34-40-62
Mémoire programme	77
Mémoire vive	7
Mémoire morte	77
Mise en référence	65-67
Modes de marches	19-35-37-65-66-67
Mot	19-78-82-83-84-86

O

Octet	82-83
Opérateur logique	20-87
Opérateur mémoire	44-46-88-90-91-93

P

Parallélisme interprété	33
Parallélisme structural	27-28-33-44-45
Partie commande (PC)	10-11-19-34-77
Partie opérative (PO)	10-34
Partie relation (PR)	10-35
Pas à pas	93
Point de vue procédé	38
Point de vue processus	34-36-37-38-39
Prédicat	19
Position de référence	11
Présélection	19-43-82-83
Procédure	11-32-35-37-65
Procédure d'arrêt d'urgence	65
Programme	32-48-77-78-80-81-82

R

RAM	77
RAZ	11-31-87
Réceptivité	8-19-20-89

Registre à décalage	71-72
Registre à désempilage	74
Registre à empilage	72
Réinitialisation du GRAFCET	57-59
Reprise de séquence	31
ROM	77

S

Saut d'étape	30
Sélection de séquence	29
Séquence répétitive	50-52-56
Séquences exclusives	29-30-87
Séquences inclusives	33
Séquences simultanées	27-87
Séquence unique	26-33
Séquenceur	37-87
Séquenceur Atlas Copco	91-92
Séquenceur Festo	93
Séquenceur Joucomatic	90-91
Séquenceur Parker	87-88
SFC	7-82
Signalisation	10-35
Situation	8-14-19
Situation courante	57-58-61-62-64
Situation initiale	8-9-11-12-13-59-70
Situation forcée	57-58-59-60
Situation vide	11-26-56-58-59-64-70
Sous-programme	35-48-49-50-51-52-53-56
Synchronisme	75-86
Syntaxe	81-82

T

Technoguides	36
Temporisateur	19-43-79-82
Transition	8-14-18-19
Transition puits	70
Transition source	70
Transition toujours vraie	19-23-27-51-70

U

Unicité de mode	65
-----------------	----

V

Variable de situation	60
-----------------------	----