

S. Moreno

Professeur certifié
de génie mécanique
spécialité "automatique et régulation"

E. Peulot

Professeur technique honoraire
agrégé de génie mécanique
spécialité "automatique"

Le GRAFCET

Conception - Implantation

dans les Automates

Programmables Industriels

Bac STI - STS - IUT - IUFM - IUP

Ecoles d'ingénieurs - Formation continue

Collection
A.Capliez



Editions Casteilla - 25, rue Monge - 75005 Paris

Chez le même éditeur

LE GEMMA

Modes de marches et d'arrêts – GRAFCET de coordination des tâches – Conception des systèmes automatisés sûrs.

S. Moreno, E. Peulot.

LE GRAFCET

Sa pratique et ses applications

J.-C. Bossy, P. Brard, P. Faugère, C. Merlaud.

AUTOMATISME APPLIQUÉ

J.-C. Bossy, D. Mérat.

LA PNEUMATIQUE DANS LES SYSTÈMES AUTOMATISÉS DE PRODUCTION.

S. Moreno, E. Peulot.



Le logo qui figure ci-contre mérite une explication. Son objet est d'alerter le lecteur sur la menace que représente pour l'avenir de l'écrit, tout particulièrement dans le domaine de l'enseignement, le développement massif du photocopillage.

Le code de la propriété intellectuelle du 1^{er} juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée dans les établissements scolaires, provoquant une baisse brutale des achats de livres, au point que la possibilité même pour les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée.

Nous rappelons donc que toute reproduction, partielle ou totale, du présent ouvrage est interdite sans autorisation de l'auteur, de son éditeur ou du Centre français d'exploitation du droit de copie (CFC 20, rue des Grands-Augustins, 75006 Paris).

ISBN : 2.7135.2371.0

© Casteilla, 2002 - 25, rue Monge, 75005 PARIS

Toute représentation, traduction, adaptation ou reproduction, même partielle, par tous procédés, en tous pays, faite sans autorisation préalable est illicite et exposerait le contre-venant à des poursuites judiciaires. Réf. Loi du 11 mars 1957.

SOMMAIRE

Introduction : bref historique	1
I. Premier contact avec le GRAFCET	3
1. Problème posé	3
2. Découpage de l'automatisme en trois parties	4
3. Situation de la machine dans les ateliers de l'entreprise	5
4. Analyse descendante de l'automatisme	6
II. Description d'un système automatisé de production (SAP)	21
1. Cahier des charges	21
2. Comment aborder la description d'un système automatisé ?	22
III. Modèle GRAFCET	29
1. Définition	29
2. Étape	29
3. Actions associées à l'étape	31
4. Transitions entre les étapes	33
5. Réceptivités associées aux transitions	34
6. Liaisons orientées	38
7. Règles de syntaxe	39
8. Règles d'évolution	39
9. Hypothèses sur les durées	46
10. Structures de base	55
11. Étude des actions ou des ordres	60
12. Macro-représentations	76
13. Forçages et figeages de situation	88
IV. Comment établir un GRAFCET correct ?	102
1. Erreurs possibles	102
2. Erreurs de syntaxe	102
3. Erreurs de forme	102
4. Erreurs de forme à corriger	104
5. Applications	106
V. Simplification du graphisme d'un GRAFCET	108
1. Recherche des transitions redondantes	108
2. Recherche des étapes redondantes	108
3. Recherche des étapes fusionnables	110
4. Conclusion sur la simplification du graphisme du GRAFCET	111
5. GRAFCET à deux étapes	111
6. Éclatement d'un GRAFCET complexe en GRAFCET simples	113
VI. Matérialisation des concepts de base du GRAFCET	116
1. Étape	116
2. Transition et réceptivité associée	116
3. Matérialisation des règles d'évolution	117
4. Réalisation des fonctions mémoire et transition	120
5. Tableau récapitulatif des mémoires matérialisant les étapes dans les divers types de structures du GRAFCET	121
VII. Structures d'implantation du GRAFCET dans les API	123
1. Quel est le problème ?	123
2. Structure basée sur la fonction mémoire	124
3. Structure informatique ou algorithmique	134
4. Structure exploitant les registres pas à pas	141
5. Structure exploitant le langage calcul sur mots	147
6. Structure exploitant les langages GRAFCET littéraux	150
7. Structure exploitant les langages GRAFCET graphiques	153
8. Conclusion sur les structures	159
9. Tableau récapitulatif des essais faits sur divers API	161
10. Intégration des forçages de situation dans les structures d'implantation du GRAFCET	162
VIII. Logiciels de Programmation du GRAFCET Assistée par Ordinateur (PAGO)	182
1. Logiciel CAPEDA de la société EURILOR	182
2. Atelier d'automatique Alograf de la société UXP	187
3. Logiciel AUTOMGEN de la société IRAI	190
IX. Application	195
1. Présentation du poste à automatiser	195
2. Analyse descendante du problème	196
X. Symbolique propre au GRAFCET, à l'organigramme et au logigramme	220
1. Structures graphiques du GRAFCET	220
2. Étapes	221
3. Macro-représentations	221
4. Ordres ou actions associés à l'étape	221
5. Transitions et réceptivités associées	222
6. Forçages de situation	223
7. Figeages de situation	223
8. Forçage à 0 des sorties et figeage de la PO	223
9. Symboles pour organigrammes (extrait NF Z 67-010)	224
10. Structures d'implantation algorithmique du GRAFCET	224
11. Symboles pour logigrammes (extraits NF C 03-312 et NF E 49-602)	225
XI. Autres méthodes d'analyse et de synthèse des automatismes	226
1. Analyse combinatoire	226
2. Méthode matricielle (méthode d'Huffman)	226
3. Description par chronogramme	227
4. Diagramme des phases	228
5. Méthode en cascade	228
6. Séquenceurs	231
7. Organiphase de la Télémécanique	233
Terminologie propre au GRAFCET et aux structures d'implantation dans les automates programmables industriels	234
Index	248
Bibliographie	252

Bref historique du GRAFCET

L'automatisation des installations constitue un des facteurs essentiels contribuant à l'amélioration de la productivité. Le progrès des techniques et de la technologie en général permet d'envisager des systèmes automatisés de plus en plus complexes. Encore faut-il disposer de méthodes d'analyse et de synthèse appropriées. Ce n'était pas le cas, en ce qui concerne les automatismes séquentiels, des méthodes proposées jusqu'aux alentours de l'année 1960.

Voici un aperçu de ces méthodes :

- **le tableau d'analyse associé aux tableaux de Karnaugh*** destiné à l'analyse des problèmes de logique combinatoire ;
- **le chronogramme*** toujours actuel et qui convient parfaitement pour décrire graphiquement les séquences liées au temps (feux de carrefour, cycle de machine à laver, etc.) ;
- **le diagramme de phases*** a été pratiqué par de grandes sociétés comme Renault. Cette méthode dérive du chronogramme. Elle nécessite une grande expérience professionnelle et sa mise en œuvre manque parfois de rigueur ;
- **la méthode d'Huffman ou méthode matricielle*** introduite en France par l'ingénieur Naslin. Cette méthode enseignée dans les établissements techniques s'est révélée très formatrice par le raisonnement logique qu'elle impose. Mais elle n'a pas trouvé de débouchés dans l'industrie, car elle est trop lourde à manipuler dès que le nombre de variables d'entrées est supérieur à six. Les tableaux d'analyse et les tableaux de Karnaugh à la base de cette méthode deviennent alors vite impraticables ;
- **le logigramme** qui est plus une méthode de synthèse qu'une méthode d'analyse. Ce genre de description s'appuie sur la norme représentative des opérateurs logiques. Des composants existent en technologies électronique et pneumatique permettant de matérialiser ces divers opérateurs logiques ;
- **l'organigramme** utilise les algorithmes de type informatique mais est mal adapté pour décrire les séquences d'automatisme à évolutions parallèles. Il peut par contre être associé au GRAFCET pour décrire des séquences où le calcul est prédominant ;
- **la technique dite de pas-à-pas** facile à mettre en œuvre mais trop souvent liée à un composant particulier (sélecteur rotatif pas à pas électromagnétique type téléphonique, programmateur pas à pas à came ou à carte perforée, électrique ou pneumatique, etc.). Ces composants ne sont plus commercialisés ;
- **la méthode en cascade*** d'origine anglo-saxonne très en faveur chez les pneumaticiens. Elle vise surtout à minimiser le nombre de variables secondaires (mémoires) matérialisées par des relais électromagnétiques bistables ou par des distributeurs à double pilotage pneumatique suivant la technologie de commande adoptée ;
- **l'organiphase*** développé par la société La Télémécanique et qui est lié aux modules logiques électroniques (modules de phase) créés par ce constructeur.

La technique pas-à-pas qui paraissait la plus prometteuse et la plus sujette à évolution a servi de point de départ à de nombreux chercheurs. Leurs études ont débouché sur **les séquenceurs*** ; à partir de 1965 différents modèles ont été commercialisés : séquenceurs pneumatiques Climax, Crouzet, Télémécanique, séquenceurs à relais bistables électromagnétiques Télémécanique, séquenceurs électroniques, pour n'en citer que quelques-uns.

La mise en pratique des séquenceurs variant avec le modèle et le constructeur, des problèmes ont surgi : divergence de choix entre les concepteurs, conflit de méthode entre les formateurs. La confusion a souvent été faite entre les séquenceurs et le GRAFCET qui faisait alors ses débuts et dont les règles d'évolution n'étaient pas encore bien assimilées par tous.

Note

* Voir chapitre XI.

Pour aller de l'avant, il fallait absolument mettre au point une méthode d'analyse indépendante du matériel. Aux Etats-Unis, les réseaux de Pétri (1962) ouvraient une première voie pour la description rigoureuse des automatismes mais cette méthode s'est, à l'usage, révélée plus théorique que pratique.

En France, au cours de l'année 1975, a été créée une Commission de normalisation de la représentation du cahier des charges d'un automatisme logique dans le cadre de l'AF CET (Association Française pour la Cybernétique Economique et Technique).

En 1977, des travaux de cette commission naissait le GRAFCET (GRAPhe Fonctionnel de Commande, Etapes, Transitions). Ensuite l'ADEPA (Agence nationale pour le DEveloppement de la Production Automatisée) a pris en charge sa diffusion sous forme d'articles et de brochures. A partir de cette date, le GRAFCET a été adopté progressivement par l'Enseignement technique, les laboratoires universitaires et plusieurs constructeurs d'automates programmables industriels (API).

En résumé, voici sur quelles bases a été conçu le GRAFCET :

- utilisation d'un procédé graphique : choix des symboles graphiques ;
- mise en évidence de chacune des situations de l'automatisme séquentiel à un moment donné : notion de situation, d'étape et d'action associées ;
- sélection des seules informations utiles à l'évolution de l'automatisme à partir d'une situation connue : notion de transition et de réceptivité ;
- définition des conditions d'évolution entre les étapes : établissement des règles d'évolution ;
- description progressive de l'automatisme : notion de point de vue (niveau) ;
- emploi d'un langage simple et accessible à l'ensemble des intervenants depuis le concepteur jusqu'à l'agent de maintenance : choix d'un vocabulaire.

Le GRAFCET a fait l'objet d'une norme française : la norme NF C 03-190 1982 rééditée en 1995.

Le GREPA (GRoupe Equipement de Production Automatisée) réunissant en équipe des industriels et des formateurs s'est fixé pour objectif l'amélioration des concepts du GRAFCET. Ces nouveaux concepts (macro-étapes et forçages de situation, notamment) sont intégrés dans les documentations UTE C 03-190 de 1990 et UTE C 03-191 de 1993. Avant cette date, ils ont été portés à la connaissance des techniciens et des enseignants par le biais d'un ouvrage intitulé "le GRAFCET, de nouveaux concepts", édité en 1985.

Il est à noter, et il faut le regretter, que la dernière édition de la norme NF C 03-190 de 1995 ignore les nouveaux concepts proposés par l'UTE.

En France, l'Inspection Générale des Sciences et Techniques industrielles, dès 1980, a introduit le GRAFCET en tant qu'outil d'enseignement des automatismes industriels dans les programmes des établissements techniques de tous niveaux. Le GRAFCET est aussi au programme des concours de recrutement des professeurs de l'Enseignement technique (CAPES et Agrégation). Son étude et ses applications font l'objet de nombreux stages de formation continue. Il est également enseigné dans de nombreux pays depuis son internationalisation par le biais du langage SFC (Sequential Function Chart ou diagramme fonctionnel en séquence) de la norme CEI 848. Le projet de norme PR NF EN 60848 a pour but d'intégrer les données de la CEI 848.

La réalisation des automatismes analysés à l'aide du GRAFCET s'est vu facilitée par l'apparition des automates programmables industriels. Les séquenceurs électriques et électroniques ont disparu du marché à l'exception des séquenceurs pneumatiques conservés pour des usages bien spécifiques, l'automatisation en ambiance explosive notamment.

Avec certains automates programmables industriels (API) la synthèse du GRAFCET est relativement facile. Ceux-ci disposent d'un langage GRAFCET alphanumérique et d'autres, ce qui est encore mieux, d'un langage GRAFCET graphique. Ces langages respectent les 5 règles d'évolution du GRAFCET normalisé.

Mais beaucoup d'API ne disposent encore que de langages du genre langage booléen, langage à contacts, langage calcul ou langage informatique. Ce n'est pas un obstacle insurmontable. Au **chapitre VII** intitulé « **Structures d'implantation du GRAFCET** », il sera démontré qu'il existe toujours, au minimum, une structure permettant l'implantation du GRAFCET dans ces API et a fortiori dans les autres.

Pour clore cette introduction, il faut noter la venue au langage GRAFCET par le biais du SFC, des constructeurs d'API étrangers, allemands et nord-américains entre autres.

I. PREMIER CONTACT AVEC LE GRAFCET

Le terme GRAFCET est un acronyme signifiant GRAPhe Fonctionnel de Commande, Etapes, Transitions. Conçu à l'origine pour être l'outil idéal de description des automatismes séquentiels, le GRAFCET a vu son usage s'étendre. Le GRAFCET est utilisé aussi bien pour l'analyse que la synthèse des automatismes tout en débordant du domaine purement séquentiel.

Une application relativement simple a été choisie ici. L'analyse méthodique du problème posé par son automatisation permettra de faire connaissance avec les concepts de base du GRAFCET et la façon de les mettre en œuvre.

1. Problème posé

Le client possède une tronçonneuse à scie circulaire à commande manuelle qu'il désire automatiser. Cette machine, pour l'essentiel, se présente ainsi :

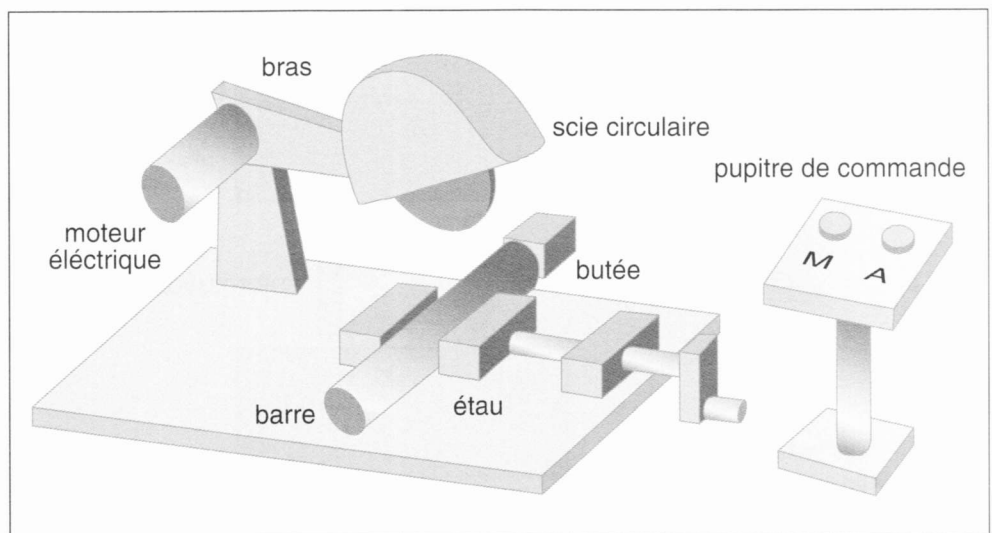


Figure 1.1
Organes essentiels de la
tronçonneuse à scie circulaire.

Sur la figure, on distingue :

- la scie circulaire entraînée par un moteur électrique,
- le bras pivotant support de la scie et du moteur,
- l'étau de serrage de la barre,
- la butée de mise à longueur,
- le pupitre de commande comportant deux boutons poussoirs pour la mise en marche et l'arrêt du moteur électrique.

Le client souhaite mécaniser puis automatiser :

- la descente et la montée du bras pivotant,
- le serrage et le desserrage de la barre.

Il faut noter que :

- mécaniser, c'est trouver un moyen pour remplacer une opération manuelle par une opération mécanique (emploi d'un actionneur),
- automatiser, c'est trouver un moyen pour faire exécuter dans le bon ordre toutes les opérations mécaniques (séquence d'actionneurs).

Pour ne pas compliquer inutilement cette première approche de la description d'un automatisme, l'automatisation de l'avance de la barre sur la butée de mise à longueur a été laissée de côté.

2. Découpage de l'automatisme en trois parties

Pour faciliter l'analyse, l'automatisme est divisé en trois parties (fig. 1.2) :

Une première partie dite Partie Opérative (en abrégé PO) regroupe les mécanismes, les actionneurs et leurs préactionneurs ainsi que les capteurs-machines nécessaires au contrôle du déplacement des actionneurs et au contrôle de présence des objets (ou des personnes).

Une deuxième partie appelée Partie Relation (en abrégé PR) regroupe les capteurs-opérateurs et les composants de signalisation visuels et/ou sonores. Le pupitre de commande sert de support aux éléments de la PR.

Une troisième partie nommée Partie Commande (en abrégé PC) regroupe les constituants et les composants destinés au traitement des informations (signaux) émises par les capteurs-opérateurs de la PR et les capteurs-machines de la PO. Les éléments de la PC sont contenus dans l'armoire de commande.

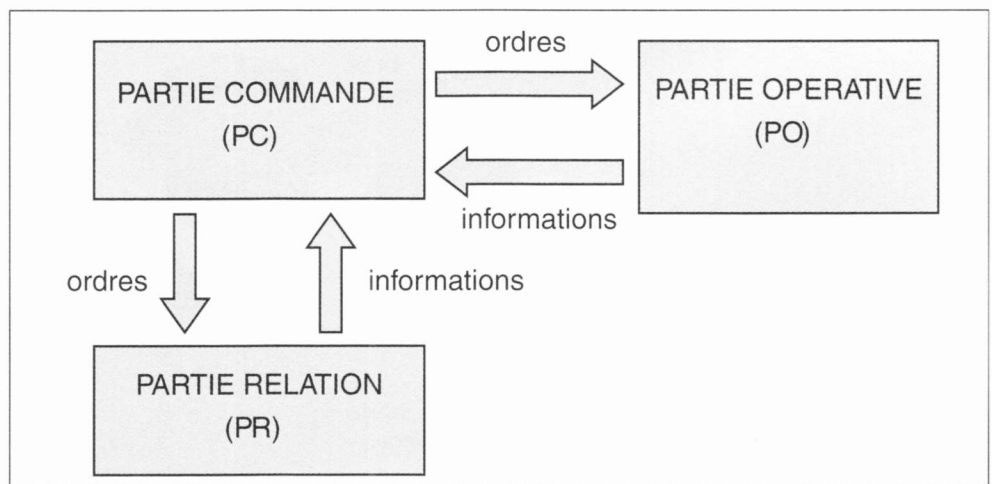


Figure 1.2
Les trois parties
d'un automatisme.

Comment s'effectue la communication entre la PO, la PR et la PC ?

- L'action exercée par le technicien sur les capteurs-opérateurs de la PR se traduit par l'envoi d'informations (signaux) vers la PC. Ce technicien chargé de la conduite de la machine est couramment désigné sous le nom d'opérateur. A titre d'exemple, l'information résultant d'une action exercée par l'opérateur sur un bouton poussoir nommé « départ de cycle » sera reçue par la PC. Ce signal, une fois traité par celle-ci, aura pour effet d'autoriser le lancement du cycle automatique de l'installation.
- Les capteurs-machines de la PO fournissent des renseignements sur la position des actionneurs et des objets à un instant donné en émettant des informations vers la PC. Ainsi l'action directe ou indirecte d'une porte sur son interrupteur « fin de course » indiquera que celle-ci est en position fermée, par exemple.
- La PC assure le traitement des informations émises par les capteurs-opérateurs de la PR et par les capteurs-machines de la PO. Les ordres résultants sont transmis aux préactionneurs de la PO selon certaines séquences (ou cycles) fonction du processus à exécuter. Des ordres sont également transmis aux composants de signalisation de la PR afin d'indiquer à l'opérateur l'état et la situation actuels de l'automatisme.

Par exemple, l'allumage d'un voyant nommé « REF » informera l'opérateur que tous les actionneurs occupent une position particulière dite de REFérence auto-riquant le départ du cycle automatique de la machine.

Autre exemple, le déclenchement d'un Klaxon alertera l'opérateur en lui signalant une anomalie de fonctionnement de la machine.

Le dialogue homme-machine est ainsi assuré de la PR vers la PO via la PC et réciproquement.

Exemples de composants de la partie opérative :

- actionneurs : vérins, moteurs, électroaimants, résistances de chauffage, etc.,
- préactionneurs : distributeurs, contacteurs, relais électromagnétiques, etc.,
- capteurs-machines : fins de course de vérins, détecteurs de position, etc.

Exemples de composants de la partie relation :

- capteurs-opérateurs : boutons poussoirs, interrupteurs, commutateurs, sélecteurs, etc.,
- composants de signalisation : voyants lumineux, gyrophares, affichages analogiques ou numériques, Klaxons, sonneries, etc.,
- composants de visualisation : écrans vidéo des terminaux et des moniteurs (monochromes ou couleur), etc.

Exemples de constituants et de composants de la partie commande :

- constituants : séquenceurs pneumatiques, automates programmables industriels (API), cartes à microprocesseur, micro-ordinateurs, systèmes numériques de contrôle commande (SNCC), etc,
- composants : modules opérateurs logiques pneumatiques, relais électromagnétiques, etc.

3. Situation de la machine dans les ateliers de l'entreprise

Un graphe arborescent permet de préciser cette situation :

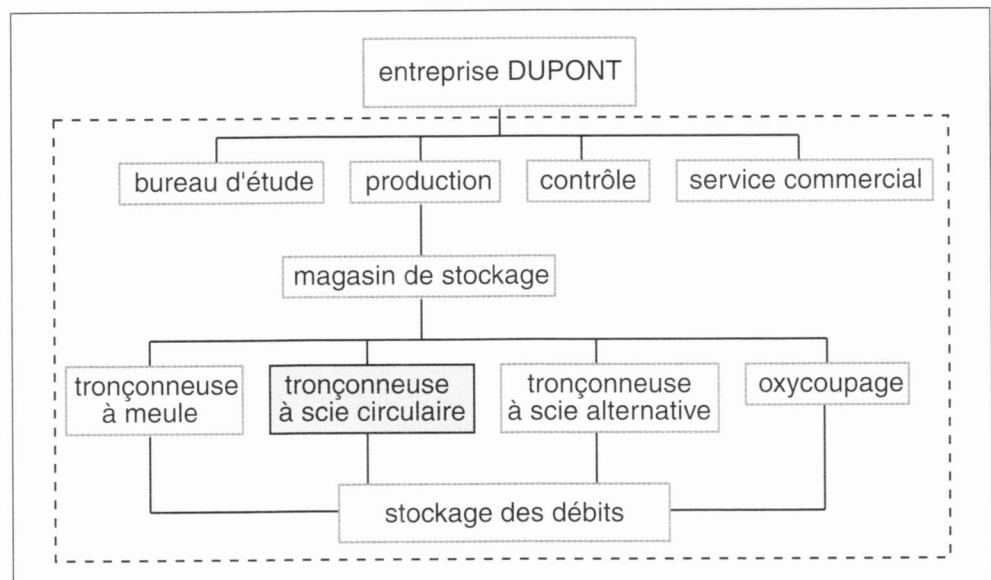


Figure 1.3
Contexte de la machine
à automatiser.

Cette représentation graphique est particulièrement recommandée. Elle donne une idée sur la place qu'occupe ou occupera la machine ou l'installation dans la structure de l'entreprise.

4. Analyse descendante de l'automatisme

L'analyse descendante permet l'approche d'un système en allant progressivement du général au particulier. Appliquée à l'étude des automatismes, cette méthode permet d'avancer dans l'analyse sans se perdre, dès le début, dans des détails inutiles. Partant de la fonction globale, l'analyse portera successivement sur :

- la recherche des fonctions,
- l'inventaire des tâches,
- la recherche de la coordination des tâches entre-elles,
- la description du cycle automatique de la machine.

Il est à noter qu'en fin d'ouvrage, au chapitre IX, est étudiée une application à caractère industriel relativement plus complexe. C'est dans ce cadre moins restreint que l'analyse descendante fait preuve de tout son intérêt.

4.1. Recherche des fonctions et des contraintes

Par recherche des fonctions, il faut entendre recherche de ce qui est à faire.

Fonction globale

L'automatisation de la tronçonneuse constitue, ici, la fonction globale.

Fonctions principales

Dans cette application, les fonctions principales du système à automatiser sont celles décrites par la figure 1.4 :

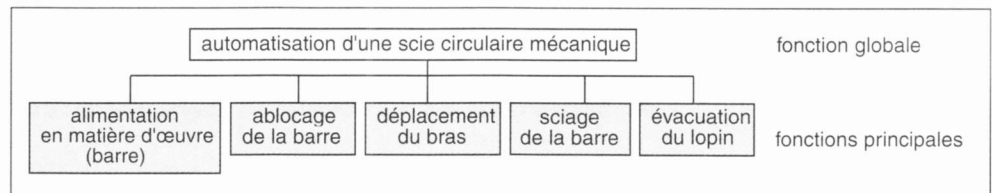


Figure 1.4
Les fonctions principales.

La fonction « alimentation en matière d'œuvre » qui restera une tâche manuelle et la fonction « évacuation du lopin » qui se fera par gravité dans une goulotte appropriée seront écartées de cette analyse.

Les fonctions principales sont soumises à des contraintes qu'il va falloir inventorier.

La levée des contraintes nécessitera des mesures et des calculs suivis de choix parmi les solutions techniques possibles.

Fonctions secondaires

Le tableau suivant (fig. 1.5) montre comment conduire la recherche des contraintes des fonctions principales. Les fonctions secondaires résultent du découpage des fonctions principales en fonctions plus élémentaires donc plus faciles à analyser.

La recherche des contraintes suppose une certaine expérience en construction mécanique et en automatique. Cet inventaire est le plus souvent le résultat d'un travail d'équipe (client et concepteurs – bureau d'étude et bureau d'automatique –). Les contraintes retenues orientent le choix vers une solution technique appropriée. Il faut ensuite procéder à des choix technologiques et dimensionnels et définir les paramètres de réglage de l'installation. Ce travail est du ressort du technicien de bureau d'étude et de celui du technicien-automaticien. Dans le cas des petites entreprises, ces deux fonctions sont souvent assurées par une

unique personne. Ceci reste également vrai s'il s'agit d'automatisations autour de PO relativement simples. Cela suppose une certaine polyvalence du dessinateur ou de l'automaticien.

fonctions principales	contraintes	fonctions secondaires
ablocage de la barre	capacité d'ouverture maximale, course de serrage, irréversibilité, effort de serrage en fonction des efforts d'avance et de coupe, vitesse.	<ul style="list-style-type: none"> → serrage → desserrage
sciage de la barre	nature de la matière d'œuvre, nature de la matière de la scie : carbure ou acier rapide, vitesse de coupe, usure, échauffement. effort de coupe, longueur de coupe mini-maxi.	<ul style="list-style-type: none"> → coupe scie <ul style="list-style-type: none"> → rotation <ul style="list-style-type: none"> → marche → arrêt → lubrification <ul style="list-style-type: none"> → marche → arrêt
déplacement du bras	nature du produit, débit, pompe, moteur. effort réduit, vitesse, course réglable. vitesse, course mini-maxi réglable.	<ul style="list-style-type: none"> → descente <ul style="list-style-type: none"> → approche rapide → remontée <ul style="list-style-type: none"> → avance lente

Figure 1.5
Recherches des contraintes
et des fonctions
secondaires.

Noter que la levée de certaines contraintes peut être faite par une fonction secondaire qui, elle-même entraîne un certain nombre de nouvelles contraintes non soupçonnées au départ.

Par exemple, les contraintes « usure et échauffement » de la fonction « coupe » conduisent à prévoir la nouvelle fonction secondaire « lubrification », elle-même soumise à ses propres contraintes : nature du produit, débit, pompe, moteur, etc.

4.2. Inventaire des tâches

On appelle tâche une action ou une suite d'actions à effectuer. Les tâches à exécuter sont liées aux fonctions secondaires définies précédemment. La fonction « lubrification » sera isolée de l'automatisme étudié et réalisée indépendamment par une tâche marche-arrêt gérée manuellement. Un contrôle sera toutefois effectué pour s'assurer que l'arrosage est effectivement en service.

- Tâche T1 : fonction serrage de la barre.
- Tâche T2 : fonction desserrage de la barre.
- Tâche T3 : fonction mise en marche de la scie.
- Tâche T4 : fonction arrêt de la scie.
- Tâche T5 : fonction descente-approche rapide du bras pivotant.
- Tâche T6 : fonction descente-avance lente du bras pivotant.
- Tâche T7 : fonction remontée du bras pivotant.

Il faut ajouter d'autres tâches qui n'apparaissent pas dans l'analyse précédente mais qui, néanmoins, sont habituelles et nécessaires dans la plupart des automatismes, à savoir :

- tâche T8 : fonction préparation du poste de travail : il s'agit de la mise en condition de la machine. Ici, cette fonction concerne la mise en énergie, le chargement de la matière d'œuvre et la vérification du plein du réservoir de lubrifiant d'arrosage ;
- tâche T9 : fonction mise en référence. Le rôle de cette fonction est de mettre les actionneurs de la machine en situation initiale de départ et de visualiser cette mise en référence.

Même si ce ne sont pas des tâches à proprement parler, il faudra prévoir les fonctions suivantes :

- une fonction protection du matériel : arrêt de la scie en cas de surcharge ou de blocage de la scie circulaire ;
- une fonction sécurité de l'opérateur : arrêt d'urgence de la scie en cas d'incident.

La tâche T8 étant une opération purement manuelle, il ne reste plus qu'à étudier la coordination des tâches T1 à T7 et T9 sans oublier de prendre en compte les fonctions de sécurité et de protection envisagées précédemment.

4.3. Coordination des tâches entre elles

Cette analyse s'appuie sur une description de type algorithmique. Pour chaque tâche, il faut se poser les questions suivantes :

- 1) Quelles sont les conditions autorisant le début de la tâche ?
(**début si** conditions vraies **alors** exécuter tâche)
- 2) Quelles sont les conditions vérifiant que la tâche est terminée ?
(**fin si** conditions vraies)
- 3) Quelle(s) tâche(s) est ou sont autorisée(s) en suivant ?
(**fin autorise** tâche(s) suivante(s))



APPLICATION

Tâche T1 (serrage de la barre) :

début si la machine est en référence (fin tâche T9)
et si la barre est en butée **et si** la pompe est en marche
et si le départ de cycle est autorisé
alors serrer la barre
fin si la barre est serrée
fin autorise le début de la tâche T3
(mise en rotation de la scie).

Tâche T2 (desserrage de la barre) :

début si fin de la tâche T4 (arrêt de la scie)
alors desserrer la barre
fin si la barre est desserrée
fin autorise le retour en référence
(tâche T9).

Tâche T3 (mise en rotation de la scie) :

début si fin de la tâche T1
alors mettre la scie en rotation **si pas** de surcharge sur la scie
et si pas de demande d'arrêt d'urgence
fin si le bras est en position haute
fin autorise le début de la tâche T5
(descente-approche rapide du bras).

Tâche T4 (arrêt de la scie) :

début si fin de la tâche T7 (remontée du bras)
alors arrêter la scie
fin si la scie est arrêtée **et si** le lopin est tombé
fin autorise le début de la tâche T2
(desserrage de la barre).

Tâche T5 (descente-approche rapide du bras) :

début si fin de la tâche T3 (mise scie en rotation)
alors effectuer la descente en rapide
fin si la course rapide est terminée
fin autorise le début de la tâche T6
(descente-avance lente du bras).

Tâche T6 (descente-avance lente du bras) :

début si fin de la tâche T5
alors effectuer avance lente
fin si le bras est en position basse
fin autorise le début de la tâche T7
(remontée du bras).

Tâche T7 (remontée du bras) :
début si fin de la tâche T6 (descente lente du bras)
alors remonter le bras
fin si le bras est en position haute
fin autorise le début de la tâche T4
(arrêt de la scie).

Tâche T9 (mise en référence) :
début si la PC est en situation initiale
et si une demande de mise en référence est formulée
et si la machine n'est pas déjà en référence
alors remonter le bras et arrêter la scie **et** desserrer la barre
et visualiser la mise en référence
fin si la machine est en référence
fin autorise le début de la tâche T1
(serrage de la barre).

Dans cette description les ordres associés à l'adverbe « alors » sont figurés par des verbes à l'infinitif. Ces ordres assurent le déclenchement des actions :

Ordres → Actions

- serrer la barre,
- desserrer la barre,
- mettre la scie en rotation,
- remonter le bras, etc.

Les conditions nécessaires à l'enchaînement des tâches apparaissent, ici, sous la forme :

- de consignes en provenance du pupitre de commande : autorisation de mise en référence et autorisation de départ de cycle données par l'opérateur ;
- de contrôles de position en provenance de la machine et permettant de savoir si les ordres - donc les actions - ont été exécutés : bras en position haute, intermédiaire ou basse, barre serrée ou non ;
- d'un contrôle de présence en provenance de la machine et permettant de détecter la présence ou non de la barre sur la butée de mise à longueur ;
- d'un contrôle de résultat permettant de savoir si la scie est en mouvement.

Les consignes, les contrôles de position, de présence et de résultat constituent par définition des signaux ou plus généralement des informations. Ces signaux sont fournis par des capteurs de type approprié où on distingue :

- les capteurs-opérateurs situés sur le pupitre de commande (PR) et manipulés par l'opérateur ;
- les capteurs-machines convenablement placés dans la PO et actionnés directement ou indirectement par les mécanismes dans une position bien déterminée de leur mouvement ou bien sensibilisés par l'absence ou la présence des objets ou par le constat d'une action en cours.

Le choix de la technologie des capteurs est fait dans un second temps de la description. De même, on ne sait encore rien des moyens techniques capables d'assurer les fonctions ou actions décrites précédemment. Ce choix reste également à faire.

A ce niveau de l'analyse, une représentation plus détaillée du découpage Partie Opérative, Partie Relation, Partie Commande devient possible (fig. 1.6).

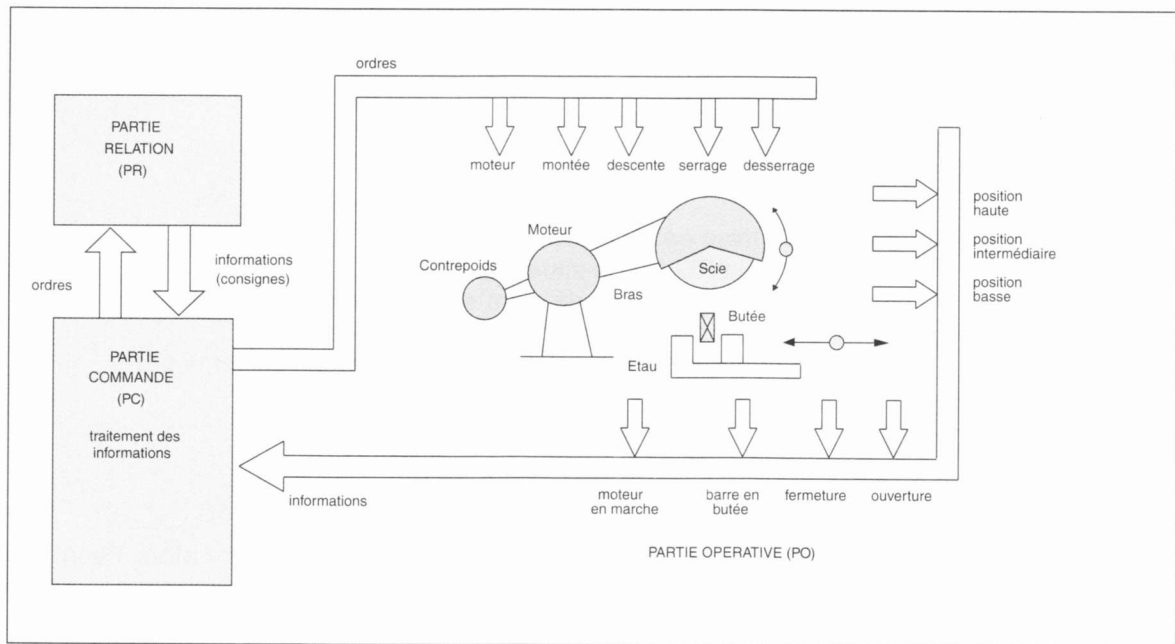


Figure 1.6
Découpage PO, PR, PC.

4.4. Description du cycle automatique de la machine

Il faut maintenant établir la séquence des tâches correspondant au cycle recherché. Grâce aux algorithmes décrits précédemment, ce travail ne doit poser aucun problème. On peut situer l'exécution de chaque tâche par rapport aux autres puisqu'on sait maintenant comment elle débute et quelle autorisation elle fournit lorsque sa réalisation est terminée.

Séquencement des tâches

test autorisations de départ données
début tâche T1
exécuter T1: serrage de la barre

test serrage effectué
début tâche T3
exécuter T3 : mise en rotation de la scie

test rotation scie vérifiée
début tâche T5
exécuter T5 : descente du bras en vitesse rapide
et maintien scie en rotation

test position intermédiaire du bras obtenue et rotation scie confirmée
début tâche T6
exécuter T6 : poursuite de la descente du bras en vitesse lente
et maintien scie en rotation

test position basse du bras obtenue et rotation scie confirmée
début tâche T7
exécuter T7 : remontée du bras en vitesse rapide et maintien scie
en rotation

test position haute du bras obtenue et rotation scie confirmé
début tâche T4
exécuter T4 : arrêt de la rotation de la scie

test scie à l'arrêt vérifié et lopin tombé
 début tâche T2
 exécuter T2 : desserrer la barre

test barre desserrée confirmé
 début tâche T9
 exécuter T9 : mise en référence si autorisation et si machine non
 en référence

Le diagramme fonctionnel (GRAF CET) permet de faire ce travail graphiquement, c'est-à-dire d'une façon plus imagée et par suite plus concise.

Exemple

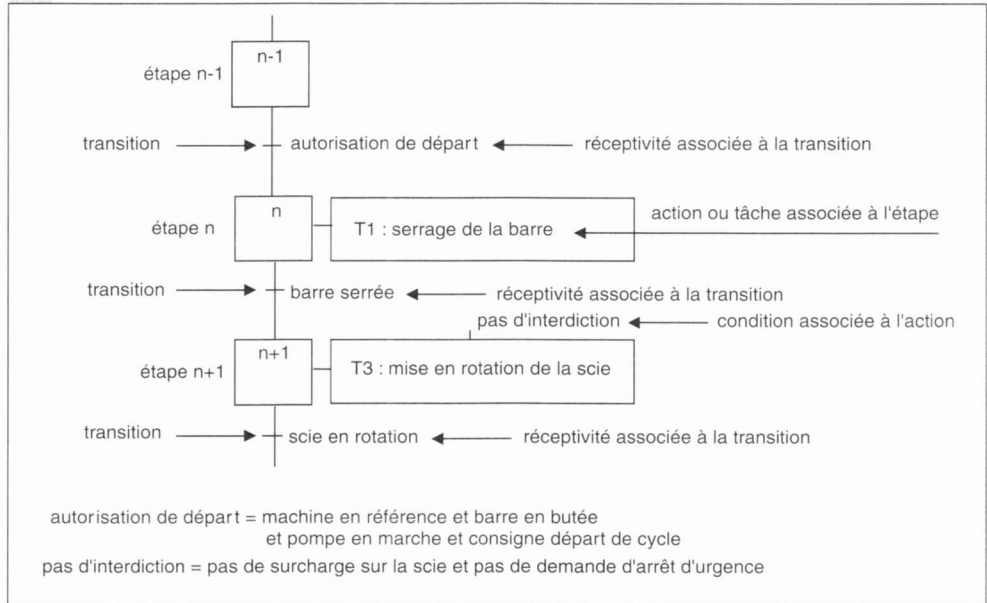


Figure 1.7
 Graphe d'exécution des
 tâches T1 et T3
 (GRAF CET partiel).

La tâche T1 est associée à une étape repérée n et figurée par un carré.

Pour que la tâche T1 soit exécutée, il faut que l'étape n soit active.

Comment fait-on pour activer l'étape n ?

L'étape n – 1 est supposée active. La transition entre l'étape n – 1 et l'étape n de ce fait se trouve validée. A un moment donné le test « autorisation de départ » devient positif ou vrai.

Dans le GRAFCET, l'expression logique du test est baptisée condition de transition ou réceptivité. La condition de transition peut donc être vraie ou fausse. Si la condition de transition est vraie, le système peut évoluer par franchissement de la transition. Cette transition est figurée par un tiret.

- Une transition est franchissable et obligatoirement franchie si elle est validée et si la condition de transition ou réceptivité associée à la transition est vraie.
- La transition est validée si l'étape précédente est active.
- Le franchissement de la transition entraîne simultanément l'activation de l'étape suivante et la désactivation de l'étape précédente.

Il s'agit là d'un extrait des 5 règles d'évolution du GRAFCET développées au chapitre III-8.

En résumé, le franchissement de la transition « étape $n - 1 \rightarrow$ étape n » a lieu dans les conditions suivantes : si l'étape précédente $n - 1$ est active (transition validée) et si la condition de transition « autorisation de départ » associée à la transition est vraie (transition franchissable), la transition « étape $n - 1 \rightarrow$ étape n » est obligatoirement franchie. Ce franchissement a pour effet d'entraîner simultanément l'activation de l'étape suivante n et la désactivation de l'étape précédente $n - 1$.

La tâche T1 « serrage de la barre » associée à l'étape est obligatoirement exécutée.

L'évolution « étape n vers étape $n + 1$ » s'opère de la même manière : le franchissement de la transition « étape $n \rightarrow$ étape $n + 1$ » a lieu obligatoirement si l'étape n est active (transition validée) et si la condition de transition « barre serrée » associée à cette transition est vraie (transition franchissable). Le franchissement a obligatoirement lieu. Ce franchissement entraîne simultanément l'activation de l'étape $n+1$ et la désactivation de l'étape n .

La tâche T3 « mise en rotation de la scie » associée à l'étape $n + 1$ est exécutable si la condition « pas d'interdiction » est vraie.

La tâche T3 est par définition **une tâche conditionnelle**.

Le cycle automatique complet est décrit par le GRAFCET de la figure 1.8. Les notions élémentaires acquises sur le graphisme et les règles d'évolution suffisent pour en comprendre le déroulement. Pour en savoir plus sur le GRAFCET et ses possibilités, voir chapitre III.

Le diagramme fonctionnel met en évidence :

- les étapes et leurs actions associées,
- les transitions et leurs conditions de transitions associées,
- les liaisons d'étape à transition et de transition à étape,
- les conditions associées aux actions.

Interprétation du GRAFCET (fig. 1.8)

Une étape est soit inactive, soit active. Si elle est active, deux cas sont possibles :

- 1) L'action associée est non conditionnelle : elle est obligatoirement exécutée.
- 2) L'action associée est conditionnelle : elle est obligatoirement exécutée si et seulement si la condition est vraie.

La situation de la partie commande est caractérisée par l'ensemble des étapes actives à l'instant considéré. Elle se note : $S = \{\text{étapes actives}\}$.

Exemples

- La situation $S = \{3\}$ signifie que dans cette situation seule l'étape 3 est active.
- La situation $S = \{5, 16\}$ signifie qu'à l'instant considéré seules les étapes 5 et 16 sont actives.

Le cycle automatique de la tronçonneuse (séquence) correspond à l'enchaînement des situations $S = \{1\}$, $S = \{2\}$, $S = \{3\}$, $S = \{4\}$, $S = \{5\}$, $S = \{6\}$, $S = \{7\}$.

L'étape 1 doublement encadrée caractérise la situation initiale. Cette étape est active dès la mise en énergie. Cette étape et son activation sont indispensables pour pouvoir démarrer la séquence. Cette étape est dite étape initiale et on notera la façon de la représenter.

L'évolution du GRAFCET correspond au passage d'une situation à une autre situation par franchissement d'une transition. Le franchissement s'effectue en respectant les règles d'évolution entrevues précédemment.

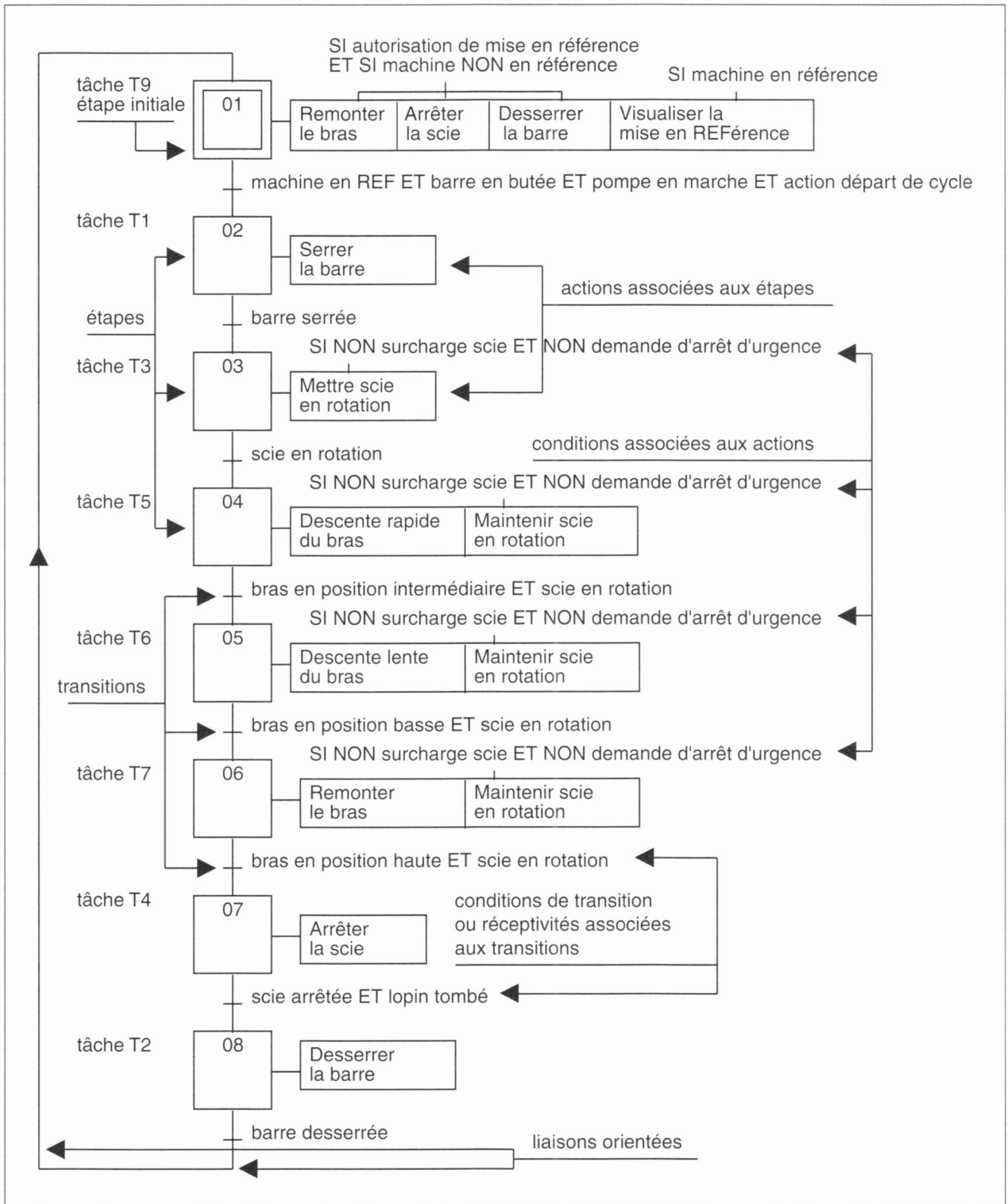


Figure 1.8
GRAFCET décrivant
le cycle automatique
de la tronçonneuse.

Exemple

La partie commande se trouvant dans la situation $S = \{6\}$ où seule l'étape 6 est active, le bras remonte, la scie continuant à tourner (actions associées). Le bras étant remonté en position haute et la scie toujours en rotation, la condition de transition ou réceptivité devient vraie (capteurs à l'état 1), la transition « étape 6 → étape 7 » est franchissable et obligatoirement franchie. La situation évolue de $S = \{6\}$ vers $S = \{7\}$ où seule l'étape 7 est active. L'évolution entraîne donc ici simultanément, l'activation de l'étape 7 et la désactivation de l'étape 6. La scie s'arrête (action associée à l'étape 7).

Autre exemple

Dans la situation $S = \{3\}$ où seule l'étape 3 est active, la scie est en marche conditionnelle, c'est-à-dire s'il n'y a pas de demande d'arrêt d'urgence et pas de surcharge sur la scie. L'action associée à l'étape 3 est **une action (ou tâche) conditionnelle**.

Les transitions sont à étudier une à une, à partir de l'étape 1. On peut suivre ainsi l'évolution de l'automatisme et vérifier que le GRAFCET décrit bien, au niveau des actions, un processus acceptable pour l'automatisation de cette machine.

La tâche T9 assurant la mise en référence des actionneurs est obtenue ici en associant conditionnellement les actions appropriées à l'étape initiale. Pour obtenir la mise en référence, il faut être dans la situation $S = \{1\}$ et avoir l'autorisation de l'opérateur (envoi de la consigne mise en référence) et s'assurer que la machine n'est pas déjà en référence (vérification de la position des actionneurs).

Il faut rappeler que les évolutions du GRAFCET obéissent à des règles précises ayant fait l'objet d'une norme, règles qui seront analysées en détail au chapitre III-8.

4.5. Options technologiques pour la partie opérative (*)

Pour progresser dans l'analyse de l'automatisme, il faut choisir les mécanismes, les actionneurs pour mouvoir ces mécanismes et les préactionneurs pour commander ces actionneurs. C'est en principe, comme il a déjà été dit, le travail d'une équipe composée de techniciens de bureau d'étude et d'automaticiens. Il s'agit de choix portant :

- 1) **Sur la technologie des actionneurs (*)** : par exemple, faut-il adopter un vérin hydraulique, un vérin pneumatique, un vérin oléopneumatique multiplicateur de pression ou bien un vérin électrique pour actionner le dispositif de serrage de la barre ?
- 2) **Sur le type de préactionneur (*)** : si on retient pour composant un vérin oléopneumatique multiplicateur de pression, faut-il le commander par un distributeur à 2 ou 3 positions ? Si l'on retient le modèle à 2 positions, doit-il être à simple pilotage (monostable) ou double pilotage (bistable), de technologie pneumatique ou électropneumatique ?
- 3) **Sur l'aspect économique** : l'objectif de cette automatisation est essentiellement d'améliorer les performances (cadence) et l'autonomie de cette machine pour un coût minimal.

Note

* Voir l'ouvrage "La Pneumatique dans les systèmes automatisés de production", Éditions Casteilla.

La scie circulaire étant à l'origine entraînée par un moteur électrique triphasé asynchrone, il n'y a pas de raisons valables pour que le concepteur actuel remette ce choix en question.

Il s'agit malgré tout d'une contrainte à laquelle doit faire face le concepteur et qui, par voie de conséquence, va orienter les autres choix technologiques. Il est à noter que le concepteur se trouve souvent placé dans une situation similaire :

- soit une partie de l'installation à automatiser existe déjà, auquel cas, d'un point de vue économique, un certain nombre d'actionneurs doivent être conservés si leur technologie ne pénalise pas les performances envisagées pour la nouvelle installation ;
- soit l'installation est entièrement à concevoir avec pour conséquence théorique une plus grande liberté de choix. Comme il existe sans doute des solutions techniques éprouvées et par suite bien adaptées au conditionnement du produit ou à la fabrication de l'objet en question, il est difficile d'aller à leur rencontre.

Dans le cas de la tronçonneuse, on a conservé le moteur électrique pour la coupe.

Pour la commande du bras et le serrage de la barre, on a opté respectivement pour :

- un vérin pneumatique associé à un régulateur hydraulique qui incorpore un dispositif d'approche rapide et d'avance lente avec butée intermédiaire réglable. Par suite les étapes 4 et 5 seront fusionnées. Ce genre de composant est proposé par plusieurs fabricants renommés ;
- un vérin oléopneumatique multiplicateur de pression pour obtenir l'effort de serrage de l'étau. Une solution pneumatique nécessiterait un vérin très encombrant et un dispositif mécanique assurant l'irréversibilité du blocage pour pallier l'élasticité de l'air comprimé. Plusieurs constructeurs ont ce genre de composant à leur catalogue.

Deux technologies sont complémentaires sinon concurrentes pour la commande des préactionneurs et donc de la partie commande* : la technologie pneumatique et la technologie électrique.

L'actionneur électrique (le moteur) est certes minoritaire par rapport aux autres actionneurs pneumatiques retenus (vérins) mais on optera ici pour la technologie électrique.

En effet, en automatique il faut toujours - notamment en ce qui concerne la PC - se laisser une large possibilité d'évolution sans complications exagérées. C'est le propre des PC de type programmable par opposition aux PC de type câblé.

Il existe actuellement des automates programmables industriels (API) relativement peu onéreux, bas de gamme performants.

L'unité centrale de l'API est un microprocesseur qui gère le programme mis dans une mémoire. Cette mémoire peut être écrite, lue et modifiée à la demande, ce qui confère une grande souplesse à la PC réalisée autour d'un API.

On pourra plus tard équiper cette machine d'un système d'avance automatique de la barre sur la butée de mise à longueur. On pourra également disposer au pupitre de commande d'un mode manuel pour la commande indépendante de chacun des actionneurs. Ces adjonctions se feront avec un minimum de modifications au niveau du câblage de l'armoire de la PC : ajout uniquement des entrées et sorties nouvelles suivi d'une révision du GRAFCET et du programme.

Quant à l'interfaçage des préactionneurs des vérins, il est obtenu sans problème avec des distributeurs à pilotage électropneumatique.

Note

** Les Technoguides édités par l'ADEPA permettent d'aboutir au meilleur choix technologique des composants et des constituants de l'automatisme en évitant les choix arbitraires voire routiniers.*

Récapitulatif du choix technologique des composants de la PO et de la PR :

Figure 1.9
Choix des actionneurs.

ACTIONNEURS		
Tâches	Symboles	Composants
montée-descente scie serrage de la barre rotation de la scie pompe	A S R P	vérin pneumatique à double effet attelé à un régulateur hydraulique vérin spécial avec multiplicateur de pression oléopneumatique moteur électrique 400 V triphasé asynchrone motopompe électrique 400 V triphasé asynchrone

Figure 1.10
Choix des préactionneurs.

PREACTIONNEURS		
Actionneurs	Symboles	Composants
du vérin A du vérin spécial S du moteur R de la motopompe P	A+,A- S+,S- KR KP	distributeur 5/2 à double pilotage électropneumatique 24 V ~ distributeur 5/2 à double pilotage électropneumatique 24 V ~ contacteur monostable 24 V ~ contacteur monostable 24 V ~

Figure 1.11
Choix des composants
de signalisation.

SIGNALISATION		
Fonctions	Symboles	Composants
tension 400 V tension 24 V machine en référence pompe en marche	V400 V24 VREF VP	voyant électrique à transformateur 400/24 V voyant électrique 24 V voyant électrique 24 V voyant électrique 24 V

Figure 1.12
Choix des capteurs-
opérateurs.

CAPTEURS-OPERATEURS		
Informations	Symboles	Composants
mise sous tension 400 V-24 V départ de cycle (DCY) mise en référence (REF) marche motopompe (P) arrêt motopompe (P) arrêt d'urgence (AU)	Q dcy BPréf BPmp BPap au	sectionneur porte-fusible tétrapolaire bouton poussoir unipolaire F (NO) bouton poussoir unipolaire F (NO) bouton poussoir unipolaire F (NO) bouton poussoir unipolaire F (NO) bouton "coup de poing" unipolaire F-O (NO-NF)

Figure 1.13
Choix des capteurs-
machines.

CAPTEURS-MACHINES		
Détection	Symboles	Composants
montée-descente du bras présence barre sur butée serrage de la barre moteur en marche motopompe en marche protection moteur R protection moteur pompe	a0,a1 pb s kr kp rtr rtp	interrupteurs de position unipolaire F (NO) interrupteur de position unipolaire F (NO) pressostat 0,4 - 5 bars inverseur unipolaire F-O (NO-NF) contact auxiliaire F (NO) du contacteur KR contact auxiliaire F (NO) du contacteur KP contact O (NF) du relais thermique RTR contact O (NF) du relais thermique RTP

4.6. Partie opérative compte tenu des options technologiques prises (fig. 1.14)

Remarque

Le pupitre de commande de cette machine est relativement dépouillé puisqu'il sert uniquement de support :

- aux quatre voyants de signalisation : mise en énergie (V400) et (V24), mise en référence (VREF), pompe en marche (VP) ;
- aux quatre boutons poussoirs : départ de cycle (dcy), mise en référence (BPréf), marche et arrêt de la pompe (BPmp) et (BPap), coup de poing d'arrêt d'urgence (au).

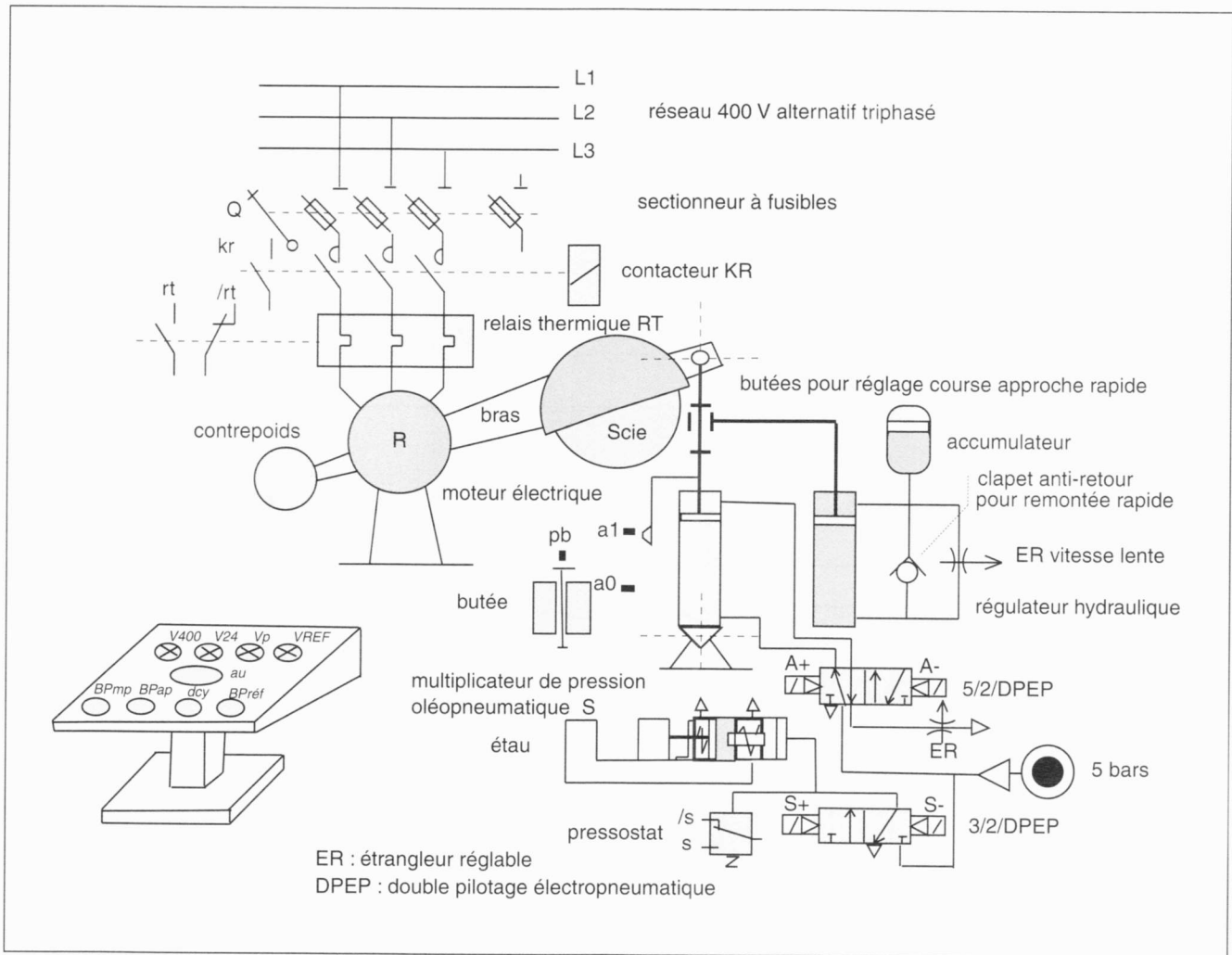


Figure 1.14
Partie opérative après choix
des composants.

Le pupitre de commande regrouperait un nombre plus important de composants si, volontairement, l'étude des modes de marche et d'arrêt n'avait été omise, ceci afin de ne pas compliquer inutilement ce premier contact avec le GRAFCET.

4.7. Relation de cause à effet entre les composants

Avant de passer à la nouvelle description par GRAFCET, on utilise un mode de représentation par diagramme-bloc (fig. 1.15).

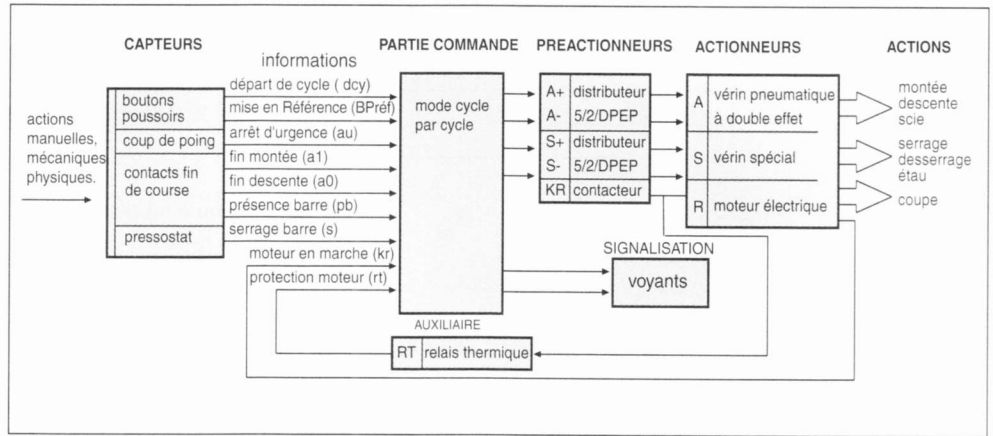
Le diagramme-bloc permet de mettre en évidence tous les composants d'un automate avec leurs entrées-sorties et leurs relations de cause à effet.

Le bloc est représenté par un rectangle. **Ce qui entre dans un bloc est la cause et ce qui en sort est l'effet.**

Cette représentation est à recommander et elle doit trouver sa place en tant que document du dossier technique de la machine ou de l'installation automatisée.

Elle permet de réunir, sur un document unique, tous les composants essentiels de l'ensemble ou par folio de chacun des sous-ensembles de l'automatisme.

Figure 1.15
Diagramme-bloc appliqué
à l'automatisme
de la tronçonneuse
sans la motopompe.



4.8. Description par GRAFCET compte tenu des options technologiques (fig. 1.16)

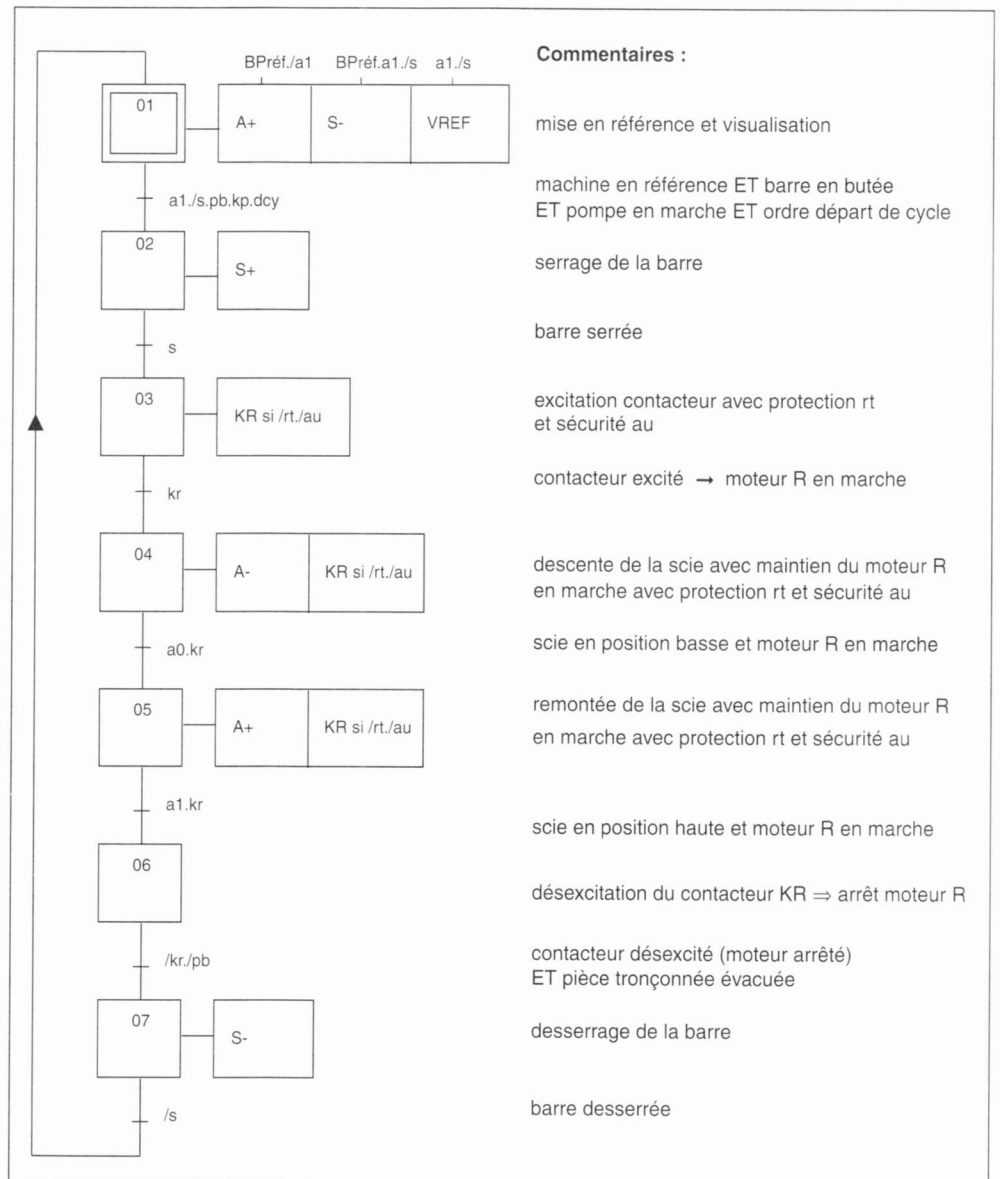


Figure 1.16
GRAFCET après
choix technologique
des composants.
Des commentaires ont été
ajoutés sur le document.
Ces commentaires sont
souhaitables voire indis-
pensables, car ils évitent
les erreurs d'interprétation
des symboles utilisés pour
les actions et les capteurs.
Par ailleurs, ils font gagner
du temps en évitant d'avoir
à consulter la table
des symboles.

A propos du GRAFCET précédent (fig. 1.16), noter l'emploi des opérateurs logiques NON, ET et OU dans les conditions de transitions (réceptivités) et dans les conditions associées aux actions et leur écriture conventionnelle :

- la barre inclinée (/) dite aussi « slash » pour le NON, le point (.) pour le ET et le plus (+) pour le OU.

Exemples

a1.kr se lit « a1 ET kr »,
/rt./au se lit « NON rt ET NON au » ou encore « rt barre ET au barre ».

La mise et le maintien en marche du moteur R pendant la descente et la remontée de la scie sont obtenus en associant l'ordre KR aux étapes successives 3, 4 et 5.

Cet ordre est en effet destiné à la bobine monostable du contacteur KR et il est nécessaire que cet ordre soit maintenu pour que le contacteur reste excité et continue ainsi d'alimenter le moteur R.

En revanche - et c'est l'évidence même - il suffit de ne plus envoyer d'ordre à la bobine KR aux étapes 6 et 7 et 1 et 2 pour que le moteur s'arrête ou soit maintenu à l'arrêt.

L'adoption d'un distributeur 2 positions, à double pilotage électropneumatique donc par nature bistable, pour commander notamment le serrage et le desserrage de la barre, résout le problème du maintien du serrage aux étapes 3, 4, 5 et 6. C'est le distributeur lui-même qui assure la mémorisation et il n'est pas nécessaire d'associer l'ordre de serrage aux étapes précitées. Le distributeur réalise ainsi une fonction opérative de commande (FOC), ici une fonction mémoire.

Remarque

La mise en référence a été améliorée et est devenue séquentielle par l'intermédiaire des conditions associées aux actions. C'est un moyen parmi d'autres de réaliser la mise en référence d'une installation.

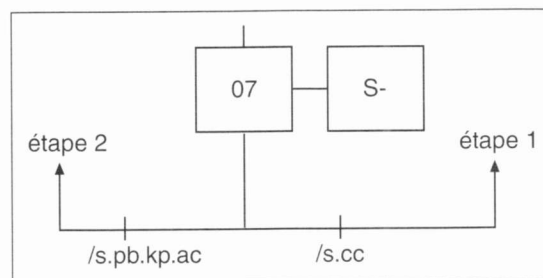


Figure 1.17
Sélection cycle automatique
continu-cycle par cycle.

En mode production, le rebouclage de l'étape 7 vers l'étape 2 est possible sans passer par l'étape 1. Il faut, pour cela, ajouter au pupitre de commande, un sélecteur de cycles pour avoir le choix entre un cycle automatique continu noté ac et un cycle par cycle noté cc.

La réceptivité associée à la transition nouvelle étape 7 → étape 2 aura pour expression : /s.pb.kp.ac tandis que la réceptivité associée à la transition étape 7 → étape 1 est modifiée et devient : /s.cc (fig. 1.17). Noter qu'ici seule la marche cycle par cycle se justifie puisque l'avance de la barre reste une opération manuelle.

La lecture du GRAFCET fournit les expressions logiques des entrées des pré-actionneurs qui sont en fait les sorties de la PC. Sachant que les signaux associés aux étapes sont notés Xi où i est le numéro de l'étape, on obtient :

pilotages :	A+ = X1.BPréf./a1+X5	S+ = X2
	A- = X4	S- = X1.BPréf.a1./s+ X7
contacteur :	KR = (X3+X4+X5)./rt./au	
voyant REF :	VREF = X1.a1./s	

Commentaires sur la recherche des expressions logiques des sorties

L'ordre de commande du pilotage A+ est associé à l'étape 1 sous la condition BPréf./a1. Il est également associé à l'étape 5, d'où son expression logique :

$$A+ = X1 \text{ ET } B\text{Préf} \text{ ET NON } a1 \text{ OU } X5 \text{ ou encore } A+ = X1.B\text{Préf.}/a1+X5$$

L'ordre de commande du contacteur KR est associé aux étapes 3, 4 et 5 sous la condition /rt./au d'où son expression logique :

$$KR = (X3+X4+X5) /rt./au$$

Le même raisonnement est à appliquer aux autres sorties.

Pour terminer cette étude, il reste à choisir :

- 1) l'automate programmable industriel le mieux adapté en s'attachant, entre autres caractéristiques, au nombre d'entrées-sorties, à la capacité mémoire, à la marque et au modèle, aux langages de programmation disponibles ;
- 2) le langage de programmation à adopter pour l'API retenu, à programmer l'automatisme compte tenu du GRAFCET de la figure 1.16 et à l'introduire dans la mémoire de l'API.

La programmation d'un GRAFCET faisant appel à la notion de structure d'implantation, étudiée au chapitre VII, la synthèse c'est-à-dire la réalisation de l'automatisme de la tronçonneuse ne sera pas étudiée ici.

Il reste maintenant à voir comment aborder la description d'un automatisme (chapitre II) puis à découvrir les autres concepts du GRAFCET et leur mise en œuvre (chapitre III).

II. DESCRIPTION D'UN SYSTÈME AUTOMATISÉ DE PRODUCTION (SAP)

1. Cahier des charges

Au sens strict du terme, il s'agit des conditions imposées au réalisateur (concepteur et fournisseur) par le client (demandeur et acheteur) en vue de la création ou de la modification d'une installation automatisée.

C'est le client qui indique les objectifs à atteindre en précisant :

- quelle est la valeur ajoutée au produit traité ?
- quelle est la cadence de production à atteindre ?
- quel est le budget disponible ou coût maximal à ne pas dépasser ?
- etc.

Mais le problème ne peut être décrit convenablement que par un spécialiste du processus.

Par processus on entend l'ensemble des opérations d'élaboration d'un produit selon un procédé déterminé au moyen d'unités de traitement et de transformation.

En revanche, la recherche du procédé relève de la compétence du concepteur.

Le réalisateur n'étant pas censé connaître tous les processus industriels (fabrication d'objets, élaboration de produits, conditionnement de produits finis, etc.) pour ne citer que quelques exemples, l'avis du spécialiste ne peut lui être que profitable.

Lorsque la faisabilité du projet est démontrée par le concepteur, c'est finalement le client qui donne son accord pour le lancement de l'étude et de sa réalisation.

Ce stade ne peut être atteint qu'après maintes discussions pouvant remettre en cause le produit et éventuellement le processus (démarche selon l'esprit productique).

Lorsque les divers intervenants ont une bonne connaissance des outils-méthodes tels que le GRAFCET, le GEMMA* (Guide d'Etude des Modes de Marche et d'Arrêt) sans oublier l'organigramme ni le chronogramme, parvenir à un accord devient plus facile.

Mais le cahier des charges ne saurait se limiter à la partie technique. Des clauses d'ordres commercial, juridique et financier y sont également consignées.

L'automaticien, plus directement concerné par l'aspect technique du projet, ne saurait ignorer ces aspects.

Note

* *Noter au passage que le GEMMA est un outil graphique de recherche et d'analyse des modes de marche et d'arrêt. Des méthodes permettent de transcrire le GEMMA en GRAFCET d'où son intérêt. Un aperçu du GEMMA et de son exploitation est donné au chapitre IX. Voir également notre ouvrage consacré au GEMMA et à la conception des SAP sûrs, des mêmes auteurs, édité dans la même collection.*

2. Comment aborder la description d'un système automatisé de production (SAP) ?

2.1. Premier type d'approche

Une première approche peut être faite en décomposant le système en trois parties interdépendantes :

1 - La Partie Opérative (PO) regroupant :

- les mécanismes,
- les actionneurs : moteurs, vérins, etc.,
- les préactionneurs : contacteurs, distributeurs, etc.,
- les capteurs-machines : fins de course, détecteurs, etc.

2 - La Partie Relation (PR) comprenant :

- le pupitre de commande à la disposition de l'opérateur pour le dialogue homme-machine et supportant :
 - les capteurs-opérateurs autorisant la mise en énergie de l'installation,
 - la sélection des modes de marche,
 - la commande manuelle des actionneurs,
 - la mise en référence,
 - le départ des cycles,
 - l'arrêt d'urgence ;
- les signalisations diverses :
 - voyants lumineux fixes ou clignotants, gyrophares,
 - afficheurs analogiques et numériques,
 - écrans vidéo monochromes ou couleur,
 - synoptiques,
 - Klaxons, buzzers, sonneries, etc.

3 - La Partie Commande (PC) regroupant les constituants et les composants assurant le traitement des signaux d'entrées en provenance de la PO et de la PR afin d'émettre les ordres de sorties destinés aux entrées des préactionneurs et aux éléments de signalisation.

Le traitement des informations par la PC peut être réalisé soit sous la forme entièrement câblée (en voie de disparition), soit sous la forme mixte câblée et programmée pour accroître la sécurité.

Cette dernière forme se prête très bien à l'intégration des API et des micro-ordinateurs industriels en tant que constituant principal de la PC. Le programme gère l'essentiel de l'automatisme.

Les sécurités câblées (arrêt d'urgence, interrupteurs de surcourse, etc.) et les sécurités mécaniques assurent la sécurité de premier niveau en cas de défaillance du ou des constituants programmés.

Au sujet de la sécurité, on constate comparativement :

- que les sécurités câblées (ou mécaniques) sont sûres mais à durée de vie limitée (nombre limité de manœuvres des composants avant usure, panne ou rupture),
- que les constituants programmables sont actuellement fiables (très peu de pannes) mais imprévisibles en cas de dysfonctionnement (sorties mises à 1 ou 0 aléatoirement).

En résumé, la combinaison des deux techniques permet d'obtenir le meilleur compromis en ce qui concerne la sécurité des personnes et du matériel.

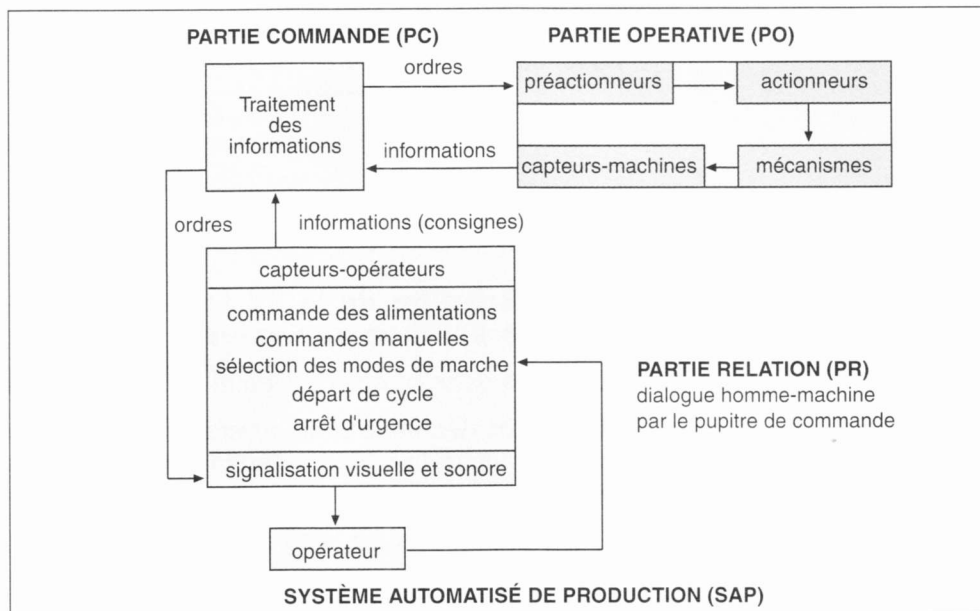


Figure 2.1
Représentation détaillée
des trois parties du
système automatisé de
production.

Remarque

Le découpage en trois parties du SAP est seulement destiné à faciliter l'analyse et la synthèse de l'automatisme. Ce découpage n'impose pas la séparation physique de ces trois sous-ensembles.

L'implantation des contacteurs (PO) dans l'armoire de commande où est logée la PC est de pratique courante.

De même, il semble commode pour les opérations de présélection de trouver les compteurs, les minuteries, etc. de la PC en face avant du pupitre de commande (PR). Le pupitre de commande lui-même peut être associé à l'armoire de commande de la PC.

En fait, l'implantation sur le site de l'armoire de commande (PC) et du pupitre (PR) par rapport à la machine ou à l'installation (PO) découle le plus souvent de contraintes ergonomiques, de contraintes liées au temps de réponse si la PC est tout pneumatique, voire de contraintes liées à l'espace disponible.

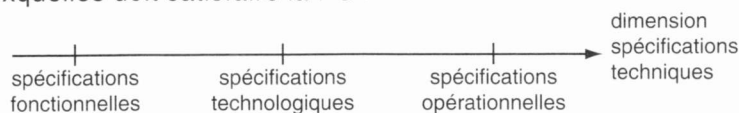
2.2. Deuxième type d'approche

L'approche suivante s'inscrit dans le cadre général d'une méthode dite d'analyse descendante. Elle fait appel à la notion de niveau de représentation d'une partie commande. La caractérisation du niveau nécessite la prise en compte de trois dimensions :

- une dimension « point de vue » caractérisant la situation de l'observateur décrivant le fonctionnement du SAP envisageable dans l'ordre suivant :



- une dimension « spécifications » caractérisant la nature des spécifications techniques auxquelles doit satisfaire la PC :



- une dimension « finesse » caractérisant le niveau de détail dans la description du fonctionnement :



A. Dimension point de vue

a. Description de la PC faite par un observateur se situant d'un point de vue externe au SAP

On parle dans ce cas de « point de vue processus ».

Il s'agit de décrire le comportement de la partie commande (PC) vis-à-vis de la partie opérative (PO) indépendamment de tout choix technologique et en tenant compte principalement des spécifications fonctionnelles.

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.

A ce niveau, le GRAFCET correspond à un Graphe de Coordination des tâches.

b. Description de la PC faite par un observateur se situant d'un point de vue interne au SAP et externe à la PC

En raccourci, on dit « point de vue PO-PR ». Il s'agit maintenant, dans un premier temps, de décrire le fonctionnement de la PC 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 à automatiser.

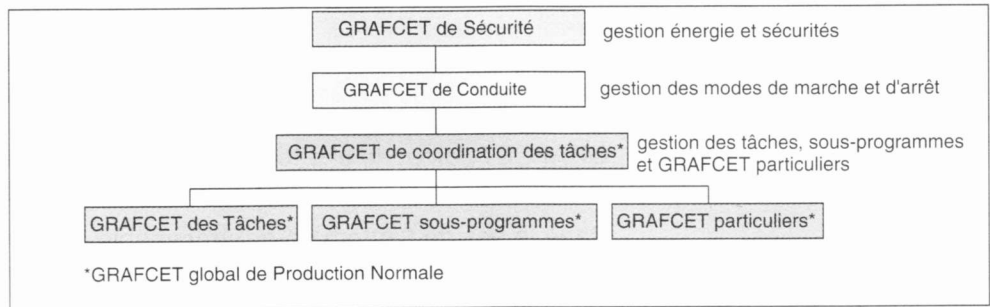
Ces options correspondent à une partie des spécifications technologiques de la partie opérative. A ce niveau, on peut structurer la description sous forme d'un ensemble de GRAFCET comprenant par exemple :

- 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'Etude 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 hiérarchisés (fig. 2.2). 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.

Figure 2.2
Hiérarchie
entre les GRAFCET.



Dans le cas d'automatisations complexes, il peut être nécessaire d'avoir un ou plusieurs GRAFCET de gestion de l'énergie et un ou plusieurs GRAFCET de sécurité hiérarchisés.

Le complément d'informations fourni par cette étude permet de définir progressivement la Partie Relation (PR), par conséquent le pupitre de commande : choix des capteurs-opérateurs et implantation sur la face avant de celui-ci.

Des capteurs non prévus dans un premier temps peuvent apparaître nécessaires au niveau de la sécurité du personnel et de la protection du matériel, par exemple. Ils devront être implantés dans la PR et dans la PO à une place appropriée.

Ces informations complémentaires correspondent à une partie des spécifications opérationnelles.

c. Description de la PC faite par un observateur se situant d'un point de vue interne à la PC

On dit aussi « point de vue PC ou réalisateur ».

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 : 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 ou inverse, 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. Le séquenceur pneumatique s'impose comme constituant lorsque l'environnement est particulier (ambiance explosive, par exemple). Indépendamment de la marque et du modèle, le séquenceur évolue de manière asynchrone. Ceci revient à dire qu'avec un séquenceur, il n'est pas possible de respecter les règles

d'évolution du GRAFCET normalisé. En effet, le GRAFCET évolue de manière synchrone : activation et désactivation simultanées des étapes lors du franchissement d'une transition. Il s'ensuit, pour prendre un cas précis, que la réalisation avec un séquenceur d'une boucle à deux étapes est impossible. Si le GRAFCET comporte une telle structure, il faut le modifier en introduisant dans la boucle une troisième étape (voir chapitre VI-3, cas de la boucle à deux étapes).

Dans le cas de l'automate programmable industriel (API), afin de faciliter la programmation, la mise au point, les modifications et la maintenance, il est souhaitable de trouver sur chacun des GRAFCET ou à défaut sur des documents d'accompagnement :

- les adresses des bits associés ou non aux étapes,
- les adresses des entrées associées aux capteurs,
- les adresses des sorties associées aux entrées des préactionneurs,
- les adresses des temporisateurs, des compteurs, etc., avec les adresses de leurs bits associés.

Un tableau d'exploitation des bits internes pour tous les calculs intermédiaires doit accompagner le GRAFCET. Des commentaires seront ajoutés au GRAFCET. Tout ceci nécessite un travail méticuleux parfois fastidieux mais généralement très efficace à l'usage, notamment à l'occasion d'un dépannage urgent.

La structure des GRAFCET devra être adaptée au langage disponible dans l'API s'il n'est pas un langage typiquement orienté GRAFCET. Les langages GRAFCET graphiques, en général, n'acceptent pas les graphismes en boucle ouverte. Il faut alors refermer le GRAFCET avec une transition et une réceptivité associée nulle de la forme $r./r = 0$ (voir exemple chapitre III-13.8).

C'est à ce niveau de la description que s'élabore progressivement le dossier technique de l'installation automatisée, dossier qui accompagnera celle-ci durant toute son existence. Ce dossier réunit tous les documents utiles à la compréhension du fonctionnement, à son exploitation (modes de marche), à sa modification ou son amélioration éventuelle et à sa maintenance. Toutes les modifications appor-

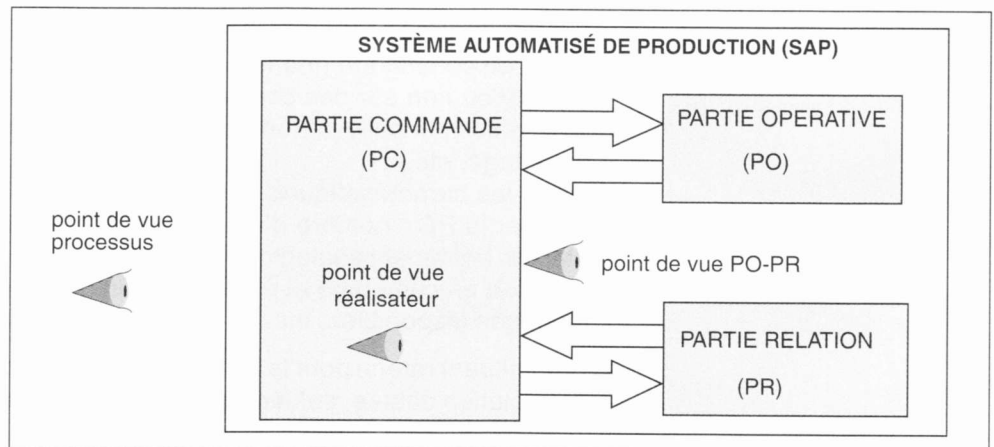


Figure 2.3
Position de l'observateur
suivant le point de
vue choisi.

Note

La conception d'un système automatisé de production (SAP) sûr, conforme aux normes de sécurité actuelles, passe par la recherche de solutions sûres à tous les stades de son étude. C'est pourquoi, le principe de l'analyse descendante par points de vue successifs est conservé mais élargi. On s'intéresse notamment, dans l'ordre, aux points de vue produit, production, procédé, processus, PO-PR, PC, réalisation, mise en service, exploitation et maintenance.

Cette nouvelle procédure est développée dans l'ouvrage consacré au GEMMA et à l'étude de la conception de SAP sûrs édité dans la même collection.

tées à l'installation pendant toute sa durée de vie devront être consignées dans ce dossier. La non-mise à jour des dossiers techniques rend en effet très difficiles certains dépannages intervenant à un moment souvent crucial pour l'entreprise.

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 sera souvent résumée de la manière suivante :

- point de vue système ou procédé = niveau 1,
- point de vue PO - PR = niveau 2,
- point de vue PC ou réalisateur = niveau 3.

La figure 2.3 montre la position de l'observateur suivant les divers points de vue. Au chapitre III-3 est présentée (figure 3.7), de manière plus concrète, l'évolution de la description par GRAFCET en fonction des niveaux qui ont été définis.

B. Dimension spécifications techniques

C'est le terme retenu par la norme pour désigner toutes les spécifications dont il a été question au paragraphe précédent. Elles sont réparties en trois groupes principaux de représentations :

- d'abord les représentations relevant des **spécifications fonctionnelles** ; ces spécifications correspondent aux fonctions ou aux tâches devant être assurées ou effectuées par l'automatisme. Elles caractérisent donc les comportements que doit avoir la PC face aux informations issues de la PO et à celles données par l'opérateur à partir du pupitre de commande de la PR ;
- ensuite les représentations relevant des **spécifications technologiques** ; ces spécifications prennent en compte les choix technologiques possibles pour les composants et les constituants de la PO, de la PR et de la PC c'est-à-dire de l'ensemble du système automatisé de production (SAP) ;
- enfin les représentations relevant des **spécifications opérationnelles** ; ces spécifications précisent les comportements que doivent avoir la PC et le système automatisé dans le contexte de production.

Elles concernent les performances globales de l'automatisme.

Elles prennent en compte les contraintes imposées par les normes de sécurité concernant le personnel et les contraintes liées à la protection du matériel.

Elles rappellent les conditions destinées à faciliter la maintenance de l'installation.

Elles intègrent les propositions de modes de marche et d'arrêt souhaitées par le client ou proposées par le concepteur afin de faciliter la conduite et les réglages de l'installation.

Elles fournissent des indications quant aux dispositions à retenir notamment en ce qui concerne le dialogue homme-machine par l'intermédiaire du pupitre de commande et/ou d'un synoptique câblé (voyants) ou programmé (écran vidéo).

C. Organisation des dimensions entre elles

Chaque niveau de point de vue peut être considéré comme un plan dans lequel on se déplace suivant « un axe spécifications techniques » ou suivant « un axe finesse » (voir figure 2.4 page suivante).

On progresse ainsi dans la description, plan par plan, par affinements successifs en partant du problème posé pour aboutir à sa réalisation.

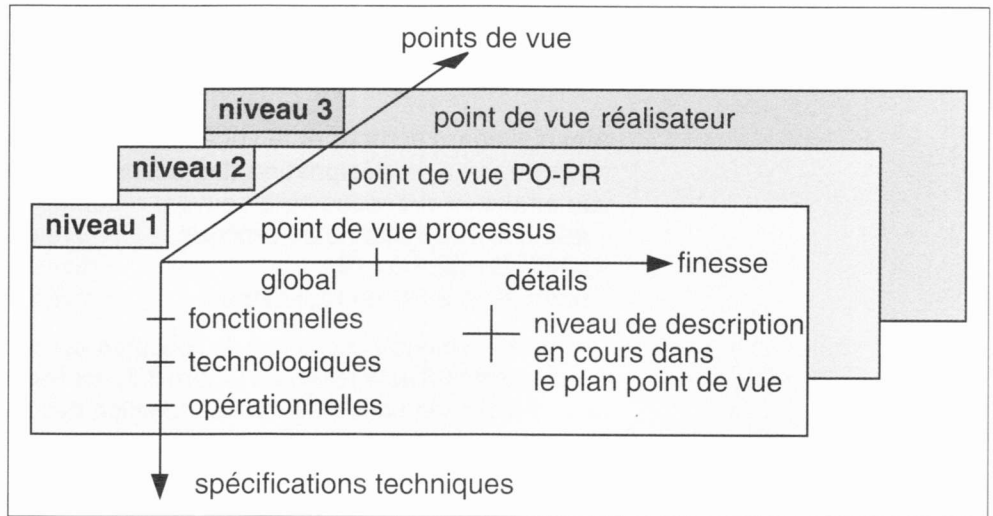


Figure 2.4
Les trois plans point de vue
ou niveaux.

Remarque

Au niveau 1, le GRAFCET correspondant décrit le processus adopté. Le terme « point de vue processus » a été préféré à celui de « point de vue système ou procédé » jugé non approprié.

Au niveau 2, la PO et la PR étant en phase de définition, la GRAFCET correspondant a pour finalité de décrire l'ordre des actions commandant les entrées de la PO et de la PR. Ce point de vue est dit « point de vue PO-PR » puisque c'est de ce point de vue qu'on examine comment automatiser la PO et la PR.

Au niveau 3, les constituants et les composants de la PC et de la PR sont définitivement choisis. Le GRAFCET correspondant décrit la PC telle qu'elle sera réalisée et réceptionnée. Le point de vue envisagé est dit plus simplement « point de vue réalisateur ».

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. 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 du niveau 1),
- du point de vue PO-PR (GRAFCET de niveau 2),
- du point de vue PC,
- 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.

Dans l'ouvrage intitulé « LE GEMMA » édité dans la même collection, sont développées les méthodes de conception de SAP sûrs. On y trouve également une application à caractère industriel dont l'analyse descendante est réalisée selon les points de vue répertoriés ci-dessus.

Note

Une application est proposée au chapitre IX. Son analyse dite descendante repose sur la description progressive du SAP selon les trois points de vue développés dans ce chapitre.

III. MODÈLE GRAFCET

1. Définition

Le GRAFCET est un modèle de représentation graphique du comportement de la partie commande d'un système automatisé.

Ce modèle est défini par un ensemble constitué :

- **d'éléments graphiques de base** comprenant :
 - les étapes,
 - les transitions,
 - les liaisons orientées.

Les liaisons orientées relient entre elles les étapes et les transitions, structurées en un réseau alterné étapes-transitions-étapes formant l'ossature séquentielle graphique du GRAFCET ;

- **d'une interprétation** traduisant le comportement de la partie commande vis-à-vis de ses entrées et de ses sorties, et caractérisée par :
 - les conditions de transitions ou réceptivités associées aux transitions,
 - les actions associées aux étapes ;
- **de 5 règles d'évolution** définissant formellement le comportement dynamique de la partie commande ;
- **de postulats sur les durées** relatives aux évolutions.

2. Étape

L'étape correspond à une situation dans laquelle le comportement de la PC ne varie pas vis-à-vis de ses entrées et de ses sorties.

L'étape se représente par un carré (ou un rectangle) repéré numériquement, le repère étant placé dans la partie supérieure du cadre (fig. 3.1). La partie inférieure peut recevoir, si besoin est, une étiquette (un commentaire ou une adresse du bit image de l'étape dans le cas d'un API, par exemple). Si la place fait défaut, l'étiquette ou le commentaire peut être placé, entre guillemets, à côté du cadre de l'étape à identifier.

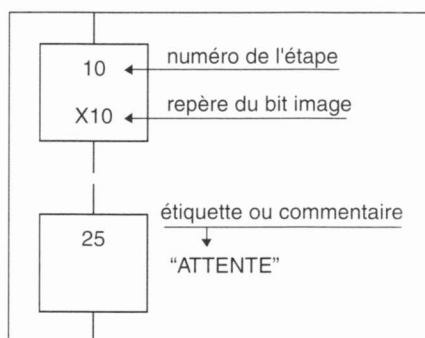


Figure 3.1
L'étape et son repérage.

A un instant donné et suivant l'évolution du système, une étape est soit active soit inactive.

Un état interne de la PC ou « **situation** » est défini par l'ensemble des étapes actives à un instant donné. Ceci peut être exprimé sous la forme : $S = \{i, j, k, \dots, l\}$ ou pour plus de précision, par $S(G_m, G_n, G_p, \dots, G_q) = \{i, j, k, \dots, l\}$ où i, j, k, \dots, l sont les étapes actives des GRAFCET $G_m, G_n, G_p, \dots, G_q$ à l'instant considéré (notation UTE C 03-191).

La situation du GRAFCET global G et d'un GRAFCET G_n où aucune étape n'est active est dite « **situation vide** ». Elle se note respectivement $S = \{ \}$ ou $S(G) = \{ \}$ et $S(G_n) = \{ \}$.

Exemple

$S = \{1, 14, 15\}$ ou encore $S(G_m, G_n, G_p) = \{1, 4, 15\}$ signifient qu'à un certain moment de l'évolution du SAP, les étapes 1, 4 et 15 des GRAFCET G_m, G_n, G_p sont actives en même temps.

La situation du GRAFCET global G et d'un GRAFCET Gn à un moment donné est dite « **situation courante** ». Elle se note respectivement $S = \{ * \}$ ou $S(G) = \{ * \}$ et $S(Gn) = \{ * \}$.

Un GRAFCET représente sous forme statique un ensemble de situations possibles. En fonctionnement dynamique, seule une situation bien déterminée existe à un moment donné.

L'analyse de toutes les situations possibles de la PC à partir du GRAFCET permet de vérifier si son comportement est conforme aux spécifications fonctionnelles du cahier des charges.

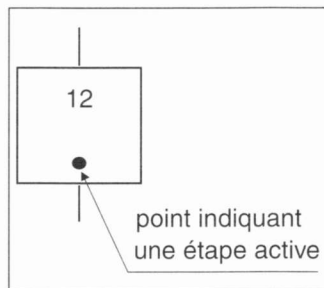


Figure 3.2
Marquage
d'une étape active.

Il est commode de pouvoir mettre en relief, en cours de discussion, directement sur le GRAFCET, quelles sont les étapes actives au moment considéré. A cet effet, on procède à un marquage de ces étapes (fig. 3.2) :

- au crayon par un point lorsqu'on travaille sur papier ou sur transparent en rétro-projection,
- par une inversion vidéo ou par un changement de couleur lorsqu'on travaille sur un écran de moniteur vidéo associé à une console de programmation.

Remarque

Une étape déjà activée peut être réactivée lors d'une évolution de situation sans que l'activité de celle-ci soit remise en cause.

Seules la ou les étapes précédentes sont désactivées (voir les règles d'évolutions).

La situation de la PC demande à être parfaitement définie lors de son démarrage. Cette situation particulière est dite situation initiale dans laquelle certaines étapes sont initialement activées.

On repère les étapes initiales en doublant le cadre symbolique de l'étape.

Certaines descriptions par GRAFCET hiérarchisés avec forçages de situations conduisent à distinguer les étapes initiales des étapes initialisables (fig. 3.3) :

- **étapes initiales** : étapes initialement activées par une procédure dépendant de la technologie de la PC.
- **étapes initialisables** : étapes non initiales activées par forçage à partir d'un GRAFCET hiérarchiquement supérieur et sans franchissement de transition.

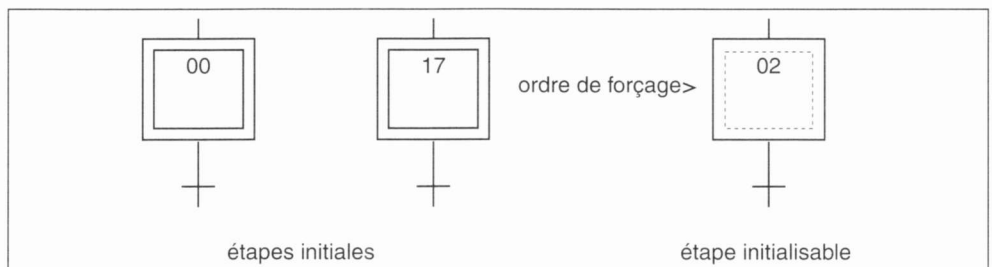


Figure 3.3
Représentation des étapes
initiales et initialisables.

Pour différencier l'étape initialisable de l'étape initiale, le cadre intérieur du symbole est dessiné en trait discontinu (pointillés ou tirets courts au choix).

Il faut souligner que sans étapes initiales activées le système ne peut pas démarrer.

Cette notion d'étape initiale et d'étape initialisable sera revue dans le cadre du forçage de situation étudié au chapitre III-13.

3. Actions associées à l'étape

Une ou plusieurs actions élémentaires ou complexes 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.

Dans le premier niveau de description (voir fig. 3.4), 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. 3.4) ou encore par un symbole (à expliciter par ailleurs) à l'intérieur d'un ou plusieurs rectangles reliés à l'étape.

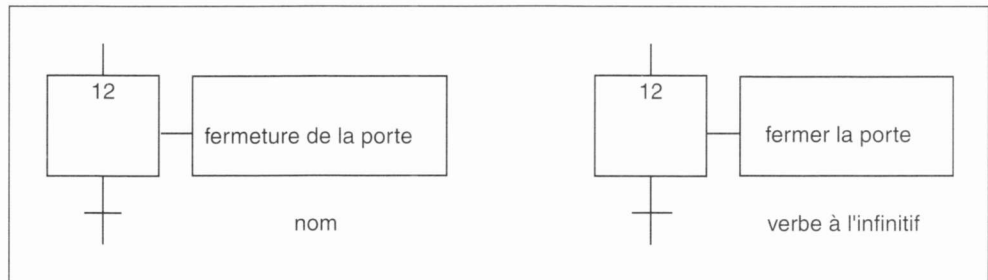


Figure 3.4
Désignation littérale
de l'action.

Dans le niveau suivant de description, 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.

Il s'agit le plus souvent d'ordres destinés aux entrées des préactionneurs : bobines de contacteurs, pilotages de distributeurs, bobines de relais électromagnétiques, etc.

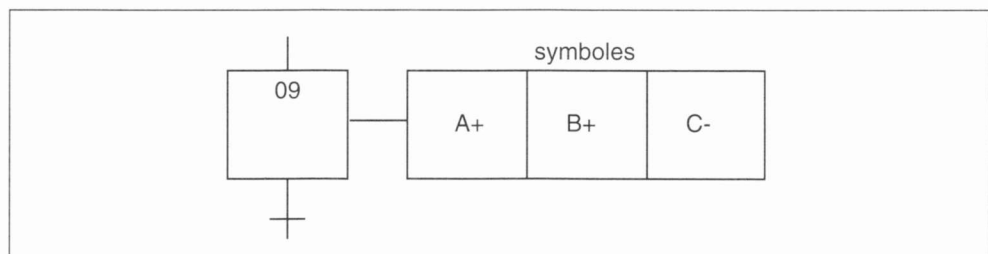


Figure 3.5
Repérage symbolique
des actions.

La notation, à ce niveau, se fait généralement de façon très symbolique, ce qui permet au GRAFCET de franchir l'obstacle des langues sur le plan international.

Un tableau récapitulatif indique alors dans la langue du pays la correspondance entre chaque symbole et l'action à exécuter ou l'ordre à émettre.

La figure 3.6 (page suivante) représente des actions associées à l'étape autorisées par la norme.

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, d'un niveau à l'autre, d'enchaîner le raisonnement sur les actions associées aux étapes, sans aucune ambiguïté.

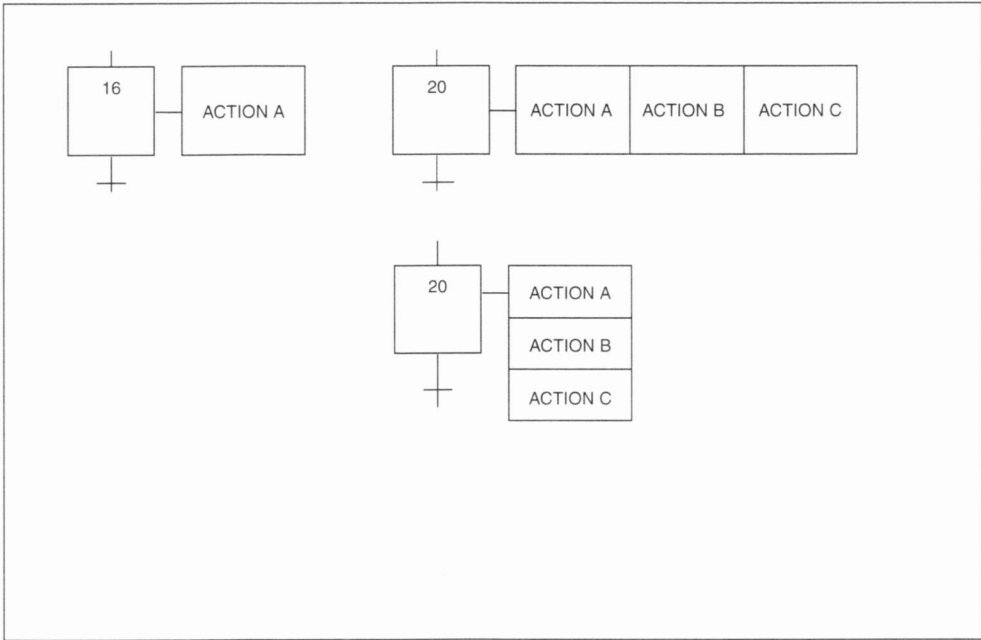


Figure 3.6
Représentation normalisée
des actions.

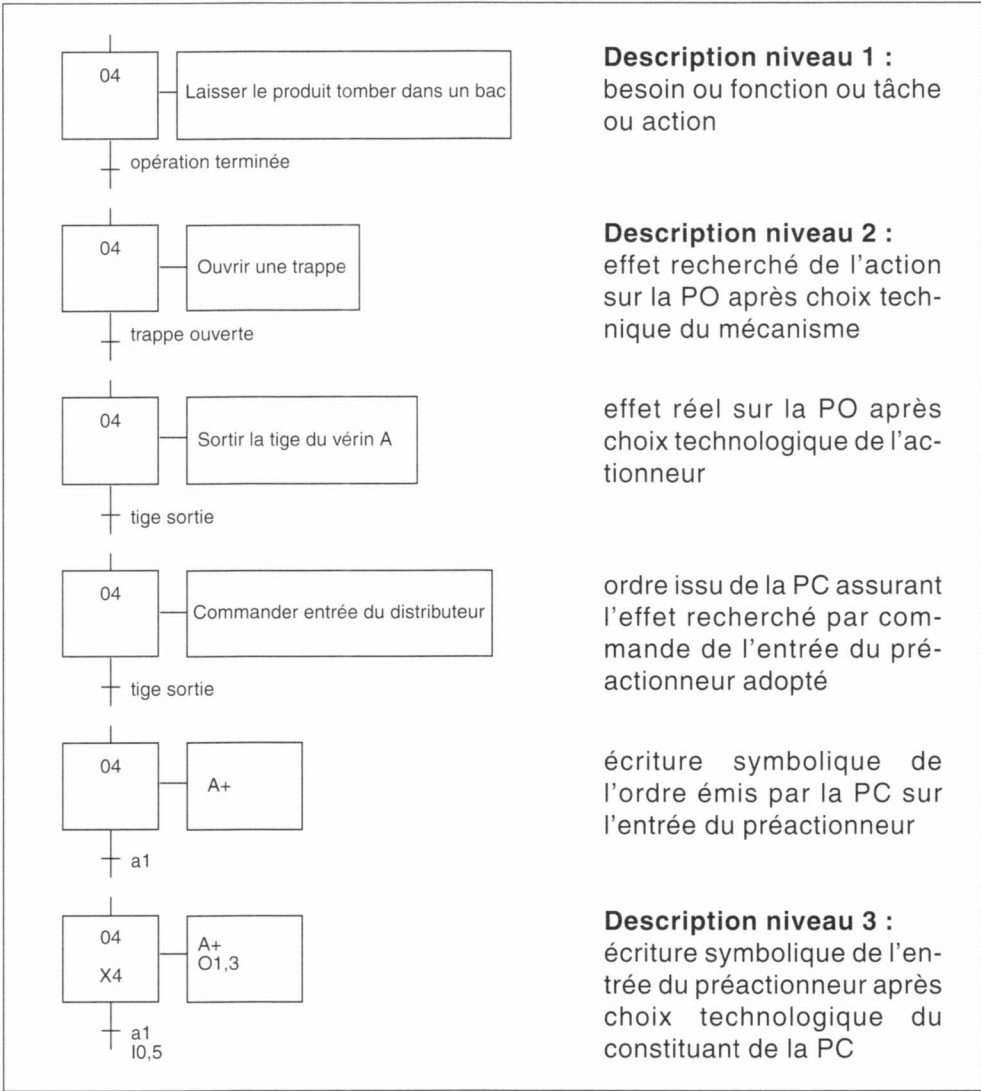


Figure 3.7
Enchaînement
du raisonnement
sur les actions.

Remarque au sujet de la notation du niveau 3

Ici le constituant retenu est un API TSX 67-40 de Schneider Télémécanique. Cet automate possède un langage GRAFCET graphique (PL7-3). Ce tracé servira de base à la programmation proprement dite à partir de la console (voir chapitre VII-7). Des précisions sont apportées sur les adresses des entrées-sorties et sur celles des bits internes, images des étapes, qui seront utilisées :

- X4 : bit image de l'étape 4,
- O1,3 : adresse de la sortie du TSX 67-40 (Output) émettant l'ordre vers A+,
- I0,5 : adresse de l'entrée du TSX 67-40 (Input) sur laquelle arrive le signal du capteur a1.

Il faut noter l'évolution parallèle de l'écriture de la condition de transition associée à la transition 4 → 5, elle-même fonction du niveau.

La nature et le classement des actions seront étudiés au chapitre III-11.

4. Transitions entre les étapes

Une transition marque la 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 de la PC.

Une transition est soit validée soit non validée.

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

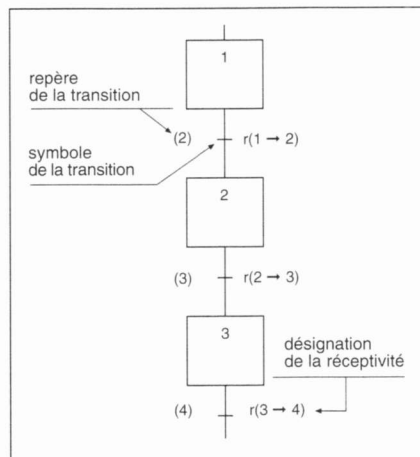


Figure 3.8
La transition
et la condition de
transition associée.

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.

Pour faciliter la discussion du GRAFCET, les transitions peuvent être repérées, entre parenthèses, à gauche du tiret.

Mais pour ne pas alourdir le GRAFCET, il est souvent plus simple de désigner la transition par la ou les étapes amont et la ou les étapes aval (fig. 3.8).

Exemples (fig. 3.9)

- 1/ TRANSITION 3 → 4 :
elle autorise l'évolution de la situation $S = \{3\}$ vers la situation $S = \{4\}$,
- 2/ TRANSITION 2, 3 → 4, 5 :
elle autorise l'évolution de la situation $S = \{2, 3\}$ vers la situation $S = \{4, 5\}$,
- 3/ TRANSITION 5 → 6, 7 :
elle autorise l'évolution de la situation $S = \{5\}$ vers la situation $S = \{6, 7\}$,
- 4/ TRANSITION 12, 13 → 14 :
elle autorise l'évolution de la situation $S = \{12, 13\}$ vers la situation $S = \{14\}$.

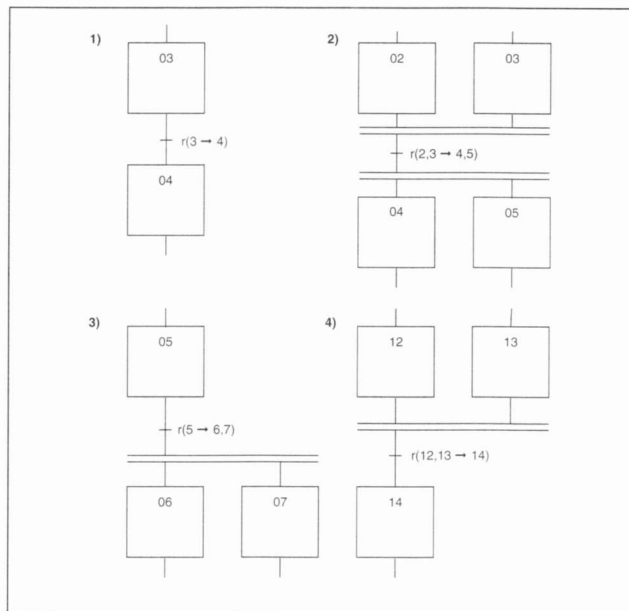


Figure 3.9
Exemples de transitions.

On remarquera que lorsque plusieurs étapes sont reliées à la même transition, les liaisons orientées correspondant à ces étapes sont regroupées en amont et en aval sur deux traits parallèles horizontaux.

5. Conditions de transitions associées aux transitions

A chaque transition est associée une proposition logique appelée **réceptivité** (NF C 03-190 de 1995) ou encore **condition de transition** (UTE C 03-190 de 1990 et CEI 848). Mais les deux termes coexistent dans la documentation UTE C 03-191 de 1993. Les conditions de transitions peuvent être représentées soit par des indications littérales, soit par des expressions logiques soit par symboles graphiques (schéma à contacts, logigramme, etc.).

La condition de transition peut être vraie ou fausse c'est-à-dire égale à 1 ou à 0.

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

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 :
 - temporisateurs émettant un signal noté $t/\text{étape}$ de lancement/durée,
 - compteurs émettant un signal noté $ci = n$ où n est la valeur de la présélection de leur contenu,
 - décompteurs émettant un signal noté $di = 0$ lors du retour à zéro de leur contenu,
 - etc. ;
- 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 égale à 1 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**.

Notations : front montant : \uparrow , front descendant : \downarrow

Exemple

$\uparrow a$ = front montant de la variable a.



Figure 3.10
Représentation graphique
des fronts montants et
descendants.

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

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

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

- opérateur NON ou Complémentation noté « / » (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 ET noté « . »,
- opérateur OU noté « + ».

Pour la discussion, la réceptivité peut être notée $r(i \rightarrow j)$ où i et j sont respectivement les étapes amont et aval ou faire l'objet d'une numérotation entre parenthèses (documentation UTE C 03-191). La notation $r(i \rightarrow j)$ quoique plus lourde est aussi plus explicite.

Exemple 1

Évolution liée à des conditions.

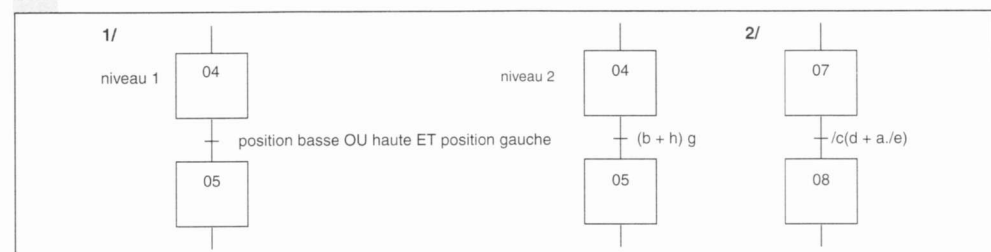


Figure 3.11
Évolutions liées
à une condition.

Remarque

Les chronogrammes des exemples 2 à 6 (pages suivantes) sont établis en tenant compte de l'hypothèse faite au sujet du délai de franchissement d'une transition, cette durée étant considérée comme non nulle (voir chapitre III-9).

Exemple 2

Évolution liée à des événements

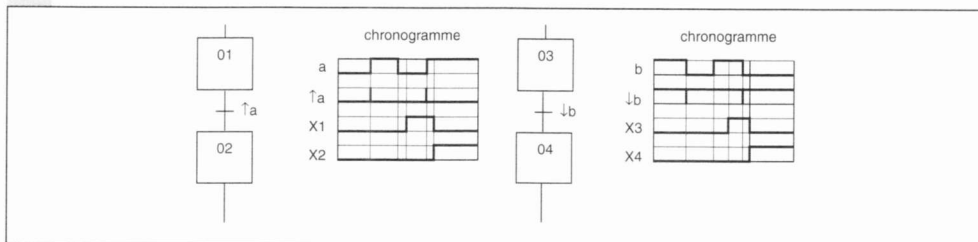


Figure 3.12
Évolutions liées
à des événements.

Exemple 3

Évolution liée à une combinaison de conditions et d'événements

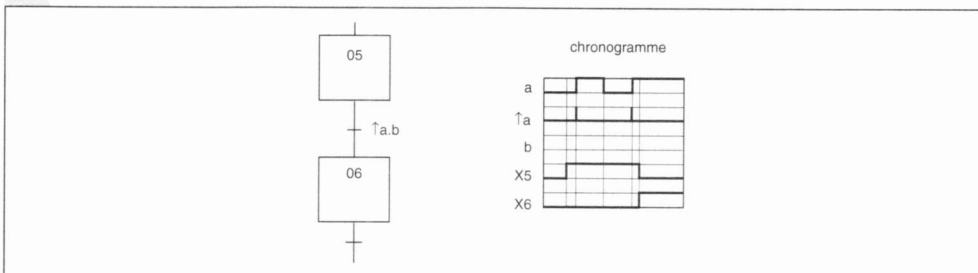


Figure 3.13
Évolution liée
à une combinaison
de conditions
et d'événements.

Exemple 4

Évolution liée à une condition interne

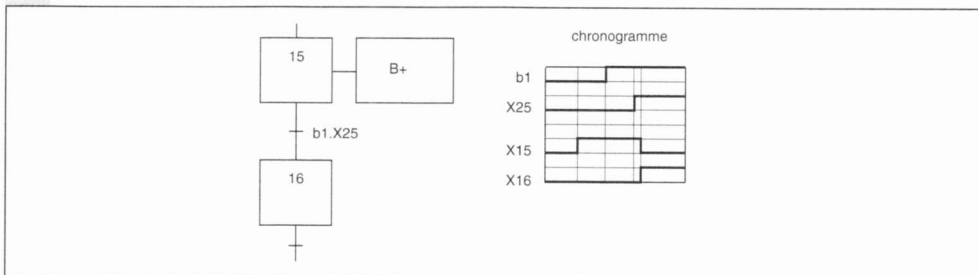


Figure 3.14
Évolution liée
à une condition interne.

Exemple 5

Évolution liée au temps

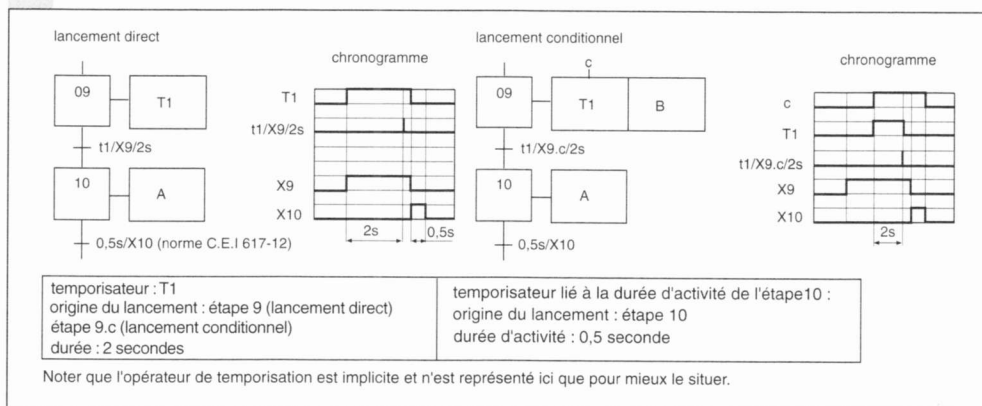


Figure 3.15
Évolution liée au temps.

Exemple 6

Évolution liée à un comptage

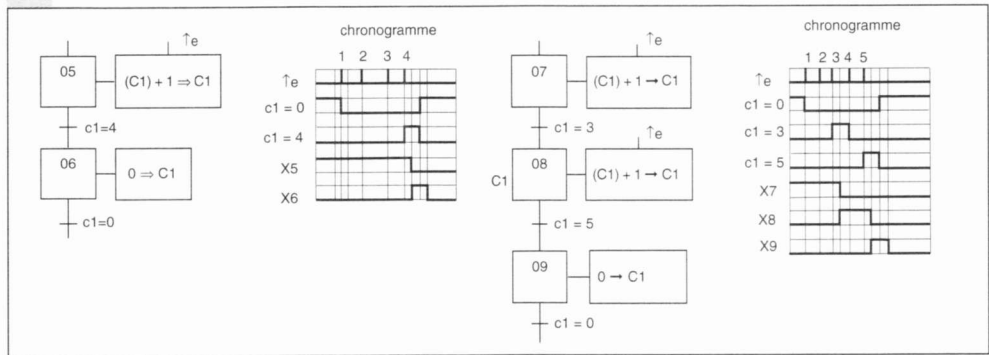


Figure 3.16
Évolution liée
à un comptage.

Exemple 7

Évolution liée à des conversions analogiques-numériques (ANA/NUM)

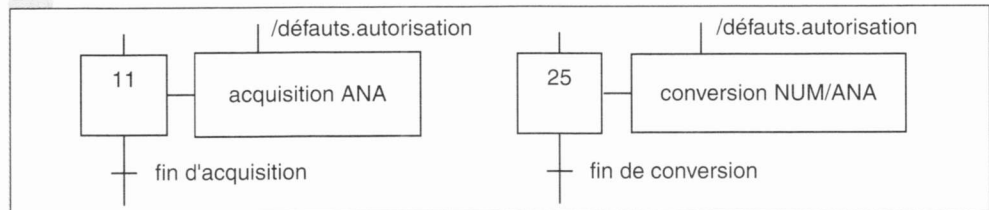


Figure 3.17
Évolution liée
à des conversions
analogiques-numériques.

Exemple 8

Évolution liée à une condition de transition toujours vraie

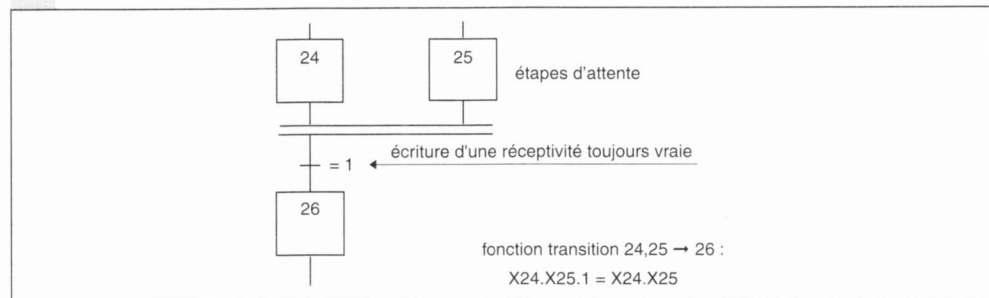


Figure 3.18
Évolution liée
à une condition de
transition toujours vraie.

A propos de la stabilité d'une situation (voir addenda en fin d'ouvrage)

Une situation est dite stable, si cette situation est conservée tant qu'aucun ordre externe, donc en provenance d'une entrée externe, n'est émis.

Il faut rappeler, d'après les postulats temporels :

- que la durée d'une **situation stable** est non nulle à l'échelle de temps externe,
- que la durée d'une **situation instable** est nulle à l'échelle de temps externe.

La condition de transition « =1 » étant un ordre interne, la situation $S = \{24,25\}$ de la figure 3.18 est instable. Toute action, image d'une sortie externe et associée à l'une de ces étapes, ne peut être exécutée.

Il ne faut donc pas associer d'actions images de sorties externes aux étapes amont d'une condition de transition toujours vraie.

Voir le cas contraire d'une sortie interne associée à l'étape amont d'une condition de transition toujours vraie, fig. 3.31.

6. 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. 3.19). 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 pas de connexion.

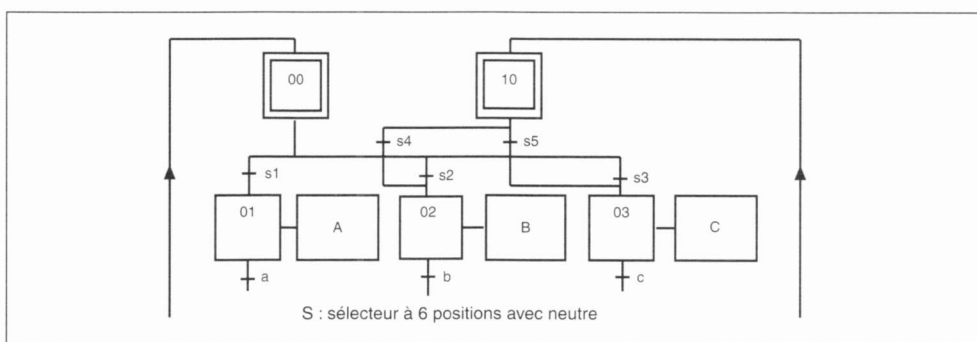


Figure 3.19
Tracé des liaisons avec
et sans connexions.

Le double trait horizontal (ISO 5807) sert à regrouper plusieurs liaisons liées à la même transition, celle-ci conditionnant l'évolution entre deux situations mettant en jeu plus de deux étapes (fig. 3.20) :

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

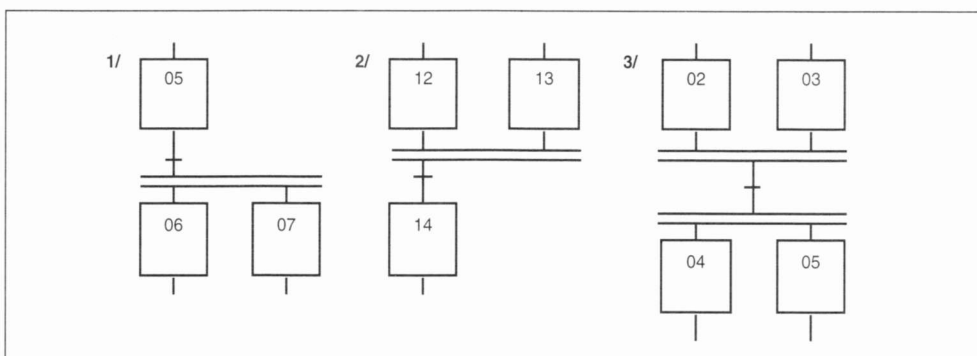


Figure 3.20
Représentation des liaisons
dans le cas d'une transition
liée à plus de 2 étapes.

A titre mnémotechnique, on peut dire :

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

7. Règles de syntaxe*

Un GRAFCET est correctement tracé s'il respecte la règle d'alternance étape-transition et transition-étape. Ceci signifie en clair que deux étapes ne peuvent être reliées directement et que deux transitions ne peuvent pas se suivre.

8. Règles d'évolution

Ces règles sont essentielles dans la définition du modèle GRAFCET.

Règle 1 relative à la situation initiale de la PC

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.

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 (forçage de situation).

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

La situation initiale doit avoir un comportement passif (non-émission d'ordre) vis-à-vis de la PO, celle-ci se trouvant dans une position de repos particulière choisie par le concepteur comme position de référence.

Par contre, il n'est pas interdit de profiter de la mise en situation initiale pour émettre des ordres internes à la PC : remise à zéro (RAZ) des compteurs, initialisation des temporisateurs, etc. Il est possible aussi d'y associer des ordres externes s'ils sont conditionnels.

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.

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).

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

Note _____

*Cf chapitre IV.

Et pour terminer, comparaison entre les trois états possibles de la PC :

1/ **PC hors énergie** : la PC n'a pas d'existence physique vis-à-vis de sa PO, le concept de situation est ici sans signification.

2/ **PC en énergie et inactive** : la situation de la PC est vide et celle-ci n'est donc pas réceptive à aucune des informations pouvant être émises par la PR et/ou par la PO,

3/ **PC en énergie et active** : au moins une étape est active (étape initiale) et la PC devient réceptive aux informations émises par la PR et/ou par la PO.

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 est nommée **fonction transition**.

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

Noter qu'une transition est validée lorsque toutes les étapes immédiatement précédentes sont actives.

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).

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 transitions les états actifs des étapes (fig. 3.21).

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

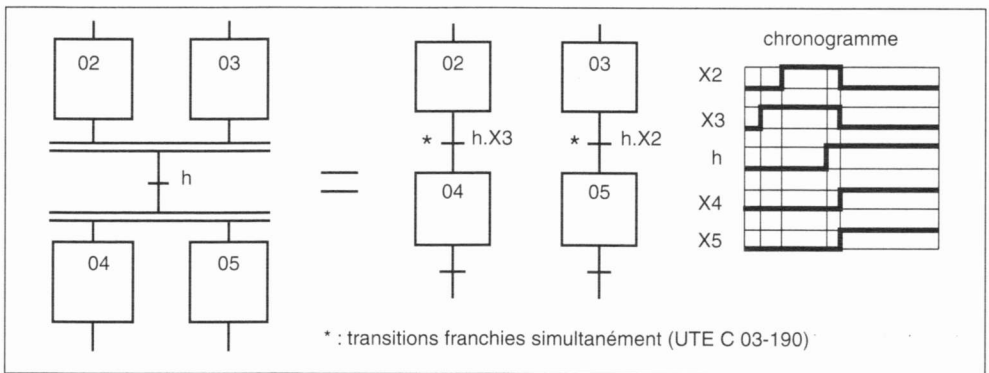


Figure 3.21
Illustration de la règle 4.

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.

Exemple illustrant les règles 2 et 3

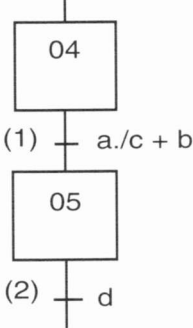
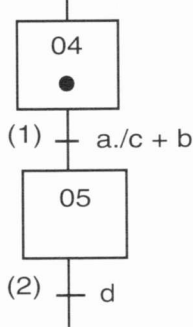
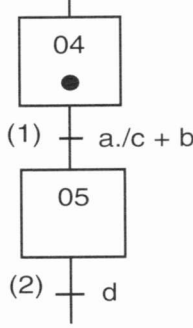
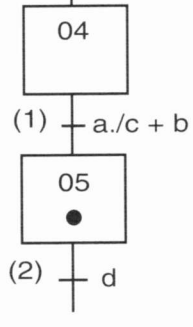
GRAF CET	état logique de la réceptivité $r(4 \rightarrow 5)$	commentaires
 <p>Diagram showing two steps: 04 and 05. Step 04 is active. Transition (1) is labeled $a./c + b$. Step 05 is inactive. Transition (2) is labeled d.</p>	0 ou 1	transition (1) non validée car l'étape 4 est inactive
 <p>Diagram showing two steps: 04 and 05. Step 04 is active and has a dot in the center. Transition (1) is labeled $a./c + b$. Step 05 is inactive. Transition (2) is labeled d.</p>	0	transition (1) validée car l'étape 4 est active mais non franchissable car la réceptivité n'est pas vraie
 <p>Diagram showing two steps: 04 and 05. Step 04 is active and has a dot in the center. Transition (1) is labeled $a./c + b$. Step 05 is inactive. Transition (2) is labeled d.</p>	1	transition (1) franchissable et obligatoirement franchie car elle est validée et la réceptivité est vraie
 <p>Diagram showing two steps: 04 and 05. Step 04 is inactive. Step 05 is active and has a dot in the center. Transition (1) is labeled $a./c + b$. Transition (2) is labeled d.</p>	0 ou 1	transition (1) franchie avec, simultanément, activation de l'étape 5 et désactivation de l'étape 4 transition (2) maintenant validée

Figure 3.22
Illustration des règles
d'évolution 2 et 3.

Autre exemple illustrant les règles 2 et 3

GRAF CET	état logique $r(5,6,7 \rightarrow 8,9)$	commentaires
	0 ou 1	transition (1) non validée car l'étape 6 est encore inactive
	0 de la réceptivité	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 3.23
Autre illustration
des règles 2 et 3.

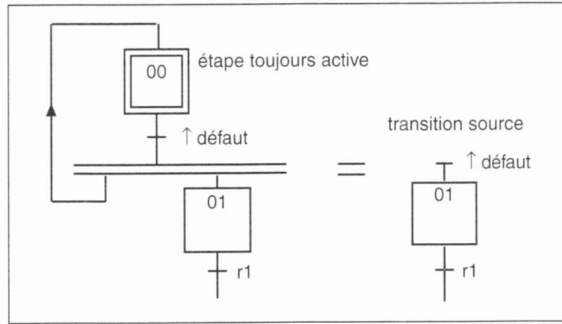
Cas d'applications particulières des règles d'évolution

1/ Transition toujours validée (fig. 3.24)

La situation initiale correspond à $S = \{0\}$. Lorsque la condition de transition associée à la transition $0 \rightarrow 1$ est vraie, le GRAFCET évolue vers la situation $S = \{0,1\}$, l'étape 0 est à la fois activée et désactivée, donc elle reste activée d'après la règle 5.

Si en cours de fonctionnement, on se trouve dans la situation $S = \{0,n\}$ avec n différent de 1, le franchissement de la transition $0 \rightarrow 1$, si la condition de transition associée est vraie, conduira vers la situation $S = \{0,1,n\}$.

Figure 3.24
Transition toujours validée.

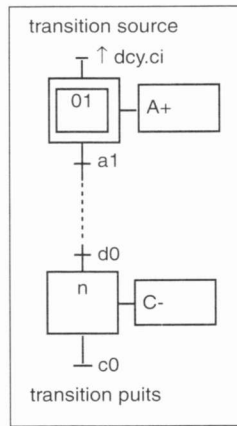


La suppression de l'étape toujours active en tête de GRAFCET est possible et la transition isolée devient alors une transition source.

Par définition une transition source est une transition sans étape précédente et par suite elle est toujours validée.

2/ Transition source et transition puits

Figure 3.25
Transition source et transition puits.



Par définition une transition puits est une transition sans étape immédiatement suivante.

La transition puits conduit à une situation vide.

Noter les expressions logiques découlant de ces transitions particulières :

activation étape 1 : $A(1) = \uparrow dcy.ci$

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

Noter qu'il peut aussi exister une étape source sans transition précédente et une étape puits sans transition suivante (voir chapitre X.2).

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 de son GRAFCET ou par forçage à 0.

3/ Registre à décalage

L'exemple est inspiré d'une chaîne d'enrobage de fromages.

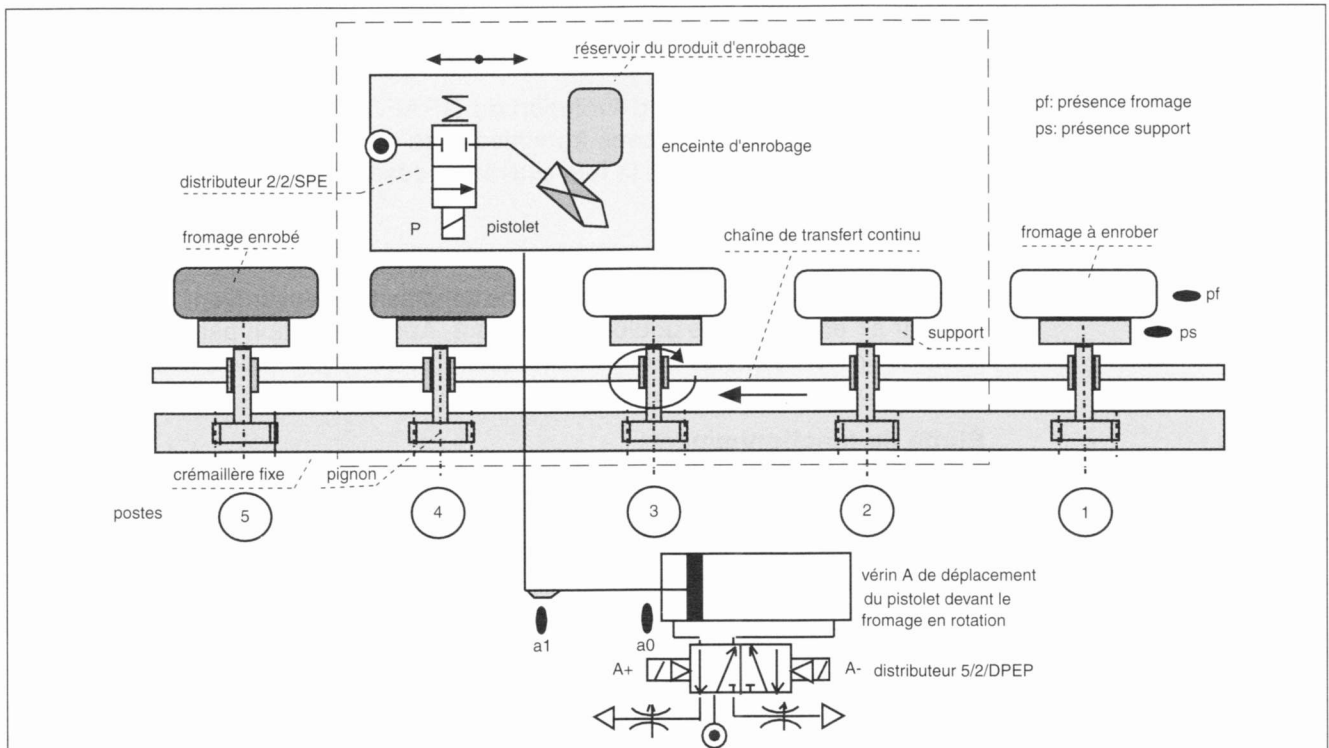
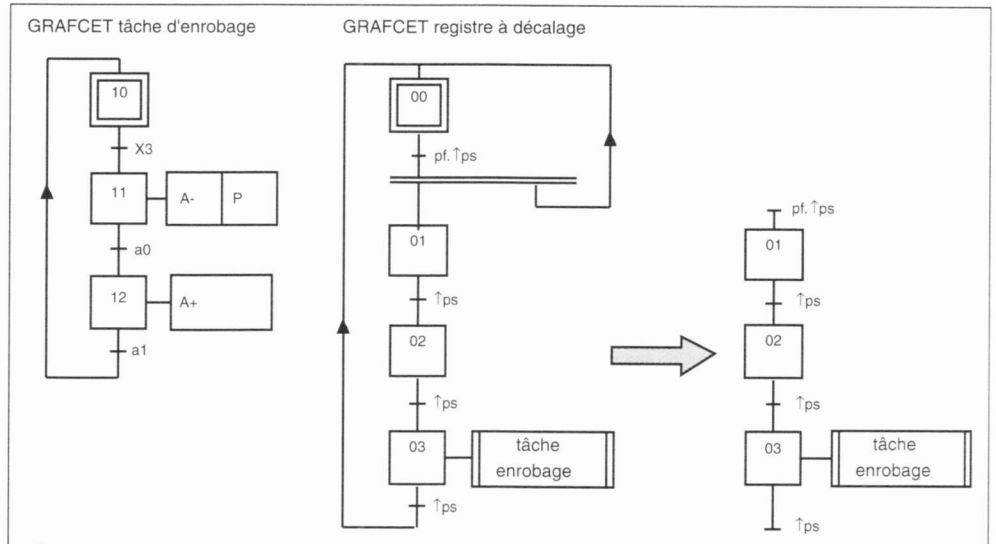


Figure 3.26 : Schéma du poste d'enrobage des fromages.

Figure 3.27
 GRAFCET de la tâche
 d'enrobage et GRAFCET
 du registre à décalage.



Étude du GRAFCET du poste d'enrobage des fromages

- $S = \{0\}$ et pf et $\uparrow ps = 1 \rightarrow S = \{0, 1\}$,
- $S = \{0, 1\}$ et pf et $\uparrow ps = 1 \rightarrow S = \{0, 1, 2\}$,
- $S = \{0, 1, 2\}$ et pf et $\uparrow ps = 1 \rightarrow S = \{0, 1, 2, 3\}$,
- $S = \{0, 1, 2, 3\}$ et $\neg pf$ et $\uparrow ps = 1 \rightarrow S = \{0, 2, 3\}$ support vide en P1,
- $S = \{0, 2, 3\}$ et pf et $\uparrow ps = 1 \rightarrow S = \{0, 1, 3\}$ support vide en P2,
- $S = \{0, 1, 3\}$ et pf et $\uparrow ps = 1 \rightarrow S = \{0, 1, 2\}$ support vide en P3.

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 l'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 ».

4/ Registre à empilage

Des cartons provenant de deux chaînes de conditionnement arrivent, suivant sélection par s_1 et s_2 , par les goulottes 1 et/ou 2. Après détection par p_1 et p_2 (informations $\downarrow p_1$ et $\downarrow p_2$), les cartons sont transférés sous le poste de marquage. Une fois tamponnés, les cartons sont évacués, 4 par 4, vers le poste de palettisation.

Étude du fonctionnement

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.

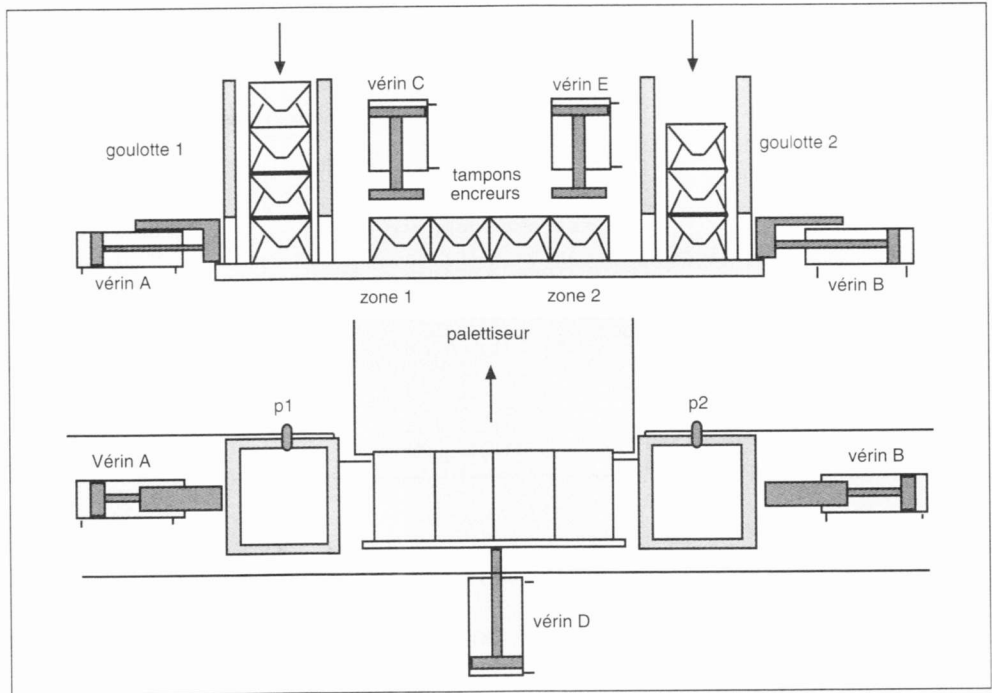


Figure 3.28
Poste d'empilage.

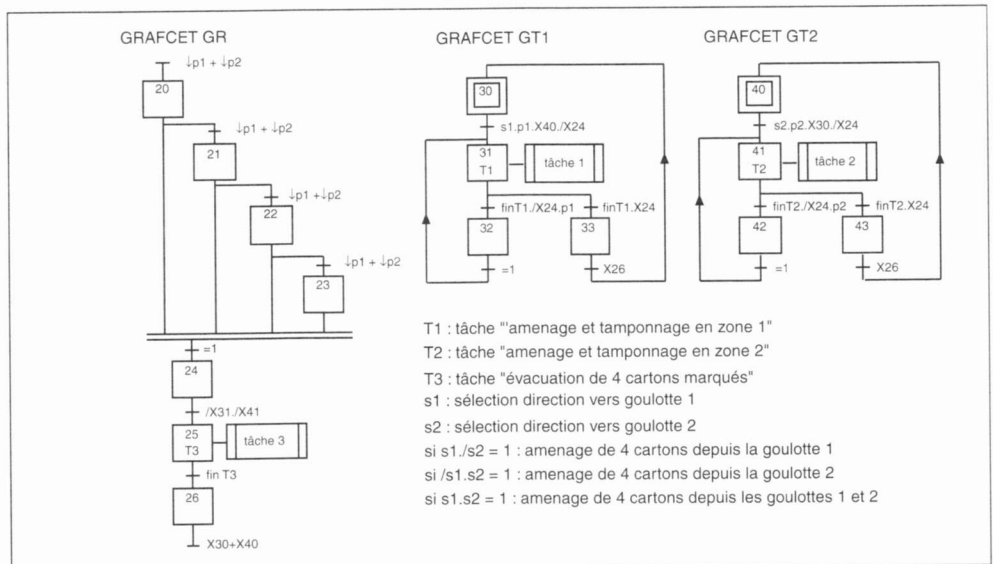


Figure 3.29
GRAFCET du poste
d'empilage.

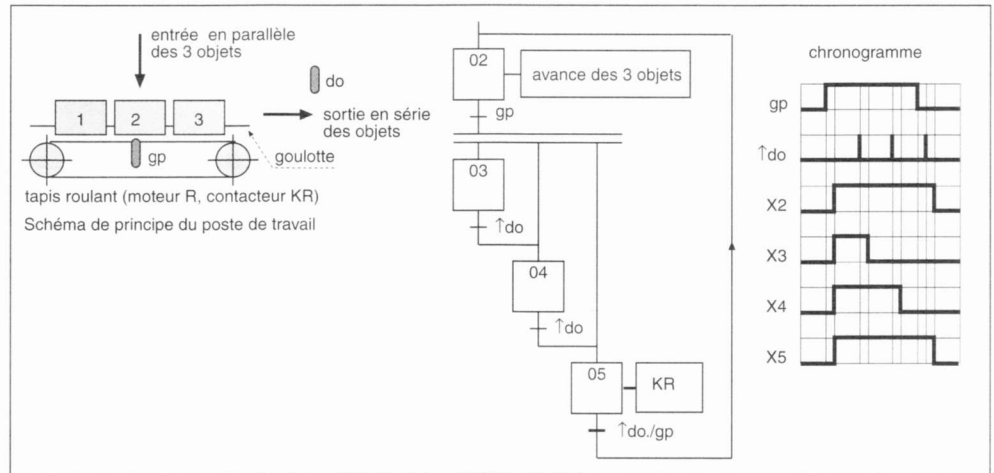
5/ Registre à déempilage (figure 3.30 page suivante)

Après avoir rempli en parallèle une goulotte avec 3 objets (information goulotte pleine ou gp) il faut sortir les objets en série, c'est-à-dire séparément les uns derrière les autres et suivre l'opération de déempilage par passage devant un détecteur d'objet (information $\uparrow do$).

Étude du fonctionnement

- Si $X2.gp = 1$ alors $S = \{3, 4, 5\}$,
- première détection en sortie série : $S = \{4, 5\}$, l'étape 4 activée et désactivée reste activée, l'étape 5 activée et réactivée reste activée, l'étape 3 est désactivée,
- deuxième détection en sortie série : $S = \{5\}$, l'étape activée et réactivée reste activée, l'étape 4 est désactivée,
- goulotte vide et troisième détection en sortie série : $S = \{2\}$.

Figure 3.30
Poste de désemballage.



EXEMPLE 1 : la bascule.
A chaque impulsion sur le bouton poussoir m, 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 : l'hypothèse faite 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. En revanche, l'action A (sortie externe) ne sera jamais exécutée (échelle de temps interne différente de l'échelle de temps externe*).

* Voir détails sur les postulats temporels, d'après la documentation UTE C 03-191, dans l'addenda page 219.

Figure 3.31
Durées relatives aux évolutions (transition et activité d'étape).

9. Postulats sur les durées

Il s'agit, ici, de compléments au modèle GRAFCET destinés à rendre les évolutions possibles et non ambiguës*.

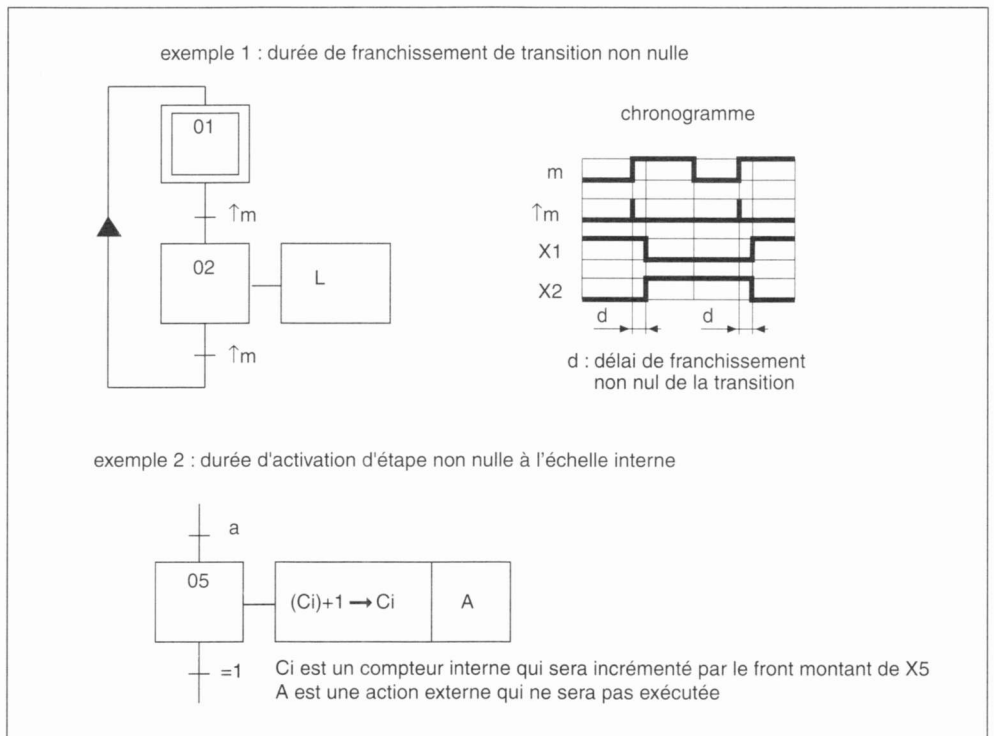
9.1. Cas des transitions et des étapes (à l'échelle de temps interne *)

1°. Durée de franchissement d'une transition

Cette durée peut être considérée aussi petite que l'on veut mais non nulle.

2°. Durée d'activation d'une étape

Conséquence du 1°, cette durée ne sera jamais nulle.



9.2. Cas des événements (fronts montants et descendants)

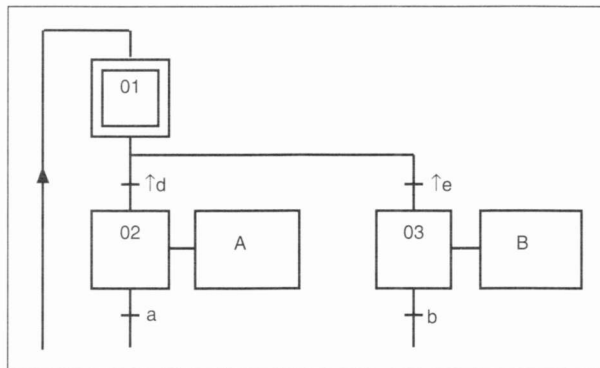
A. Simultanéité d'événements externes à la PC

Le modèle GRAFCET exclut formellement la simultanéité d'occurrence de deux événements externes non corrélés.

Par occurrence, on entend ici l'existence au même moment de deux événements distincts.

Deux événements sont corrélés, c'est-à-dire qu'ils ont une relation logique réciproque, dans les deux cas suivant :

- s'ils sont associés à la même variable : $\uparrow a$ et $\downarrow a$,
- s'ils sont associés à deux variables logiques ayant une intersection commune :
 $X = \uparrow a + c$ si par exemple $b = 1$ et $c = 0$ alors X et Y sont corrélés.
 $Y = \uparrow a . b$



Conséquence : si deux événements ne sont jamais en occurrence, la structure de la figure 3.32 correspond à une sélection de séquence.

Figure 3.32
Sélection de séquence
si les deux événements ne
sont pas en occurrence.

B. Simultanéité d'événements internes à la PC

Le modèle GRAFCET impose la simultanéité d'occurrence de plusieurs événements internes. La règle 3 relative à l'évolution de la situation impose donc la simultanéité de l'activation et de la désactivation des étapes aval et amont de la transition franchie. Il en est de même de la règle 4 relative aux évolutions simultanées.

Quelques propriétés des combinaisons de conditions et d'événements

- $\uparrow(a+b)$ est différent de $\uparrow a + \uparrow b$ (fig. 3.33)

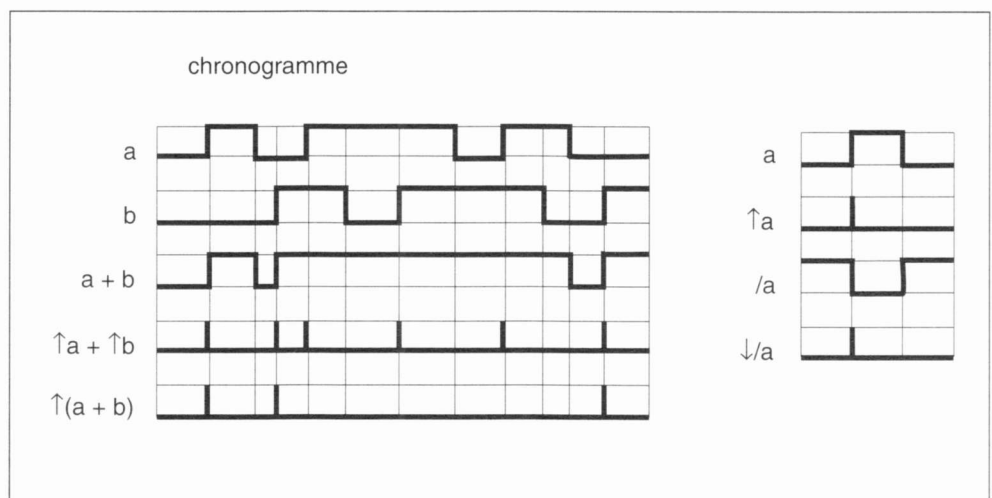


Figure 3.33
 $\uparrow(a+b)$ est différent de
 $\uparrow a + \uparrow b$
et $\uparrow a = \downarrow /a$

- $\uparrow a = \downarrow /a$ est vraie (fig. 3.33)
- $\uparrow(a.b)$ est différent de $\uparrow a . \uparrow b$ (fig. 3.34)

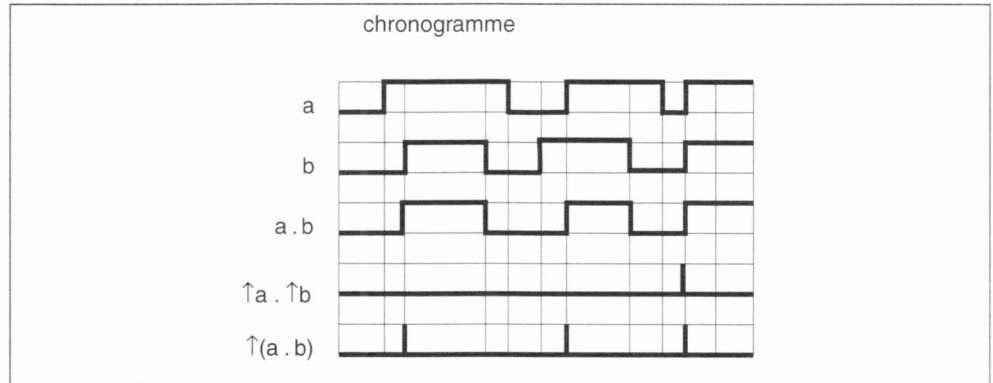


Figure 3.34
 $\uparrow(a.b)$ est différent de $\uparrow a . \uparrow b$.
 Cette propriété n'est vraie
 que si les variables a et b
 sont associées à
 des événements internes.

9.3. Délais d'un point de vue PC ou réalisateur

A ce niveau, la PC a une existence physique et il faut tenir compte des composants retenus pour la réaliser :

- les postulats formulés sur la durée non nulle de franchissement d'une transition et sur la durée non nulle d'activation d'une étape sont en accord avec les temps de réponse des composants et des constituants dans une réalisation aussi bien du type câblé que programmé ;
- la simultanéité d'événements externes serait admissible dans une réalisation programmée (API, par exemple) si l'intervalle de temps entre deux scrutations des entrées était supérieur à l'intervalle de temps séparant les événements. Mais cette hypothèse est à écarter vu la durée de plus en plus faible des cycles de scrutation des API et des micro-ordinateurs actuels ;
- la simultanéité de changement d'état de variables internes (cas des étapes) qui correspond à une évolution synchrone est impossible dans des réalisations câblées du type séquenceur à cause du temps de réponse des modules matérialisant les étapes : relais électromagnétiques bistables ou distributeurs à double pilotage suivant la technologie employée, électrique ou pneumatique.

Aussi fait-on une entorse aux règles 2, 3 et 4 et accepte-t-on une évolution de type asynchrone pour les réalisations à base de séquenceur (voir chapitre VI.3).

9.4. Problème des fronts montants et descendants

A. Détection directe en vérifiant les états successifs de la variable

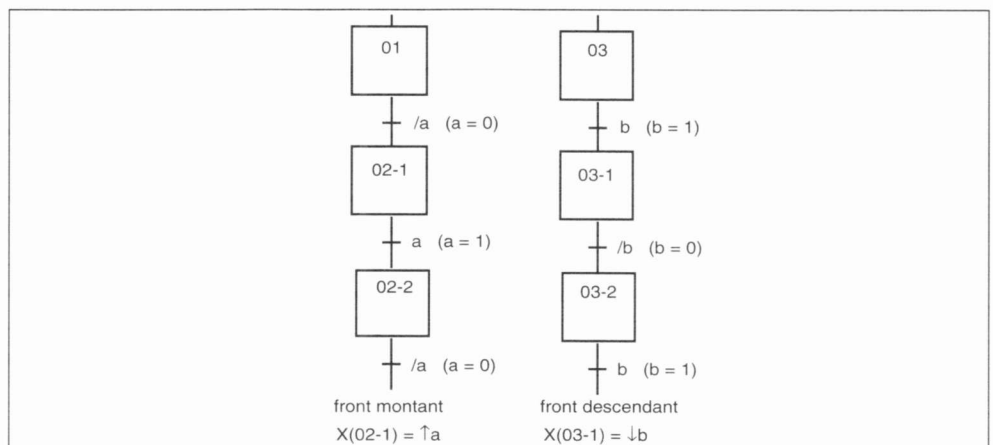


Figure 3.35
 Détection directe des fronts
 montants et descendants.

B. Détection par GRAFCET particulier

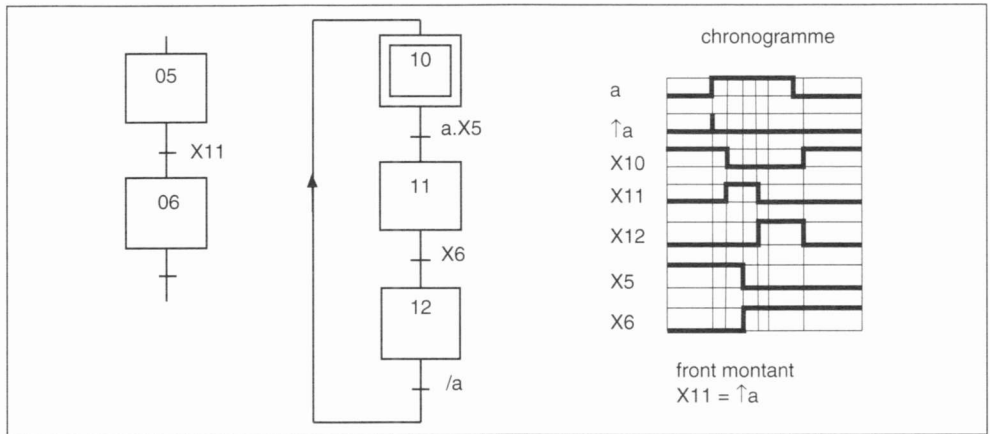


Figure 3.36
Détection d'un front montant par GRAFCET particulier.

C. Détection en langage à contacts d'API

Dans la plupart des API, la mise à 1 d'un bit interne pendant un cycle de scrutation du programme suffit à matérialiser le front montant d'une variable externe ou interne à la PC.

Exemple : variable externe : capteur a, front montant interne : B1

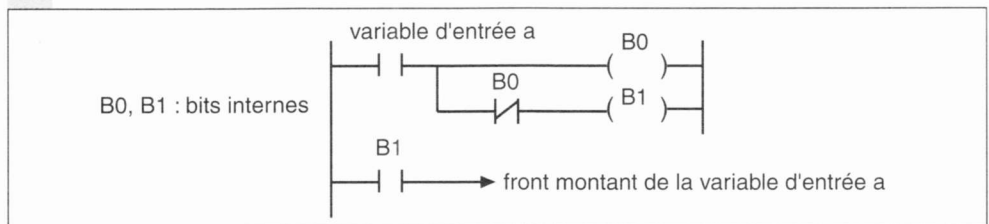


Figure 3.37
Détection d'un front montant par programmation en langage à contacts.

Noter que certains API disposent, dans leur langage, de fronts montants pré-programmés :

- langage des PB de Schneider Electric (zone 08E0 à 08EF, par exemple),
- langage PL7-3 des TSX SchneiderTélémeccanique.

D. Détection d'un front associé à une condition

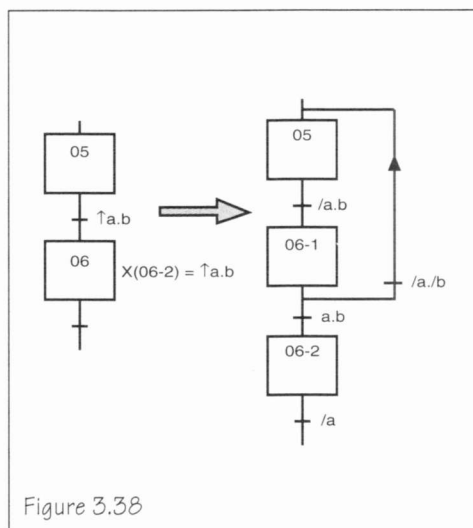


Figure 3.38
Détection directe d'un front montant associé à une condition.

Figure 3.39
Détection par GRAFCET particulier d'un front montant associé à une condition.

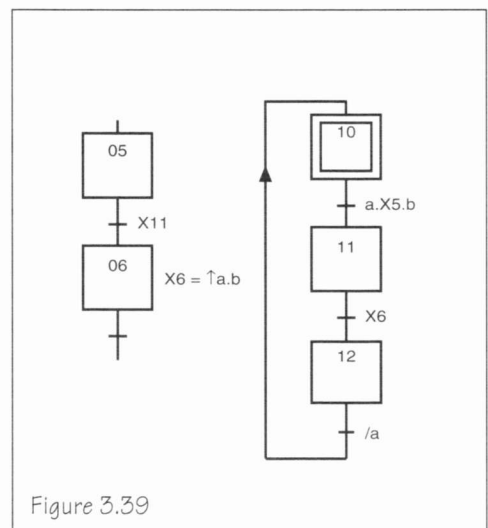


Figure 3.39

E. Détection de fronts associés dans une relation logique

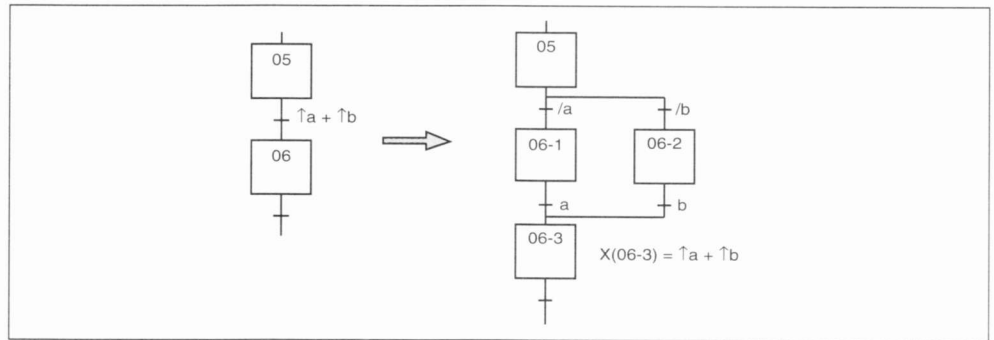


Figure 3.40
Détection directe
de fronts montants associés
dans une relation logique.

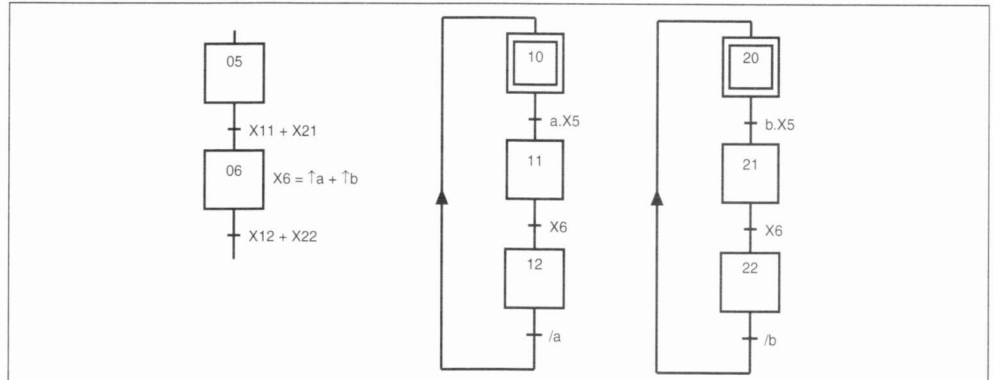


Figure 3.41
Détection par
GRAFCET particulier de
fronts montants associés
dans une relation logique.

F. Détection de changements d'états non synchronisés

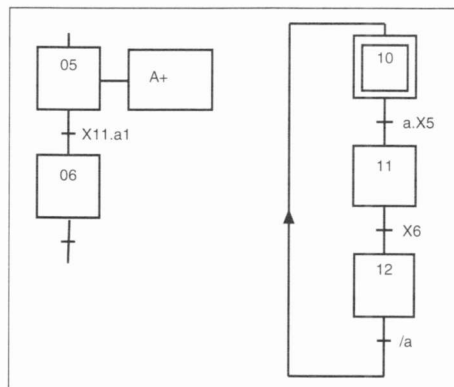


Figure 3.42
Détection par GRAFCET
particulier de changements
d'états non synchronisés.
L'étape 11 mémorise le front
montant sur a dans
l'attente de la condition a1.

Pour évoluer, la PC doit parfois prendre en compte un signal fugitif (comparable à un front) un certain temps après sa disparition. La solution passe alors par la mise en mémoire de cet événement.

9.5. Temporisateurs

A. Opérateur temporisateur du modèle GRAFCET

Cet opérateur implicite et interne à la PC est défini formellement dans le GRAFCET :

- par l'origine de la temporisation toujours liée à l'activation d'une étape et à son passage de l'état inactif ($X_i = 0$) à l'état actif ($X_i = 1$), soit $\uparrow X_i$ (front montant) ;
- par la variable de temporisation notée $t/X_i/q$ où X_i est l'étape d'origine de lancement de la temporisation et q sa durée exprimée avec l'unité adéquate ;
- par le fait que la variable de temporisation est forcée à 0 pendant toute la durée de la mesure du temps puis forcée à 1 à la fin de cette mesure.

Il découle de cette définition que si l'étape i est désactivée puis réactivée pendant la mesure du temps, le nouveau front montant $\uparrow X_i$ annule la mesure en cours et relance une nouvelle mesure.

Si l'étape i est activée et désactivée simultanément, puisque d'après la règle 5 elle reste activée, la mesure du temps se poursuit normalement (pas de front montant $\uparrow X_i$).

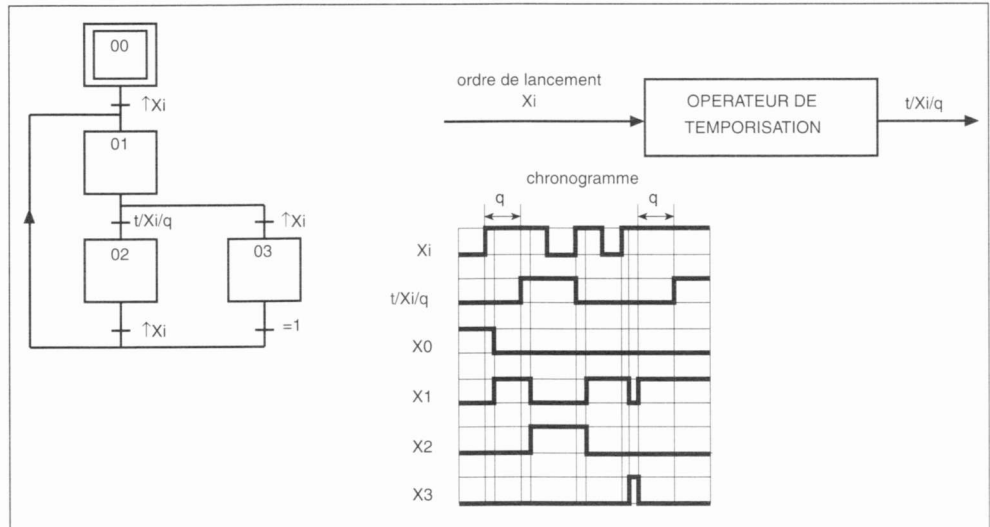


Figure 3.43
Description
par GRAFCET de l'opérateur
temporisateur formel
et fictif.

B. Opérateur temporisateur d'un point de vue PC ou réalisateur

La mesure du temps en automatique fait appel :

- soit à des temporisateurs analogiques (fig. 3.44) :
 - du type électronique (circuit RC) : charge électrique d'un condensateur C au travers d'une résistance R limitant le courant de charge. Au bout d'un temps t la tension aux bornes du condensateur atteint une valeur image du niveau logique 1, par exemple $5\text{ V} = \text{niveau } 1 (< 4,5\text{ V} = \text{niveau } 0)$;
 - du type pneumatique : le retard est obtenu par un amortisseur pneumatique à vitesse réglable (relais électromagnétique à contact temporisé pneumatiquement) ;
 - du type mécanique : déplacement ou rotation d'un doigt entraîné à vitesse constante par un motoréducteur synchrone. Ce doigt agit en fin de temporisation sur un capteur fin de course (minuterie électromécaniques) ;

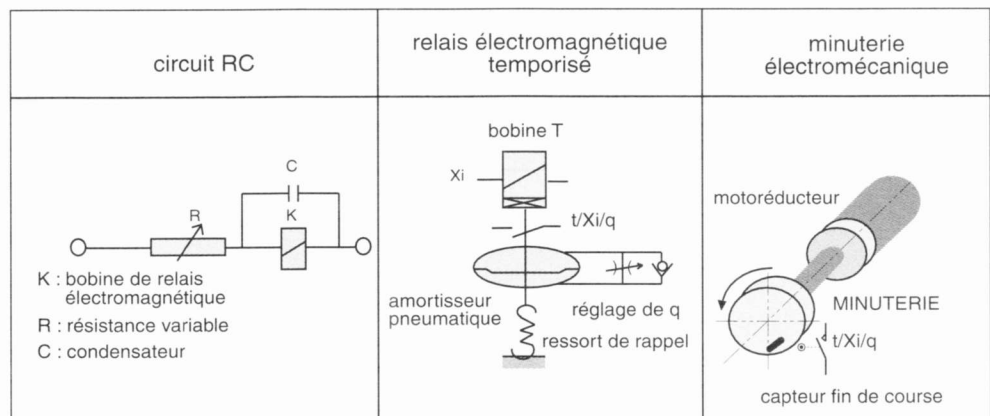


Figure 3.44
Temporisateurs analogiques.

- soit à des temporisateurs numériques obtenus (fig. 3.45) :
 - par décomptage ou comptage d'impulsions délivrées par une horloge à fréquence connue, (fixe ou programmable et stable dite base de temps).
 - Lorsque le nombre d'impulsions décomptées ou comptées est égal respectivement à 0 ou à une valeur présélectionnée n , la sortie d'un comparateur est mise à 1.
 - Cette sortie $di = 0$ ou $ci = n$ est l'image de la variable de temporisation t .

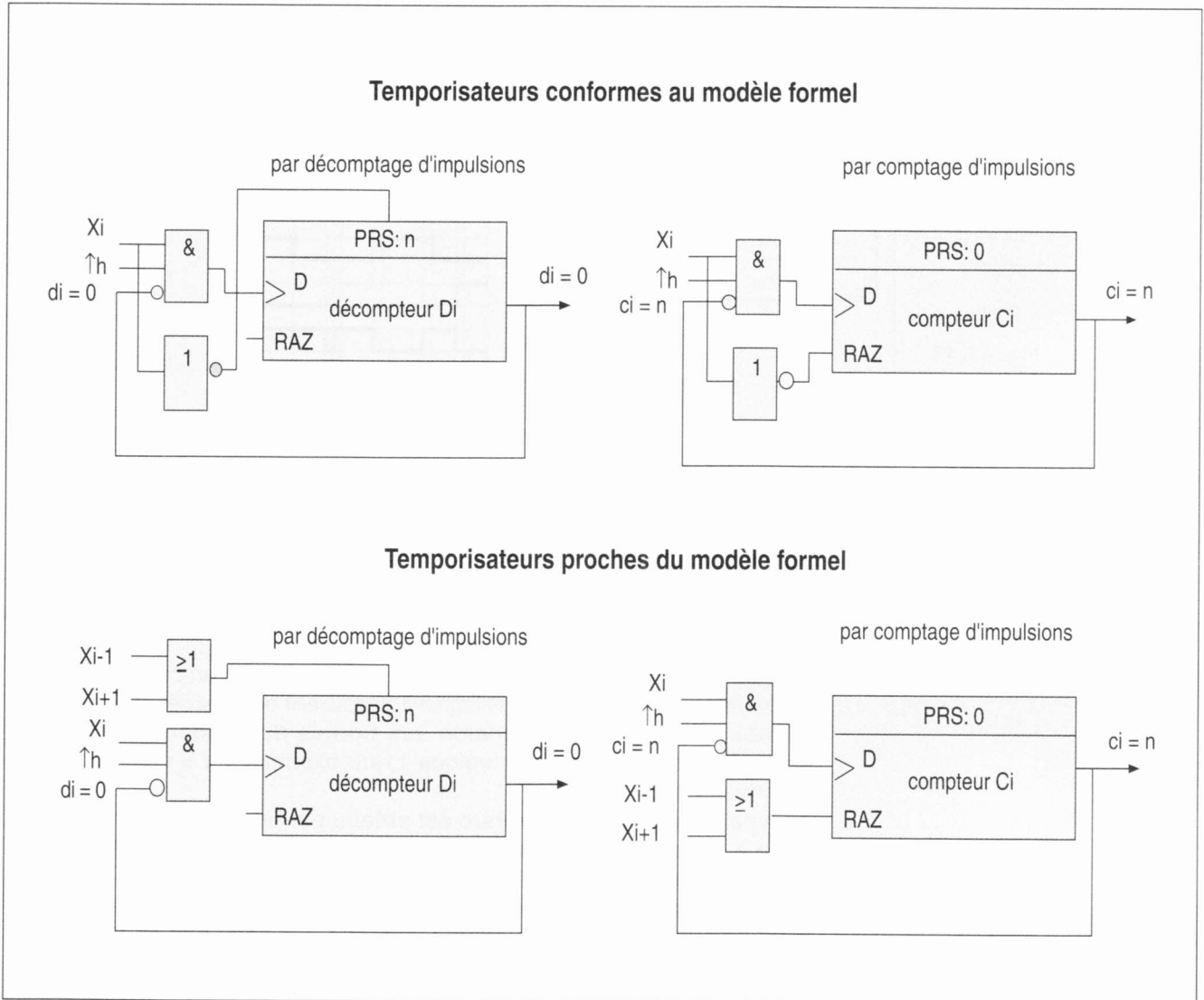
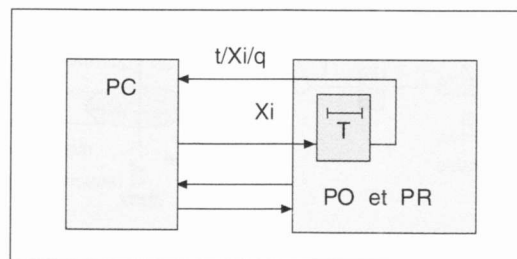


Figure 3.45
Réalisation
des temporisateurs à l'aide
de compteurs et de
décompteurs.

- Les temporisateurs existent sous forme de constituants isolés (fig. 3.46) :
- relais électromagnétiques à temporisation pneumatique ou électronique,
 - minuteries électromécaniques ou électroniques.

Figure 3.46
Fonction opérative
de traitement (FOT).



Ils sont alors vus de la PC comme appartenant à la PO, l'ordre de lancement venant de la PC.

En fin de temporisation, le temporisateur émet une information $t/X_i/q$ sur l'entrée de la PC. Le temporisateur dans ce cas réalise une fonction opérative de traitement (FOT).

Ils existent aussi sous la forme de blocs-fonctions temporisateurs intégrés dans le logiciel de l'API.

Dans les langages PL7-2 et PL7-3 de Schneider Télémécanique, le temporisateur est identifié par la lettre T et un numéro i .

En fin de temporisation, le temporisateur fournit le signal Ti,D exploitable directement. La sortie temps écoulé D est exploitée pour commander un bit interne Bi .

Les variables Ti,D et Bi au choix représentent l'image de la variable de temporisation $t/Xi/q$.

Pour obtenir ce résultat, il faut paramétrer le temporisateur en choisissant une base de temps BT (minute, seconde ou 1/10 de seconde) et une valeur n de façon que $q = BT.n$ (fig. 3.47).

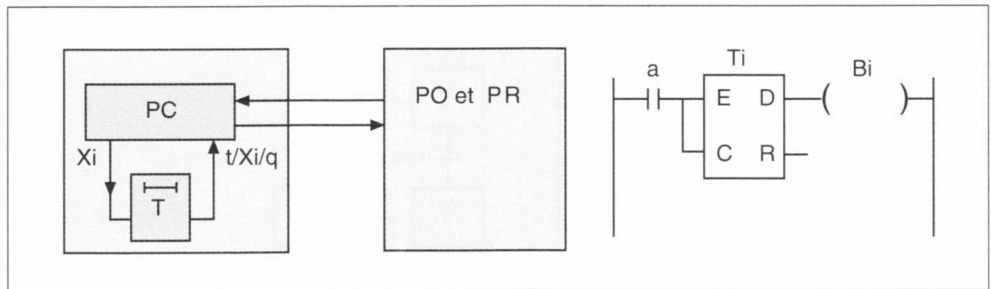


Figure 3.47
Temporisateurs intégrés
au logiciel de l'API

Les temporisateurs peuvent être créés :

- soit par l'intermédiaire du logiciel de l'API à partir de blocs-fonctions décompteurs ou compteurs (fig. 3.45),
- soit par l'intermédiaire de mots en langage à contacts (fig. 3.48).

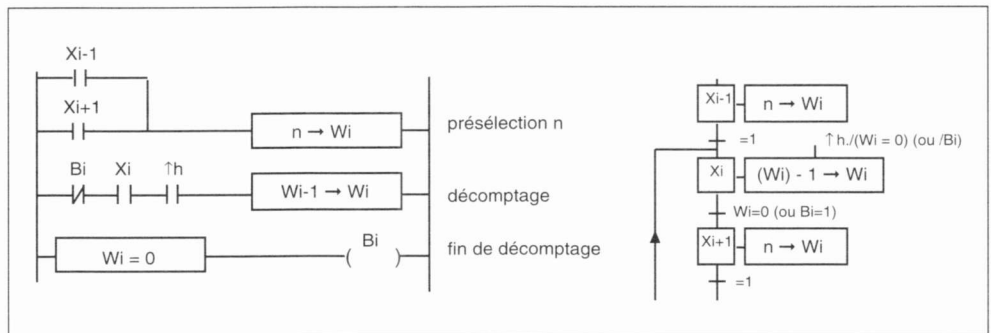


Figure 3.48
Temporisateur par
décomptage sur mot
en langage à contacts.

Les temporisateurs numériques créés à la demande sont construits en respectant, au mieux, le modèle formel.

Il faut d'abord faire le choix entre un décompteur Di ou un compteur Ci ou entre une instruction de décomptage ou de comptage si on travaille sur un mot Wi .

Ensuite il reste à choisir :

- la base de temps $\uparrow h$,
- la valeur de présélection : n pour un décompteur ou 0 (RAZ) pour un compteur,
- la variable de temporisation $t/Xi/q$ associée à $di = 0$ ou $ci = n$ ou $Wi = 0$ ou $Wi = n$.

Quelques exemples d'intégration des temporisateurs dans un GRAFCET

Exemple 1 : GRAFCET avec temporisateur externe à la PC (fig. 3.49)

Le temporisateur est matérialisé par un constituant de la PO, jouant le rôle de fonction opérative de traitement : relais électromagnétique temporisé, minuterie, horloge, etc.

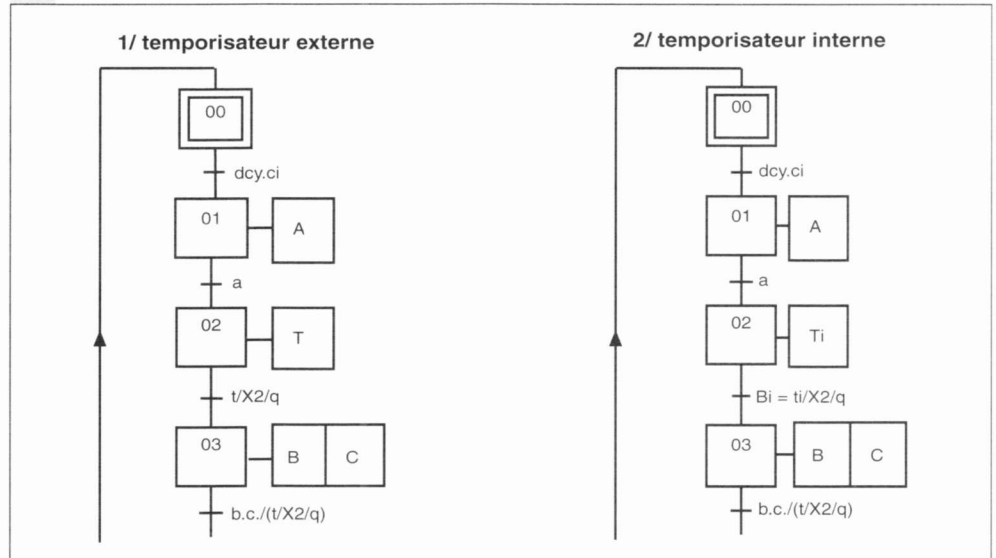


Figure 3.49
GRAFCET avec temporisateur externe ou interne à la PC.

Exemple 2 : GRAFCET avec temporisateur interne à la PC (fig. 3.49)

Le temporisateur est intégré à la PC : le logiciel dispose de blocs-fonctions temporisateurs appelables par leurs adresses (voir aussi fig. 3.47).

Exemple 3 : GRAFCET avec temporisation numérique (fig. 3.50)

Le temporisateur est créé à partir d'un bloc-fonction décompteur D_i ou d'un bloc-fonction compteur C_i ou d'un mot W_i et d'une base de temps H . La réalisation est proche du modèle formel (voir fig. 3.45).

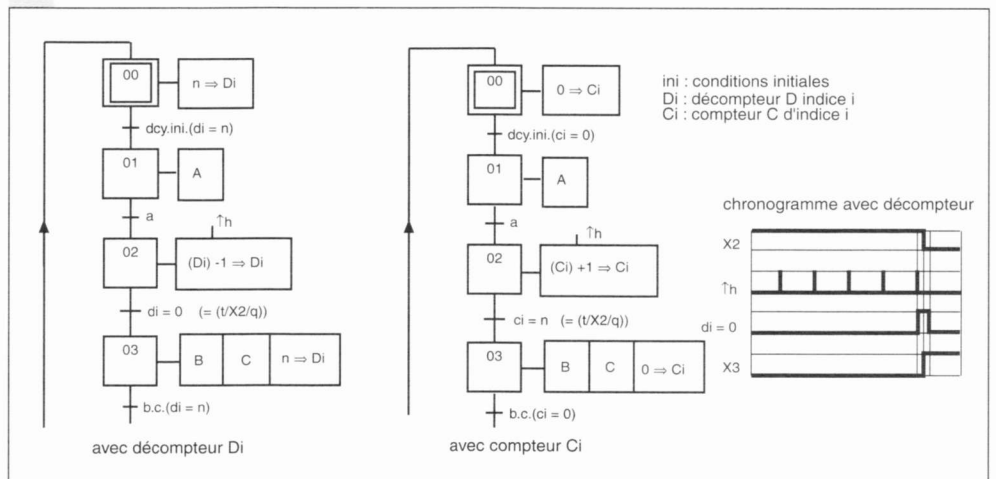


Figure 3.50
Temporisateurs numériques créés à partir d'un décompteur ou d'un compteur.

10. Structures de base

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

10.1. Séquence unique

La séquence unique est composée d'une suite d'étapes alignées verticalement pouvant être activées les unes après les autres.

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

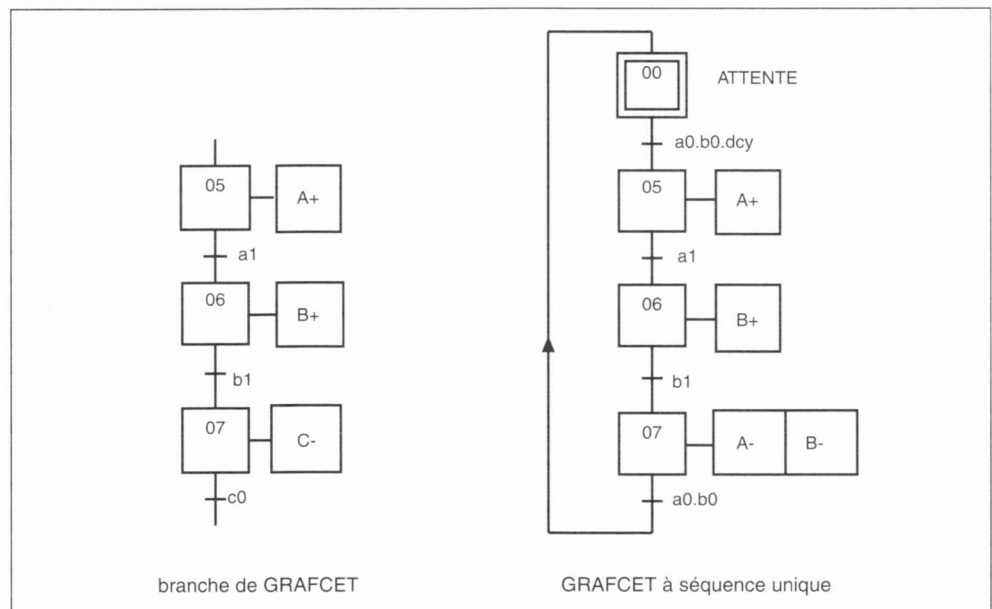
La séquence unique peut correspondre soit à une branche de GRAFCET soit à un GRAFCET complet.

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 à plusieurs séquences avec une seule étape active à la fois est dit GRAFCET à étape active unique.

Une séquence avec une ou plusieurs étapes actives à la fois est dite séquence unique active.

Une séquence avec aucune étape active est dite séquence unique inactive.



10.2. Séquences simultanées

Si le franchissement d'une transition conduit à activer plusieurs étapes en même temps, ces étapes déclencheront des séquences dont les évolutions seront à la fois simultanées et indépendantes. Pour désigner ce type de structure, on dit qu'on se trouve en présence d'un **parallélisme structural**.

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

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

Avant la publication de la norme 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.

E^xemple de séquences simultanées

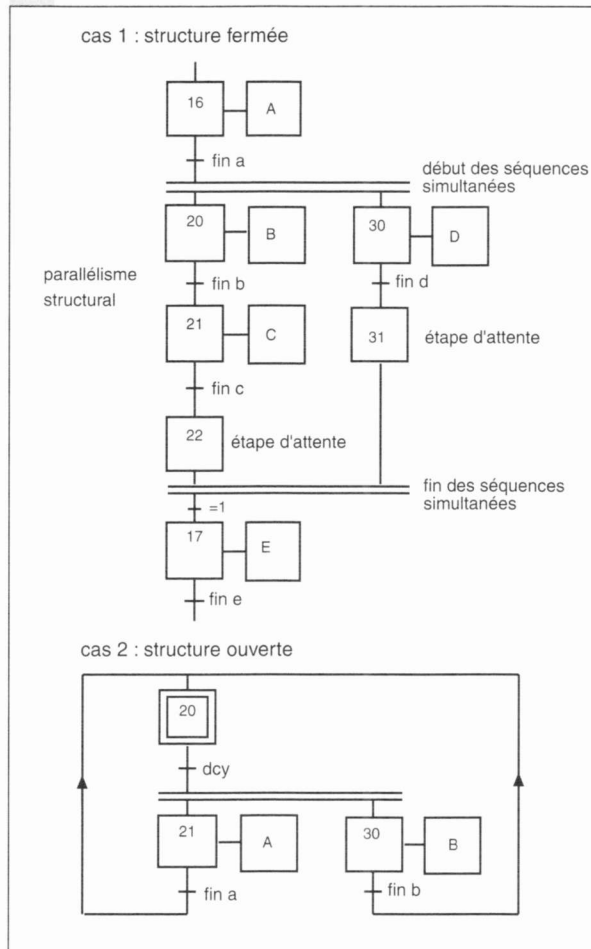


Figure 3.52
Parallélisme structural.

situation $S = \{20, 30\}$. Si la condition de transition "dcy" devient à nouveau vraie, la situation du GRAFCET évolue de $S = \{20, 30\}$ vers la situation $S = \{21, 30\}$. L'étape 30 activée et réactivée reste activée. L'évolution est du même genre si étant dans la situation $S = \{21, 30\}$ la condition de transition "fin b" devient vraie. Si étant dans la situation $S = \{21, 30\}$ les conditions de transitions "fin a" et "fin b" sont vraies simultanément, l'évolution est comparable à celle de la sortie du parallélisme structural du cas 1. On se retrouve en situation $S = \{20\}$.

CAS 1 : structure fermée

Le système étant dans la situation $S = \{16\}$, si la condition de transition "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. Lorsque les étapes 22 et 31 sont actives ($S = \{22, 31\}$), la transition $22, 31 \rightarrow 17$ 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 : structure ouverte

Le GRAFCET étant dans la situation $S = \{20\}$, si la condition de transition "dcy" devient vraie, la situation du GRAFCET évolue vers la situation $S = \{21, 30\}$. Si la condition de transition "fin a" devient vraie, la situation du GRAFCET évolue de $S = \{21, 30\}$ vers la

R^emarque

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.

10.3. Sélections de séquence

Une sélection de séquences ou un choix d'évolutions entre plusieurs étapes ou plusieurs séquences se représente, à partir d'une ou plusieurs étapes, par autant de transitions validables et franchissables qu'il y a d'évolutions possibles.

A. Exemples de séquences exclusives

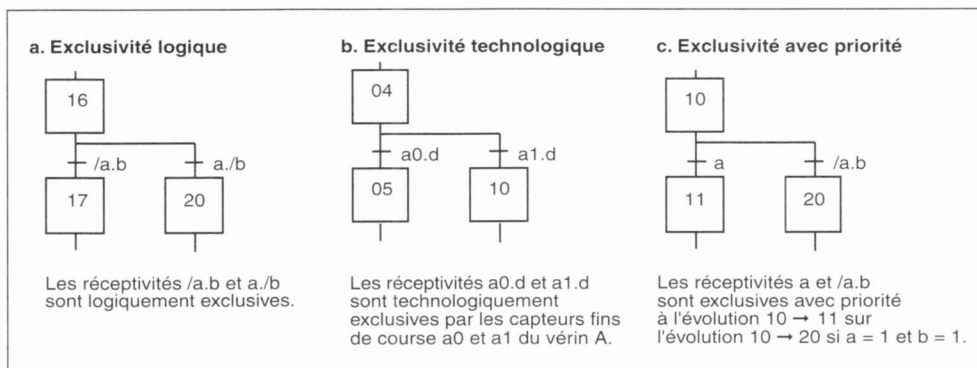


Figure 3.53
Sélections de séquences.

B. Sélection entre deux évolutions à partir de deux étapes

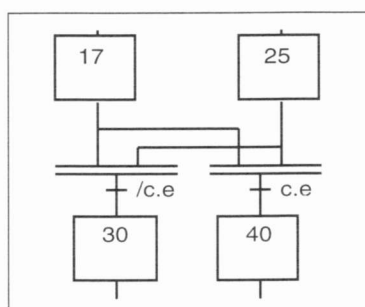


Figure 3.54
Sélection entre deux évolutions.

Évolutions possibles : $S = \{17, 25\}$ vers $S = \{30\}$ ou $S = \{17, 25\}$ vers $S = \{40\}$ suivant l'état de la variable c.

C. Cas particuliers

a. Saut d'étapes

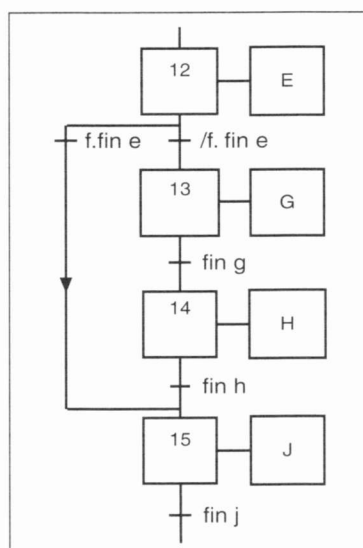


Figure 3.55
Saut d'étapes.

Ce type d'aiguillage permet de sauter une ou plusieurs étapes lorsque les actions associées à ces étapes sont sans objet à un moment donné.

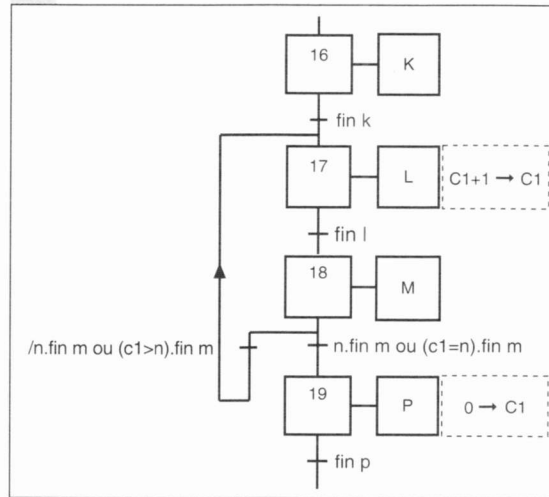
Exemple (fig. 3.55)

- Saut de l'étape 12 vers l'étape 15 si la condition de transition fin f.e est vraie,
- sélection par la variable f,
- les actions G et H ne seront pas exécutées tant que la variable f restera à l'état 1.

b. Reprise de séquence

Ce type d'aiguillage permet de reprendre une ou plusieurs fois la même séquence tant qu'une condition donnée n'est pas remplie.

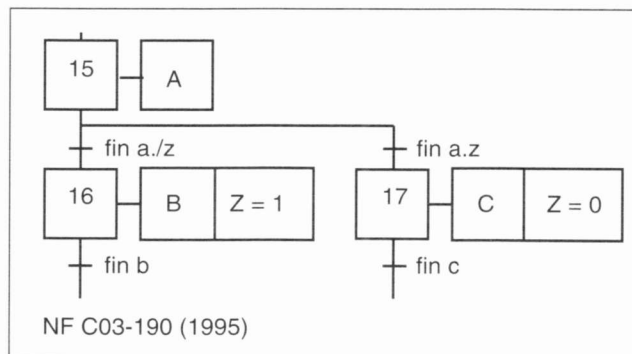
Exemple (fig. 3.56)



- Reprise de la séquence 17-18 si la condition de transition /n.fin m (ou (c1<n).fin m est vraie),
- la sélection est liée à la variable n (ou au test du contenu du compteur C1),
- les actions L et M sont successivement exécutées tant que l'état de la variable n est égal à 0 (ou que le contenu du compteur C1 reste inférieur à n).

Figure 3.56
Reprise de séquence.

c. Sélection alternative par drapeau



L'action Z associée à l'étape 16 et à l'étape 17 joue le rôle de drapeau (flag) ou plus simplement d'aiguilleur automatique.

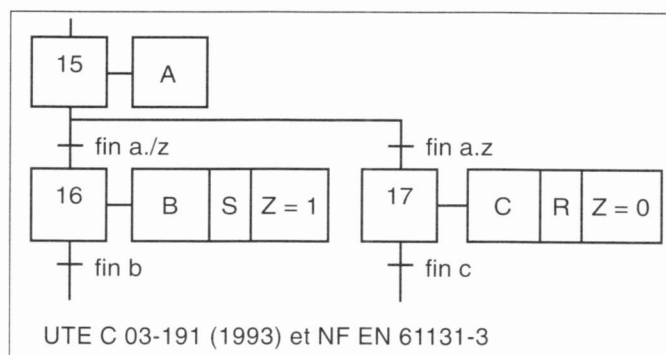
Si on se trouve dans la situation $S = \{15\}$ avec $\text{fin a./z} = 1$ donc avec $z = 0$, on évolue vers la situation $S = \{16\}$. L'ordre $Z = 1$ met la variable z à l'état 1.

Figure 3.57
Sélection alternative par drapeau.

Au prochain passage dans la situation $S = \{15\}$ comme $z = 1$ on évoluera vers la situation $S = \{17\}$. L'ordre $Z = 0$ 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 l'exploiter dans les programmes des API.



Autre notation de la mise à 1 ou 0 du bit drapeau suivant la documentation UTE C 03-191 et la norme française EN 61131-3 (fig. 3.58) :

Les ordres S et R signifient respectivement Set (mise à l'état 1) et Reset (mise à l'état 0).

Figure 3.58
Notation des ordres Set et Reset du bit drapeau.

d. Parallélisme interprété

Les divergences des exemples ci-dessus sont toutes exclusives, ce qui assure une évolution absolument déterminée lorsque la condition de transition devient vraie.

Il existe pourtant des cas où on peut tolérer d'avoir des séquences inclusives. Si on accepte ce genre de solution, il faut impérativement réfléchir à ses conséquences. On parle alors de parallélisme interprété par opposition au parallélisme structural.

1^{er} cas

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

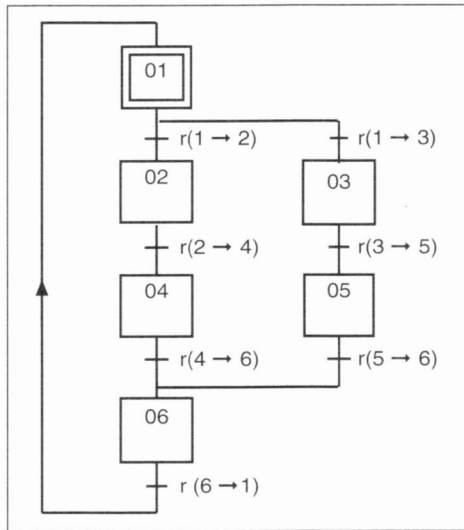


Figure 3.59
Parallélisme interprété.

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

Le troisième cas correspond typiquement au parallélisme interprété.

Ce dernier cas conduit souvent à des GRAFCET au graphisme simplifié mais il faut prendre la peine de vérifier si toutes les évolutions prévues sont possibles.

e. Comparaison entre le parallélisme structural et le parallélisme interprété

Ces deux GRAFCET (figure 3.60 page suivante) ont un comportement identique sur le plan des évolutions. On constate, ici, que le parallélisme interprété présente un nombre plus réduit de liaisons orientées.

On remarquera que dans les deux types de parallélisme, on s'assure de la parfaite exécution des évolutions en vérifiant l'inactivité de certaines étapes. Dans le cas du parallélisme interprété, la condition de transition $/X2./X4./X3./X5$ liée à la transition $6 \rightarrow 1$ contrôle l'évolution parallèle des séquences 2-4 et 3-5

déclenchées si $m1 = m2 = 1$ à partir de la situation $S = \{1\}$. Le retour à l'étape 1 n'est possible que si ces deux évolutions ont été complètement effectuées.

L'étape 6 du parallélisme interprété risque d'être activée puis réactivée avant d'être désactivée. Elle restera donc activée tant que la condition de transition $/X2./X4./X3./X5$ ne sera pas vraie.

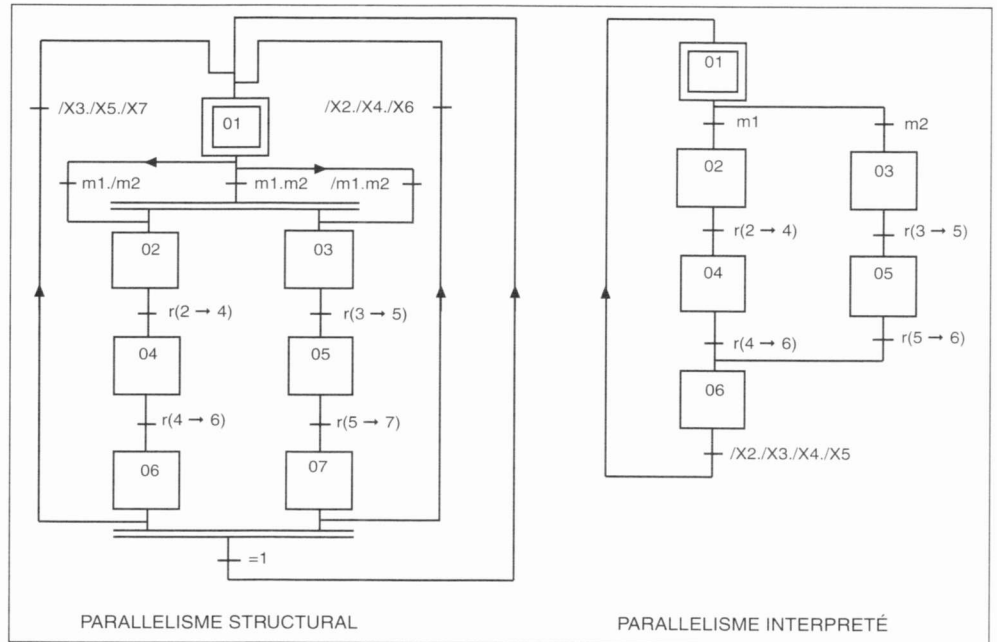


Figure 3.60
Comparaison entre les deux
types de parallélisme.

11. Étude des actions ou des ordres

11.1. Nature des actions associées à l'étape

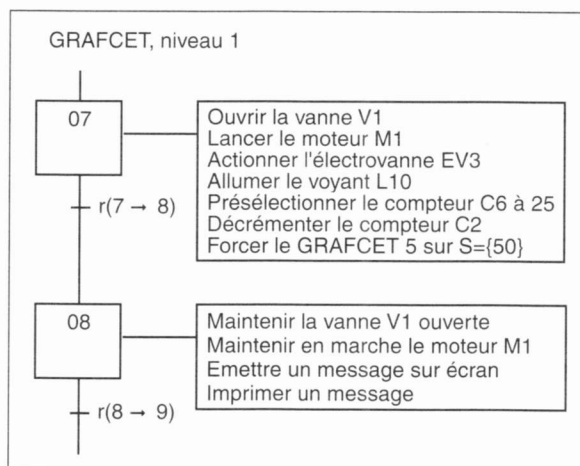


Figure 3.61
Exemples d'actions
possibles dans un GRAFCET.

La définition plus ou moins précise des actions est liée au niveau de description selon le point de vue de l'observateur du système automatisé.

Lorsque le dernier niveau est atteint (point de vue réalisateur) les actions correspondent à des ordres émis par la PC vers les entrées des interfaces de la PO et de la PR.

Les actions peuvent être de natures diverses.

En dehors des actions V1 et M1, les autres actions associées à l'étape 7 cessent à l'étape 8 puisqu'elles ne sont pas associées à cette étape. On peut donc dire dans certains cas que c'est l'absence d'ordre qui traduit une action ou un effet (cas des préactionneurs monostables).

Les actions peuvent être de nature binaire (TOR pour Tout Ou Rien) ou de nature analogique (acquisition, restitution, régulation, etc.) ou de nature numérique (entrées/ sorties sur mots, calculs sur mots, transcodages, etc.) ou de nature alpha-numérique (réception et envoi de messages, édition de journal de bord, etc.).

Le GRAFCET s'utilise comme outil de description d'un algorithme informatique au même titre qu'un organigramme pour analyser puis développer un programme de calcul. Comme actuellement l'automatisation de haut niveau mêle indistinctement tous les types d'automatismes, à savoir :

- les automatismes combinatoires et séquentiels,
- les régulations et les asservissements,
- les traitements numériques,
- les traitements de texte (journaux de bord, messages, etc.),

le GRAFCET se présente comme un outil capable d'assurer la coordination de systèmes automatisés complexes.

En conclusion, le GRAFCET peut être considéré comme un outil de description universel des automatismes.

Exemple 1 : gestion par GRAFCET de tâches de calcul et de traitement de texte

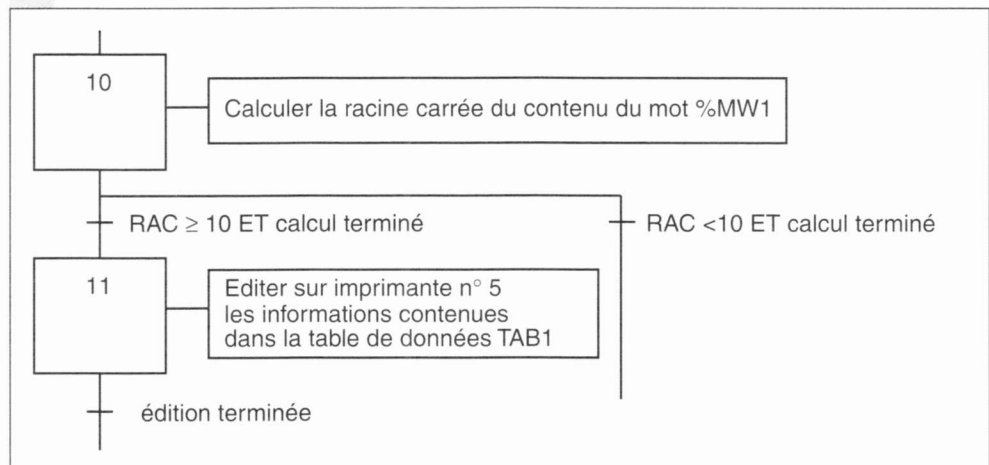


Figure 3.62
GRAFCET gérant des tâches de calcul et de traitement de texte.

Exemple 2 : coordination par GRAFCET de tâches de régulation et d'asservissement

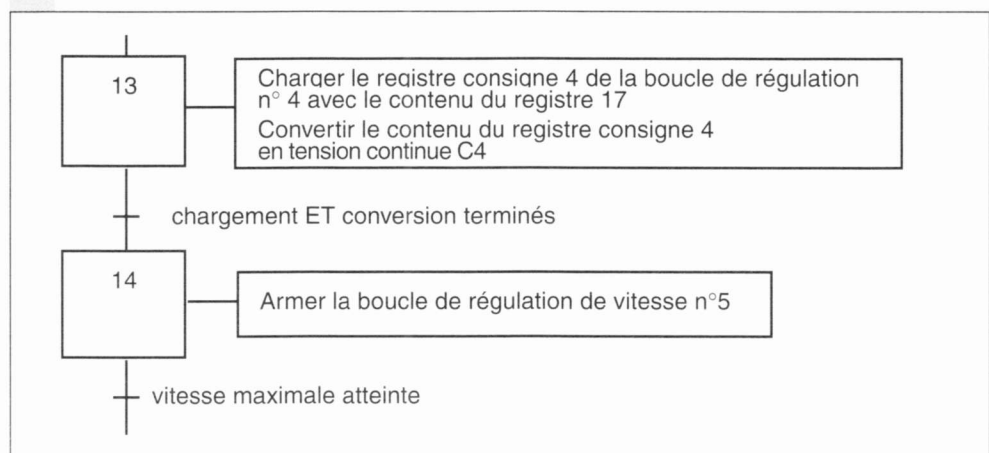


Figure 3.63
GRAFCET de gestion de tâches de régulation et d'asservissement.

11.2. Classification des actions ou des ordres

Au niveau 1, le terme « action » recouvre aussi bien une fonction à assurer qu'une tâche à exécuter tandis qu'aux niveaux 2 et 3 il s'agit surtout d'ordres à donner aux préactionneurs. Il faut rappeler également que les actions associées à l'étape ne sont exécutables que si au moins l'étape est active.

Il semble donc intéressant de prendre comme critère de comparaison la durée d'activation de l'étape. On distingue alors :

- les actions continues ou à niveau,
- les actions conditionnelles (symbole C pour Conditioned),
- les actions conditionnelles retardées (symbole D pour Delayed),
- les actions conditionnelles à durée limitée (symbole L pour Limited),
- les actions impulsionnelles ou ponctuelles (symbole P pour Pulsed),
- les actions maintenues ou mémorisées (symbole S pour Set et R pour Reset).

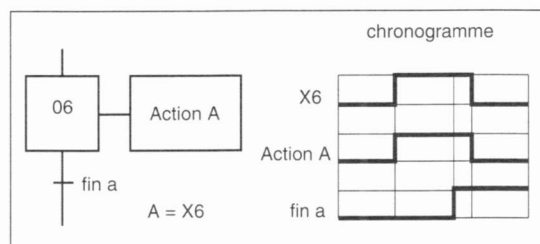
Les symboles recommandés sont placés dans une case à gauche de celle de l'action.

Les étapes émettent aussi des ordres destinés à modifier la situation interne de la PC ; ordres qu'on peut classer ainsi :

- ordres de lancement de GRAFCET sous-programmes ou de tâches,
- ordres de lancement de branches de GRAFCET (macro-étapes),
- ordres de forçage et de figeage de situation,
- ordres de sélection des modes de marche et d'arrêt,
- ordres de demande de calcul ou de traitement de texte.

11.3. Détails des actions

A. Action continue ou à niveau



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

Figure 3.64
Action continue ou à niveau.

B. Action conditionnelle (symbole C)

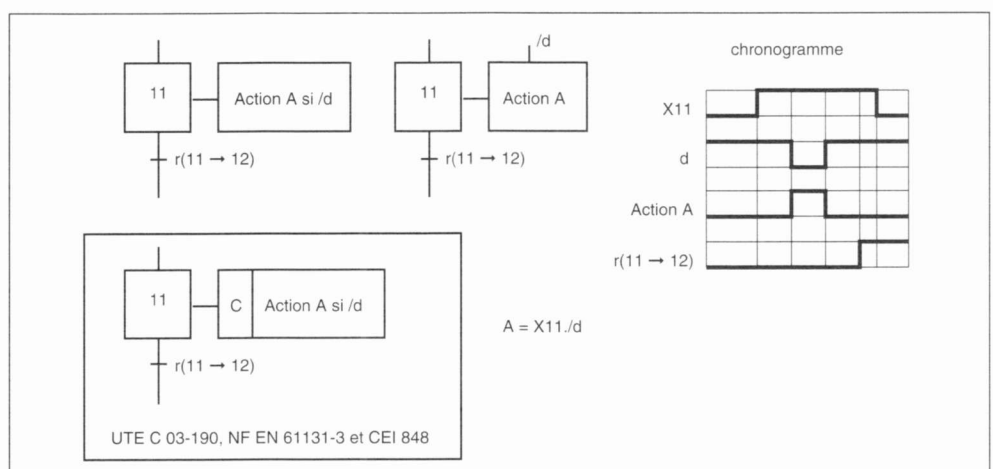


Figure 3.65
Action conditionnelle.

L'exécution de l'action ne devient effective que lorsque l'étape à laquelle elle est associée devient active et que la condition logique 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 est seulement sensible à cette condition.

C. Action retardée et action à durée limitée (symboles D et L)

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

La notation générale t/étape de lancement/durée est applicable.

A titre de rappel t/X15/5s signifie que ce signal sera émis 5 secondes après le début de l'activation de l'étape 15.

Dans l'exemple (fig. 3.66), 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.

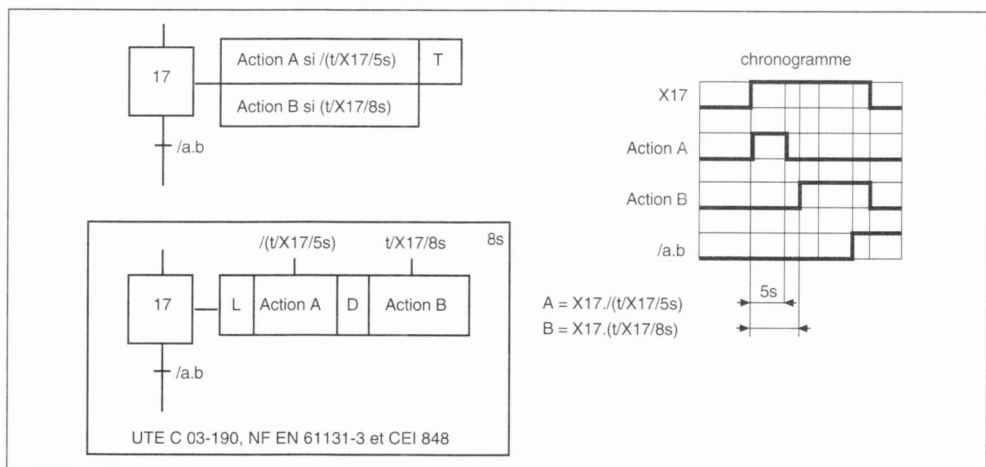


Figure 3.66
Action retardée
et action à durée limitée.

Noter que l'opérateur temporisateur T est implicite et il n'est donc pas obligatoire de le faire figurer sur le GRAFCET. Sa mise en place est toutefois conseillée lorsqu'il s'agit d'un composant ou d'un opérateur dont le symbole et l'adresse doivent apparaître sur le GRAFCET de niveau 3.

D. Action impulsionnelle ou ponctuelle (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 pouvoir exécuter l'action en question, par exemple :

- l'incrémenter 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 :

$$\uparrow X_i \text{ ou } \downarrow X_i \text{ ou } \uparrow e \text{ ou } \downarrow e$$

Noter que 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.

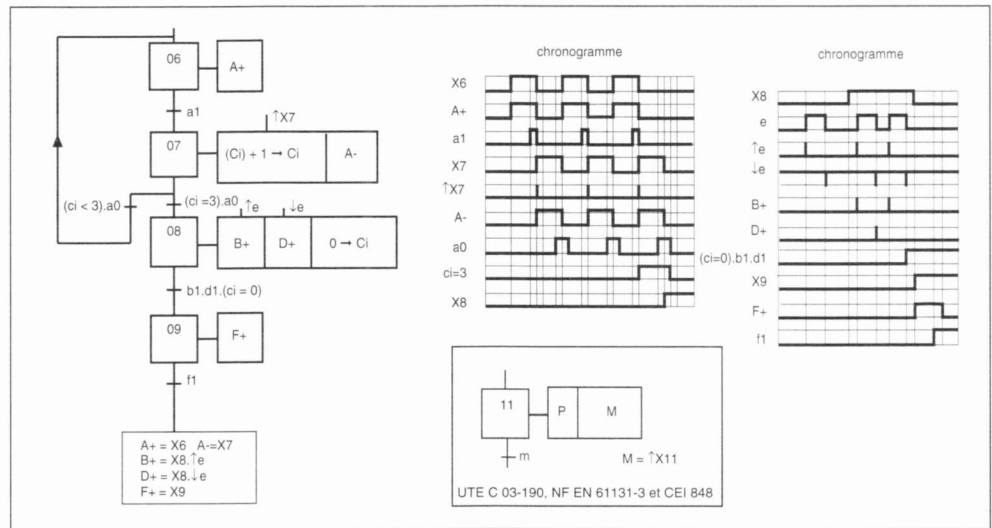


Figure 3.67
Action ponctuelle
ou impulsionnelle.

E. Action maintenue ou mémorisée (symboles S et R)

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

Plusieurs descriptions sont possibles :

- en associant l'action à toutes les étapes concernées, la solution étant de la forme : action = $X_m + X_n + X_p + \dots$,
- en associant l'action à une étape d'un parallélisme structural,
- en mémorisant l'action par l'intermédiaire d'un GRAFCET particulier à cette action,
- en exploitant les instructions SET et RESET propres à la PC (API),
- en exploitant un opérateur mémoire extérieur à la PC et faisant office **de fonction opérative de commande (FOC)**. Tel est le cas :

- des composants bistables (relais électromagnétique bistable, distributeur à double pilotage à deux positions, etc.),
- des réalisations câblées avec des composants monostables (relais électromagnétiques monostables auto-alimentés, modules logiques pneumatiques ou électroniques montés en rétro-action, etc. (fig. 3.68)).

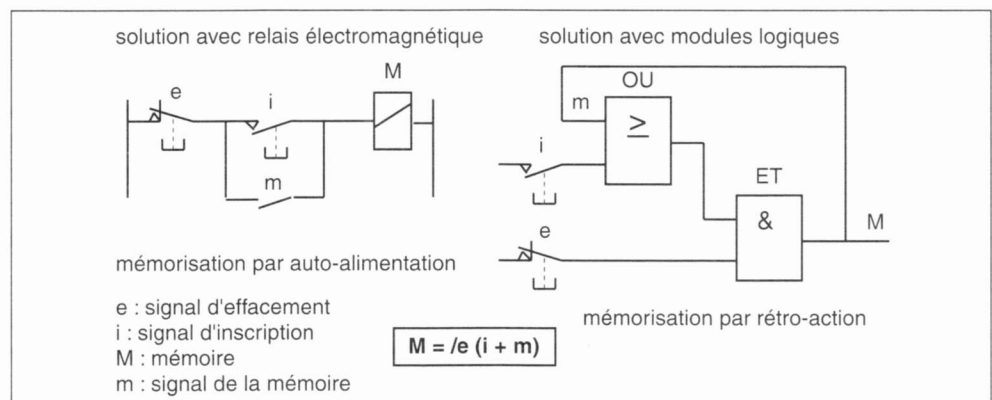


Figure 3.68
Mémoires câblées.

Exemple 1 : mémorisation par association à des étapes consécutives

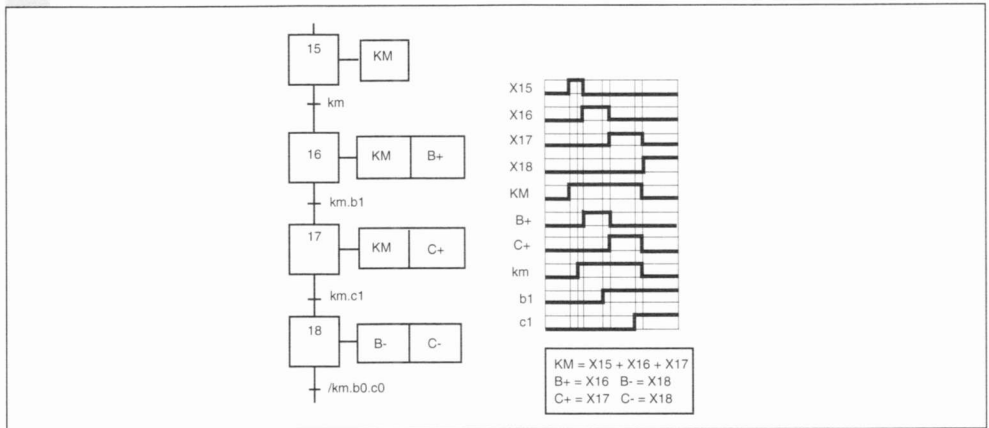


Figure 3.69
Mémorisation
par association
à des étapes consécutives.

Exemple 2 : mémorisation par association à une étape dans un parallélisme structural

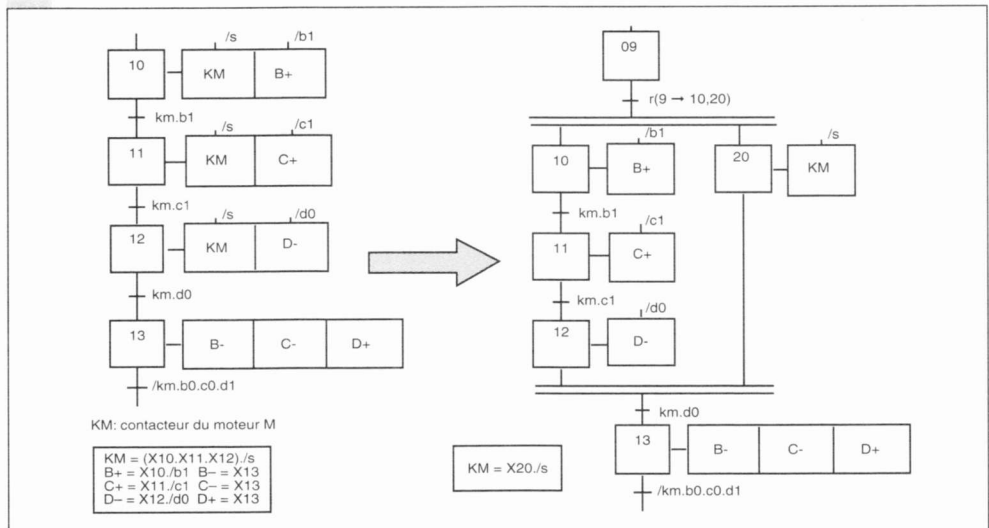


Figure 3.70
Mémorisation
par association
à une étape dans
un parallélisme structural.

Exemple 3 : mémorisation par lancement d'un GRAFCET particulier

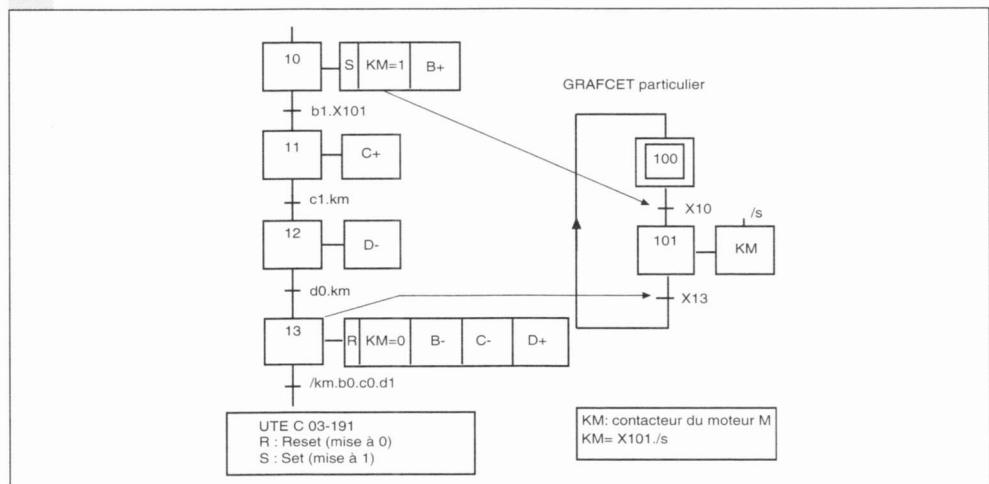


Figure 3.71
Mémorisation
par lancement
d'un GRAFCET particulier.

Exemple 4 : mémorisation par instructions propres à la PC (API)

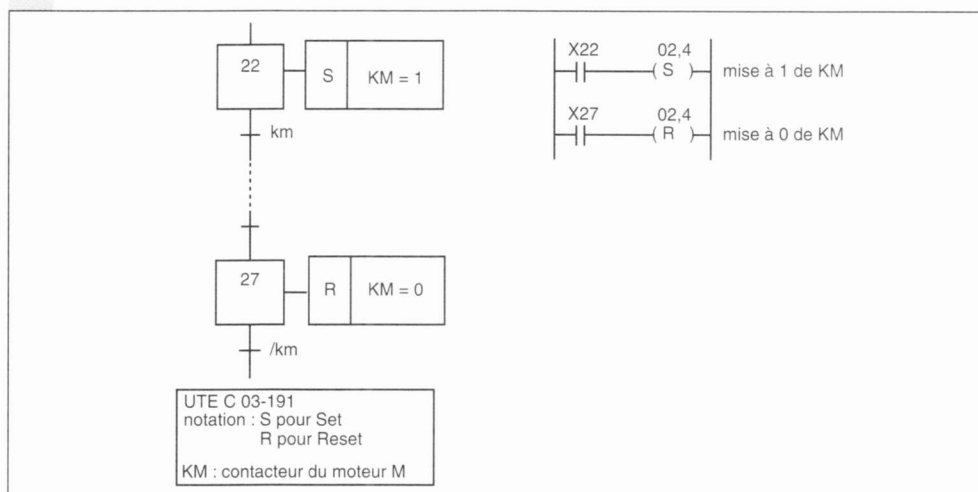


Figure 3.72
Mémorisation
par instructions
propres à la PC.

Exemple 5 : mémorisation par un opérateur mémoire extérieur à la PC (fonction opérative associée, UTE C 03-191)

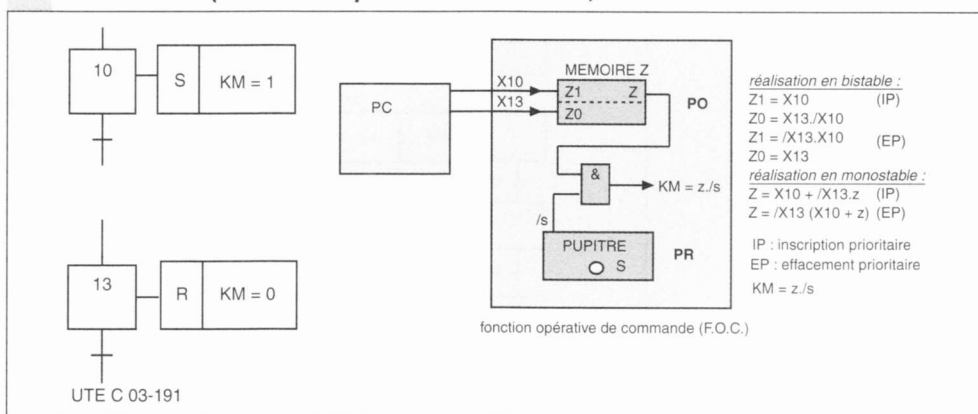


Figure 3.73
Mémorisation par un
opérateur extérieur à la PC.

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

- soit directement avec un composant bistable : relais électromagnétique bistable, distributeur à double pilotage, etc.,
- soit indirectement, par câblage, avec des composants monostables : relais électromagnétiques monostables auto-alimentés, modules logiques pneumatiques ou électroniques montés en rétro-action, etc. (voir fig. 3.68).

Remarque

Dans la documentation UTE C 03-190 et les normes EN 61131-3 et CEI 848, 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.

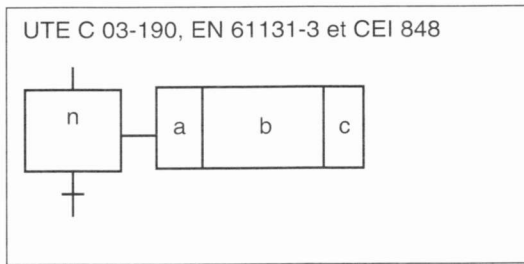


Figure 3.74
Sections de la case
d'action.

Plusieurs lettres symboles peuvent être associées à la même action. Dans ce cas, faire très attention à l'ordre d'écriture (fig. 3.75).

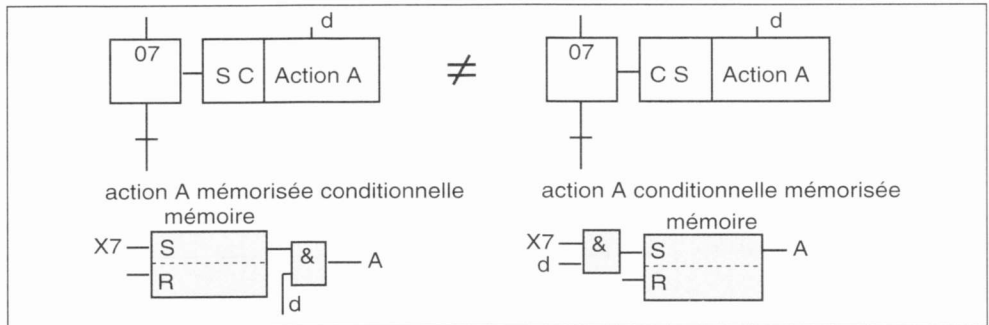


Figure 3.75
Association
d'identificateurs.

La société Schneider Télémécanique 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).

F. Ordres de lancement d'une séquence répétitive (GRAFSET sous-programme)

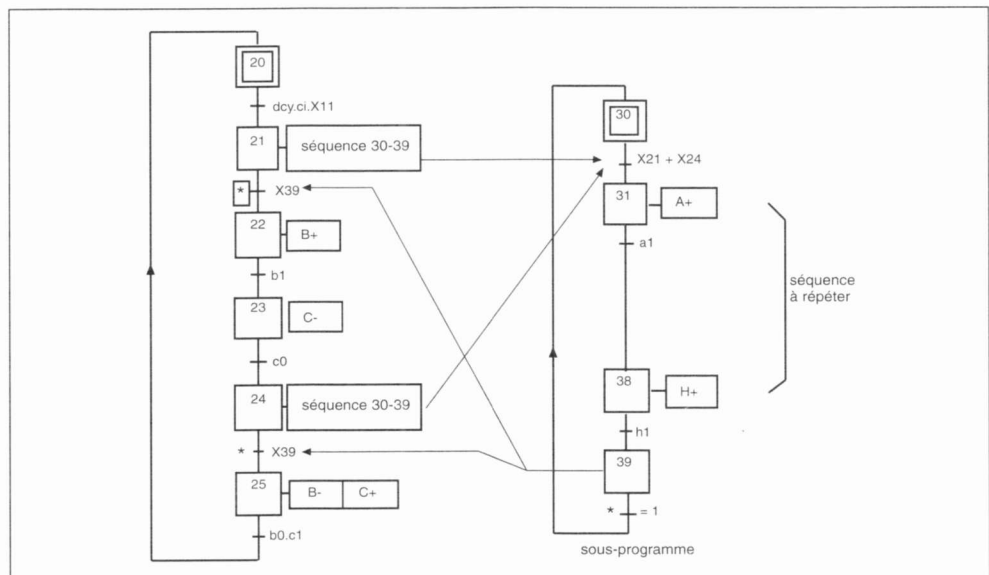


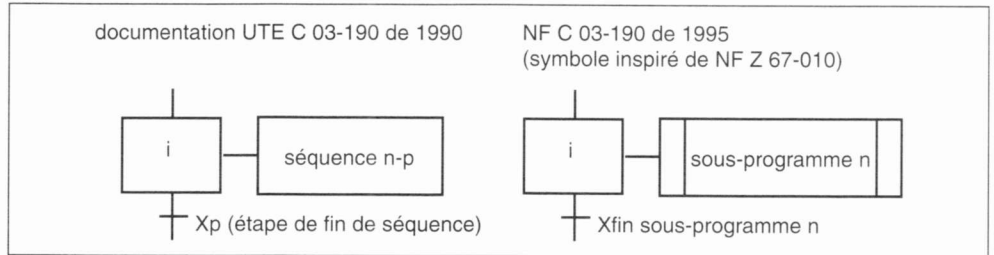
Figure 3.76
Ordre de lancement d'un
GRAFSET sous-programme
(UTE C 03-190).

Dans l'exemple précédent, le GRAFCET sous-programme 1 est lancé dès que l'étape 21 du GRAFCET principal est active. L'activation de la dernière étape du sous-programme (condition X39) assure la relance des évolutions du GRAFCET principal avec passage de la situation $S = \{21, 39\}$ à la situation $S = \{22, 30\}$. La condition de transition = 1 entraîne en effet la réinitialisation du GRAFCET sous-programme dès que l'étape 39 est active (documentation UTE C03-190).

Le même cycle recommence lorsque l'étape 24 devient active, le sous-programme 1 est relancé, la condition X39 redémarre le GRAFCET principal et réinitialise le GRAFCET sous-programme 1.

La coordination entre les deux GRAFCET est, ici, du type asynchrone dite aussi du type appel-réponse.

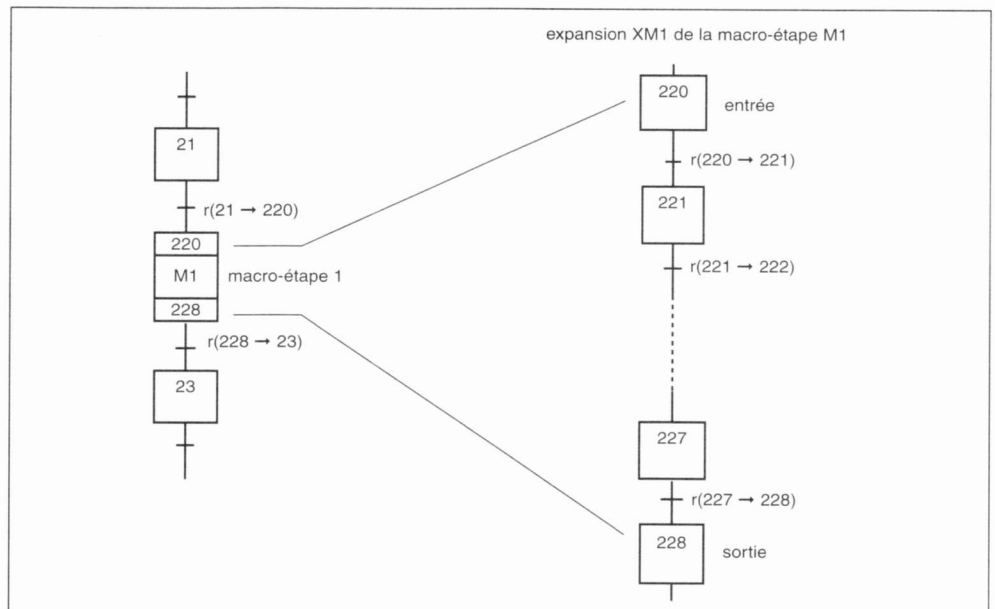
Figure 3.77
Symbole de l'ordre
de lancement
d'un sous-programme.



G. Ordre de lancement de branches de GRAFCET (macro-étape)

La macro-étape offre la possibilité de condenser la description en développant à part les détails de la branche qu'elle représente (expansion de la macro-étape).

Figure 3.78
Macro-étape
et son expansion.



Dans l'exemple donné, la macro-étape M1 est activée par le franchissement de la transition $21 \rightarrow M1$ ou $21 \rightarrow 220$. La branche de GRAFCET (220-228) matérialisant l'expansion XM1 de la macro-étape évolue alors indépendamment de la macro-étape M1.

Note

L'étude détaillée des GRAFCET sous-programmes et de leurs lancements est faite au chapitre III.12.1. L'étude détaillée des macro-étapes est abordée au chapitre III.12.2.

En fin d'évolution de l'expansion XM1, l'activation de l'étape 228 autorise le franchissement de la transition M1 → 23 ce qui entraîne simultanément :

- l'activation de l'étape 23,
- la désactivation de l'étape 228, c'est-à-dire de la macro-étape M1.

L'expansion XM1 passe dans une situation vide.

H. Ordre de forçage de situation (documentation UTE C 03-191)

L'ordre de forçage de situation émis par un GRAFCET hiérarchiquement supérieur permet de modifier la situation courante d'un GRAFCET hiérarchiquement inférieur, sans qu'il y ait franchissement de transition.

L'ordre émis est prioritaire sur toutes les conditions d'évolution et a pour effet d'activer la ou les étapes correspondant à la situation forcée et de désactiver les autres étapes du GRAFCET forcé.

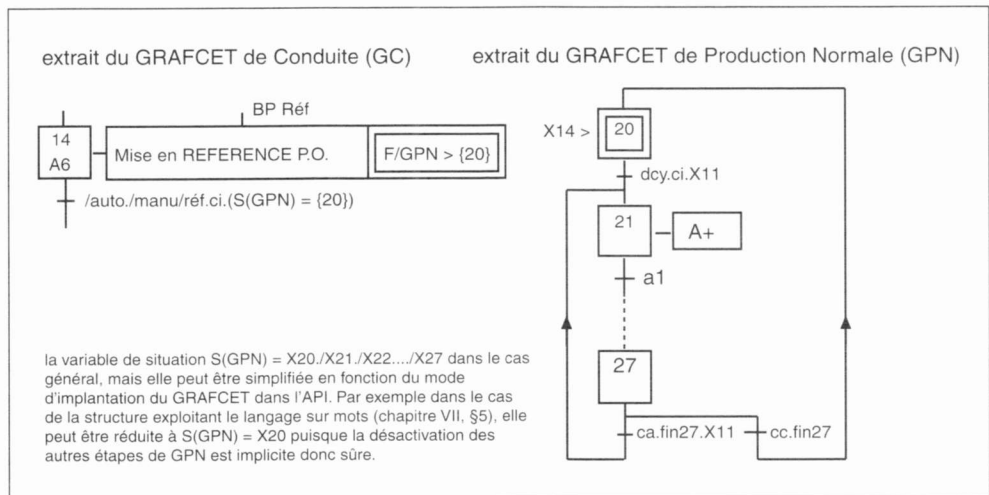


Figure 3.79
Forçage de situation.

Dans l'exemple de la figure 3.79, la notation F/GPN > {20} signifie : forçage du GRAFCET GPN dans la situation S = {20}.

Ce forçage est obtenu en activant l'étape 20 et en désactivant les étapes 21 à 27 par l'ordre X14.

La notation X14 > à gauche de l'étape 20 rappelle qu'un ordre de forçage sera émis par l'étape 14 pour amener le GRAFCET GPN dans la situation S(GPN) = {20}.

L'ordre de forçage est prioritaire et a pour effet tant qu'il est émis :

- de maintenir actives les étapes activées de la situation forcée,
- de maintenir inactives les étapes désactivées.

I. Ordre de figeage de situation

L'ordre de figeage de situation émis par un GRAFCET hiérarchiquement supérieur permet de bloquer, dans sa situation courante, les évolutions d'un GRAFCET hiérarchiquement inférieur.

Dans l'exemple de la figure 3.80 page suivante : la notation F/GPN > { * } signifie figeage du GRAFCET GPN dans sa situation courante.

La notation X12 > F/GPN > { * } à droite de l'étape 20 rappelle qu'un ordre de figeage du GRAFCET GPN sera émis par l'étape 12 avec pour effet de bloquer immédiatement les évolutions du GRAFCET GPN dans sa situation courante.

Remarque : on adoptera la notation Xi(Gn) > pour, en cas d'ambiguïté, préciser à quel GRAFCET appartient l'étape i.

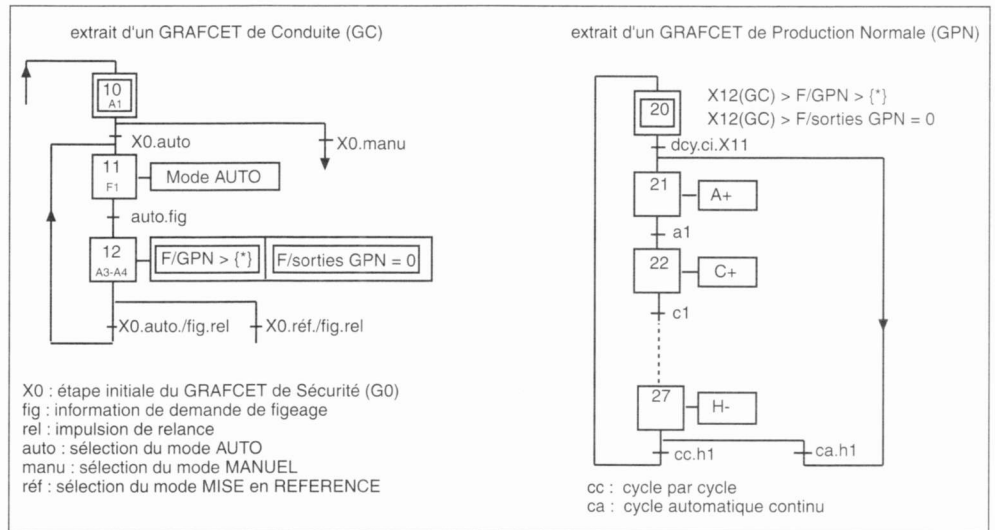


Figure 3.80
Figeage de situation.

Le figeage des évolutions de l'ensemble des GRAFCET de Production Normale GPN peut être obtenu :

- soit par la mise à 0 de ses conditions de transitions par l'intermédiaire du signal /X12 émis par l'étape 12,
- soit par un ordre spécifique de figeage propre au constituant retenu pour réaliser la PC

L'intérêt du figeage de situation, de l'ensemble des GRAFCET de Production Normale, est de permettre d'abord l'arrêt puis la reprise éventuelle du cycle de production à partir de la situation courante figée.

Noter que les étapes de la situation courante restent activées, ce qui peut poser certains problèmes en ce qui concerne les actions associées. Par conséquent, la demande de figeage est souvent accompagnée d'une demande de mise à 0 des actions associées aux étapes du ou de l'ensemble des GRAFCET figés.

Dans l'exemple précédent, si les ordres émis par les étapes de l'ensemble des GRAFCET de Production Normale sont destinés à la commande de bobines de distributeurs, de contacteurs ou de relais électromagnétiques, le forçage à 0 de ces ordres permettra de désactiver les bobines associées aux étapes actives de la situation figée.

Les notations proposées dans ce cas sont :

F/sorties GPN = 0

pour indiquer le forçage à 0 des actions listées ou non associées aux étapes du GRAFCET GP et :

X12 > F/sorties GPN = 0

placé à droite de l'étape 20, pour rappeler qu'un ordre de forçage à 0 des actions (ou sorties) listées ou non est émis par l'étape 12 du GRAFCET de Conduite GC. Si nécessaire la liste des actions à forcer à 0 doit être détaillée à côté du GRAFCET correspondant.

Une autre solution au problème du figeage assure systématiquement la mise à 0 des sorties. On l'obtient en forçant l'ensemble des GRAFCET de Production Normale dans une situation vide ce qui a pour effet :

- de figer la Production Normale puisqu'elle ne peut plus évoluer, aucune transition n'étant validable,
- de mettre systématiquement à 0 toutes les actions associées à l'ensemble des GRAFCET de Production Normale.

Pour reprendre l'évolution à partir de la situation d'arrêt (situation existant au moment du forçage de la situation vide), il faut pouvoir :

- mémoriser la situation courante au moment du forçage,
- forcer le GRAFCET, sur demande, sur cette ancienne situation.

Mais dans tous les cas, un risque apparaît au niveau de la PO : le figeage immédiat de la PO, s'il est souhaité pour des raisons de sécurité, ne sera effectif que si un choix approprié des préactionneurs et des actionneurs a été fait :

- préactionneurs de type bistable,
- actionneurs à verrouillage mécanique ou pneumatique (bloqueurs).

Notation proposée pour le forçage à 0 des sorties listées ou non de la PO associé à l'étape i :

F/ sorties PO = 0

et pour rappeler que cet ordre a été émis par l'étape i : **Xi > F/sorties PO = 0.**

Note

Les forçages et les figeages de situation sont étudiés plus en détail au chapitre III.13.

J. Ordres de sélection des modes de marche et d'arrêt

Dans le cas le plus simple, les modes de marche et d'arrêt sont gérés par deux GRAFCET hiérarchiquement supérieurs aux autres GRAFCET.

Le premier GRAFCET est appelé GRAFCET de Sécurité (GS).

Il gère l'énergie sur la PO et les procédures de sécurité.

Le second GRAFCET est appelé GRAFCET de Conduite (GC).

Il gère les modes de marche et d'arrêt.

Le GRAFCET GS est hiérarchiquement supérieur au GRAFCET GC. Chacun des modes ou états est associé à une situation particulière et unique de ces deux GRAFCET (recherche de l'unicité de mode).

Si l'automatisation est plus complexe, le GRAFCET de sécurité peut être éclaté en un ou plusieurs GRAFCET de gestion de l'énergie et en un ou plusieurs GRAFCET de sécurité suivant une hiérarchie à établir.

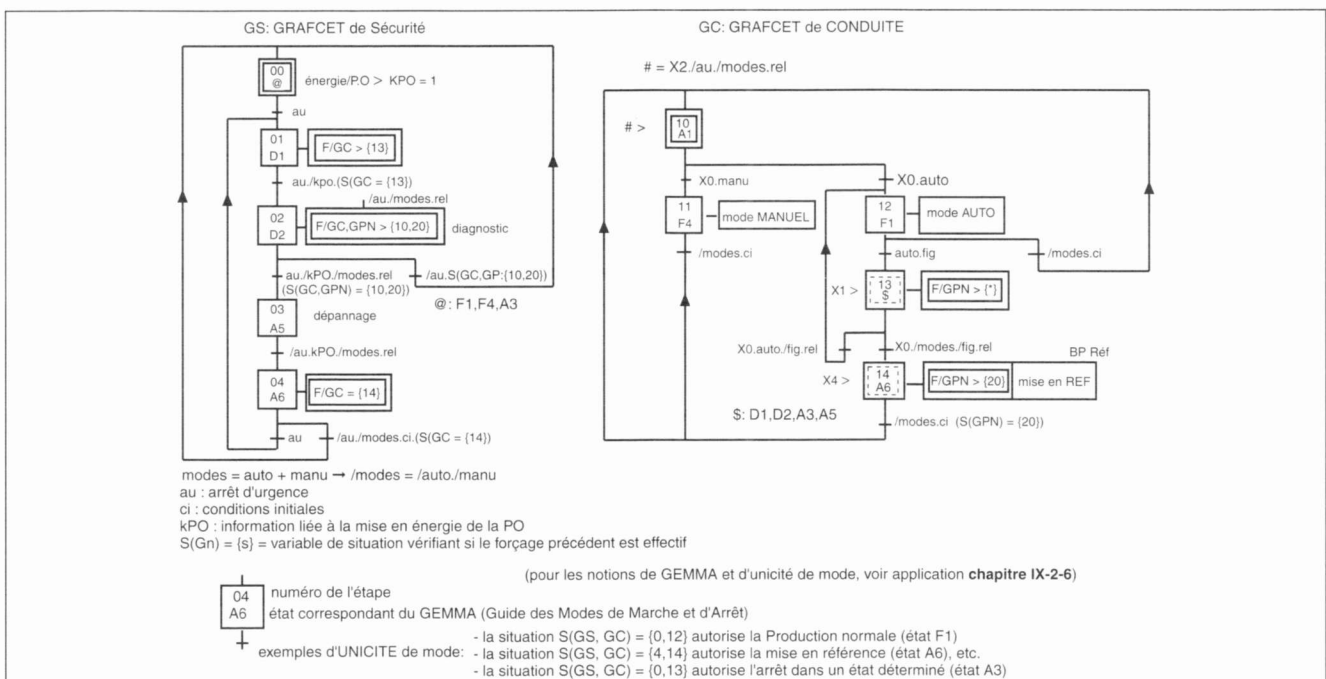


Figure 3.81 : Ordre de sélection des modes de marche et d'arrêt.

Le système est en production normale (état F1 du GEMMA) lorsque les GRAFCET GS et GC se trouvent dans la situation S (GS, GC) = {0, 12}. Afin de rappeler la nécessité d'avoir l'étape 12 active pour être dans l'état F1 de production normale, on associe un ordre de sélection figuré ici symboliquement par l'étiquette "Mode AUTO" (état F1 du GEMMA) par opposition au "Mode MANUEL" (état F4 du GEMMA) correspondant à la situation S(GS,GC) = {0, 11}.

Un aperçu sur la recherche des modes de marche et d'arrêt est donné au chapitre IX.2.2, alinéa 9.

K. Ordres de demande de calcul

Avec ce type d'ordre, on se donne la possibilité de coordonner le traitement de la partie séquentielle avec le traitement de la partie calcul d'un automate dans le cas d'une description entièrement par GRAFCET ou dans le cas d'une description combinée GRAFCET et organigramme.

Ce traitement mixte nécessite une PC réalisée soit avec un API pouvant effectuer des calculs soit en associant un API sans possibilité de calcul avec un autre constituant ayant cette faculté : carte à microprocesseur travaillant en langage machine ou micro-ordinateur travaillant en langage évolué (BASIC, PASCAL, C, etc.).

Exemple 1 : extrait d'un GRAFCET gérant une régulation

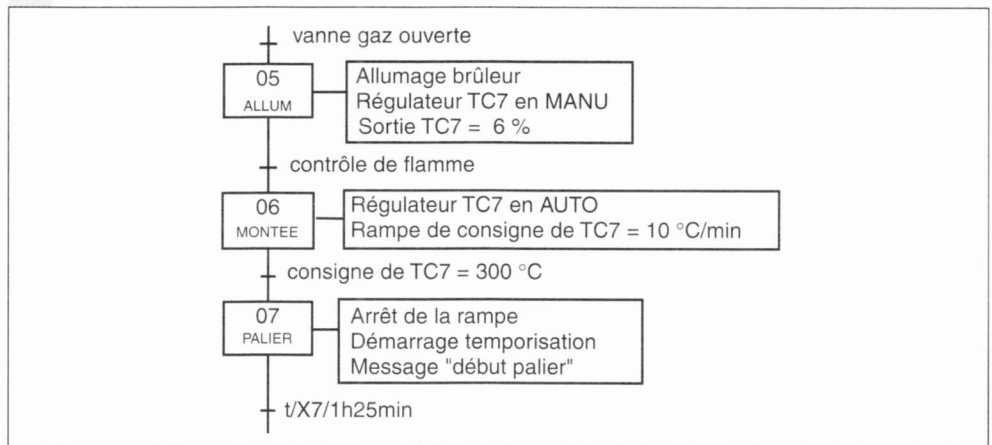


Figure 3.82
Calcul avec coordination
directe par GRAFCET.

Exemple 2 : déclenchement d'une alarme en fonction du nombre ou du % de pièces mauvaises PM

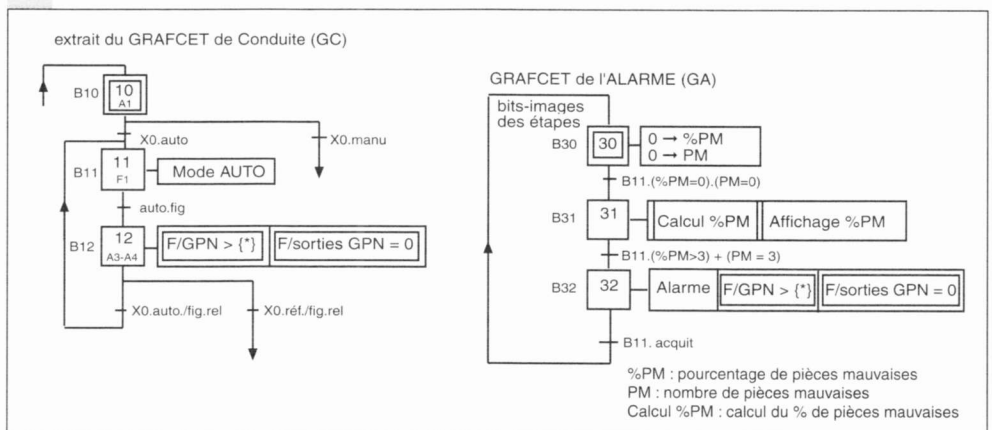


Figure 3.83
Coordination d'un GRAFCET
et d'un organigramme
(fig. 3.84).

Partie séquentielle (langage littéral SMC) :

.....

S99.B101 = /B101. oscillateur
 S100.B31.B101 = L1. saut au programme calcul tous les deux cycles
 si l'étape 31 est active

S101.....

.....

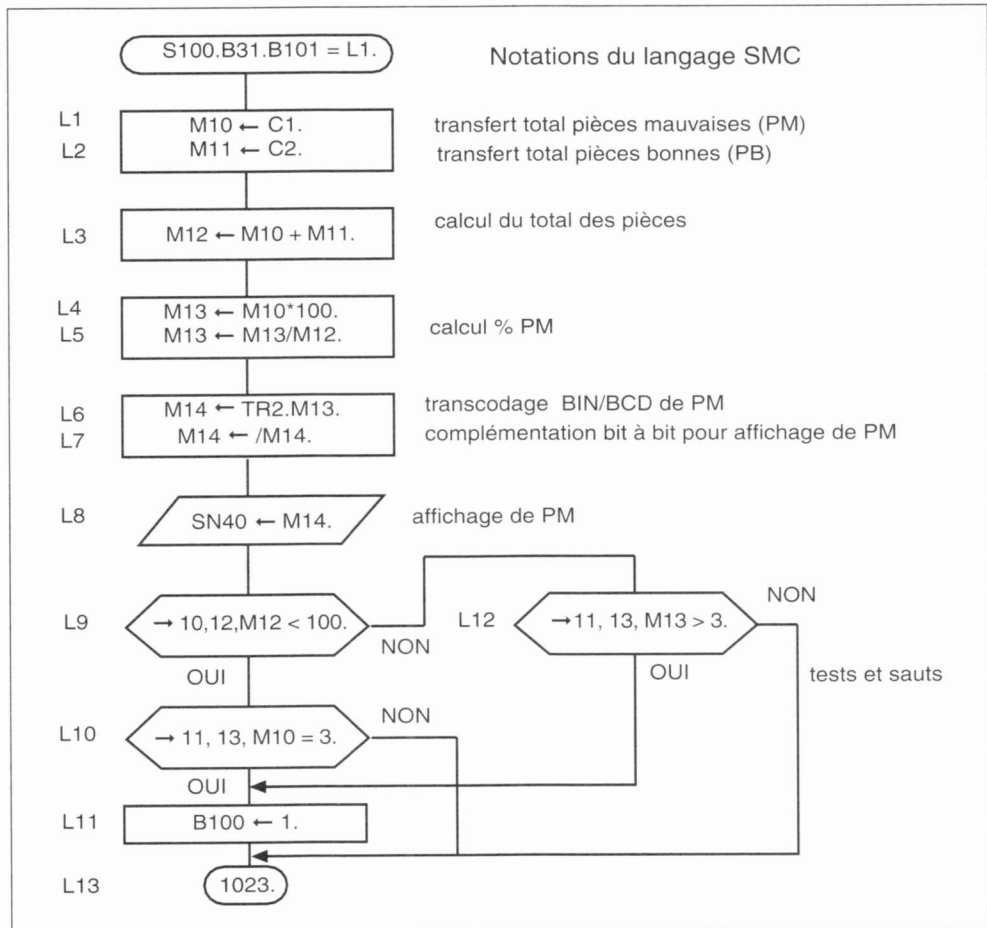


Figure 3.84
 Analyse du programme
 calcul par organigramme.

Commentaires du traitement de la partie calcul :

- S100 : saut au programme calcul si étape 31 active (B31 = 1) et bit 101 = 1
- L1, L2 : acquisition du contenu des compteurs C1 (pièces mauvaises PM) et C2 (pièces bonnes PB)
- L3 : calcul total des pièces TP (TP = TM+TB)
- L4, L5 : calcul du % PM (%PM = PM*100)/TP)
- L6 : transcodage BN-BCD (Binaire → Binaire Codé Décimal)
- L7 : complémentation pour affichage numérique
- L8 : affectation à une carte numérique pour sortie affichage
- L9 : test si total pièces TP inférieur à 100, si OUI test si PM = 3, si NON, aller en 12
- L10 : test si nombre de pièces mauvaises PM = 3, si OUI mise à 1 bit alarme B100, si NON aller en 13
- L11 : mise à 1 bit alarme B100
- L12 : test si %PM > 3, si OUI mise à 1 bit alarme B100, si NON aller en 13
- L13 : retour au séquentiel (adresse 1023)

Le programme de la partie calcul écrit en langage littéral découle directement de l'organigramme :

L1.M10 ← C1.
 L2.M11 ← C2.
 L3.M12 ← M10+M11.
 etc.

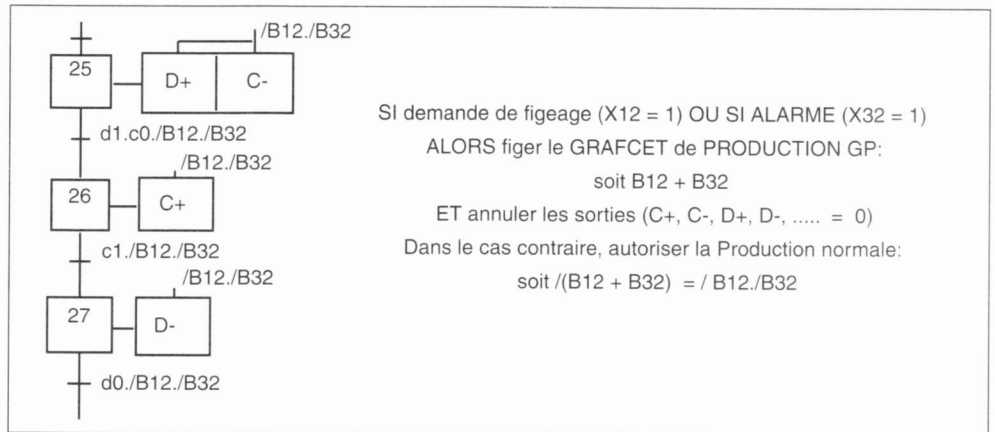


Figure 3.85
 Extrait du GRAFCET
 de production normale.

Exemple 3 : niveau 3, traitement du calcul en langage à contacts

Le problème envisagé est le même que précédemment. La description par GRAFCET de niveau 3 facilite l'analyse préparatoire d'un programme calcul en langage à contacts, ici le langage à contacts des API TSX Schneider Télémécanique. Elle permet notamment de faire abstraction des instructions de saut, souvent source d'erreurs dans ce type de description.

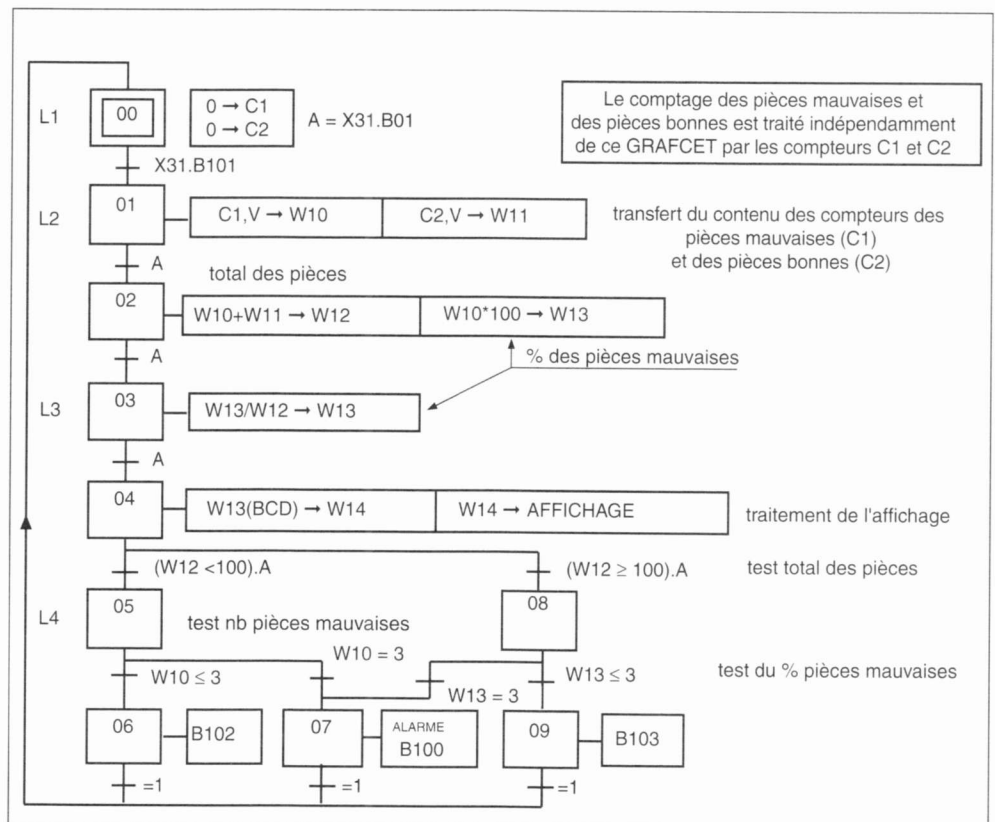


Figure 3.86 : Analyse
 préparatoire par GRAFCET.

Réalisation en langage à contacts sur mots (fig. 3.87)

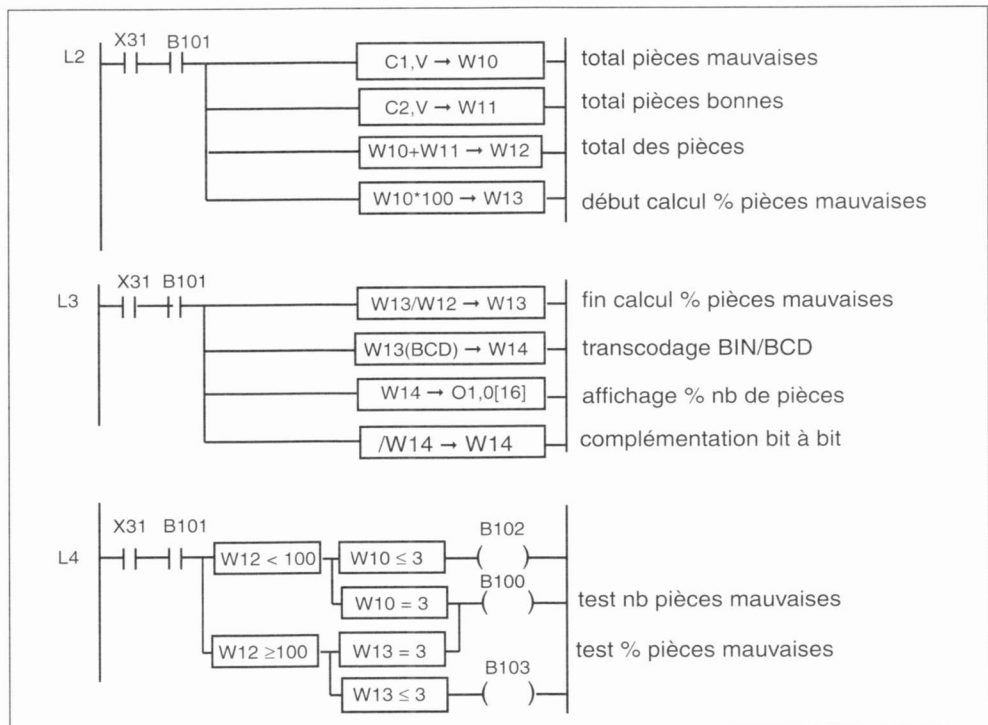


Figure 3.87
Réalisation en langage à contacts sur mots (réseau L1 de mise à 0 des compteurs C1 et C2 non représenté).

Remarques concernant l'exécution d'un réseau de contacts en langage PL7-2

Le programme est exécuté réseau par réseau séquentiellement selon l'ordre d'écriture en mémoire et non selon l'ordre croissant des numéros des étiquettes (labels).

La lecture de l'ensemble des réseaux se fait de haut en bas sauf en cas de saut en arrière commandé par une bobine J (jump). Ce type de saut est déconseillé : risque de déclenchement du chien de garde si l'exécution de la boucle interne dure plus de 150 ms (débordement).

L'exécution d'un réseau s'effectue séquence par séquence, de haut en bas et de gauche à droite. Une séquence est délimitée à droite par une connexion verticale ou par le coin supérieur gauche d'un bloc fonction.

Application au réseau L4 de la figure 3.87 :

L'exécution se fera dans l'ordre suivant (zone test) :

- 1 - (X31-B101),
- 2 - (W12 < 100), (W12 ≥ 100),
- 3 - (W10 ≤ 3), (W10 = 3), (W13 = 3), (W13 ≤ 3).

Les bobines sont exécutées après traitement de la zone test. En respectant cet ordre d'exécution, le système met à jour :

- l'état logique de chaque contact en fonction de l'entrée associée, acquise en début de cycle,
- l'état logique des objets bits associés aux bobines afin de mettre à jour les sorties en fin de cycle.

Pour qu'une bobine et son contact aient la même valeur dans le même cycle, il faut qu'ils appartiennent à deux réseaux différents. Le réseau où se trouve la bobine devra précéder celui contenant son contact.

12. Macro-représentations

Avec la notion de 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 regroupées en tâches ou en sous-programmes :

– **par tâches on entend une suite d'actions correspondant à l'exécution d'une fonction bien déterminée :**

dosage de produits, mélange de ces produits, etc. ou encore à l'exécution d'un mode de marche bien particulier : séquence de préparation des postes avant passage en production normale (marches de préparation F2 du GEMMA), séquence de clôture (marches de clôture F3 du GEMMA), etc.,

– **par sous-programmes, on entend une suite d'actions ou de tâches devant être exécutées plusieurs fois dans le même cycle de production :**

tâche de perçage répétée n fois pour percer n trous sur la même pièce, par exemple.

Deux concepts sont en concurrence :

- le concept d'étape de lancement d'un GRAFCET sous-programme ou d'un GRAFCET de tâche,
- le concept de macro-étape associée à son expansion.

12.1. GRAFCET sous-programme

La notion de sous-programme est empruntée au langage informatique. Des instructions spéciales sont prévues pour appeler un sous-programme à partir d'un programme principal puis pour revenir à celui-ci lorsque le sous-programme est terminé. On rencontre ce type d'instructions dans les langages assembleurs des microprocesseurs et dans les langages évolués.

Exemples

En langage assembleur pour microprocesseurs :
instruction JSR (Jump to SubRoutine),
instruction RTS (ReTurn from Subroutine).

En langage BASIC :
instruction GOSUB (GO to SUBroutine),
instruction RETURN (RETURN from subroutine).

Dans les automatismes séquentiels, il est fréquent de rencontrer des séquences répétitives dans le même cycle. Dans l'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. 3.88).

L'organisation type sous-programme peut être également exploitée pour décrire des tâches séparées, répétitives ou non, appelées à partir d'un GRAFCET de gestion ou de coordination des tâches d'un niveau supérieur.

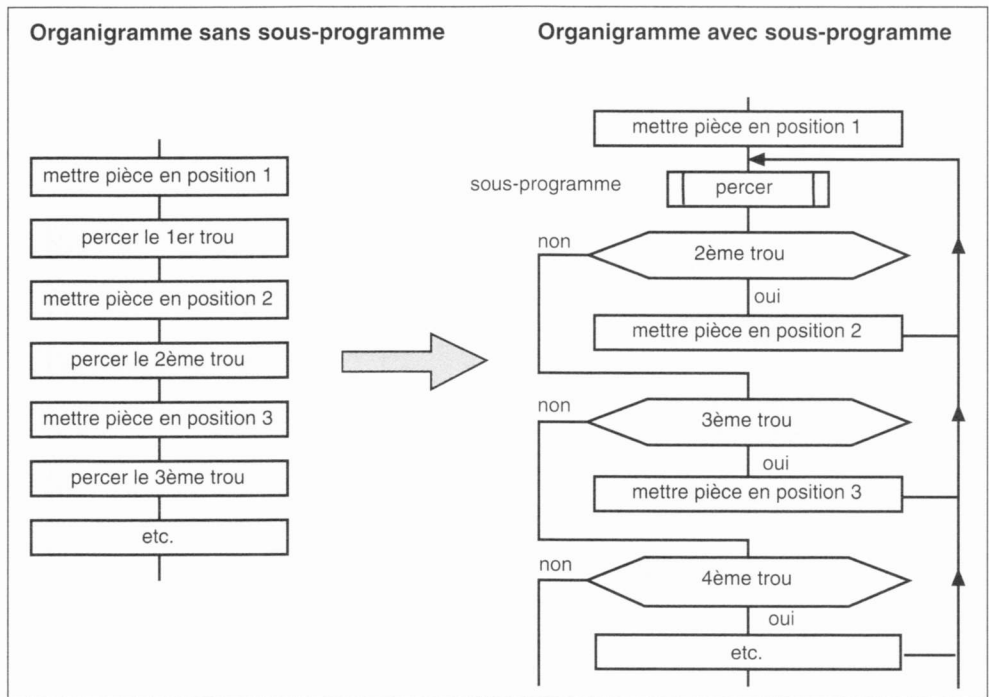


Figure 3.88
Organigrammes sans
et avec sous-programme.

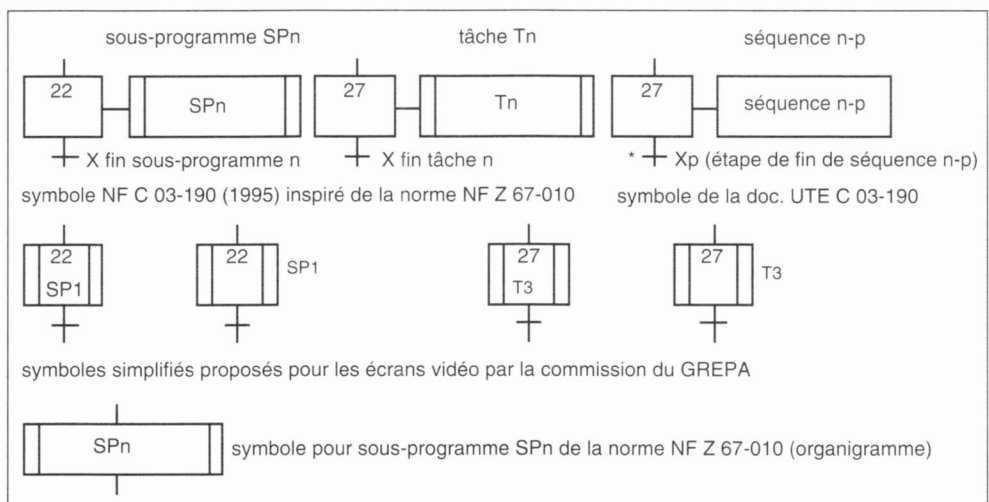


Figure 3.89
Symboles associés
au lancement de
sous-programmes
ou de tâches.

Noter que le symbole proposé par la documentation UTE C03-190 suppose la séquence répétitive définie, ce qui n'est pas le cas dans l'analyse descendante. Le symbole NF C 03-190, par contre, permet de reporter à plus tard la description détaillée du sous-programme. De plus, avec ce symbole, les lancements de sous-programmes sont plus aisément repérables dans le GRAFCET.

A. Structure d'un GRAFCET sous-programme (séquence répétitive)

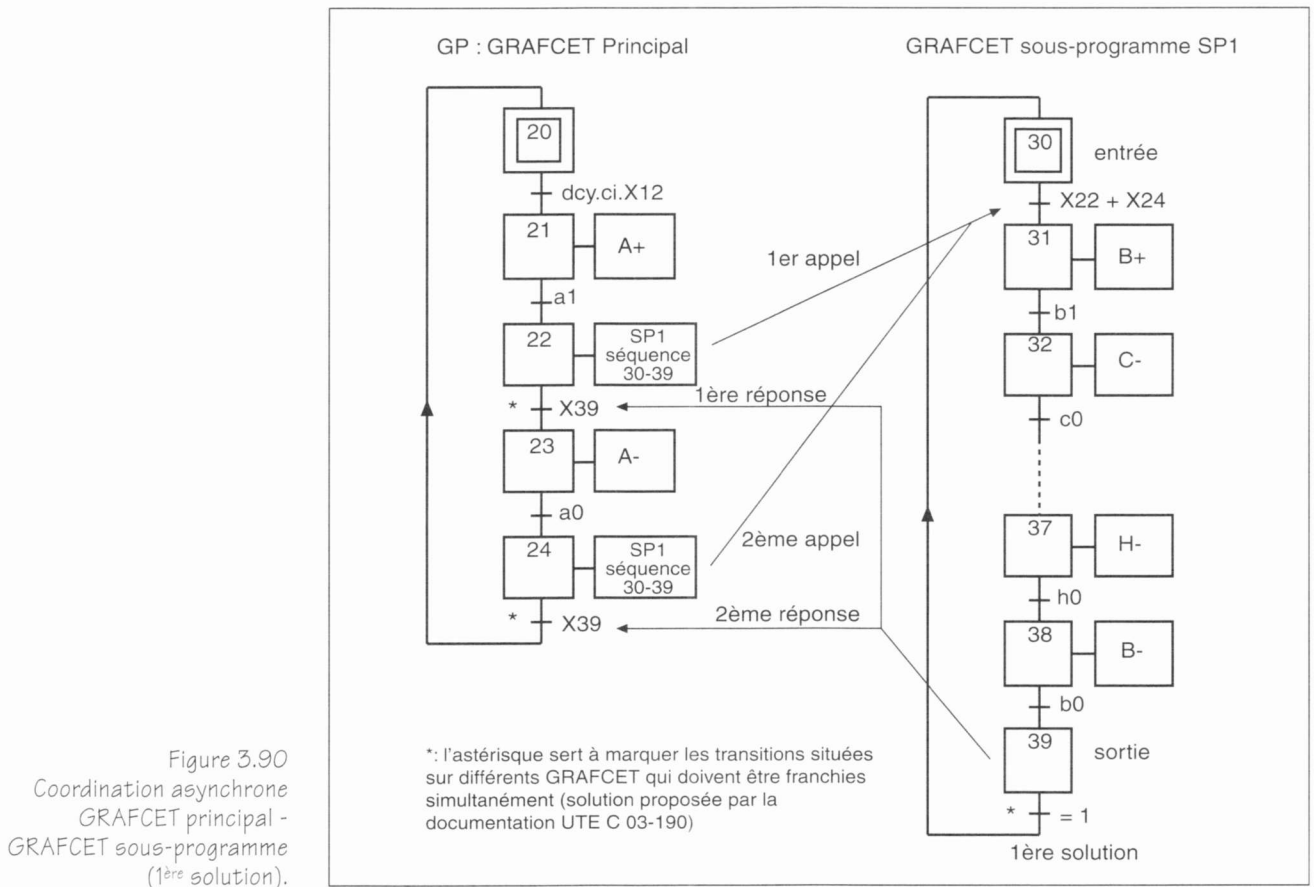
Cette structure est caractérisée :

- 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.

Reste à résoudre le problème de la coordination entre le GRAFCET principal et le GRAFCET sous-programme.

a. Coordination asynchrone entre GRAFCET principal et GRAFCET sous-programme



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 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.

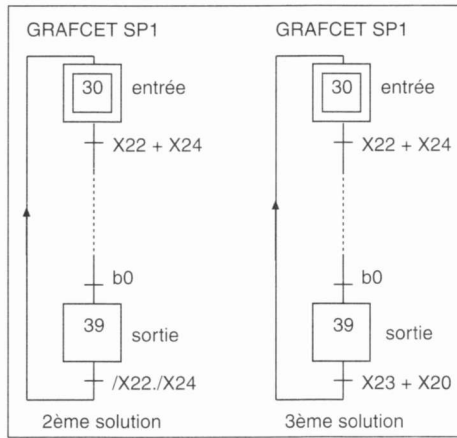


Figure 3.91
Solutions 2 et 3
comparées de retour au
GRAFCET principal.

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 (fig. 3.90),
- soit par le contrôle de la désactivation des étapes de lancement : condition /X22./X24,

- soit par le contrôle de l'activation des étapes suivantes des étapes de lancement : 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.

Cette solution est employée de préférence dans cet ouvrage compte tenu des problèmes posés par une condition de transition toujours vraie (= 1) dans certaines réalisations (voir figure 3.31 et addenda page 219).

b. Coordination synchrone entre GRAFCET principal et GRAFCET sous-programme sans étape de sortie

La solution UTE C 03-190 basée sur l'application de la règle d'évolution 4 peut être poussée plus loin en éliminant l'étape de sortie.

Dans l'exemple fig. 3.92, lorsque l'étape 38 est active et que la condition de transition est vraie, les transitions 22 → 23 et 38 → 30 sont franchissables et simultanément franchies. Il en est de même des transitions 24 → 20 et 38 → 30 à la fin de la deuxième exécution de SP1.

Cette structure apporte un gain en nombre d'étapes utilisées si la description fait appel à de nombreux sous-programmes. Elle peut être utile pour certaines réalisations programmées (API) au nombre d'étapes limité.

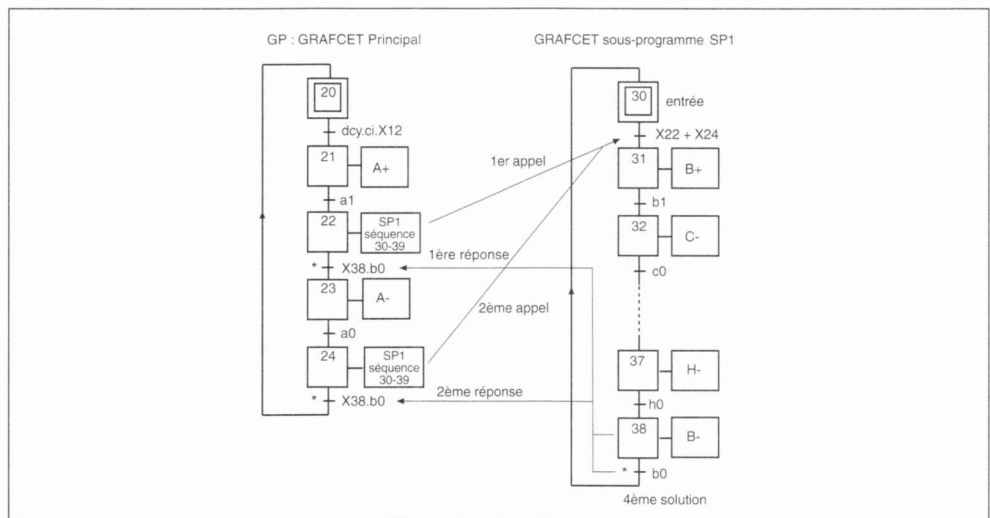


Figure 3.92
Coordination synchrone
entre GRAFCET principal et
GRAFCET sous-programme.
(4^{ème} solution)

Noter pour terminer qu'un GRAFCET sous-programme peut lui-même contenir une macro-représentation de lancement d'un autre GRAFCET sous-programme (imbrications des sous-programmes) (fig. 3.93).

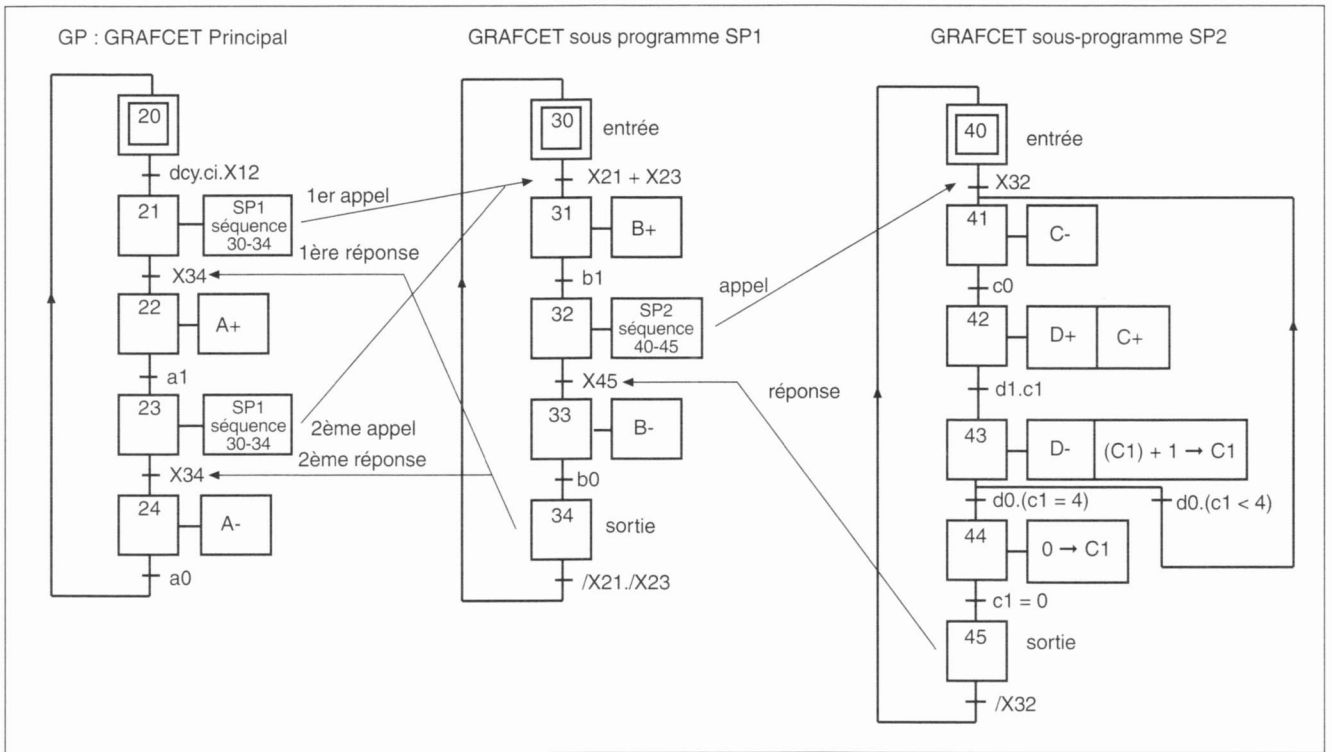


Figure 3.93
Imbrication de GRAFCET sous-programmes.

B. Structure d'un GRAFCET de tâche (fig. 3.94 et 3.95).

Cette structure ne présente aucune différence notable avec celle d'un GRAFCET sous-programme. Le GRAFCET de tâche (séquence non répétitive) est lancé à partir d'une macro-représentation du GRAFCET de gestion ou de coordination des tâches.

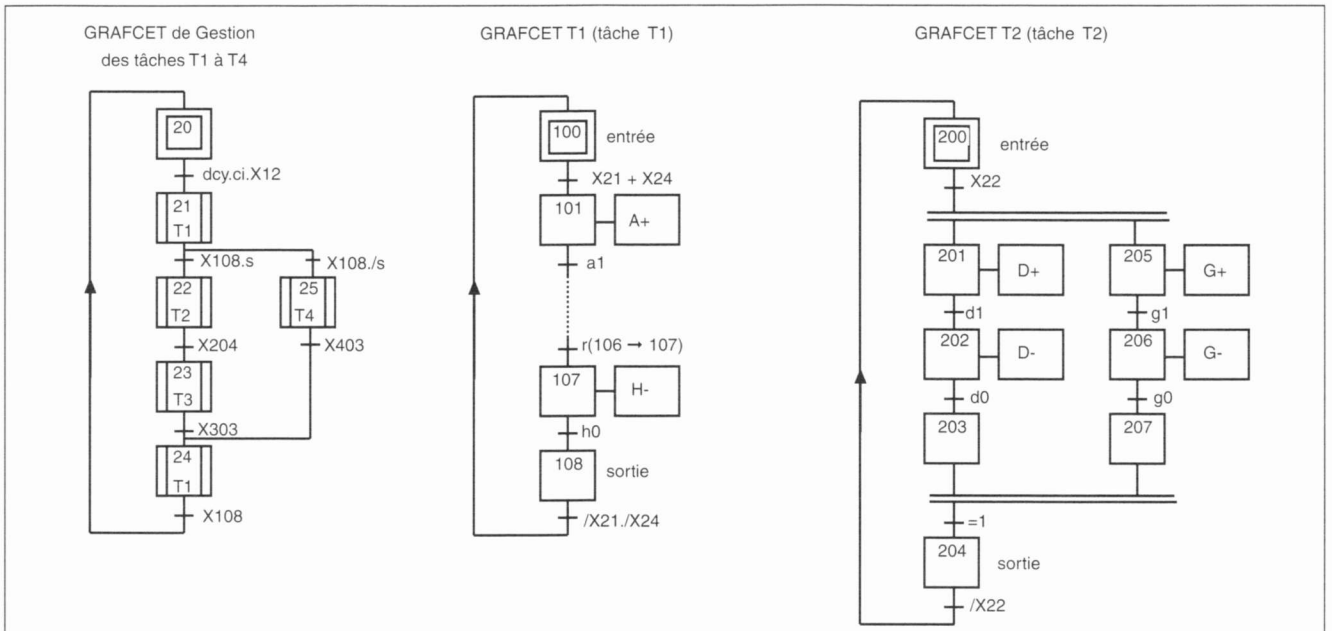


Figure 3.94 : GRAFCET principal de gestion de tâches et GRAFCET des tâches T1 et T2 (symboles GREPA).

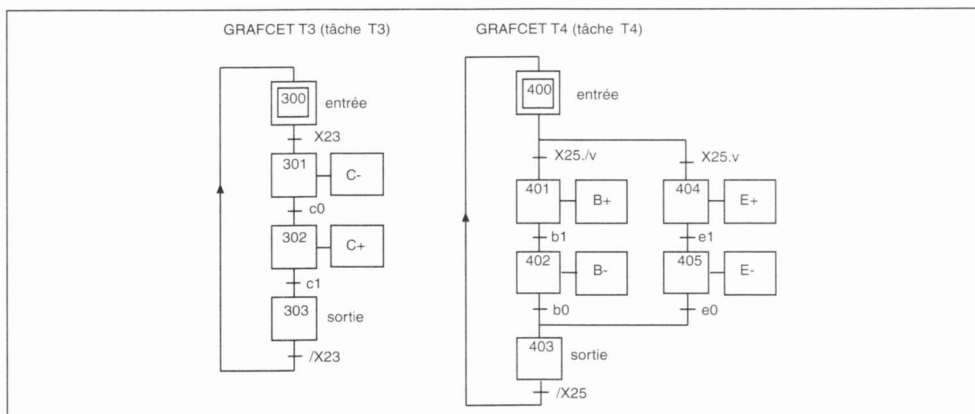


Figure 3.95
GRAFCET
des tâches T3 et T4.

Remarque

La notation « séquence n-p » (fig. 3.89) dans le symbole UTE C 03-190 de la séquence répétitive suppose cette séquence définie avec son début (étape n) et sa fin (étape p). Lorsqu'on pratique l'analyse descendante, c'est rarement le cas. En effet, par principe, le tracé du GRAFCET principal ou du GRAFCET de coordination des tâches, dans un premier temps, n'implique pas la connaissance du détail des GRAFCET sous-programmes. A ce niveau, la séquence sera notée provisoirement **SPn** pour sous-programme n ou **Tn** s'il s'agit de tâches*.

Noter que cette question ne se pose pas avec le symbole NF C 03-190 inspiré de la NF Z 67-010.

* Voir application chapitre IX, figures 9.7 à 9.10 et 9.23 à 9.26

12.2. Macro-étape et expansion

A. Définition de la macro-étape

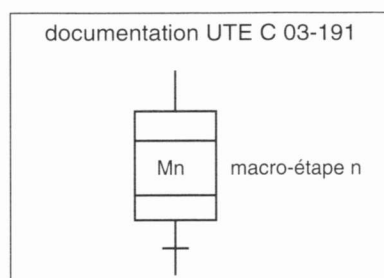


Figure 3.96
Symbole de la macro-étape.

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.

Symbole d'une macro-étape (fig. 3.96) : la macro-étape est figurée par un rectangle divisé en trois parties par deux traits horizontaux.

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.

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.

B. 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 uniquement si celle-ci est franchissable (voir fig.3.102). C'est pourquoi la condition de transition généralement associée à cette transition est toujours vraie (=1) ;
- 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 lui associer une action ;
- l'étape de sortie sert à concrétiser la fin de l'exécution de l'expansion. On ne doit pas lui associer d'action externe à cause de l'instabilité liée à la réceptivité =1 du GRAFCET de gestion (fig. 3.97) ;
- 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 ;
- la macro-étape peut prendre quatre états :
 - état repos : aucune étape active dans l'expansion de la macro-étape,
 - état actif : lancement et évolution de l'expansion en cours,
 - état final : évolution de l'expansion terminée avec son étape de sortie active,
 - état désactivé : retour à l'état repos si l'étape de sortie de l'expansion est active et si la transition aval de la macro-étape est franchissable.

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

L'application envisagée 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, situation qu'elle retrouverait en fin d'évolution (voir XM4, fig. 3.107).

APPLICATION

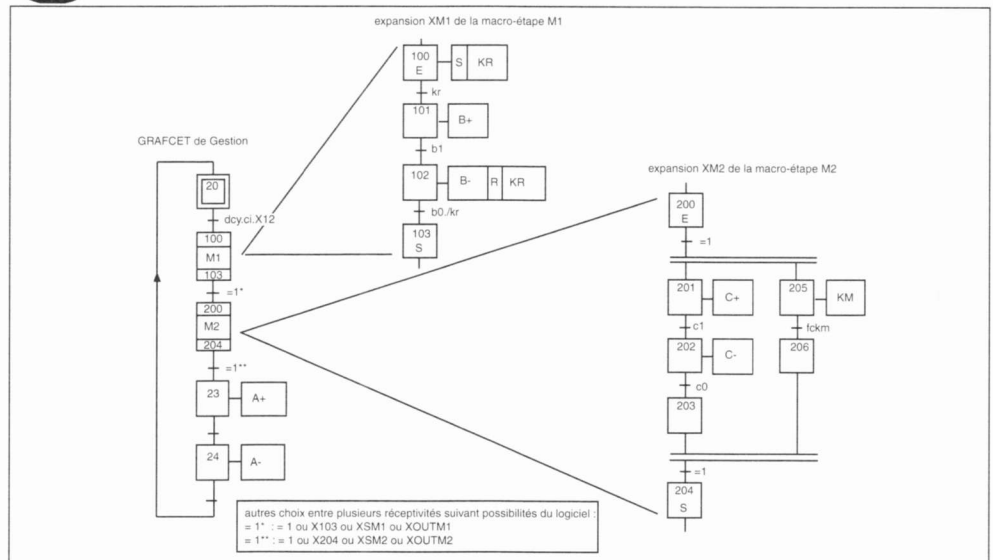


Figure 3.97
Macro-étapes
et leurs expansions.

L'expansion XM1 en attente est au repos (situation vide). Le franchissement de la transition 20 → 21 met la macro-étape M1 à l'état actif. L'étape 100 de l'expansion XM1 s'active et l'étape 20 du GRAFCET principal se désactive. 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.

12.3. Intérêt et limites des macro-représentations

L'analyse descendante d'un système procède par niveaux successifs permettant d'affiner progressivement la connaissance de l'automatisme, 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'un sous-programme ou d'une tâche est sans effet sur la macro-représentation à laquelle chacun ou chacune 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 séquences ou les tâches répétitives permet un gain appréciable de place en mémoire.

Les macro-étapes et leurs expansions associées sont sans grand intérêt 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.

Noter qu'une séquence répétitive devant être exécutée n fois dans le même cycle nécessitera 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.

12.4. Étude comparative des deux concepts de macro-représentation

L'exemple choisi est extrait de l'automatisme d'une machine à transfert rotatif. L'étude est limitée au poste de transfert proprement dit.

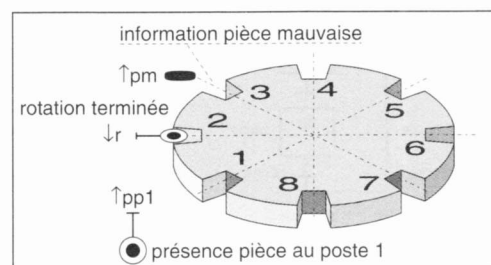


Figure 3.98
Transfert rotatif.

Inventaire des postes :

- 1 : chargement manuel (tâche 1)
- 2 : usinage 1 (tâche 2)
- 3 : contrôle (tâche 3)
- 4 : usinage 2 (tâche 4)
- 5 : usinage 3 (tâche 5)
- 6 : éjection pièce mauvaise (tâche 6)
- 7 : usinage 4 (tâche 7)
- 8 : éjection pièce bonne (tâche 8)

L'indexage du plateau fournit l'information $\downarrow r$ (rotation plateau terminée), le chargement manuel l'information $\uparrow pp1$ (présence pièce au poste 1) et le contrôle au poste 3 l'information $\uparrow pm$ (pièce mauvaise).

Cette dernière information est mémorisée par l'étape 1 du registre à décalage (GRAF CET GR). Ce registre émet des ordres pour interdire ou autoriser l'exécution des tâches 2 à 8 (GRAF CET G2 à G8) :

- X2 pour autoriser le saut des étapes 41 à 44 du poste 4,
- X3 pour autoriser le saut des étapes 51 à 56 du poste 5,
- X4 pour autoriser la séquence 61 à 63 du poste 6,
- X5 pour autoriser le saut des étapes 71 à 77 du poste 7,
- X6 pour autoriser le saut des étapes 81 à 83 du poste 8.

A. Solution avec macro-représentations du type sous-programme ou tâche (fig. 3.99, 3.100 et 3.101)

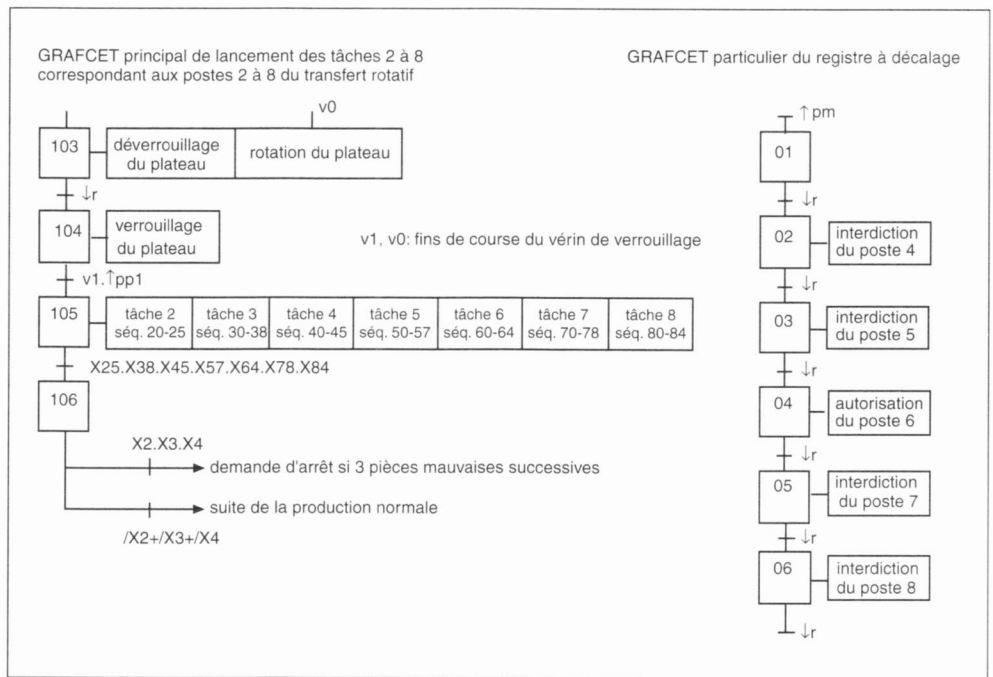


Figure 3.99
Solution par GRAFCET de gestion des tâches.

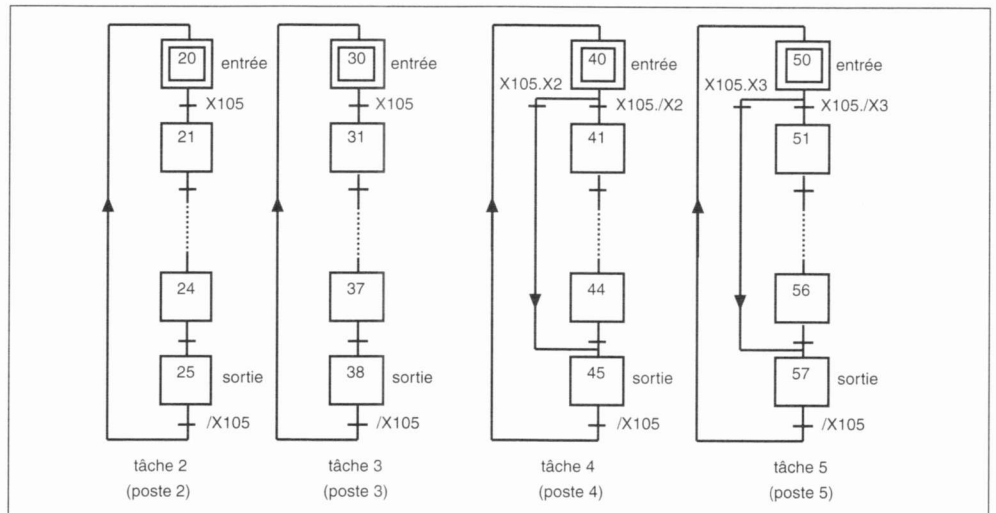


Figure 3.100
GRAF CET des sous-programmes des tâches T2, T3, T4 et T5.

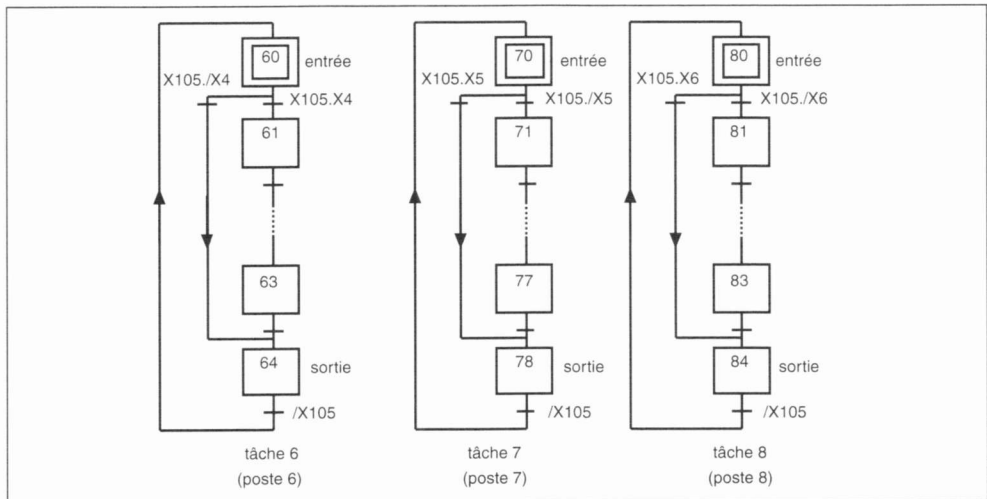


Figure 3.101
GRAFCET
des sous-programmes
des tâches T6, T7 et T8.

Le système étant dans la situation $S = \{104, 20, 30, 40, 50, 60, 70, 80\}$ la transition $104 \rightarrow 105$ est validée et franchie si le plateau est verrouillé ($v1=1$) et si une pièce brute est présente au poste 1 ($\uparrow pp1$). Le système passe dans la situation $S = \{105, 20, 30, 40, 50, 60, 70, 80\}$. L'activation de l'étape 105 assure le lancement simultané des GRAFCET des tâches 2 à 8. Le GRAFCET de la tâche 6 sera lancé mais aucun ordre ne sera émis du fait du saut des étapes 61 à 63 (transition $60 \rightarrow 64$ validée et autorisée par la condition de transition vraie $X105./X4$).

Lorsque tous les GRAFCET des tâches 2 à 8 ont terminé leurs évolutions, le système se trouve dans la situation $S = \{105, 25, 38, 45, 57, 64, 78, 84\}$. La transition $105 \rightarrow 106$ étant validée et la condition de transition $X25.X38.X45.X57.X64.X78.X84$ vraie, la transition est franchie.

Si une ou plusieurs pièces mauvaises sont détectées successivement, une ou plusieurs étapes du GRAFCET registre seront activées. En conséquence, certaines des tâches ne seront pas exécutées, les étapes utiles étant court-circuitées par le saut autorisé respectivement par les signaux $X2, X3, X5$ et $X6$.

B. Solution avec macro-représentations du type macro-étape

Le GRAFCET du registre à décalage reste identique. On notera que les macro-étapes sont situées dans un parallélisme structural du GRAFCET de gestion des tâches. (fig. 3.102). Noter qu'il n'est pas nécessaire de prévoir des étapes d'attente derrière les macro-étapes. En effet, la désactivation des macro-étapes M2 à M8 et le franchissement de la transition aval ne peuvent avoir lieu que si toutes les étapes de sortie des expansions sont actives.

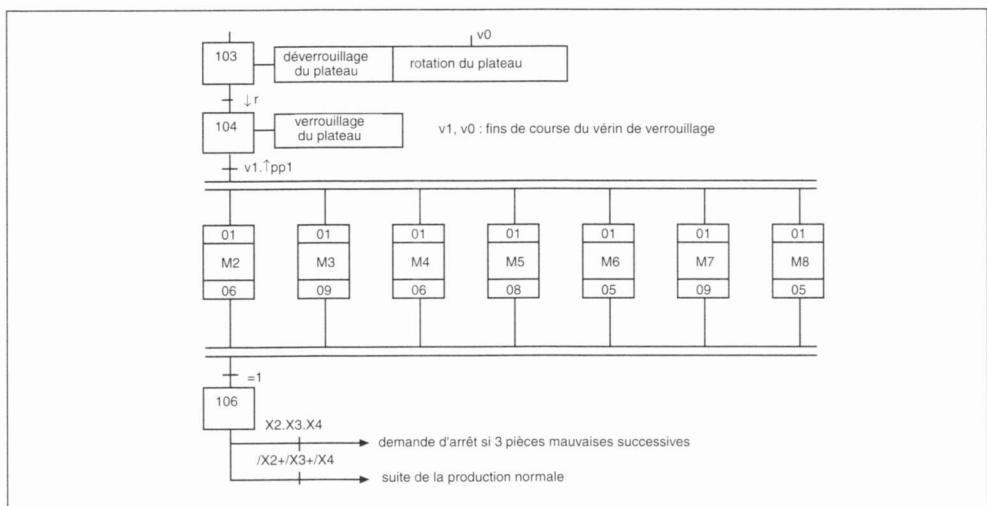
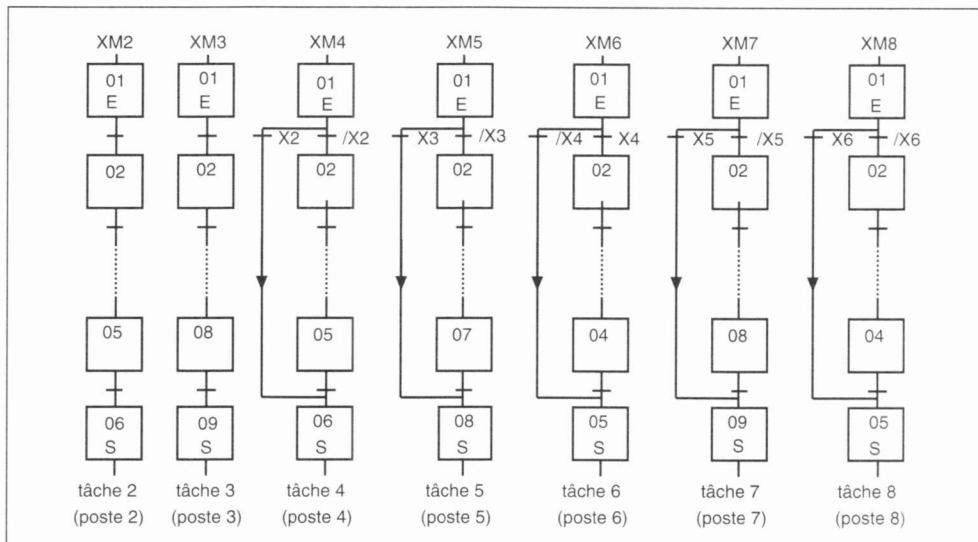


Figure 3.102
GRAFCET principal
avec les macro-étapes
M2 à M8
des tâches T2 à T8.

Figure 3.103
 Expansions XM2 à XM8
 des macro-étapes M2 à M8
 des tâches T2 à T8.



On retrouve le même principe, déjà vu, pour interdire ou non les exécutions complètes des tâches par les ordres émis par les étapes du GRAFCET du registre à décalage.

Les actions associées aux étapes des expansions ne sont indiquées généralement que sous forme de commentaires. Elles seront gérées en traitement postérieur dans le cas du langage PL7-3 V5, par exemple.

Il faut rappeler également que la solution par macro-étapes ne présente d'intérêt que si l'API dispose d'un langage autorisant la programmation de ce type de macro-représentation et si ces tâches ne sont pas répétitives.

12.5. Langage GRAFCET PL7-3 des API Schneider Télémécanique

Le traitement séquentiel de ce langage GRAFCET graphique possède des instructions relatives aux macro-étapes et à leurs expansions.

Le traitement séquentiel est structuré :

- en un sous-ensemble GRAPHE principal de gestion des macro-étapes (8 pages et 128 étapes au maximum),
- en 64 sous-ensembles ou expansions de macro-étapes (8 pages et 128 étapes au maximum chacune).

Le nombre total d'étapes reste limité à 512 dont 64 étapes au maximum activables simultanément.

Une page correspond à un écran et peut admettre 48 étapes au maximum.

Lorsque l'ensemble de la description par GRAFCET ne peut pas se représenter sur une seule page, le langage PL7-3 permet (fig. 3.104) :

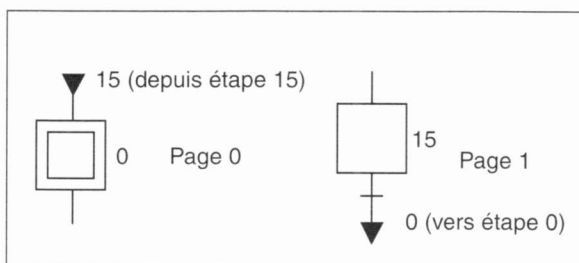


Figure 3.104
 Renvois entre deux pages.

- soit de répartir le GRAFCET sur plusieurs pages en utilisant des symboles de renvoi,

- soit de procéder par macro-représentations du type macro-étape.

Le langage PL7-3 distingue deux types de macro-étapes (fig. 3.105) :

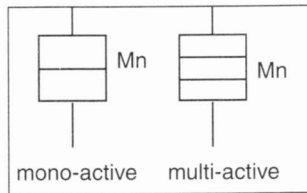


Figure 3.105
Symboles Télémécanique
des macro-étapes.

– **la macro-étape mono-active** dont l'expansion comporte soit une séquence unique soit plusieurs séquences exclusives. Un bit système SY25 permet de surveiller si l'expansion reste bien mono-active (une seule étape active à la fois) au cas où les réceptivités d'une divergence en OU deviendraient incidemment non exclusives ;

– **la macro-étape multi-active** dont l'expansion comporte des séquences parallèles (parallélisme structural ou parallélisme interprété).

Les actions associées aux étapes et les réceptivités associées aux transitions peuvent être programmées en langage graphique à contacts ou en langage littéral.

Quatre types d'actions sont autorisées :

- **action continue** : action commandée durant toute l'activité de l'étape n et notée S(C)n,
- **action continue temporisée** : action commandée pendant une durée de l'activité de l'étape n et notée S(C)n, durée,
- **action à l'activation** : action impulsionnelle commandée par le front montant de l'étape n et notée S(A)n,
- **action à la désactivation** : action impulsionnelle commandée par le front descendant de l'étape et notée S(D)n.

Exemples de programmation exploitant le concept de macro-étape

Inspiré du manuel du PL7-3 Télémécanique (extraits des écrans) :

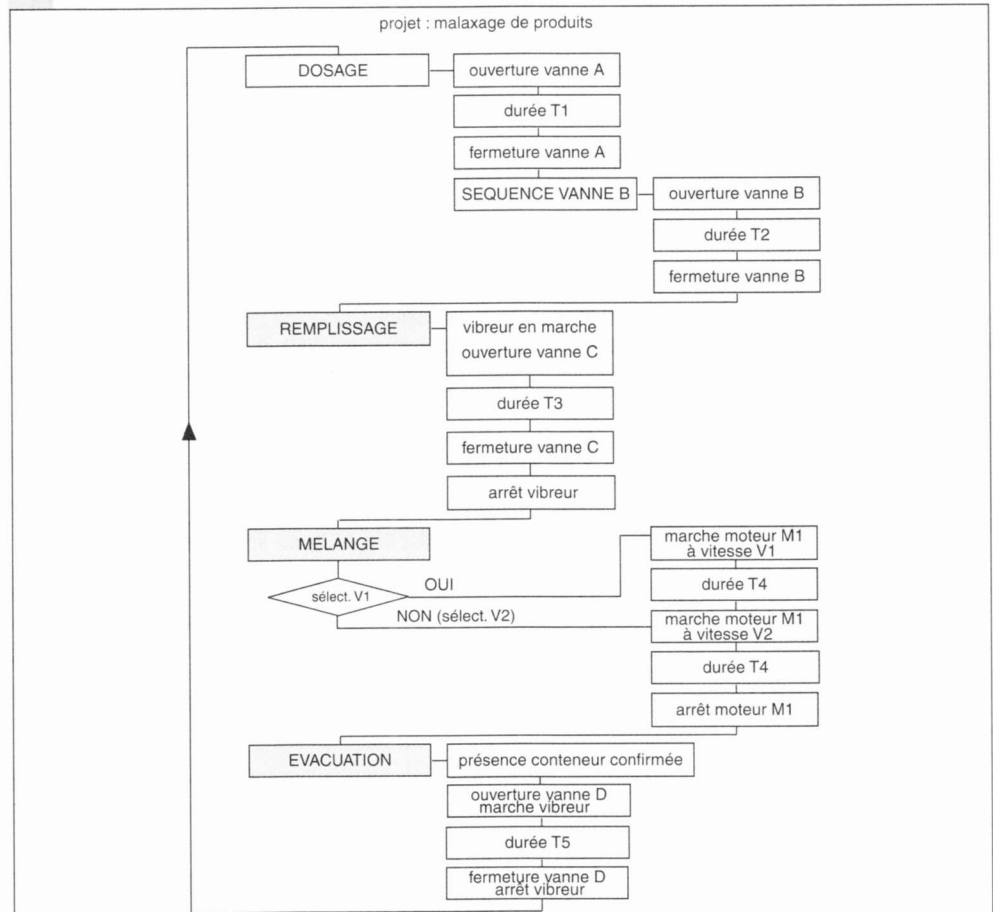


Figure 3.106
Organigramme préparatoire.

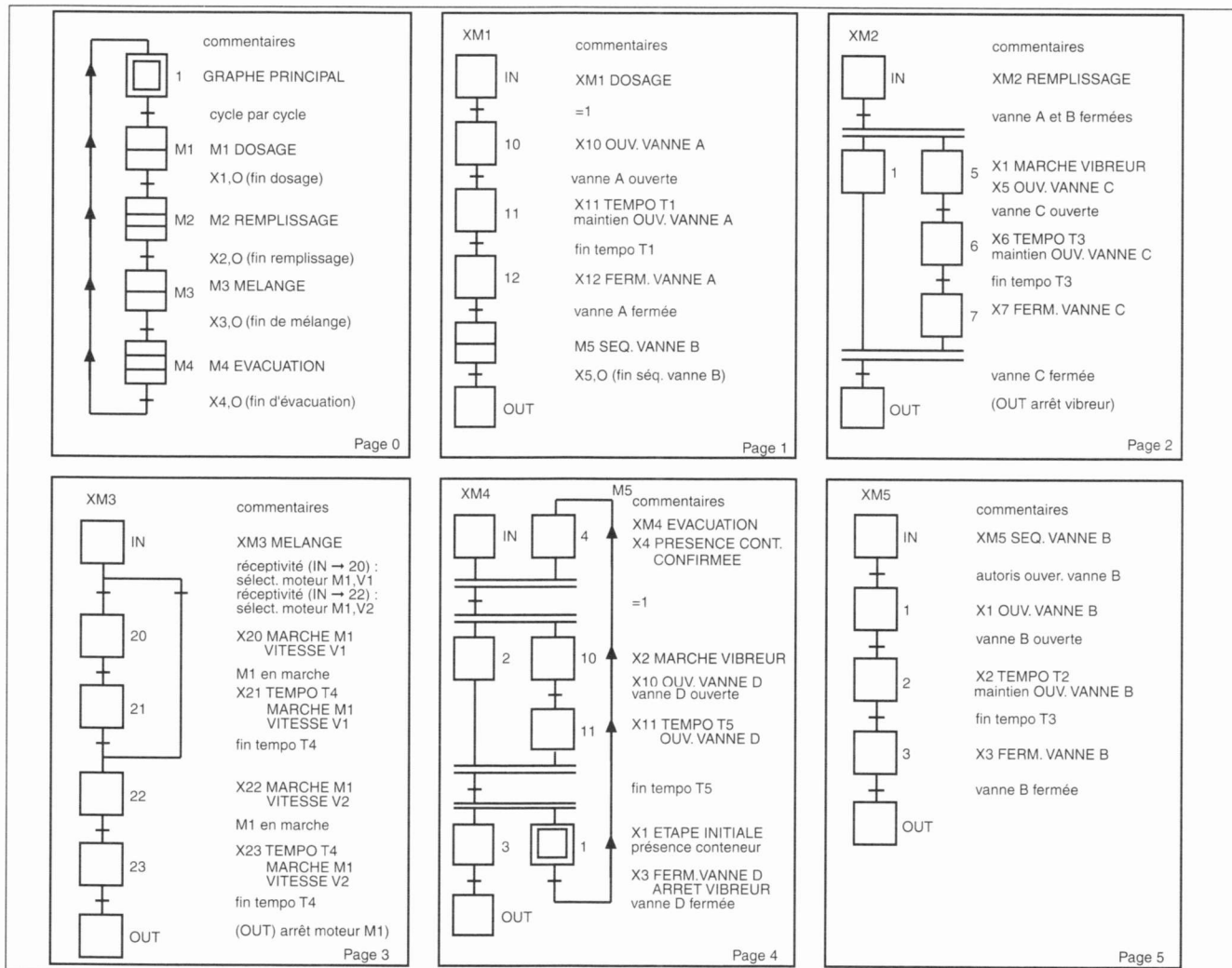


Figure 3.107 : GRAFCET de gestion des macro-étapes M1 à M5 et des expansions correspondantes.

Les actions associées aux étapes et précisées dans les commentaires sont traitées indépendamment du graphe des expansions, en traitement postérieur, au choix en langage à contacts ou en langage littéral. Chaque condition de transition est traitée à partir d'un zoom sur le symbole de la transition.

Remarque qu'aucune action n'est associée aux étapes d'entrée (IN) et de sortie (OUT). La notation (OUT arrêt moteur M1) en commentaire, page 3, n'est mise que pour rappeler qu'effectivement le moteur M1 s'arrêtera par défaut d'action à ce niveau.

13. Forçages et figeages de situation (UTE C 03-191)

13.1. Définition du forçage de situation

Par forçage de situation on entend le passage imposé de la situation courante du GRAFCET désigné à une situation déterminée différente de celle qu'aurait atteint celui-ci s'il avait évolué normalement.

La situation du GRAFCET forcé évolue alors sans franchissement de transition.

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

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

E_xemple de hiérarchie simple couramment adoptée

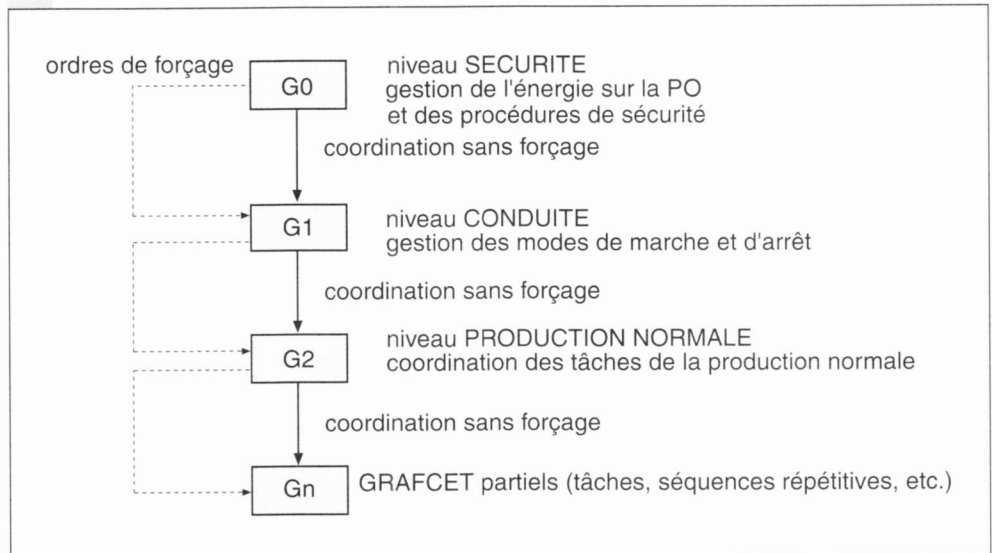


Figure 3.108
Hiérarchie des GRAFCET.

Il y a lieu maintenant de voir l'effet résultant d'un ordre de forçage sur la situation du GRAFCET forcé.

A. Situation courante forcée dans une situation précisée non vide

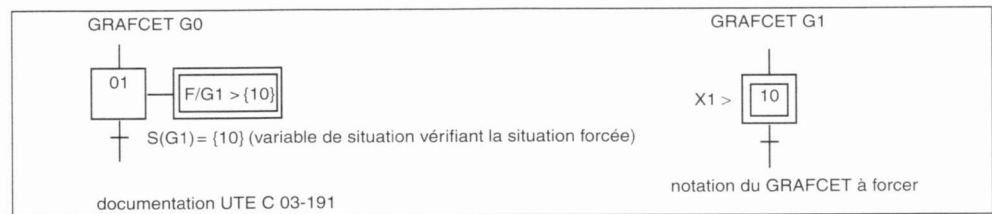


Figure 3.109
Forçage d'une situation
non vide.

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

Notations du forçage d'une situation non vide :

- de l'ordre de forçage du GRAFCET Gn : **F/Gn > {situation forcée}**
- du GRAFCET Gn forcé : **origine de l'ordre de forçage >**
(à accoler à gauche de l'étape à forcer à 1)

Remarque : le symbole > est disponible dans la plupart des éditeurs de GRAFCET des logiciels de PGO, ce qui n'est pas le cas du symbole → pourtant plus explicite.

B. Situation courante forcée dans une situation vide

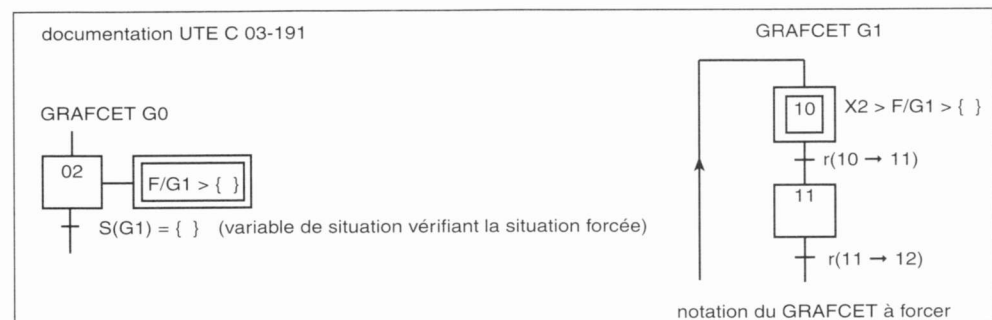


Figure 3.110
Forçage d'une situation vide.

L'activation de l'étape 1 du GRAFCET G0 force le GRAFCET G1 sur la situation vide $S(G1) = \{ \}$ de ce GRAFCET. Toutes les étapes du GRAFCET G1 sont désactivées y compris les étapes initiales.

Notations du forçage d'une situation vide :

- de l'ordre de forçage du GRAFCET G_n : **F/Gn** > { }
- du GRAFCET G_n forcé : **origine de l'ordre de forçage** > **F/Gn** > { }

(à accoler à droite de l'étape initiale du GRAFCET forcé).

L'intérêt du concept de forçage de situation d'un GRAFCET apparaît notamment lors du traitement des procédures de sécurité au niveau surveillance.

E

xemple comparatif

Description n'exploitant pas le concept de forçage de situation

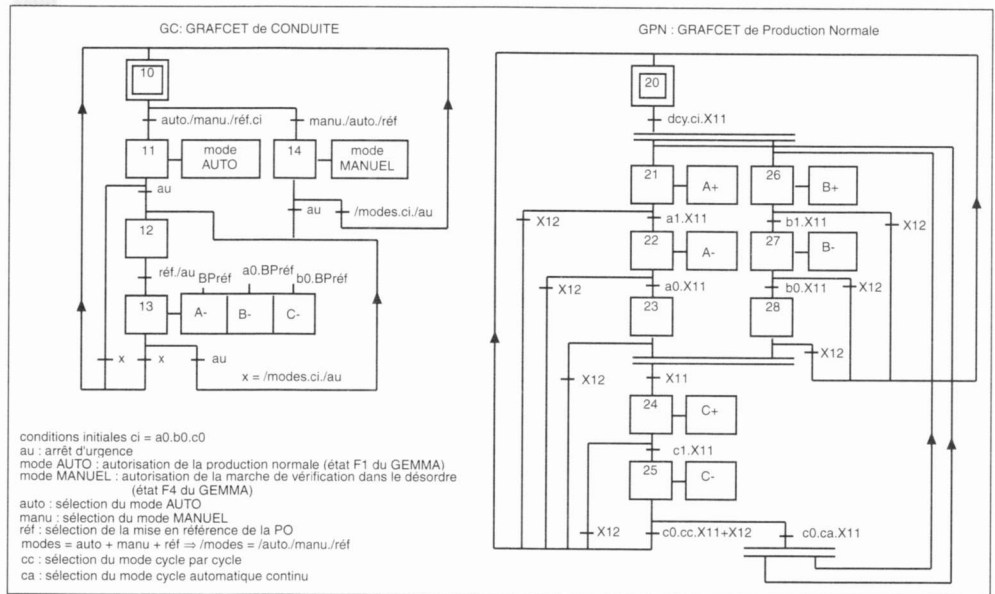


Figure 3.111
Description sans forçage
de situation.

Description exploitant le concept de forçage de situation

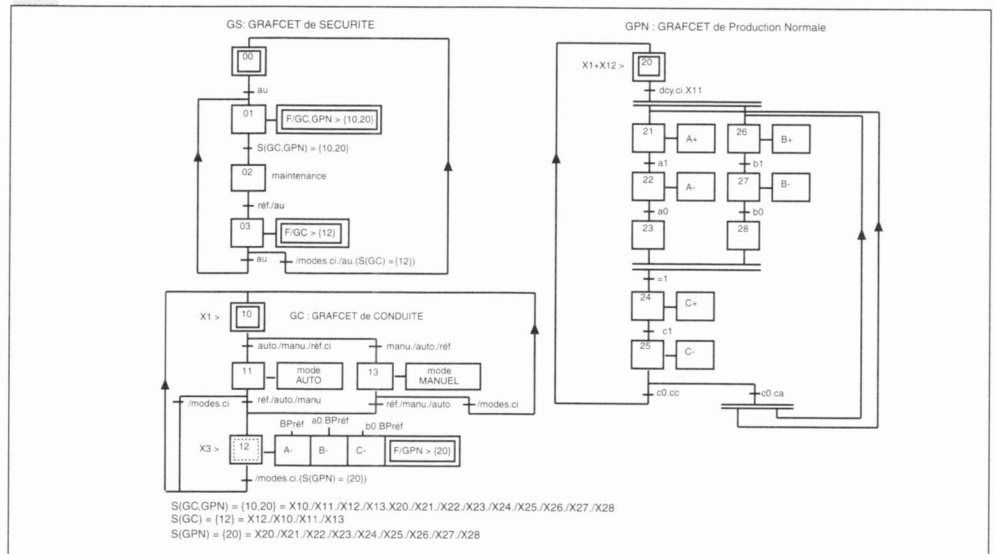


Figure 3.112
Description exploitant
le concept de forçage.

Conclusion

La comparaison est favorable au concept de forçage qui clarifie la description et améliore sa compréhension.

13.2. Ordre de forçage

Règles d'évolution par forçage

- 1 - Un GRAFCET ne peut être forcé que par un GRAFCET hiérarchiquement supérieur.
- 2 - Un GRAFCET inférieur ne peut être forcé que dans une seule situation à la fois à partir d'un ou plusieurs GRAFCET supérieurs.
- 3 - L'ordre de forçage est prioritaire sur les autres conditions assurant l'évolution du GRAFCET forcé.
- 4 - 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

$F/G2 > \{10\}$ est un ordre de forçage du GRAFCET G2 dans la situation imposée $S(G2) = \{10\}$. 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 10 et la désactivation des autres étapes du GRAFCET G2.

- 5 - 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.

Noter que dans certains cas, on peut simplifier l'écriture des forçages :

- si la situation forcée ne prête à aucune confusion
exemple : $F/G > \{10, 20\}$ force la situation $S(G) = \{10, 20\}$,
- si le forçage s'applique à plusieurs GRAFCET
exemple : $F/G1, G2 > \{10, 20\}$ 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
exemple : $F/G1 > \{INIT\}$ force G1 dans sa situation initiale.
 $F/G1-G5 > \{INIT\}$ force G1, G2, G3, G4, G5 dans leurs situations initiales.

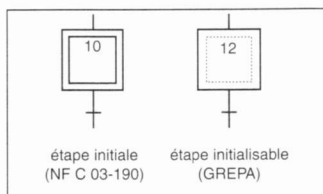


Figure 3.113
Symboles
des étapes particulières.

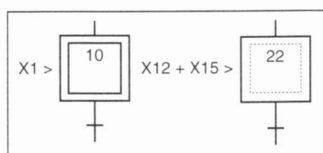


Figure 3.114
Repérage de l'origine
de l'ordre de forçage

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. 3.113).

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. 3.114).

Le symbole général proposé est : $X_i >$

Et si la référence à l'étape i ne suffit pas pour préciser l'origine du forçage, on écrit : $X_i(G_n) >$

Exemple de notations possibles (fig. 3.114) :

$X_1 >$ ou $X_1(GS) >$, $X_2 + X_{15} >$ ou $X_2(GS) + X_{15}(GC) >$

E_xemple

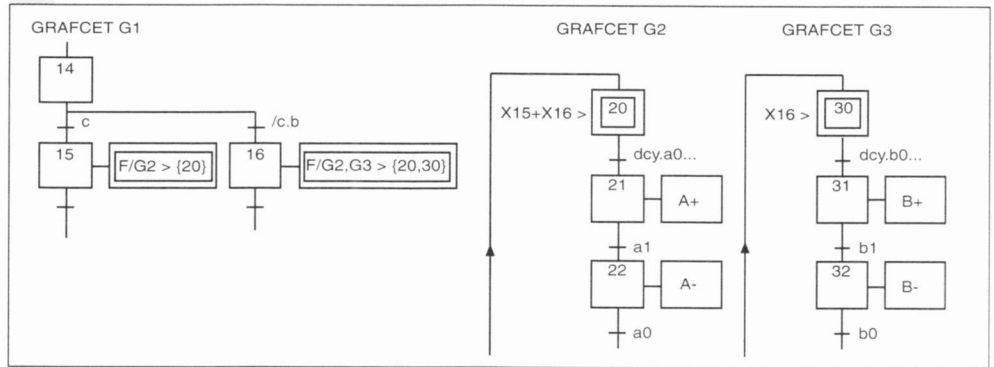


Figure 3.115
Ordres de forçage
et GRAFCET forcés.

Suivant la situation du GRAFCET G1, le système aura deux comportements distincts :

si $S(G2) = \{15\}$: initialisation du GRAFCET G2,

si $S(G2,G3) = \{16\}$: initialisation des GRAFCET G2 et G3.

Un ordre de forçage peut être conditionnel.

E_xemple (extrait de l'application chapitre III.13.7) (fig. 3.116)

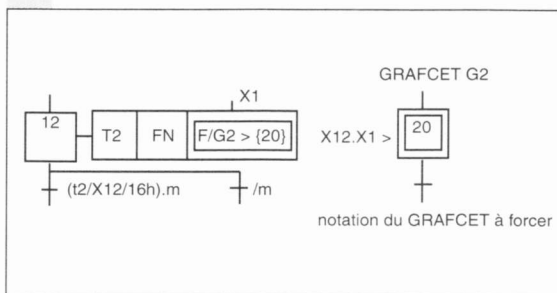


Figure 3.116
Ordre de forçage
conditionnel.

Activation de l'étape 20 :

$A(20) = X12.X1 + \dots$

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

Le forçage du GRAFCET G2 dans la situation $S(G2) = \{20\}$ sera effectif si l'étape 12 est active (feux normaux (FN)) et si le GRAFCET G0 est dans la situation $S(G0) = \{1\}$ c'est-à-dire en position feux tricolores jour (FT jour).

13.3. Figeages de situation

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.

A. Situation figée dans la situation courante

L'activation de l'étape 13 du GRAFCET G1 fige la situation du GRAFCET G2 dans sa situation courante.

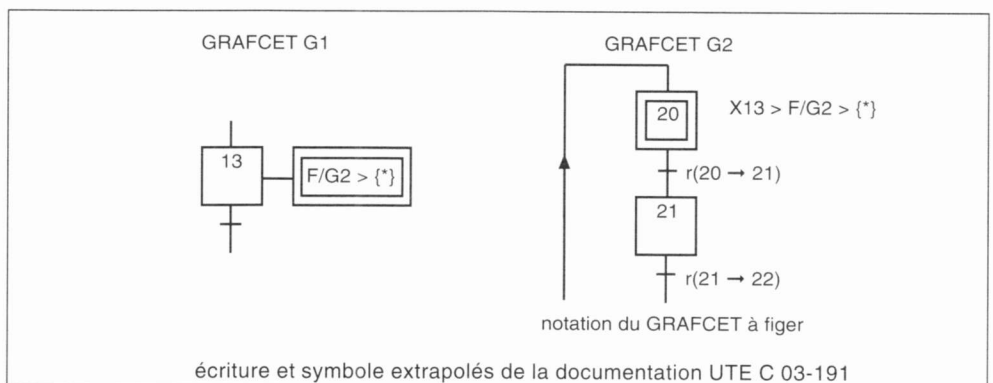


Figure 3.117
Ordre de figeage
dans la situation courante.

écriture et symbole extrapolés de la documentation UTE C 03-191

Les étapes actives restent activées, les étapes inactives restent désactivées.

Notations du figeage dans la situation courante :

- de l'ordre de figeage du GRAFCET Gn : $F/Gn > \{ * \}$
- du GRAFCET Gn à figer : **origine de l'ordre de figeage** $> F/Gn > \{ * \}$
(à accoler à droite de l'étape initiale du GRAFCET Gn à figer).

B. Situation évoluant vers une situation de figeage prédéterminée

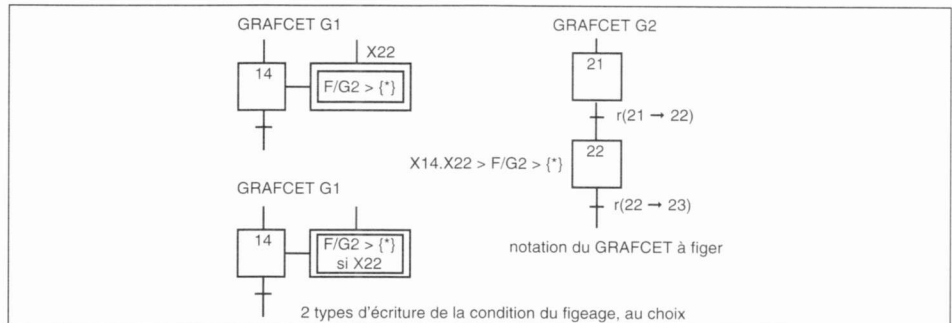


Figure 3.118
Ordre de figeage dans une situation prédéterminée.

La situation du GRAFCET G2 continue d'évoluer jusqu'à la situation $S(G2) = \{22\}$ où elle se figera. Il s'agit en fait d'un figeage conditionné par la situation de figeage à atteindre.

Notations du figeage dans une situation prédéterminée :

- de l'ordre de figeage du GRAFCET Gn sur l'étape Xi : $F/Gn > \{ * \}$ si Xi
- du GRAFCET à figer : **origine de l'ordre de figeage** $Xi > F/Gn > \{ * \}$
(à accoler de préférence à gauche de l'étape (initiale ou initialisable) où se fera le figeage)

13.4. Problèmes entraînés par le figeage de situation

A. 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, si c'est possible et si on le souhaite, 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.

Rappel

L'ordre de figeage entraîne :

- le maintien à l'état actif des étapes actives,
- le maintien à l'état inactif des états inactives.

Concrètement le figeage peut être obtenu :

- soit par la mise à 0 des réceptivités du GRAFCET désigné,
- soit par un ordre spécifique de figeage propre au constituant retenu pour réaliser la PC.

Il faut réfléchir au problème posé par le figeage de la situation d'un GRAFCET de production en ce qui concerne la PO. Si le figeage est obtenu, les ordres émis par les 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 ou des GRAFCET de Production Normale (GPN), émettre un ordre de forçage à l'état 0 des sorties de la PO ou limité à celles du ou des GRAFCET de Production Normale GPN.

Le forçage à 0 des sorties de la PO ou des GRAFCET GPN, 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.).

Voir § 13.7 les notations proposées pour le forçage à 0 des sorties de la PO, de la PR et du ou des GRAFCET GPN.

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 GPN (figure 3.119) 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. 3.120),
- soit en exploitant un bit système approprié (SY9 ou S9 pour les API Schneider Télémécanique),
- soit en combinant les procédés afin d'obtenir une redondance améliorant la sécurité.

Exemple (avec GRAFCET de Conduite simplifié)

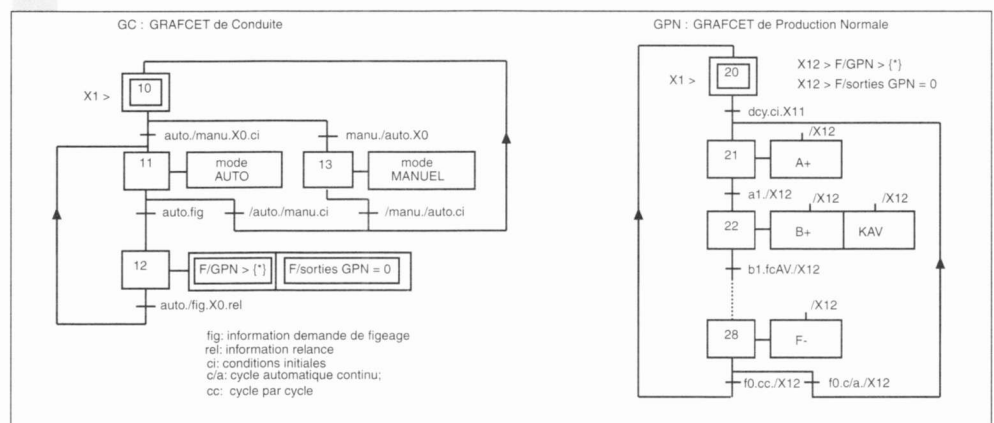


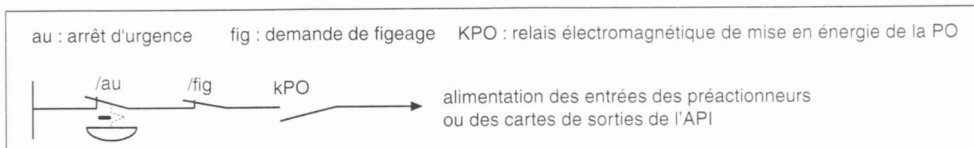
Figure 3.119
Exemple de figeage assorti
d'un forçage à 0 des sorties.

Le figeage du GRAFCET GPN de l'exemple (fig. 3.119) est obtenu par la mise à zéro des des réceptivités si l'étape 12 est active (information /X12). 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.

Forçage à 0 des sorties de la PO, exemple de solution câblée possible

Figure 3.120
Solution câblée du forçage
à 0 des sorties.



B. Cas du figeage 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 ci-dessus dans une situation vide.

Notation (écriture extrapolée de la doc. UTE C 03-191) : $SV(Gn) = \{ * \}$

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 doc. UTE C 03-191) : $RT(Gn) = \{ * \}$

N.B. : SV : pour SaVe ou SauVegarde, RT : pour ResTore ou ResTauraton.

Cette procédure particulièrement intéressante 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 des TSX Schneider Télémécanique possèdent des instructions autorisant ce type de procédure (voir applications 2 et 3, chapitre VII, § 10.7).

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.

13.5. Réalisation des forçages et des figeages de situation

D'une manière générale :

- le forçage de situation s'obtient en activant les étapes citées dans la situation à forcer et en désactivant toutes les autres étapes du GRAFCET désigné, par l'étape émettant l'ordre de forçage soit Xf.
- L'activation d'une étape forcée à 1 doit être prioritaire sur sa désactivation.

De même la désactivation d'une étape forcée à 0 doit être prioritaire sur son activation.

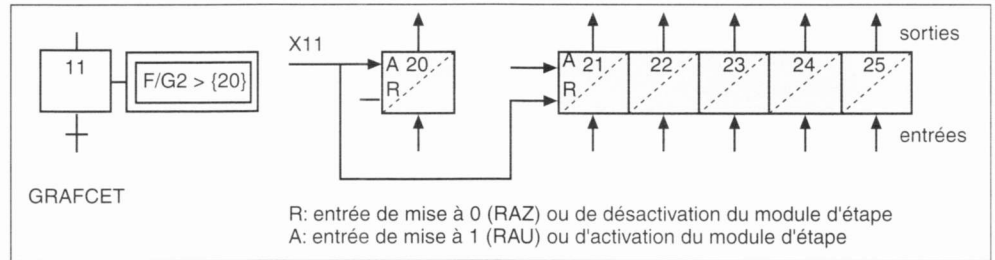
- le forçage d'une situation vide s'obtient en désactivant toutes les étapes du GRAFCET désigné par l'étape émettant l'ordre de forçage soit Xf. La désactivation de toutes les étapes forcées à 0 doit être prioritaire sur leurs activations.

Le figeage dans la situation courante s'obtient, entre autres procédés, en conditionnant les réceptivités du GRAFCET désigné par le complément de l'étape émettant l'ordre de figeage soit /Xf.

Mais il peut exister une manière particulière et mieux appropriée d'obtenir le forçage ou le figeage compte tenu du constituant retenu pour réaliser la PC.

A. Avec les séquenceurs pneumatiques

Figure 3.121
Forçage de situation
avec un séquenceur
pneumatique.



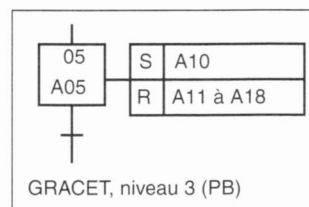
Un séquenceur est obtenu en associant autant de modules d'étapes (mémoires d'étapes) qu'il y a d'étapes dans le GRAFCET à concrétiser.

Le forçage d'une situation vide est obtenu facilement par mise à 1 de l'entrée de remise à zéro commune (R pour RAZ) à tous les modules. Le forçage d'une situation particulière nécessite la dissociation du séquenceur de manière à isoler le module d'étape à activer (mise à 1 de l'entrée A ou RAU), les autres modules d'étapes étant désactivés par l'entrée commune RAZ (fig. 3.121).

B. Avec les automates programmables industriels (API)

a. Cas du langage des PB (APRIL)

Figure 3.122
Ordre de forçage
de situation
du langage des PB.



Ce langage GRAFCET alphanumérique comporte deux instructions de forçage :

- FORU : forçage à 1 du bit interne associé à l'étape,
- FORZ : forçage à 0 du bit interne associé à l'étape.

APPLICATION (fig. 3.122)

- 0C60 SI A05 test du bit A05 image de l'étape 5
- 0C61 FORU 0A10 activation du bit A10 image de l'étape 10
- 0C63 SI A05 test du bit A05 image de l'étape 5
- 0C64 FORZ 0A11 désactivation du bit A11 image de l'étape 11
- 0C66 0A18 jusqu'au bit A18 image de l'étape 18

b. Cas des langages GRAFCET PL7-2 et PL7-3 Schneider Télémécanique

Des bits systèmes notés SY sont disponibles pour assurer les différents forçages mais avec des nuances à souligner :

- la mise à 1 de SY21 (Set) assure la réinitialisation de tous les GRAFCET,
- la mise à 1 de SY22 (Set) assure la désactivation de tous les GRAFCET,
- la mise à 1 de SY23 (Set) autorise l'activation par l'instruction Set Xi ou la désactivation par l'instruction Reset Xi de l'étape i voulue.
- le maintien à 1 de SY23 fige l'ensemble des GRAFCET sur la situation courante d'où la nécessité de prévoir son déverrouillage après un forçage de situation si on veut que l'automatisme puisse continuer à évoluer.

La remise à 0 des bits SY21, SY22 et SY23 est assurée par le système à condition qu'ils ne soient pas maintenus à 1 par un autre signal (PL7-2).

On notera surtout que les bits SY21 et SY22 ont un effet général, ils ne peuvent donc agir sur un GRAFCET partiel. Il faudra bien réfléchir avant d'en faire usage.

L'exploitation des bits système se fait en traitement préliminaire, le traitement séquentiel étant réservé au GRAFCET et le traitement postérieur aux sorties.

APPLICATION

Forçage du GRAFCET GPN ($F/GPN > \{10\}$) par l'étape 5 du GRAFCET GC (fig. 3.123)

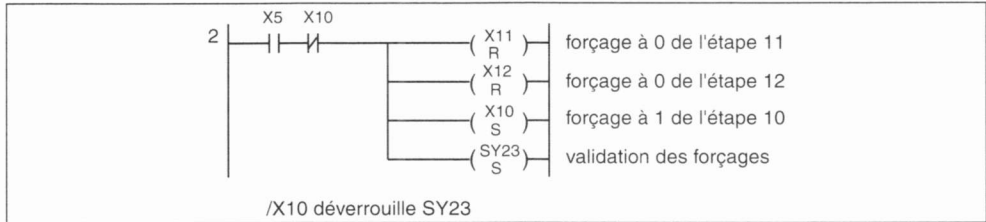


Figure 3.123
Réseau de forçage
en langage
à contacts PL7-2.

c. Cas d'autres logiciels

Logiciel PL7-3 V5 (voir chapitre VII. 10.7, application 2),
Logiciel ORPHEE (voir chapitre VII. 10.7, application 3),
Logiciel AUTOMGEN (voir chapitre VIII. 3).

13.6. Variables de situation

Après un forçage de situation, il faut s'assurer que la situation forcée est bien obtenue (compte rendu du forçage). La reprise des évolutions du GRAFCET supérieur émetteur de l'ordre de forçage doit normalement dépendre de la situation atteinte par le GRAFCET forcé.

Exemple

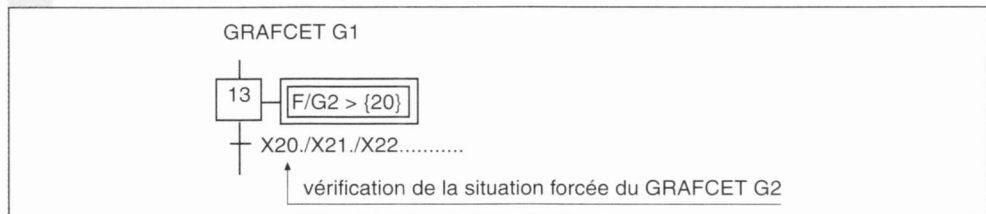


Figure 3.124
Variable de situation.

On peut remplacer l'expression logique $X20./X21./X22.....$ par $S(G2) = \{20\}$.

S(G2) : {20} est appelée une variable de situation

La notation proposée, plus générale pour un GRAFCET Gn, s'écrit ainsi :

S(Gn) = {situation forcée}

Cette notation n'est en fait qu'une simplification d'écriture dans le GRAFCET et elle devra obligatoirement être explicitée donc traitée par ailleurs.

Par exemple :

$S(G2) = \{24\} = /X20./X21./X22./X23.X24./X25./X26.....$

$S(G3) = \{ \} = /X30./X31./X32./X33.....$

$S(G4) = \{INIT\} = X40./X41./X42./X43.....$

La variable de situation peut se limiter au test de la ou des étapes forcées à 1 si le langage GRAFCET implanté dans l'API dispose d'une structure de forçage appropriée et sûre (cas de la structure exploitant le langage sur mot (chapitre VII, § 10-5).

13.7. Forçage des sorties de la PO et de la PR

On a vu précédemment l'intérêt présenté par le forçage à 0 des sorties de la PO dans le cas du figeage de situation. Ce type de forçage constitue une mesure de sécurité souvent associée à une procédure d'arrêt d'urgence.

Le forçage à 0 des sorties de la PO est également appliqué momentanément lors de la mise en énergie du constituant de la PC pour isoler les sorties pendant cette période critique.

Lorsque l'API dispose d'un « chien de garde », le constructeur conseille de câbler les sorties en série avec le contact du relais de cette protection. Le « chien de garde » contrôle la bonne exécution des programmes et en cas d'anomalie agit sur un relais noté CdG. Il s'agit là d'une autre forme de forçage à 0 des sorties. Dans le langage de certains API, un bit système est disponible pour neutraliser les sorties en cas de besoin (bit système SY9 des API Schneider Télémécanique, par exemple).

Notations (proposées par extrapolation de la documentation UTE C 03-191)

- de l'action de forçage à l'état 0 des sorties listées ou non :
- de la PO : **F/sorties PO = 0**
- de la PR : **F/sorties PR = 0 (ou 1)**
- du GRAFCET Gn : **F/sorties Gn = 0**
- de l'ordre de forçage à l'état 0 des sorties listées ou non :
- de la PO : **origine de l'ordre de forçage > F/sorties PO = 0**
- de la PR : **origine de l'ordre de forçage > F/sorties PR = 0 (ou 1)**
- d'un GRAFCET Gn : **origine de l'ordre de forçage > F/sorties Gn = 0**

Exemple d'écriture dans le GRAFCET

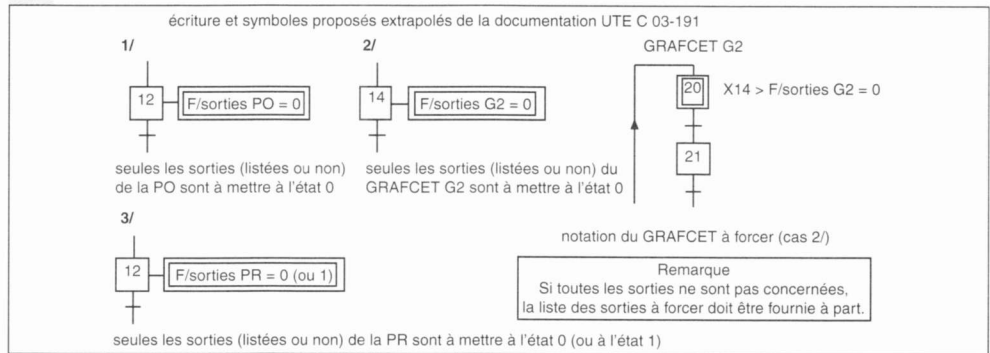


Figure 3.125
Forçage des sorties
de la PO et de la PR.

Le forçage à l'état 1 des sorties de la PR est également envisageable, par exemple pour tester l'ensemble des voyants d'un panneau de signalisation. La notation proposée sera modifiée en conséquence. Si seules certaines sorties de la PO ou de la PR sont à forcer, la liste correspondante doit être fournie (voir la remarque, fig.3.125). Le forçage à 0 ou à 1 des entrées-sorties à partir de la console de programmation des API ne doit pas être confondu avec le forçage des sorties étudié ci-dessus. Ce n'est qu'une possibilité de test supplémentaire offerte aux techniciens chargés de la mise au point ou de la maintenance de l'installation automatisée.

13.8. Exemple d'application des forçages de situation

L'exemple choisi concerne la gestion de feux de carrefour (fig. 3.126).

• Cahier des charges

Une horloge gère les feux tricolores et les feux clignotants dans les modes dits feux normaux FN et feux clignotants CL :

- de 6 à 22 heures → feux tricolores FT,
- de 22 à 6 heures → feux clignotants CL.

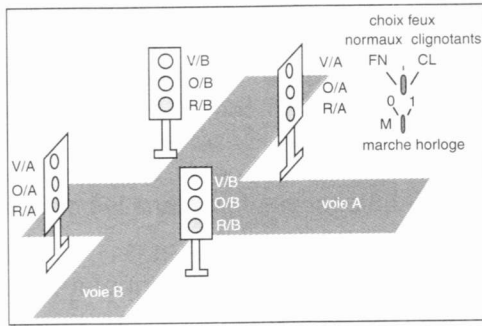


Figure 3.126
Schéma de principe du
carrefour à protéger.

On veut, sur demande de l'agent de service (information CL), pendant le jour, passer instantanément en mode clignotant.

• **Solution proposée**

La description fait appel à un ensemble de quatre GRAFCET hiérarchisés (fig. 3.127) :

- un GRAFCET G0 de gestion des modes FN et CL (fig. 3.128),

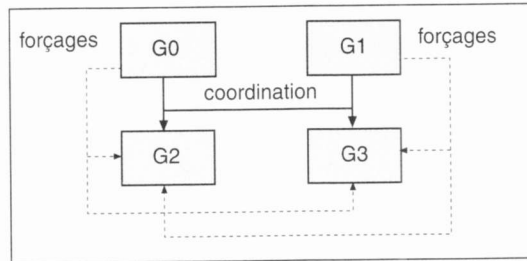


Figure 3.127
GRAFCET hiérarchisés.

- un GRAFCET G0 gère les modes FN et CL (fig. 3.128),

- un GRAFCET G2 décrit les évolutions des feux clignotants (fig. 3.129),

- un GRAFCET G3 décrit les évolutions des feux tricolores (fig. 3.129).

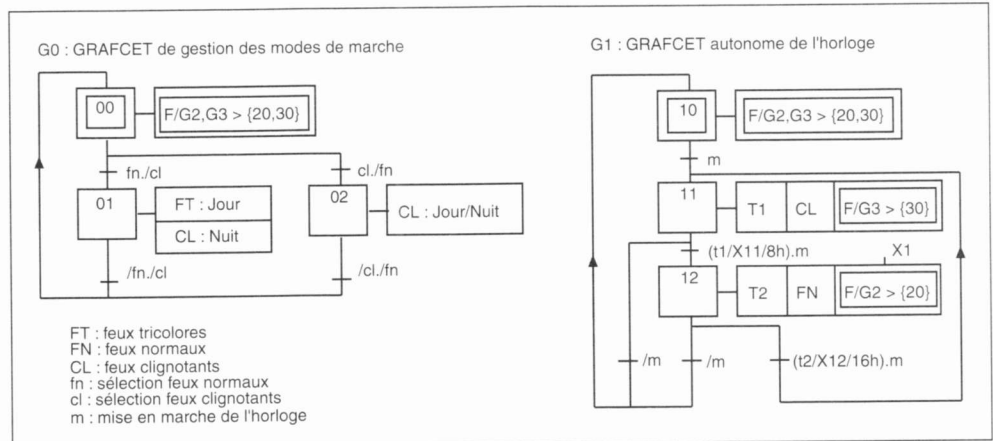


Figure 3.128
GRAFCET de gestion des
modes de marche
et GRAFCET de l'horloge.

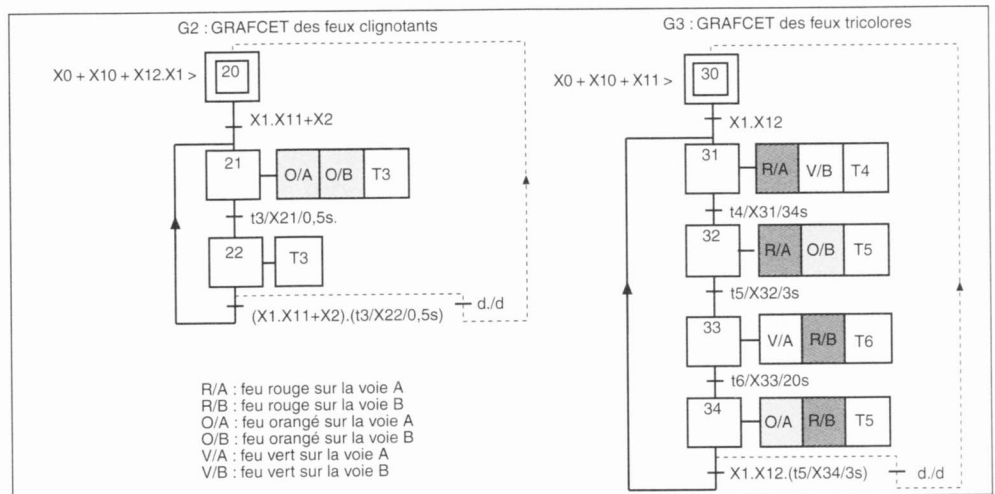


Figure 3.129
GRAFCET
des feux tricolores
et des feux clignotants.

• Commentaires

L'arrêt de l'horloge assuré par /m réinitialise le GRAFCET G1. La situation $S(G1 = \{10\})$ force les GRAFCET G2 et G3 dans leurs situations initiales $S(G2, G3) = \{20, 30\}$. Le retour au neutre du sélecteur de modes (fn, cl) met le GRAFCET G0 dans la situation $S(G0) = \{0\}$ ce qui a pour effet de forcer les GRAFCET G2 et G3 dans leurs situations initiales. Dans les deux cas les feux sont éteints.

La situation $S(G0, G1) = \{1, 11\}$ ou $S(G0, G1) = \{2, 11\}$ des GRAFCET G0 et G1 autorise le fonctionnement des feux clignotants et force la réinitialisation du GRAFCET G3 des feux tricolores.

La situation $S(G0, G1) = \{1, 12\}$ des GRAFCET G0 et G1 autorise le fonctionnement des feux tricolores et force la réinitialisation conditionnelle (X1) du GRAFCET G2 des feux clignotants.

Les temporisateurs T3 et T5 sont utilisés à deux reprises puisqu'on a besoin deux fois du retard de 0,5 seconde et deux fois du retard de 3 secondes.

Remarquer le non-rebouclage direct des GRAFCET G2 et G3 sur leurs étapes initiales. Ce sont les actions de forçages qui assurent, ici, le rebouclage indirect sur ces étapes.

Remarque

Certains langages GRAFCET alphanumériques ou graphiques implantés dans les API n'acceptent pas les graphismes en boucle ouverte.

On procède alors à un rebouclage fictif (voir tracé en pointillé) par l'intermédiaire d'une transition à laquelle est associée une réceptivité toujours nulle, par exemple d./d.

Cette application constituant un excellent exercice testable sur API, on pourra le compléter et ajouter aux modes de marche, une marche étape par étape E/E (sélection du mode E/E et appui fugitif sur un bouton BPe/e) et un test global des lampes des feux (sélection du mode TEST et appui sur un bouton BPtest).

Les variables de situation pourront être ajoutées dans les transitions appropriées des GRAFCET G0 et G1.

Rappel

La fonction transition a pour expression le produit logique des variables conditionnant le franchissement de la transition : étapes précédentes actives et réceptivité vraie.

Tableau récapitulatif des expressions logiques tirées du GRAFCET

expressions logiques des feux		forçages de situation
$R/A = X31 + X32$	$R/B = X33 + X34$	$F/G2 > \{20\} = X0 + X10 + X12.X1$
$O/A = X21 + X34$	$O/B = X21 + X32$	$F/G3 > \{30\} = X0 + X10 + X11$
$V/A = X33$	$V/B = X31$	temporisateurs
fonctions transitions		$T1 = X11$
$ft(20 \rightarrow 21) = X20.(X1.X11 + X2)$		$T2 = X12$
$ft(21 \rightarrow 22) = X21.(t3/X21/0,5s)$		$T3 = X21 + X22$
etc.		$T4 = X31$
$ft(30 \rightarrow 31) = X30.X1.X12$		$T5 = X32 + X34$
$ft(31 \rightarrow 32) = X31.(t4/X31/34s)$		$T6 = X33$
etc.		

activations et désactivations des étapes

$A(20) = F/G2 > \{20\}$	forçage à 1 prioritaire
$D(20) = /(F/G2 > \{20\}).ft(20 \rightarrow 21)$	
$A(21) = /(F/G2 > \{20\}).(ft(20 \rightarrow 21) + ft(22 \rightarrow 21))$	forçage à 0 prioritaire
$D(21) = (F/G2 > \{20\}) + ft(21 \rightarrow 22)$	
etc.	
$A(30) = (F/G3 > \{30\})$	forçage à 1 prioritaire
$D(30) = /(F/G3 > \{30\}).ft(30 \rightarrow 31)$	
$A(31) = /(F/G3 > \{30\}).(ft(30 \rightarrow 31) + ft(34 \rightarrow 31))$	forçage à 0 prioritaire
$D(31) = (FS \{30\}) + ft(31 \rightarrow 32)$	
etc.	

Pour plus de renseignements sur la recherche des expressions logiques à partir du GRAFCET, voir le chapitre VI.

Remarques concernant les notations des forçages et des figeages de situation

L'écriture manuelle des ordres de forçage et de figeage sur un document ne pose aucun problème.

Il en est de même de l'écriture sur l'écran en mode alphanumérique avec un logiciel de dessin (en traitement de texte avec le DRAW de Word, par exemple).

Par contre, il faut que l'éditeur du langage de l'API ou du logiciel de PGOA puisse exploiter les instructions de forçage lors de la compilation.

Si ce n'est pas le cas, on se contentera d'écrire les ordres de forçage et de figeage dans la zone « commentaires » généralement prévue dans ces logiciels.

Quant à la réalisation, elle fera appel à l'une des structures développées au chapitre VII, §10.

Noter que, quel que soit le logiciel utilisé, pour le traitement du figeage de situation et du forçage des sorties, il appartient à l'automaticien d'introduire les conditions adéquates dans les GRAFCET du SAP à réaliser.

Remarque concernant la documentation UTE C 03-191

Ce fascicule propose une nouvelle notation pour les forçages de la forme $F/GRAFCET > \{situation \text{ à forcer}\}$ écrite dans le GRAFCET dans une case d'action doublement encadrée.

Noter que la mise à jour de la norme C 03-190 de 1995 n'intègre pas cette proposition.

Noter que le symbole caractère « > » a été retenu de préférence au symbole « → » qui fait partie des caractères spéciaux. Généralement, il n'est pas directement disponible sur les claviers de console ou de micro-ordinateurs.

IV. COMMENT ÉTABLIR UN GRAFCET CORRECT ?

1. Erreurs possibles

Le GRAFCET (associé au GEMMA) permet de décrire progressivement l'automatisme d'un système (analyse descendante) en respectant le cahier des charges proposé par le client.

En principe aucun arbitraire ne doit présider au choix des séquences et de leurs enchaînements ou de leurs coordinations pas plus qu'au choix technologique des composants et des constituants. Encore faut-il avoir des idées sur les systèmes, savoir faire des choix comparatifs et également bien connaître la syntaxe du GRAFCET afin d'éviter des erreurs.

Les erreurs peuvent être de deux sortes :

- d'une part des **erreurs d'analyse**,
- d'autre part des **erreurs d'écriture**.

Pour éviter les erreurs d'analyse, il faut :

- comprendre le cahier des charges et le processus à décrire,
- organiser une hiérarchie entre les GRAFCET,
- choisir des structures d'évolution appropriées à la description,
- vérifier les coordinations entre GRAFCET,
- choisir la situation initiale de l'ensemble des GRAFCET,
- choisir la position de référence de la PO,
- choisir entre les divers types d'actions possibles,
- exprimer correctement les réceptivités,
- etc.

Quant aux erreurs d'écriture, il faut distinguer **les erreurs de syntaxe des erreurs de forme**.

Il est important de souligner qu'un langage GRAFCET graphique implanté dans un API et conforme au GRAFCET normalisé, ne tolère pas les erreurs d'écriture.

Le langage génère un message d'erreur pour attirer l'attention du programmeur. Il est impossible de valider un programme comportant des erreurs de graphisme. Les erreurs concernant les entrées-sorties sont détectées si une table des symboles a préalablement été établie.

2. Erreurs de syntaxe

Il s'agit essentiellement du non-respect de la règle d'alternance étape-transition et transition-étape (fig. 4.1).

3. Erreurs de forme

Pour éliminer ce type d'erreurs, on peut utiliser une méthode basée sur la représentation du GRAFCET à l'aide du graphisme des **réseaux de Petri**. Chaque étape est figurée par un cercle et la transition par une barre. On ne peut jamais aller d'un cercle à un autre sans rencontrer une barre ce qui traduit le respect de la règle d'alternance étape-transition et transition-étape.

La méthode en question consiste à tracer le GRAFCET avec le symbolisme des réseaux de Petri puis à le traduire en GRAFCET normalisé.

Exemple

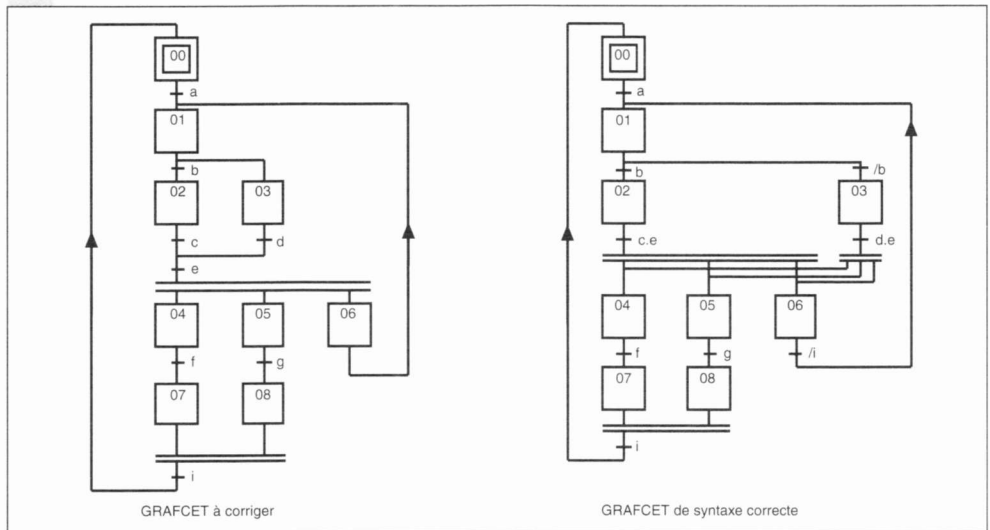


Figure 4.1
Erreurs de syntaxe.

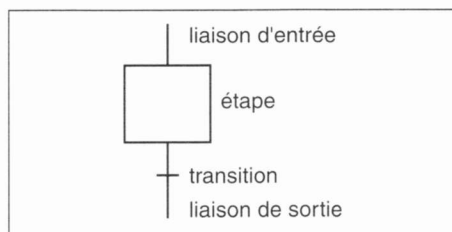


Figure 4.2
Ensemble indissociable
du GRAFCET.

Dans la représentation proposée dans la norme du GRAFCET, le symbole de l'étape (un carré), le symbole de la transition (une barre), les traits de liaisons d'arrivée et de départ forment un tout indissociable (fig. 4.2).

Dans le graphisme de Petri (fig. 4.3) :

- plusieurs flèches arrivant ou partant d'une étape (cercle) traduisent respectivement une convergence ou une divergence en OU,
- plusieurs flèches arrivant ou partant d'une transition (barre) traduisent respectivement une convergence et une divergence en ET.

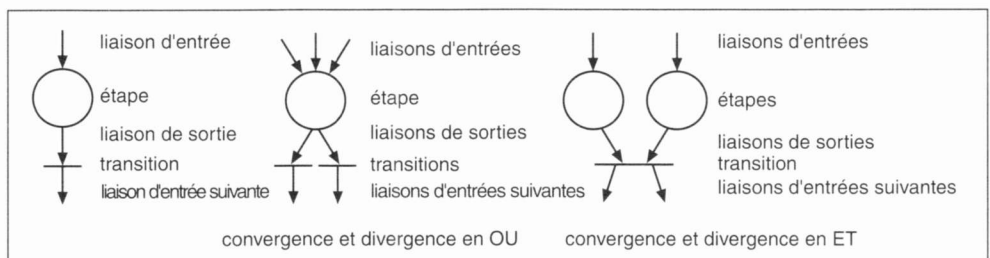


Figure 4.3
Le graphisme de Petri.

Comparaison des graphismes du GRAFCET et de Petri (fig. 4.4)

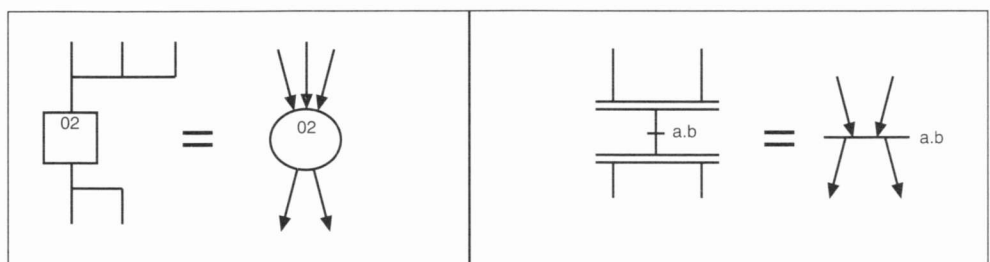


Figure 4.4
Graphismes comparés
du GRAFCET
et des réseaux de Petri.

Partant de cette observation, on peut remédier à de nombreuses erreurs de forme.

4. Erreurs de forme à corriger (fig. 4.5 et 4.6)

(Inspiré d'une étude de M.Blanchard : Comprendre et maîtriser le GRAFCET)

Figure 4.5
Erreurs de forme à corriger
(1).

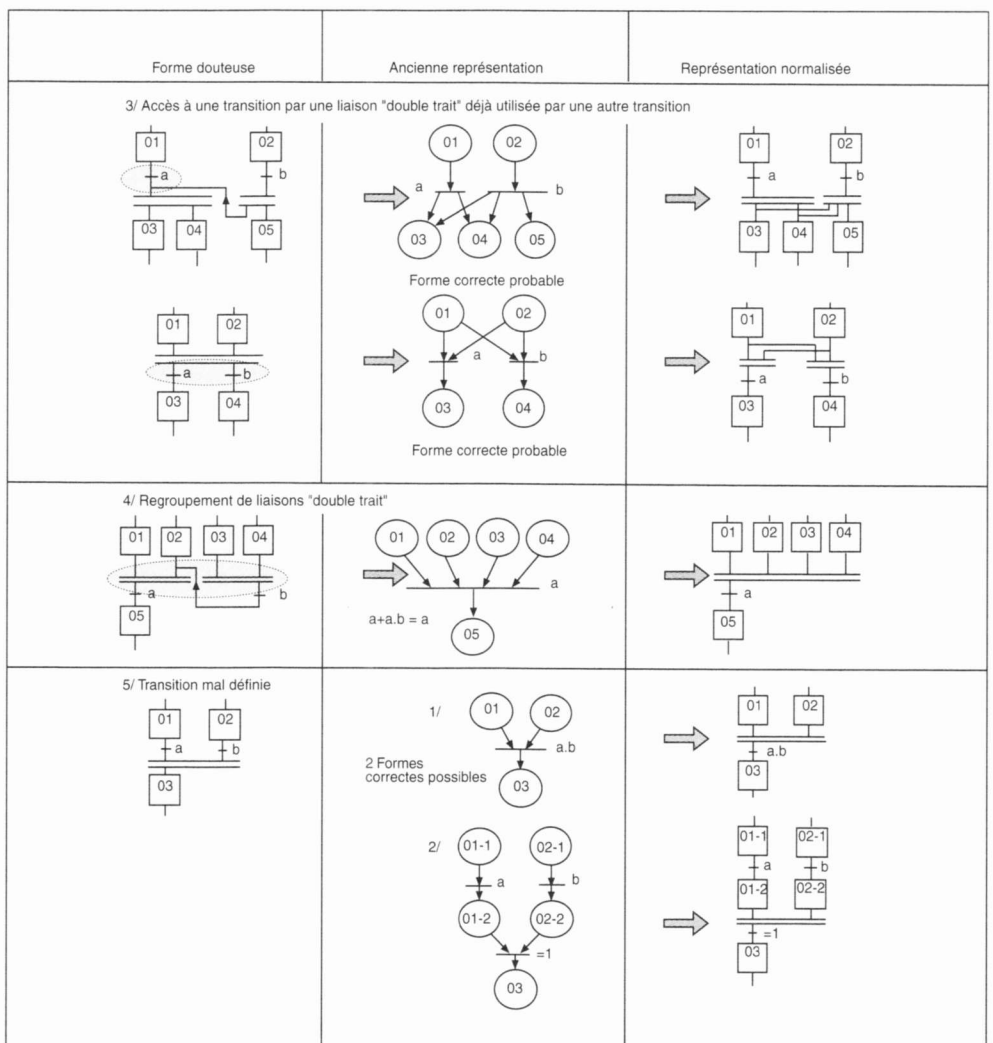
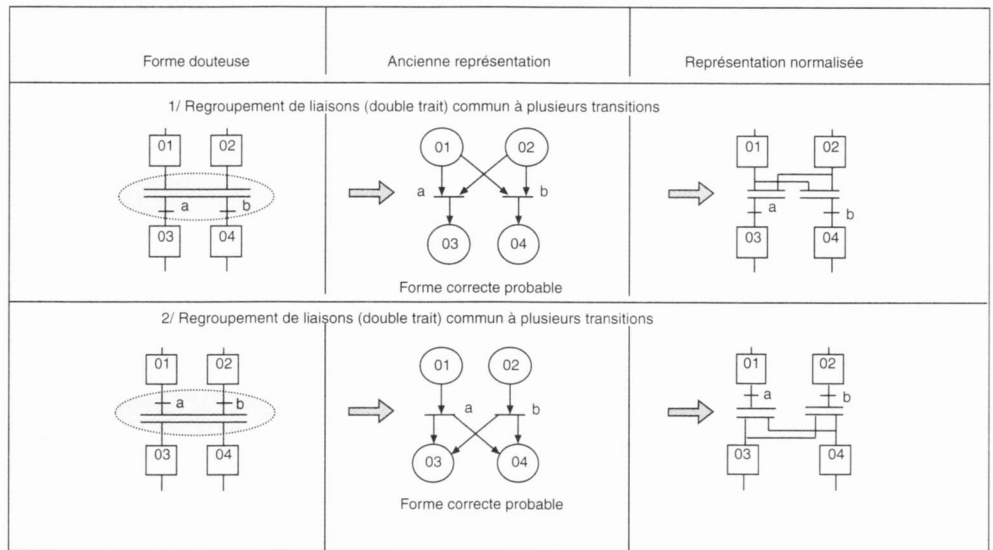


Figure 4.6
Erreurs de forme à corriger
(2).

Retour sur les formes 2, 4 et 5

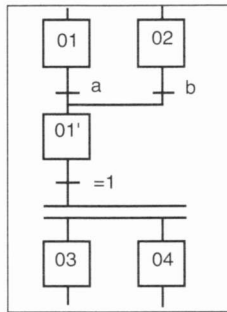


Figure 4.7
Cas de la forme 2.

• Cas de la forme 2

En ajoutant une étape supplémentaire (étape 01'), on simplifie le graphisme du GRAFCET (fig. 4.7).

• Cas de la forme 4

Une mauvaise écriture entraîne des interprétations différentes.

Exemples

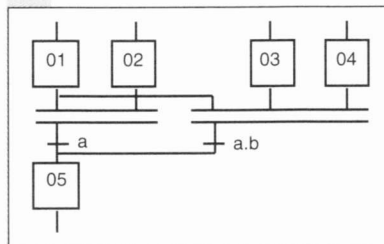


Figure 4.8
Cas de la forme 4 (1).

Si X1 et X2 et a=1 alors évolution vers 5
si X1 et X3 et X4 et a.b = 1 alors évolution vers 5
d'où l'expression de l'activation de l'étape 5 :

$$A(5) = X1.X2.a + X1.X3.X4.a.b$$

$$= a.X1(X2 + X3.X4.b)$$

Rien à signaler ici.

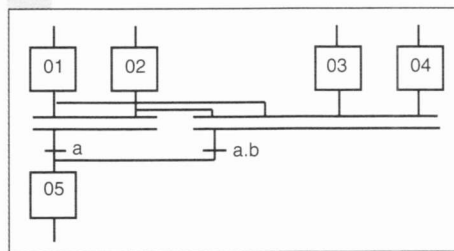


Figure 4.9
Cas de la forme 4 (2).

Si X1 et X2 et a = 1 alors évolution vers 5
si X1 et X2 et X3 et X4 et a.b=1 alors évolution vers 5
d'où l'expression de l'activation de l'étape 5 :

$$A(5) = X1.X2.a + X1.X2.X3.X4.a.b$$

$$= X1.X2.a(1 + X3.X4.b) = X1.X2.a,$$

le terme entre parenthèses valant 1 (relation logique remarquable de la forme $1 + a = 1$).

Les étapes 3, 4 et la réceptivité b sont inutiles.

• Cas de la forme 5 (fig. 4.10)

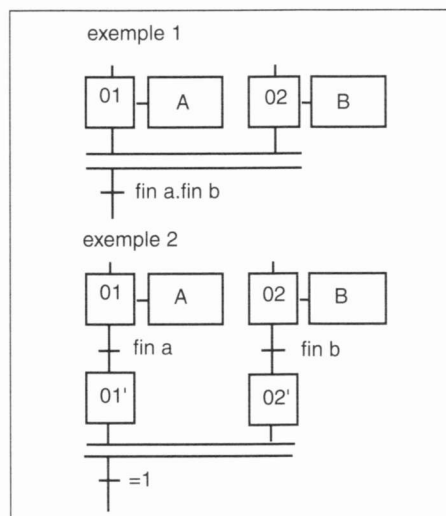


Figure 4.10
Cas de la forme 5.

S'il y a des actions associées aux étapes 1 et 2 le fonctionnement n'est pas le même.

Dans l'exemple 1, il faut que les deux actions associées aux étapes 1 et 2 soient terminées pour que l'évolution 1,2 → 3 soit possible.

Une des actions terminée reste à l'état 1 en attendant la fin de l'autre.

Dans l'exemple 2, des étapes d'attente ont été ajoutées pour pallier au problème précédent.

• Cas particulier d'une boucle sur une seule étape (fig. 4.11).

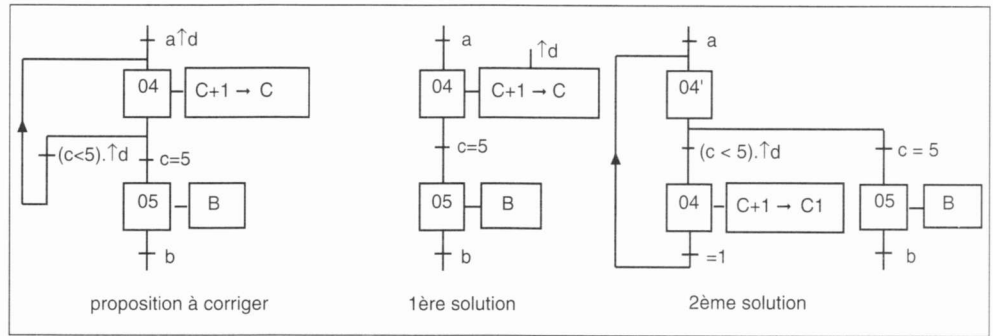


Figure 4.11
Cas d'une boucle
à une seule étape.

Le compteur a pour objet, ici, de dénombrer le nombre d'impulsions fournies par le détecteur d. Dans la proposition à corriger l'étape 4 étant toujours activée puisque, d'après la règle 5, une étape à la fois activée et désactivée reste activée, le compteur reste bloqué à 1 ((C) = 1).

Avec la 1^{re} solution, le comptage s'effectue au niveau de l'étape 4, le front montant de d étant appliqué comme condition à l'action associée.

Dans la 2^e solution, on a introduit une étape supplémentaire 4' de façon à obtenir une boucle à deux étapes. Le front montant de d est associé à la réceptivité (4' → 4).

On préférera la 1^{re} solution pour la simplicité de son graphisme.

5. Applications

Il a été donné au paragraphe précédant un aperçu des erreurs de forme à ne pas commettre. Les exercices suivants sont des exemples de description délicats à traiter, où des erreurs d'écriture sont toujours possibles.

APPLICATION 1

Décrire les évolutions suivantes :

- étape 1 vers étape 5 et étape 7 avec la réceptivité r1,
- étape 1 et 2 vers les étapes 7 et 8 avec la réceptivité r2.

Solution

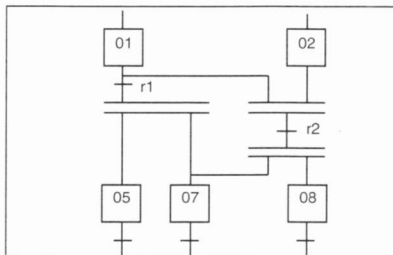


Figure 4.12
Solution de l'application 1.

APPLICATION 2

Décrire les évolutions suivantes :

- étape 1 vers étape 5 ou vers étape 7 ou vers étape 9 avec les réceptivités respectives a, b, e,
- étape 1 et étape 3 vers étape 6 ou vers étape 8 avec les réceptivités respectives c, d,
- étape 4 vers étape 9 avec la réceptivité f.

Solution (fig. 4.13)

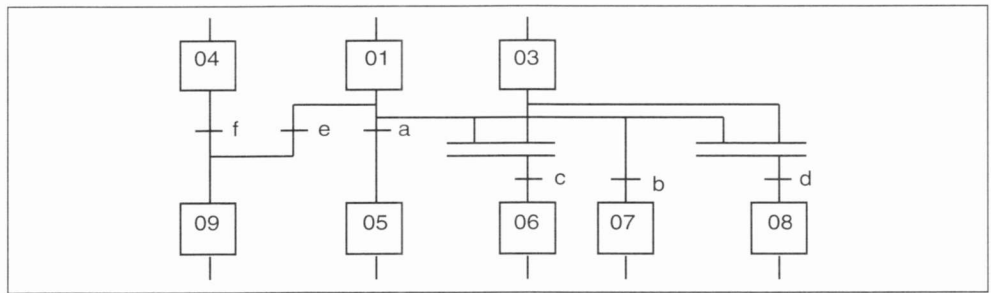


Figure 4.13
Solution de l'application 2.

APPLICATION 3

Le vérin programmeur ci-contre est équipé de capteurs fin de course et de capteurs intermédiaires à galet escamotable (fig. 4.14) : a0, a11, a12, a13, a14

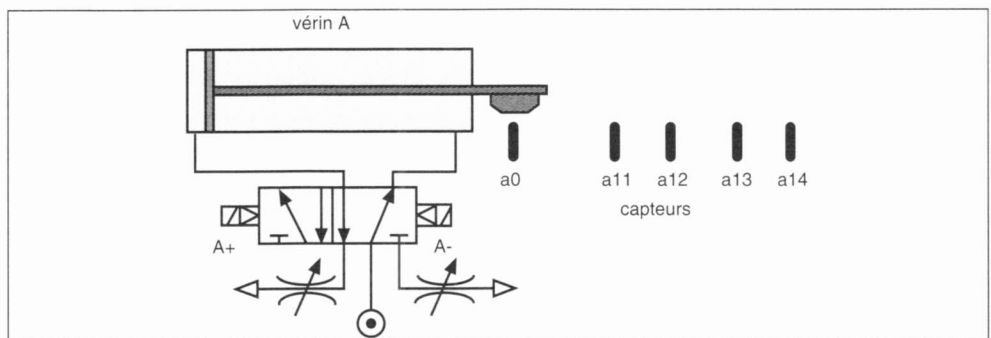


Figure 4.14
Vérin programmeur.

L'étape 10 émet l'ordre A+ : la tige du vérin sort.

Les capteurs intermédiaires déclenchent les séquences suivantes (fig. 4.15) :

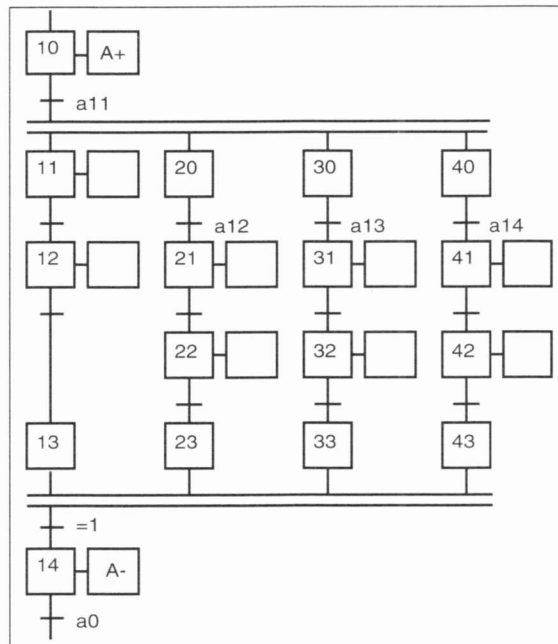


Figure 4.15
GRAFCET
du vérin programmeur.

a11 : séquence 11-12

a12 : séquence 21-22,

a13 : séquence 31-32,

a14 : séquence 41-42.

Lorsque les 4 séquences parallèles sont terminées, la transition 13, 23, 33, 43 → 14 est franchie.

L'ordre A- est émis par l'étape 14.

La tige du vérin rentre.

Les étapes 20, 30 et 40 mémorisent l'évolution 10 → 11.

Les étapes 13, 23, 33, 43 mémorisent la fin des évolutions des séquences parallèles.

Ce sont des étapes d'attente.

V. SIMPLIFICATION DU GRAPHISME D'UN GRAFCET

L'objet de ce chapitre est surtout de montrer que le tracé d'un GRAFCET peut comporter des redondances (transitions, étapes) qu'on pourra éliminer si on sait les découvrir.

Dans un autre ordre d'idée certaines étapes peuvent être fusionnées pour en réduire le nombre, ce qui peut être intéressant dans certains modes de réalisation comme les séquenceurs ou les API bas de gamme au nombre limité de bits internes et de bits d'étape. Si le GRAFCET est réduit à deux étapes, une solution câblée simple reste possible (voir chapitre V.5).

Noter que la simplification du GRAFCET ne doit pas entraîner de modification du processus original de l'automatisme.

1. Recherche des transitions redondantes

Des transitions sont identiques si elles ont mêmes étapes d'entrée et mêmes étapes de sortie et des réceptivités identiques ou non.

Des transitions identiques peuvent être regroupées en une transition unique. A cette nouvelle transition est associée une réceptivité qui est la somme logique, éventuellement simplifiée, de toutes les réceptivités associées aux transitions identiques.

Exemple

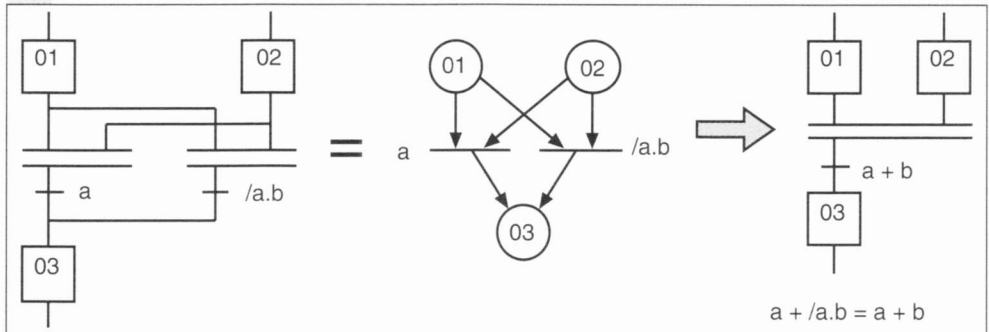


Figure 5.1
Transitions redondantes.

2. Recherche des étapes redondantes

Une étape redondante est par définition une étape superflue. Elle peut donc être supprimée, ses actions associées étant reportées sur une autre étape du GRAFCET.

Le nouveau GRAFCET résultant de cette simplification doit être équivalent à l'original quant au respect du cahier des charges de l'automatisme.

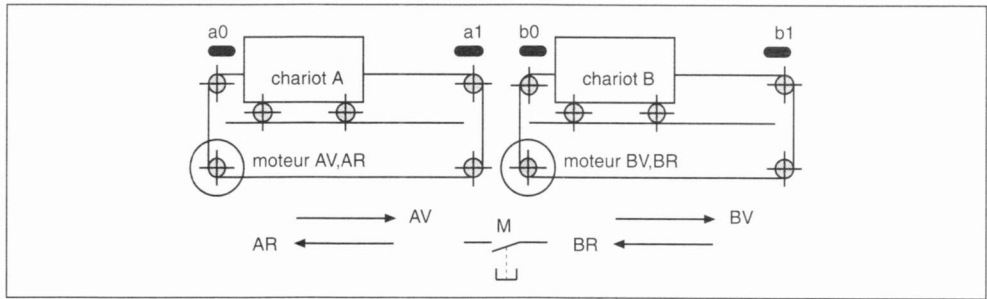
La méthode proposée est basée sur l'analyse des situations du GRAFCET et la recherche des étapes implicites d'autres étapes.



APPLICATION 1 (fig. 5.2)

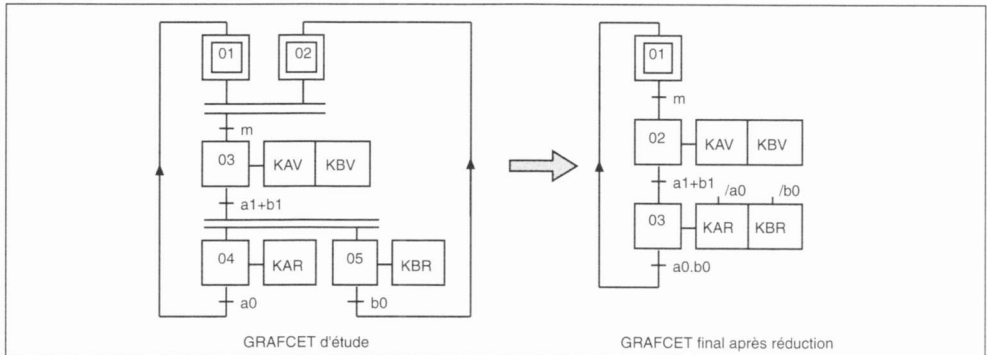
Soit deux chariots démarrant ensemble lorsqu'on appuie sur un bouton poussoir M : on désire que les deux chariots reviennent simultanément à leur point de départ lorsque au moins un des chariots atteint l'extrémité droite de son rail.

Figure 5.2
Application 1.



• Description par GRAFCET et simplification (fig. 5.3)

Figure 5.3
GRAFCET avant
et après simplification.



• Analyse des situations possibles

$S1 = \{1, 2\}$, $S2 = \{3\}$, $S3 = \{4, 5\}$, $S4 = \{4, 2\}$, $S5 = \{5, 1\}$.

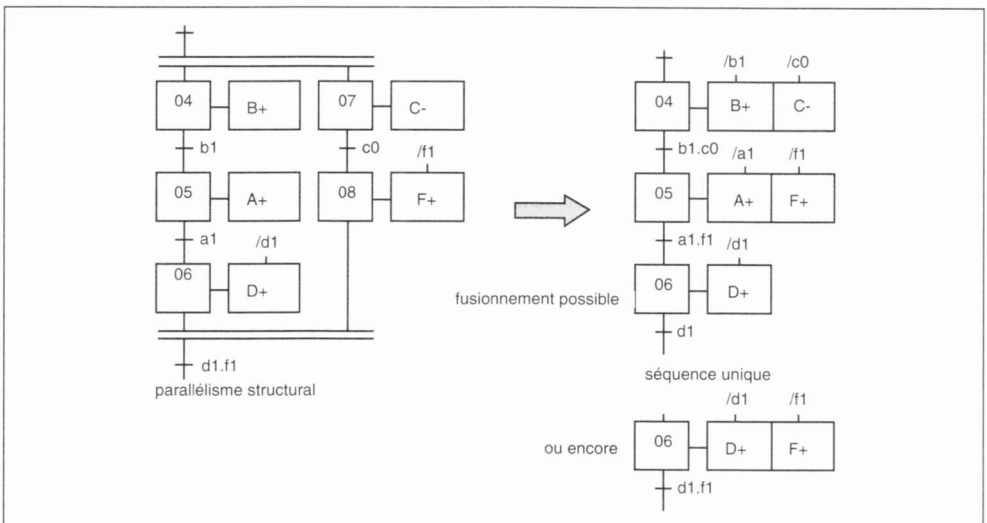
Les situations $S4$ et $S5$ peuvent être exclues sans nuire au fonctionnement à condition de verrouiller KAR par $/a0$ et KBR par $/b0$ pour arrêter chacun des moteurs en fin de course. On aboutit ainsi à un GRAFCET à étape active unique.

Rappel

Une situation est caractérisée par les étapes actives à un instant donné des évolutions du GRAFCET. La notation rappelée ici est $St = \{m, n\}$ où m et n sont les étapes actives et t le repère de la situation.

 APPLICATION 2 (fig. 5.4)

Figure 5.4
Application 2.



Situations possibles

$S1 = \{4,7\}$, $S2 = \{4,8\}$, $S3 = \{5,7\}$, $S4 = \{5,8\}$, $S5 = \{6,7\}$, $S6 = \{6,8\}$.

Deux fusionnements sont envisageables en considérant qu'il faut nécessairement partir de la situation S1 pour aboutir à la situation S6.

Mais on observera qu'il y a modification des évolutions. Ainsi dans la version simplifiée l'action A+ est conditionnée par la fin de l'action C-.

La même remarque peut être faite pour les actions D+ et F+.

Cette simplification n'est donc pas conseillée car elle remet en cause le fonctionnement primitif.

APPLICATION 3 (fig. 5.5)

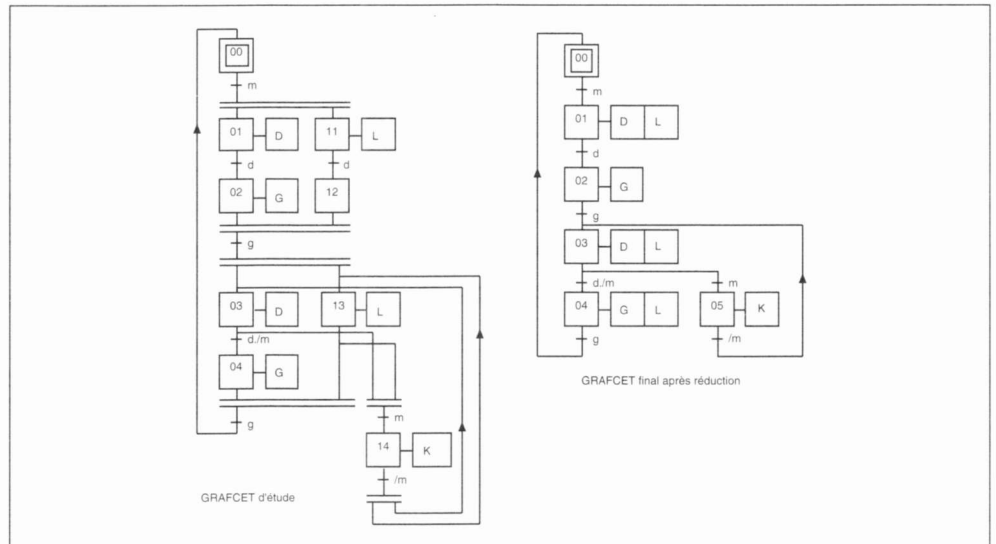


Figure 5.5
Application 3.

L'analyse de toutes les situations possibles du GRAFCET peut être faite, ici, sans trop de difficultés. Pour des GRAFCET complexes, cette recherche risque de devenir vite fastidieuse.

Il faut rechercher toutes les situations possibles :

$S1 = \{0\}$, $S2 = \{1,6\}$, $S3 = \{2,7\}$, $S4 = \{3,8\}$, $S5 = \{4,8\}$, $S6 = \{5\}$.

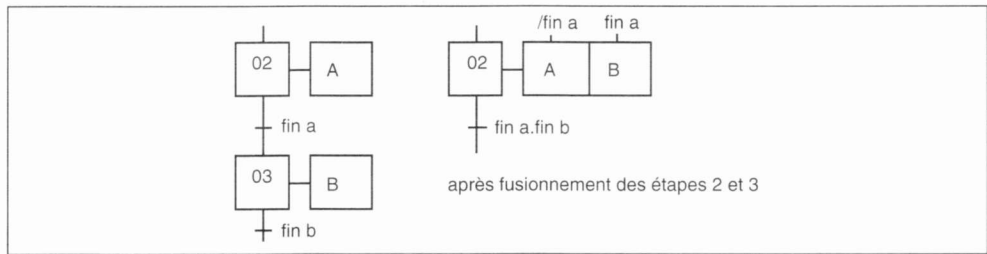
On constate que les étapes 1 et 6, et 2 et 7 sont actives ensemble et jamais isolément ni avec d'autres étapes. On dit que 1 est implicite de 6 et vice versa. Il en est de même pour 2 et 7. Les étapes 6 et 7 peuvent être supprimées. L'étape 8 ne peut être active que si 3 ou 4 sont actives. L'étape 8 est implicite des étapes impliquantes 3 et 4. L'étape 8 peut être éliminée.

Le nouveau GRAFCET ne comporte plus que 6 étapes au lieu de 9 initialement.

3. Recherche des étapes fusionnables

Après avoir éliminé les redondances (étapes et transitions), on peut éventuellement fusionner des étapes successives en introduisant des conditions au niveau des actions. La réceptivité qui disparaît devient une condition au niveau des étapes fusionnées.

Figure 5.6
Fusionnement d'étapes.



Noter qu'avant le fusionnement, l'automatisme est réceptif au signal fin a et au signal fin b (changement de situation) et qu'après le fusionnement il devient sensible aux signaux /fin a et fin a (pas de changement de situation) (fig. 5.6).

4. Conclusion sur la simplification du graphisme du GRAFCET

La réduction conduit :

- par la suppression des transitions redondantes à regrouper les réceptivités,
- par la suppression des étapes redondantes à regrouper les actions, et tendre vers un graphe à étape active unique,
- par fusionnement des étapes à regrouper les actions et à introduire des verrouillages entre les actions et rendre le système moins réceptif et plus sensible.

Remarque

La simplification du graphisme du GRAFCET n'est pas synonyme de meilleure compréhension du fonctionnement de l'automatisme.

D'autre part, en dehors des réalisations par séquenceurs où la réduction du nombre de modules d'étapes justifie ce travail supplémentaire, il s'avère, la plupart du temps, inutile dans le cas des réalisations avec API. Le nombre de bits internes images des étapes est fixe et la plupart du temps surabondant si l'automate programmable a été bien choisi.

5. GRAFCET à deux étapes

Il se trouve des cas où le fusionnement réduit le GRAFCET à deux étapes. Ces deux étapes peuvent alors être matérialisées par l'état inactif et l'état actif d'une seule mémoire et autoriser ainsi une réalisation du type câblé (voir fig. 3.68). Le composant matérialisant la mémoire peut lui-même être du type monostable ou bistable suivant qu'on accepte ou non une désactivation à la coupure d'énergie.

APPLICATION 1

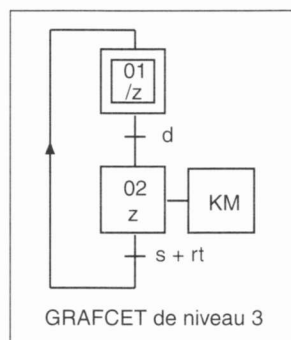


Figure 5.7
GRAFCET à deux étapes.

Commande d'un moteur électrique à un seul sens de marche (fig. 5.7)

relais bistable : $I(Z)=d./s./rt$ $E(Z)=s + rt$

relais monostable : $Z=/s./rt(d + z)$

contacteur du moteur M : $KM=z$

Si on substitue KM à Z on obtient :

$KM = /s./rt(d+km)$ en monostable

Les signaux d'inscription (I) et d'effacement (E) sont identifiés avec les réceptivités respectives.

La mémoire adoptée ici par souci de sécurité est du type à effacement prioritaire.

APPLICATION 2

Filtre pour capteur de niveau à flotteur d'un barrage (fig. 5.8)

Il s'agit de détecter le niveau bas et le niveau haut d'un liquide sans être influencé par les vagues.

Si le capteur f est encore à l'état 1 cinq secondes après être passé de l'état 0 à l'état 1, on considère que le niveau est haut.

De même si le capteur f est encore à l'état 0 deux secondes après être passé de l'état 1 à l'état 0, on considère que le niveau est bas.

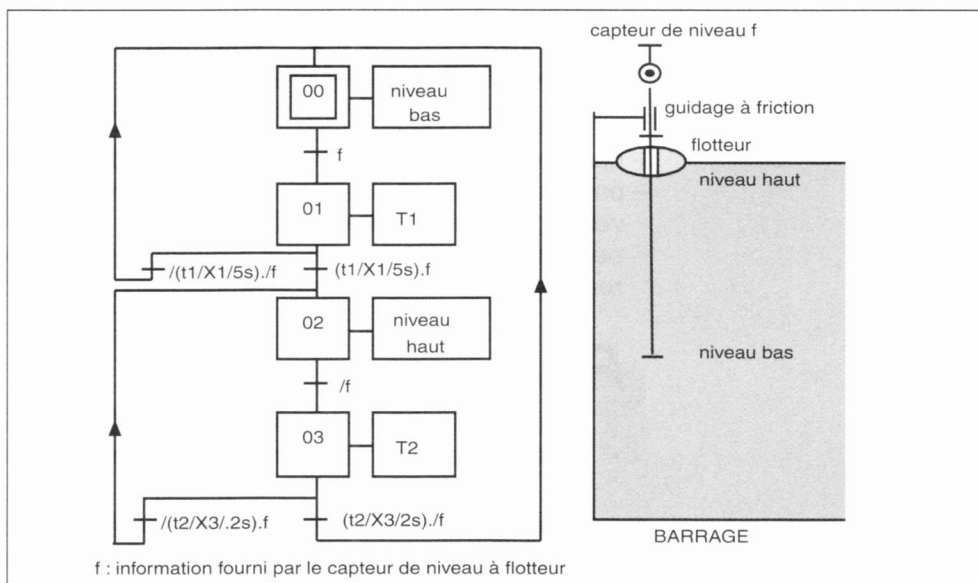


Figure 5.8
Filtre pour capteur
de niveau d'eau.

• Minimisation du nombre des étapes

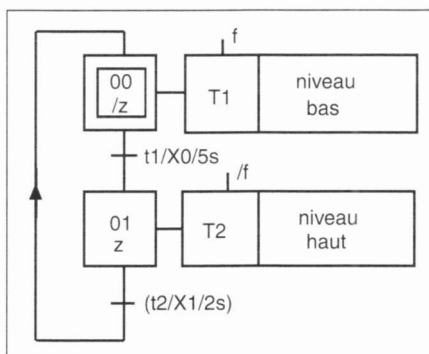


Figure 5.9
GRAFCET réduit du filtre
pour capteur de niveau.

Les étapes 0 et 1 ainsi que les étapes 2 et 3 peuvent être fusionnées. On obtient le GRAFCET à deux étapes suivant (fig. 5.9) :

Les étapes fusionnées (0-1) et (2-3) sont identifiées avec respectivement l'état 0 et l'état 1 de la variable Z (/z et z).

• Relations logiques des sorties

Pour simplifier l'écriture, poser :

$$t1/X0/5s = t1$$

$$t2/X1/2s = t2$$

d'où :

$I(Z) = t1$	$E(Z) = t2$
$Z = /t2(t1 + z)$ (effacement prioritaire)	
niveau haut : z	
niveau bas : /z	
$T1 = /z.f$	$T2 = z.f$

• Réalisation câblée et programmée (fig. 5.10)

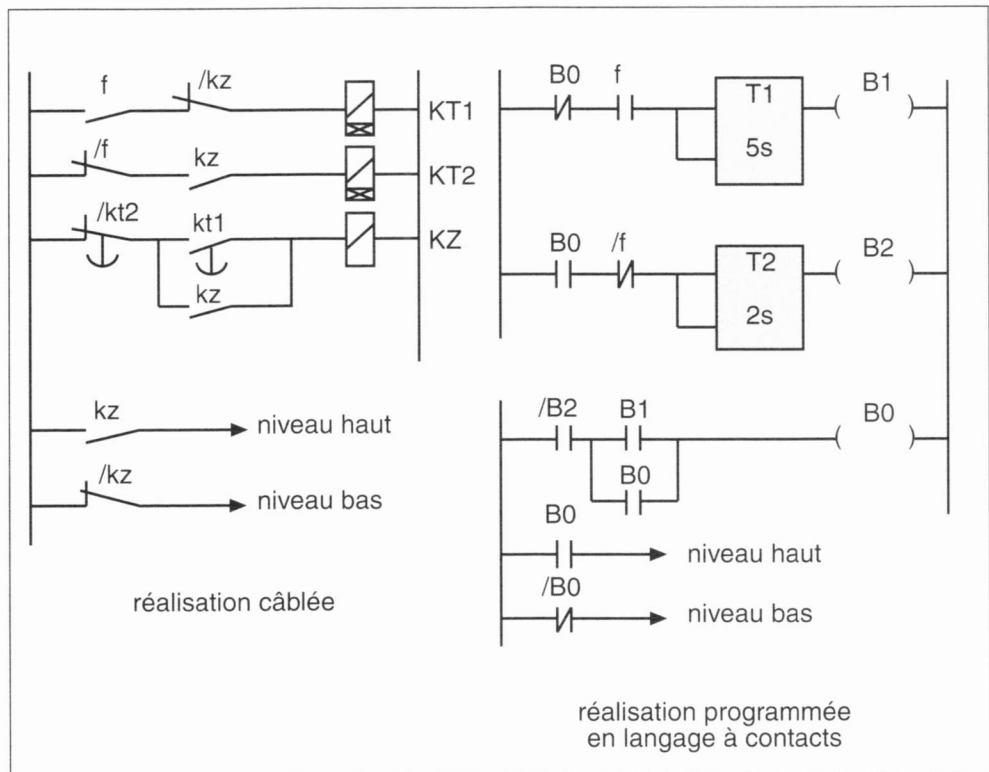


Figure 5.10
Réalisation câblée
et programmée
en langage à contacts.

6. Éclatement d'un GRAFCET complexe en GRAFCET simples

Lorsqu'on prend comme critère d'analyse, la réalisation d'une production en temps minimal ou ce qui revient au même la recherche d'une production à cadence maximale, on débouche sur un GRAFCET de coordination des tâches au graphisme relativement compliqué (*).

La structure du GRAFCET comporte des étapes mémorisant les fins de certaines tâches et autorisant la relance d'autres tâches en temps masqué.

Un exemple typique : dans une centrale à béton, le dosage des agrégats et du ciment se fait, pour le client suivant, pendant que s'opère le malaxage du mortier du client précédent.

Cette préparation en temps masqué permet de traiter un maximum de clients par journée de travail.

Il peut être intéressant de faire éclater le GRAFCET de coordination complexe en plusieurs GRAFCET plus faciles à traiter notamment par un langage GRAFCET graphique implanté dans un API.

L'exemple de la figure 5.11 est destiné à montrer le principe de la décomposition d'un GRAFCET complexe, en procédant :

- à l'isolement des étapes regroupables en GRAFCET simples,
- à la recherche des conditions d'évolutions assurant la coordination des tâches entre-elles,
- à la comparaison des évolutions du GRAFCET original et du GRAFCET éclaté.

Note

(*) Une méthode d'élaboration du GRAFCET de coordination des tâches est proposée dans l'ouvrage intitulé « LE GEMMA » édité dans la même collection.

Exemple

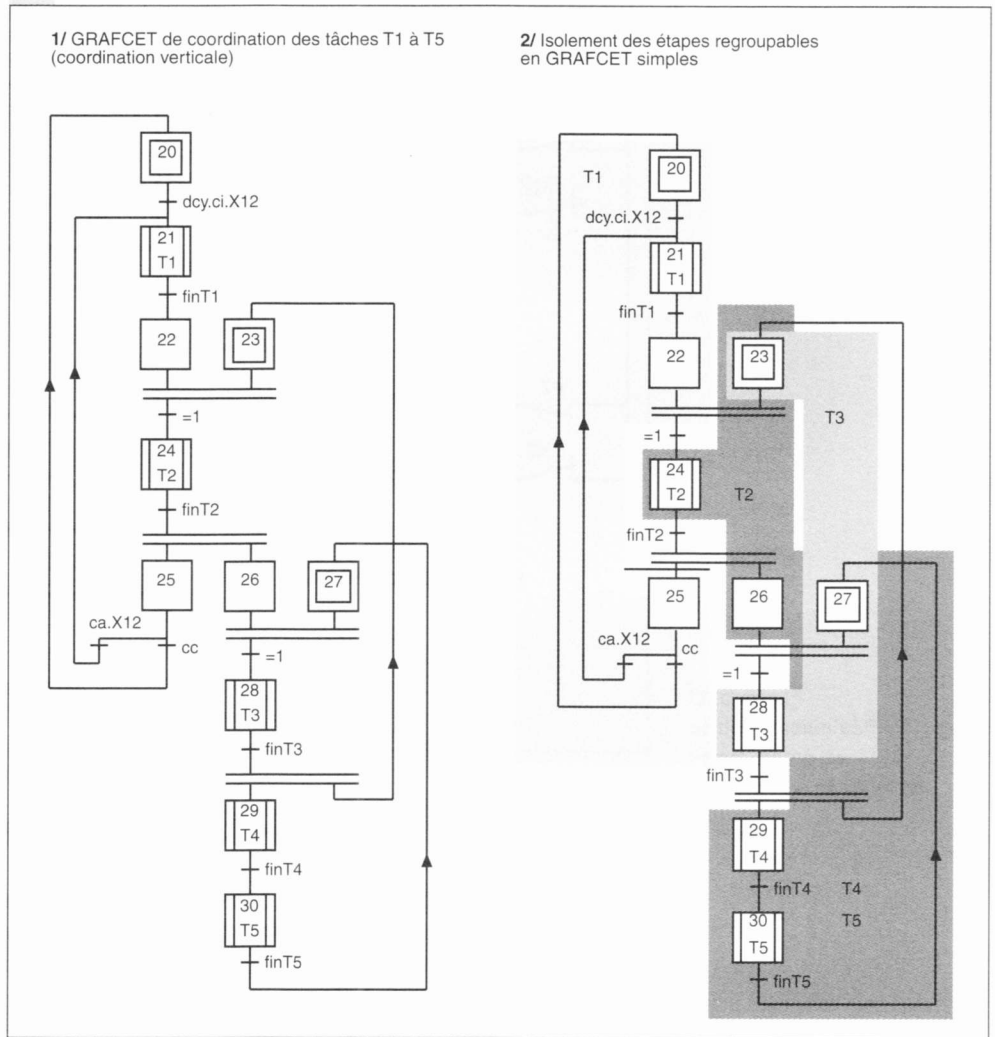


Figure 5.11
Éclatement
d'un GRAFCET complexe
en GRAFCET simples.

• Coordination horizontale des GRAFCET (fig. 5.12)

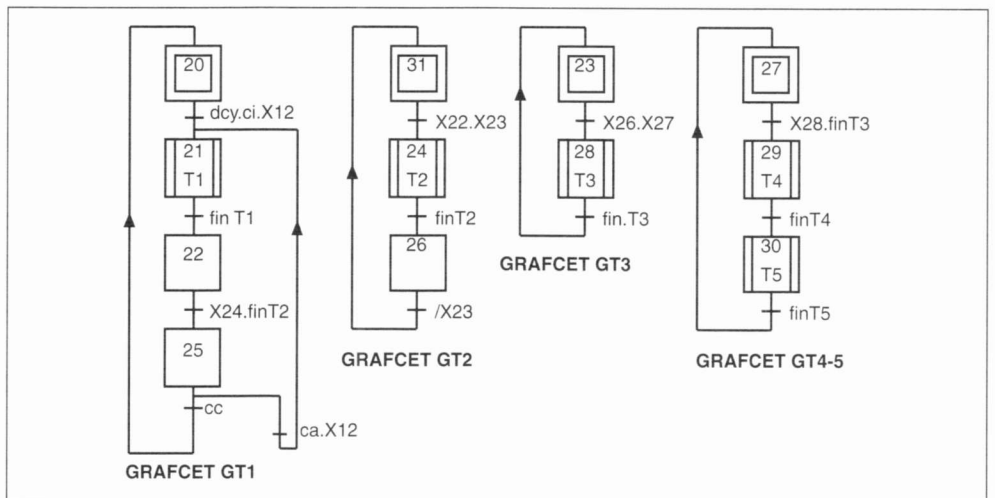


Figure 5.12
Coordination horizontale
des GRAFCET.

• Commentaires

Isolement des étapes gérant chacune des tâches :

- tâche T1 : 20-21-22-25 → GRAFCET de la tâche T1 (GT1)
- tâche T2 : 23-24-26 → GRAFCET de la tâche T2 (GT2)
- tâche T3 : 23-27-28 → GRAFCET de la tâche T3 (GT3)
- tâches T4, T5 : 27-29-30 → GRAFCET des tâches T4 et T5 (GT4-5)

On constate que les étapes 23 et 27 interviennent dans des GRAFCET différents et vont donc poser quelques problèmes. Pour s'en sortir, prendre une étape supplémentaire (étape 31) et procéder comme suit :

- l'étape 23 prise comme étape initiale du GRAFCET GT3 mémorise la fin de la tâche T3,
- l'étape 31 est prise comme étape initiale du GRAFCET GT2 en remplacement de l'étape 23 déjà utilisée,
- l'étape 27 est prise comme étape initiale du GRAFCET GT4-5 et mémorise la fin de la tâche T5,
- l'étape 27 intervient seulement comme condition au lancement de la tâche T3 dans la transition 23 → 28.

Comparaison des évolutions du GRAFCET original et des GRAFCET séparés des tâches T1, T2, T3, T4 et T5 :

- si $X_{20}.dcy.ci.X_{12} = 1$ alors lancement de la tâche T1,
- fin de la tâche T1 mémorisée par l'étape 22,
- si $X_{22}.X_{23} = 1$ alors lancement de la tâche T2,
- fin de la tâche T2 mémorisée par les étapes 25 et 26,
- relance possible de la tâche T1 si $X_{25}.ca.X_{12} = 1$,
- si $X_{26}.X_{27} = 1$ alors lancement de la tâche T3,
- fin de tâche T3 mémorisée par l'étape 23 et lancement de la tâche T4,
- si $X_{29}.finT4 = 1$ alors lancement tâche T5,
- fin de la tâche T5 mémorisée par l'étape 27,
- si $X_{26}.X_{27} = 1$ alors relance de la tâche T3.

On peut donc considérer que la coordination des tâches est identique malgré une structuration différente des GRAFCET.

L'objectif recherché est donc atteint.

VI. MATÉRIALISATION DES CONCEPTS DE BASE DU GRAFCET

1. Étape

Rappel

Une étape est soit active soit inactive. Elle est activée par le franchissement de la (ou des) transition(s) amont. Elle est désactivée par le franchissement de la (ou d'une des) transition(s) aval.

L'opérateur logique, image de l'étape, doit posséder deux états stables (état 0 et état 1) afin de mémoriser l'activation et la désactivation de l'étape.

Le profil défini ci-dessus est celui d'un opérateur réalisant la FONCTION MEMOIRE (fig. 6.1).

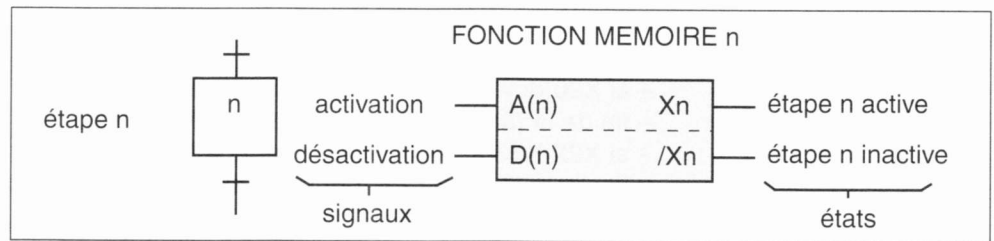


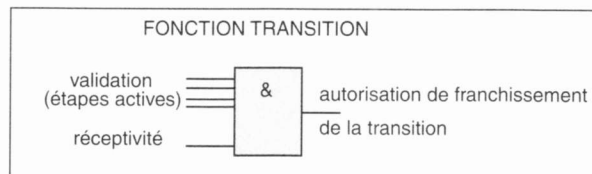
Figure 6.1
Fonction mémoire matérialisant une étape.

2. Transition et réceptivité associée

Rappel

- Une transition est soit validée soit non validée,
- elle est validée lorsque la ou les étapes immédiatement précédentes sont actives,
- elle est franchie et obligatoirement franchie si elle est validée et si la réceptivité associée est vraie.

Cette condition d'évolution est appelée **fonction transition** (fig. 6.2).



Elle se note : $ft(m \rightarrow n)$
où m est l'étape amont et n l'étape aval.

Figure 6.2
Fonction transition.

Dans le cas de plusieurs étapes amont et aval, elle se note :

$$ft(m, n \rightarrow p, q)$$

où m et n sont les étapes amont et p et q les étapes aval.

De même les réceptivités associées se notent :

$$r(m \rightarrow n) \text{ et } r(m, n \rightarrow p, q)$$

Les fonctions transitions ont pour expressions logiques :

$$ft(m \rightarrow n) = X_m \cdot r(m \rightarrow n)$$

$$ft(mn \rightarrow p, q) = X_m \cdot X_n \cdot r(m, n \rightarrow p, q)$$

3. Matérialisation des règles d'évolution

• Règle 1 concernant la situation initiale

Il est souhaitable que la réalisation, lors de la mise en énergie de la PC, permette l'activation automatique des mémoires matérialisant les étapes initiales, celles-ci concrétisant par définition la situation initiale. Mais il faut noter que la règle 1 n'en fait aucune obligation.

On pourra donc envisager dans certaines réalisations, une initialisation manuelle par action directe sur les composants (modules d'étapes des séquenceurs pneumatiques, par exemple) ou indirecte à partir du pupitre de commande par action sur un bouton poussoir INIT.

Le concept de forçage de situation peut être exploité pour forcer la situation initiale de la PC à partir d'un ordre émis par une étape accessible par une transition source (fig. 6.3).

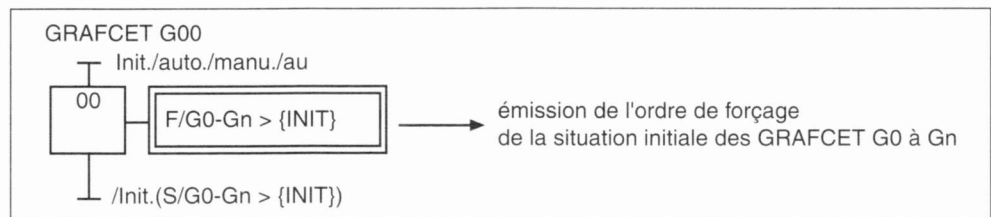


Figure 6.3
GRAFCET d'initialisation.

• Règles 2 et 3 concernant les évolutions du GRAFCET

La synchronisation, c'est-à-dire la simultanéité de l'activation de l'étape suivante et de la désactivation de l'étape précédente pose des problèmes de réalisation surtout avec les séquenceurs du fait du temps de réponse non négligeable des composants des modules d'étapes.

• Principe de l'évolution synchrone dans le cas des règles 2 et 3 (fig. 6.4)

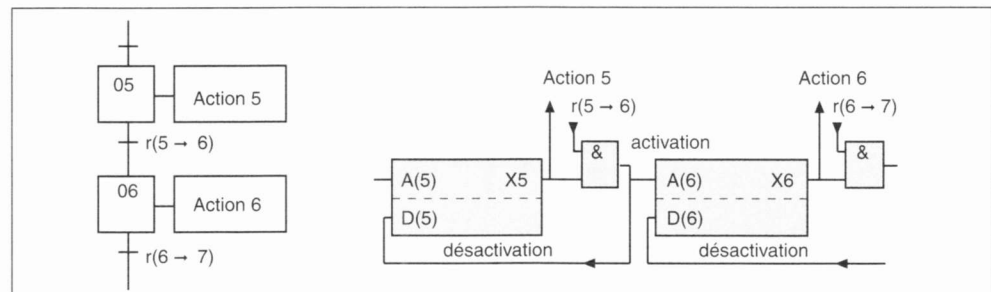


Figure 6.4
Principe de l'évolution synchrone dans le cas des règles 2 et 3.

• Comment obtenir une évolution synchrone ?

En ce qui concerne les API, la solution passe :

- soit par l'implantation de langages GRAFCET dont les algorithmes et le traitement interne assurent le respect des règles d'évolution du GRAFCET normalisé,
- soit par l'implantation de structures adaptées aux divers langages disponibles autres que le GRAFCET (voir à ce sujet, le chapitre VII consacré à l'implantation du GRAFCET dans les API).

Pour les séquenceurs, on fait une entorse aux règles d'évolution du GRAFCET en adoptant une évolution dite **évolution asynchrone** :

- la fonction transition amont active l'étape suivante,
- l'information étape suivante désactive l'étape précédente.

• Principe de l'évolution asynchrone dans le cas des règles 2 et 3 (fig 6.5)

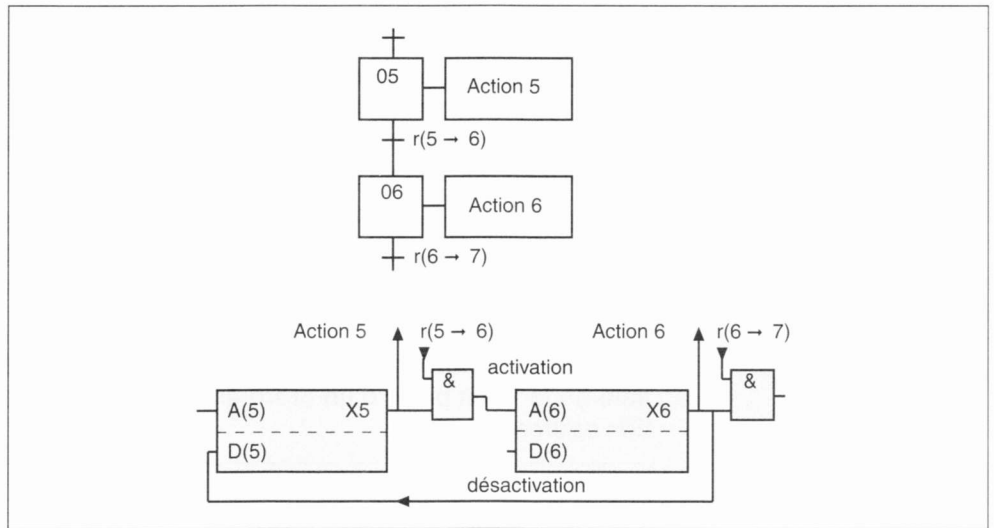
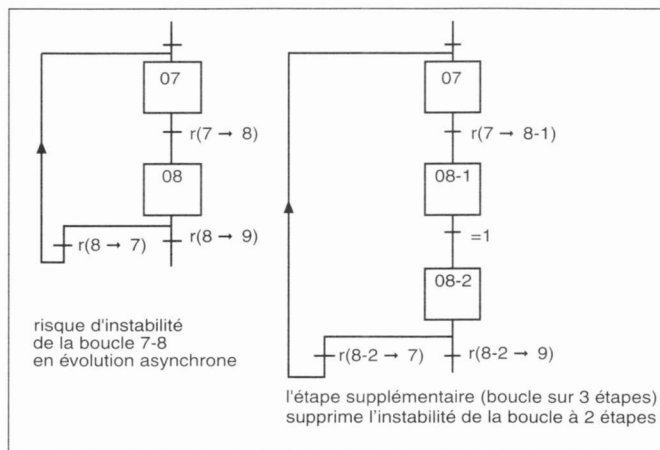


Figure 6.5
Principe de l'évolution
asynchrone dans le cas
des règles 2 et 3.

Du fait du non-respect des règles 2 et 3, certaines configurations du GRAFCET ne sont pas réalisables, notamment la reprise de séquence sur deux étapes. Il faut rajouter une étape.

• Cas de la boucle à deux étapes

L'évolution asynchrone d'une boucle à deux étapes (ici boucle 7-8-7) est généralement instable. Ceci est dû à la présence du même signal dans les expressions de l'activation et de la désactivation de chaque étape :



$$A(7) = \dots + X8.(r(8 \rightarrow 7))$$

$$D(7) = X8$$

$$A(8) = X7.(r(7 \rightarrow 8))$$

$$D(8) = X7 + \dots$$

L'introduction de l'étape 8-1 et d'une réceptivité toujours vraie (=1) règle le problème de l'instabilité de la boucle (fig. 6.6).

Figure 6.6
Boucle à deux étapes.

Noter qu'aucune action ne doit être associée à l'étape 8-1 car la réceptivité = 1 de la transition 8-1 → 8-2 rend la situation $S = \{8-1\}$ instable (voir chapitre III.9.1, exemple 8).

• Règle 4 concernant les évolutions simultanées

On se trouve confronté au même problème que pour les règles 2 et 3 en ce qui concerne les séquenceurs. Il se résout de la même manière :

- évolution asynchrone pour les réalisations type séquenceur,
- évolution synchrone en utilisant une structure appropriée pour les API ne possédant pas de langage GRAFCET.

• Principe de l'évolution synchrone dans le cas des règles 3 et 4 (fig. 6.7)

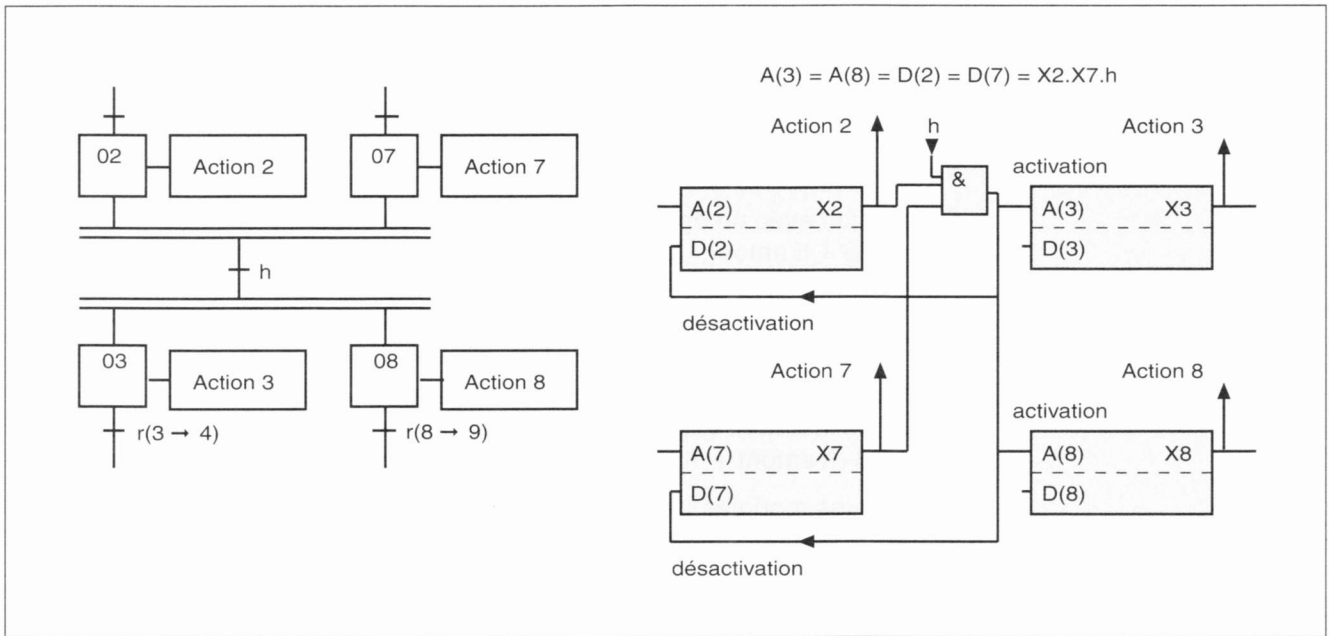


Figure 6.7
Principe de l'évolution synchrone dans le cas des règles 3 et 4.

• Principe de l'évolution asynchrone dans le cas des règles 3 et 4 (fig. 6.8)

Pour obtenir une évolution asynchrone, il suffit de prendre la condition $X3.X8$ pour désactiver les mémoires des étapes 2 et 7 au lieu de $X2.X7.h$.

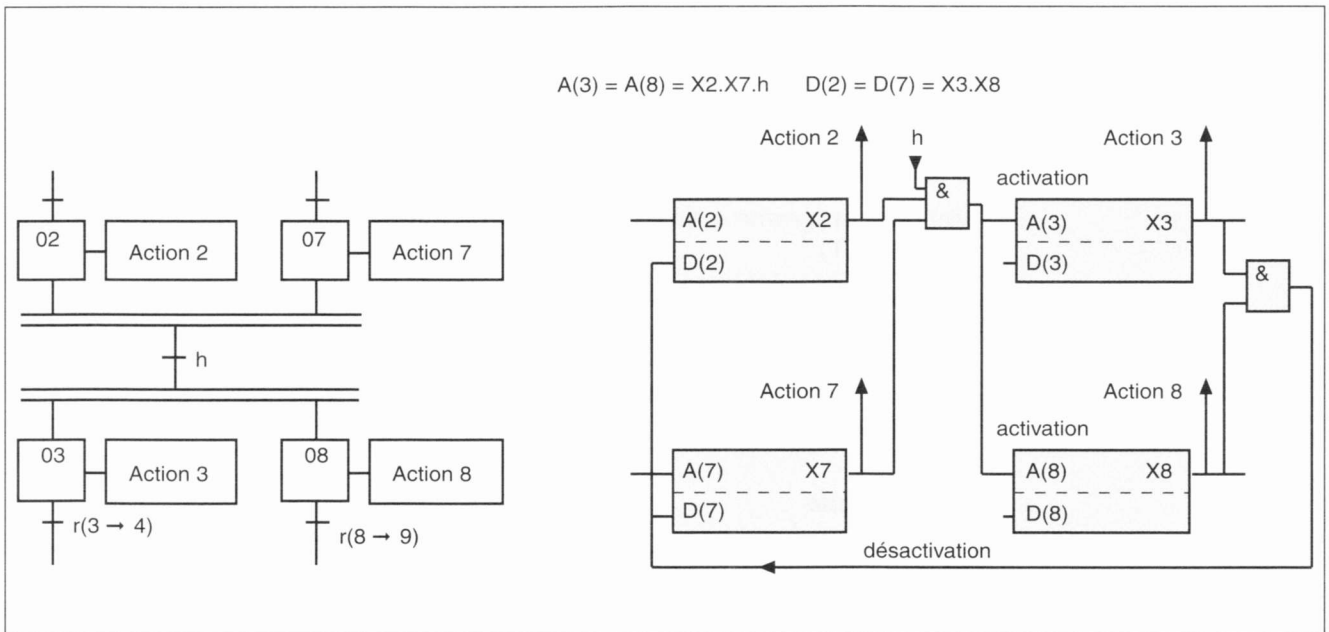


Figure 6.8
Principe de l'évolution asynchrone dans le cas des règles 3 et 4.

• Règle 5 concernant l'activation et la désactivation simultanée d'une même étape

Dans ce cas particulier, l'activation doit être prioritaire sur la désactivation. Bien que cette règle ne soit strictement applicable que dans de très rares cas, on généralise son application.

La fonction mémoire image de l'étape doit être une mémoire à inscription prioritaire.

APPLICATION

- Réalisation avec un opérateur bistable
 $A(n) = ft \text{ amont}$
 $D(n) = /ft \text{ amont.ft aval}$
- Réalisation avec un opérateur monostable
 $Xn = ft \text{ amont} + /ft \text{ aval. } Xn$
- Réalisation avec un opérateur bistable avec forçage
 $A(n) = FS + ft \text{ amont}$ | priorité au forçage à 1 (étapes à activer)
 $D(n) = /FS./ft \text{ amont.ft aval}$ |
 $A(p) = /FS.ft \text{ amont}$ | priorité au forçage à 0 (étapes à désactiver)
 $D(p) = FS + /ft \text{ amont.ft aval}$ |
- Réalisation avec un opérateur monostable avec forçage
 $Xn = FS + ft \text{ amont} + /ft \text{ aval. } Xn$ | priorité au forçage à 1 (étapes à activer)
 $Xp = /FS (ft \text{ amont} + /ft \text{ aval. } Xp)$ | priorité au forçage à 0 (étapes à désactiver)

Noter que les modules d'étapes des séquenceurs sont pour la plupart réalisés avec des opérateurs mémoire à effacement prioritaire.

Mais cela n'est guère gênant, les problèmes traités avec ce genre de composants étant toujours relativement simples.

4. Réalisation des fonctions mémoire et transition

L'opérateur mémoire qui matérialise l'étape du GRAFCET peut être créé :

- à partir de composants commercialement disponibles (cas des séquenceurs) :
 - en technologie électrique : relais électromagnétique bistable.
 - en technologie pneumatique : distributeur 2 positions à double pilotage.
- ou bien programmé en exploitant un bit interne (cas des API sans langage GRAFCET).

La fonction transition est obtenue :

- soit par câblage (montage en série des contacts ou utilisation d'un opérateur logique ET),
- soit par programmation de la fonction elle-même (calcul d'un bit interne, par exemple).

Remarque

Les langages GRAFCET graphiques implantés dans certains API libèrent en partie l'automaticien de cette phase de synthèse :

- le GRAFCET est reproduit graphiquement sur l'écran vidéo de la console ou du moniteur assorti d'un contrôle des erreurs de syntaxe,
- les réceptivités et les actions sont programmées en langage littéral ou en langage graphique à contacts.

État actuel de la technologie

Les réalisations à base de composants discrets ou modulaires sont délaissées au profit des réalisations programmées.

La fiabilité des automates programmables industriels (API) et des micro-ordinateurs industriels n'étant plus à mettre en doute, le présent et l'avenir appartiennent aux solutions programmées (*).

Des langages (ou logiciels) orientés « automatisme » sont proposés aux concepteurs parmi lesquels on trouve :

- des langages GRAFCET graphiques,
- des langages graphiques à contacts avec bits monostables ou bistables,
- des langages calcul sur mots,
- des langages avec registres,
- des langages littéraux opérant sur bits et sur mots avec ou sans instructions GRAFCET,
- etc.

Dans le chapitre suivant consacré aux méthodes d'implantation du GRAFCET dans les API, il va être montré comment utiliser au mieux ces langages et comment matérialiser les étapes par l'intermédiaire de mémoires obtenues à partir de bits, de mots ou de registres, entre-autres.

Les modules pneumatiques et les séquenceurs pneumatiques sont toujours commercialisés par quelques sociétés.

Ces composants ont des domaines d'applications très spécifiques :

- ambiances explosives,
- ambiances poussiéreuses,
- ambiances très humides, par exemple.

Note

(*) Voir comment améliorer la sûreté des API par redondance en consultant l'ouvrage intitulé « LE GEMMA » édité dans la même collection.

5. Tableau récapitulatif des mémoires matérialisant les étapes dans les divers types de structures du GRAFCET

(fig. 6.9)

Notations utilisées :

A(n) : activation étape n

D(n) : désactivation étape n

r(m → n) : réceptivité liée à la transition étape m → n

Xn : mémoire matérialisant l'étape n.

Remarque

Il est utile de rappeler que l'évolution asynchrone ne respecte pas les règles d'évolution du GRAFCET normalisé. Son usage peut être toléré pour matérialiser des GRAFCET à étape active unique.

Les séquenceurs, à cause du temps de réponse des mémoires des modules d'étapes, sont à évolution asynchrone (voir chapitre XI.6).

liaisons types	structure du GRAFCÉT	évolution synchrone	évolution asynchrone
branche		$A(2) = X1.(r(1 \rightarrow 2))$ $D(2) = X2.(r(2 \rightarrow 3))$	$A(2) = X1.(r(1 \rightarrow 2))$ $D(2) = X3$ $X2 = X1.(r(1 \rightarrow 2)) + /X3.X2$ $X2 = X1.(r(1$
entrée de sélection de séquences exclusives ou non		$A(2) = X1.(r(1 \rightarrow 2))$ $D(2) = X2.(r(2 \rightarrow 4))$ $A(3) = X1.(r(1 \rightarrow 3))$ $D(3) = X3.(r(3 \rightarrow 5))$	$A(2) = X1.(r(1 \rightarrow 2))$ $D(2) = X4$ $A(3) = X1.(r(1 \rightarrow 3))$ $D(3) = X5$ $X2 = X1.(r(1 \rightarrow 2)) + /X4.X2$ $X3 = X1.(r(1 \rightarrow 3)) + /X5.X3$
sortie de sélection de séquences		$A(3) = X1.(r(1 \rightarrow 3)) + X2.(r(2 \rightarrow 3))$ $D(3) = X3.(r(3 \rightarrow 4))$	$A(3) = X1.(r(1 \rightarrow 3)) + X2.r(2 \rightarrow 3))$ $D(3) = X4$ $X3 = X1.(r(1 \rightarrow 3)) + X2.(r(2 \rightarrow 3)) + /X4.X3$
entrée de séquences simultanées		$A(2) = A(3) = X1.(r(1 \rightarrow 2,3))$ $D(2) = X2.(r(2 \rightarrow 4))$ $D(3) = X3.(r(3 \rightarrow 5))$	$A(2) = A(3) = X1.(r(1 \rightarrow 2,3))$ $D(2) = X4$ $D(3) = X5$ $X2 = X1.(r(1 \rightarrow 2,3)) + /X4.X2$ $X3 = X1.(r(1 \rightarrow 2,3)) + /X5.X3$
sortie de séquences simultanées		$A(3) = X1.X2.(r(1,2 \rightarrow 3))$ $D(3) = X3.(r(3 \rightarrow 4))$	$A(3) = X1.X2.(r(1,2 \rightarrow 3))$ $D(3) = X4$ $X3 = X1.X2.(r(1,2 \rightarrow 3)) + /X4.X3$
sortie de séquences simultanées avec réceptivité toujours vraie		$A(3) = X1.X2$ $D(1) = D(2) = X1.X2$ $D(3) = X3.(r(3 \rightarrow 4))$	$A(3) = X1.X2$ $D(1) = D(2) = X3$ $X3 = X1.X2 + /X4.X3$

Figure 6.9 : Tableau récapitulatif des mémoires d'étapes et de leurs signaux de commande.

VII. STRUCTURES D'IMPLANTATION DU GRAFCET DANS LES API

1. Quel est le problème ?

Le GRAFCET a été conçu à l'origine par ses créateurs essentiellement comme un outil de description des automatismes.

Il est donc normal qu'il soit utilisé comme outil principal pour conduire l'analyse d'un processus.

Reste le problème de la synthèse de l'automatisme, autrement dit de sa réalisation.

Si l'on dispose d'un langage GRAFCET graphique implanté dans l'automate programmable industriel (API) et respectant les règles d'évolution définies par la norme, la synthèse posera peu de problèmes. Il suffira, pour ce qui est du GRAFCET de niveau 3, de reproduire celui-ci sur l'écran de la console de l'API ou sur l'écran du moniteur du micro-ordinateur. Il faudra malgré tout tenir compte des spécificités du langage graphique. Les réceptivités et les sorties seront ensuite programmées en langage littéral ou en langage à contacts.

Si l'on ne dispose pas d'un langage GRAFCET littéral ou graphique, il faut utiliser une structure d'implantation adaptée au langage disponible (voir note en bas de page).

L'avantage d'une structure d'implantation est qu'elle se répète, dans sa forme, identique à elle-même pour toutes les étapes du GRAFCET. Le travail du programmeur est ainsi grandement facilité.

La structure respectant les 5 règles d'évolution, le programme tournera sans aléa. Ainsi le fonctionnement de l'installation correspondra au cahier des charges de l'automatisme si, évidemment, l'étude préparatoire (analyse) a été correctement conduite.

Organisation de l'étude des structures

Seront étudiées et exploitées dans l'ordre :

1. la structure basée sur la fonction mémoire,
2. la structure informatique ou algorithmique,
3. la structure exploitant les registres pas à pas,
4. la structure exploitant le langage calcul sur mots,
5. la structure exploitant le langage GRAFCET littéral,
6. la structure exploitant le langage GRAFCET graphique.

Le poste de dosage d'un produit schématisé sur la figure 7.1 page suivante a été choisi comme support. Son GRAFCET est intéressant. D'une part il est simple à comprendre et d'autre part il comporte une boucle à deux étapes.

Ce genre de boucle est le plus sujet à caution. Lorsque le programme d'une telle boucle tourne correctement, on obtient implicitement la preuve que les règles 2, 3 et 4 sont respectées, c'est-à-dire que l'évolution est bien du type synchrone.

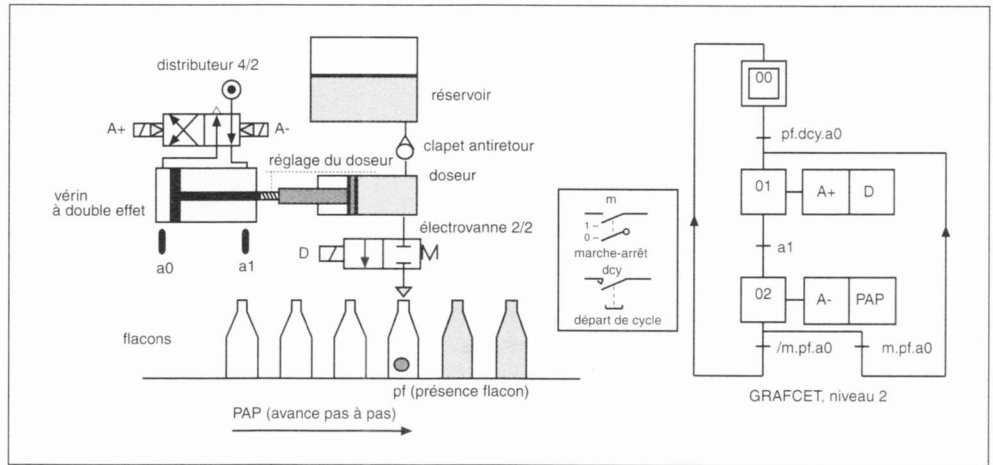
Note

Qu'entend-on par structure d'implantation du GRAFCET ?

Une structure d'implantation du GRAFCET regroupe la suite ordonnée des opérations permettant de programmer l'activation et la désactivation d'une étape ou autrement dit de programmer le franchissement d'une transition.

APPLICATION : dosage de produit liquide (fig. 7.1)

Figure 7.1
Poste de dosage
d'un produit liquide.
En simulation, ce poste
fonctionne correctement
dans la mesure où la
variable pf prend l'état 0
avant que la variable a0
passe à l'état 1 (transition
2 → 0 ou 1). Dans la pra-
tique, il faudrait utiliser le
front montant de pf (↑ pf),
ce qui n'a pas été fait ici
pour simplifier la solution.



2. Structure basée sur la fonction mémoire

L'étape est matérialisée par une mémoire réalisée avec un bit interne monostable ou bistable, au choix. La structure repose sur le principe suivant :

pour respecter les règles d'évolution 2, 3 et 4, il faut que l'expression logique de l'activation de l'étape suivante soit la même que celle de la désactivation de l'étape précédente.

Cette expression logique correspond à la fonction transition (en abrégé ft).

Recherche des fonctions transition

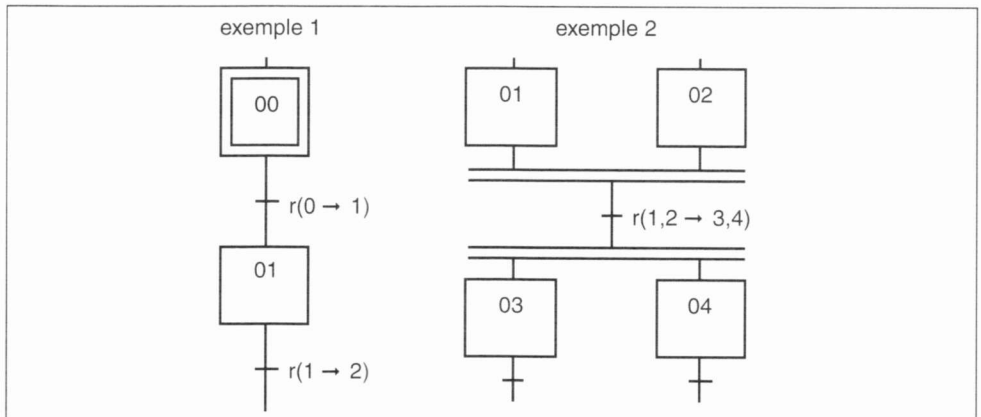


Figure 7.2
Exemples pour la recherche
des fonctions transitions.

Exemple 1

fonction transition : $ft(0 \rightarrow 1) = X0.r(0 \rightarrow 1)$
activation de l'étape 1 : $A(1) = X0.r(0 \rightarrow 1)$
désactivation de l'étape 0 : $D(0) = X0.r(0 \rightarrow 1)$

Exemple 2

fonction transition : $ft(1,2 \rightarrow 3,4) = X1.X2.r(1,2 \rightarrow 3,4)$
activation étapes 3 et 4 : $A(3,4) = X1.X2.r(1,2 \rightarrow 3,4)$
désactivation étapes 1 et 2 : $D(1,2) = X1.X2.r(1,2 \rightarrow 3,4)$

Pour respecter la règle d'évolution 5, il faut opter pour une mémoire à inscription prioritaire.

Pour initialiser la mémoire-image de l'étape initiale (règle 1), il faut mettre le bit correspondant à l'état 1 dès le démarrage du programme.

La boucle à deux étapes programmée sans précaution risque d'être instable sur certains automates.

Dans ce cas, il suffit de calculer préalablement la fonction transition. On ajoute ainsi un cycle supplémentaire dans l'exécution du programme. L'exécution se déroule ainsi :

- calcul de la fonction transition et mise en bit interne B_i dans le cycle C ,
- exploitation du bit interne, image de la fonction transition, dans le cycle $C+1$ pour activer l'étape suivante et désactiver l'étape précédente,
- la transition étant franchie, la fonction transition est annulée dans le cycle $C+2$.

L'évolution synchrone est obtenue dans le cycle $C+1$.

La figure 7.3 montre le principe de l'évolution synchrone en fonction des cycles d'exécution du programme. La fonction transition est mémorisée durant le cycle C (mémoire fictive) et c'est au cycle $C+1$ qu'elle assure **simultanément**, par l'intermédiaire du bit B_i , l'activation de l'étape 6 et la désactivation de l'étape 5.

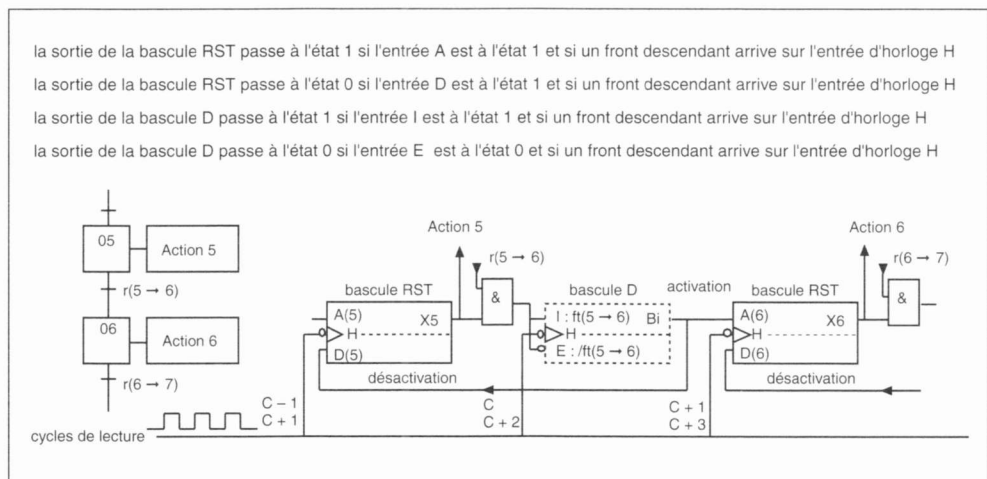


Figure 7.3
Evolution synchrone
par calcul de la fonction
transition.

Rappel

La fonction transition vérifie que la transition est validée et que la réceptivité associée est vraie. Son expression est de la forme :

$$ft(n \rightarrow m) = X_n.r(n \rightarrow m).$$

Le calcul de la fonction transition se justifie essentiellement dans le cas des boucles à deux étapes. Malgré tout, dans les applications suivantes les fonctions transitions ont été systématiquement calculées afin de conserver le même type de structure dans tout le programme (voir note).

Note

CADEPA, un logiciel de PGOA (Programmation du GRAFCET Assistée par Ordinateur) utilise ce type de structure et calcule toutes les fonctions transitions lors de la traduction en langage constructeur. Pour économiser les bits internes, CADEPA peut aussi, au choix, travailler sur des bits de mots (voir chapitre VIII-1, application 1).

Structure liée à l'étape n, matérialisée par le bit Bn de la mémoire n
(fig. 7.4)

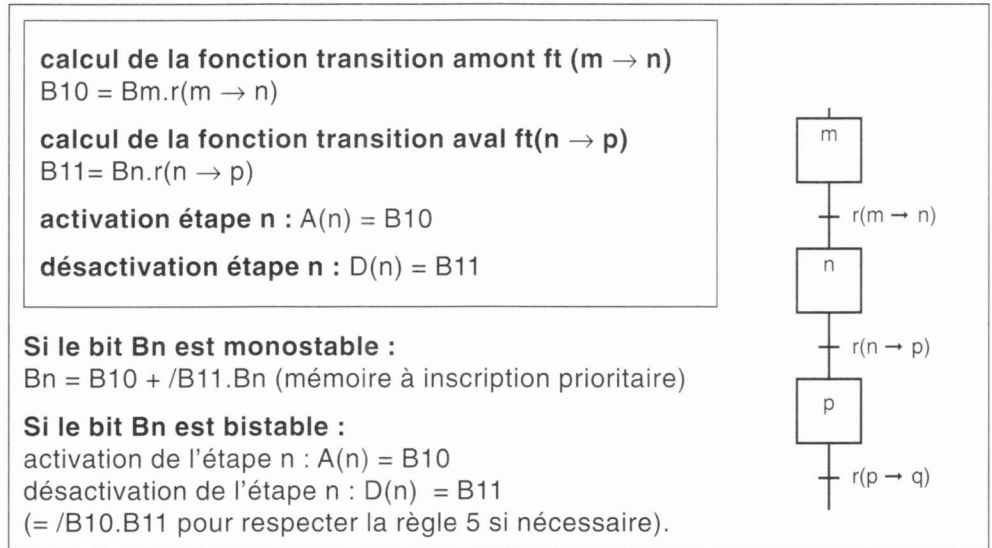


Figure 7.4
Branche de GRAFCET.

2.1. Implantation en monostable

Application au GRAFCET support

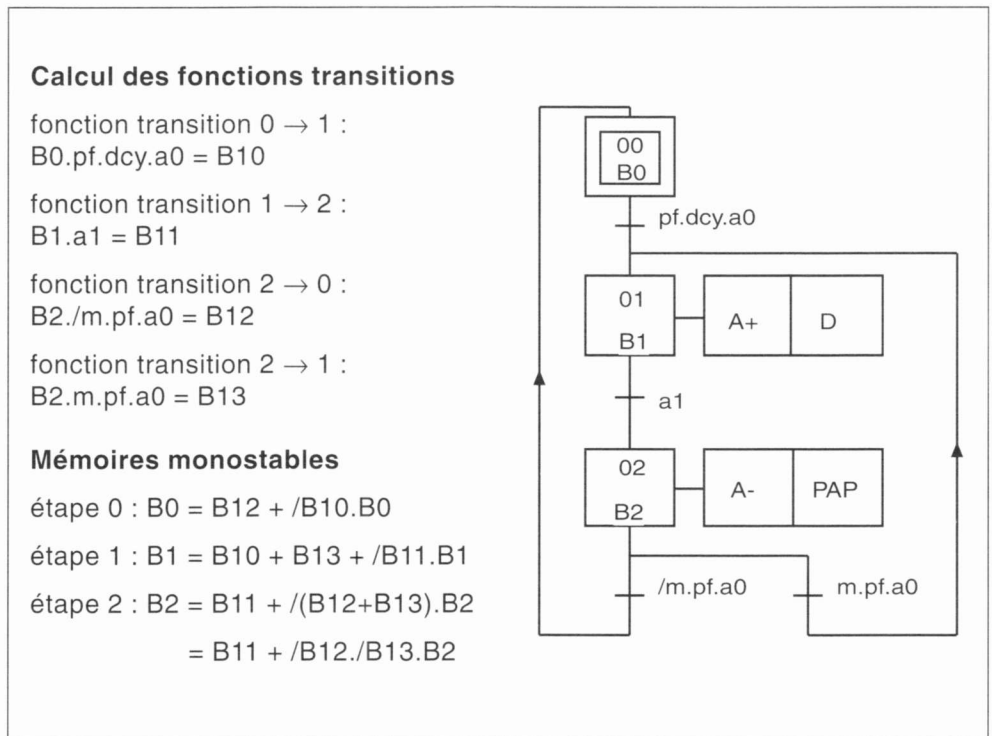


Figure 7.5
GRAFCET support.

Cette structure s'implante dans tous les API. Elle peut donc être considérée comme une structure universelle d'implantation du GRAFCET. Elle peut être traitée en langage booléen, en langage assembleur ou littéral, en langage à contacts ou en langage logigramme.

Exemple de mémoire d'étape à programmer : $B_n = B_{10} + /B_{11}.B_n$

langage booléen

$B_n = B_{10} + /B_{11}.B_n$

langage assembleur

LDAC B11

AND Bn

OR B10

STO Bn

langage à contacts et langage logigramme

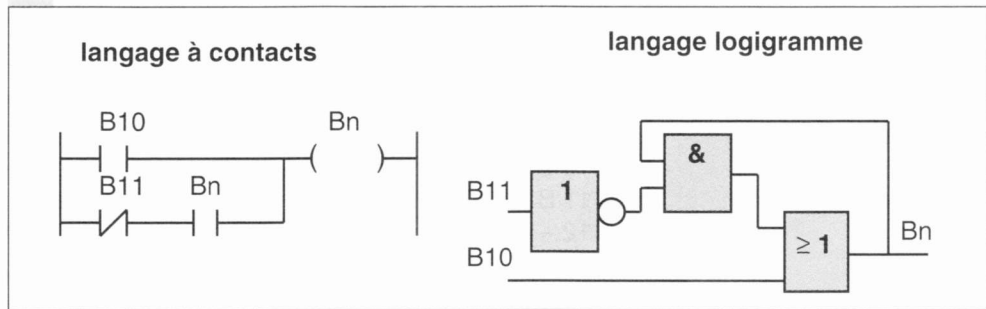


Figure 7.6
Structure programmée
en langage à contacts
et en logigramme.



APPLICATION 1 : exploitation des bits monostables dans les SMC (APRIL)

Le programme peut être entré au clavier en langage booléen puis lu en langage à contacts ou entré directement dans ce dernier langage.

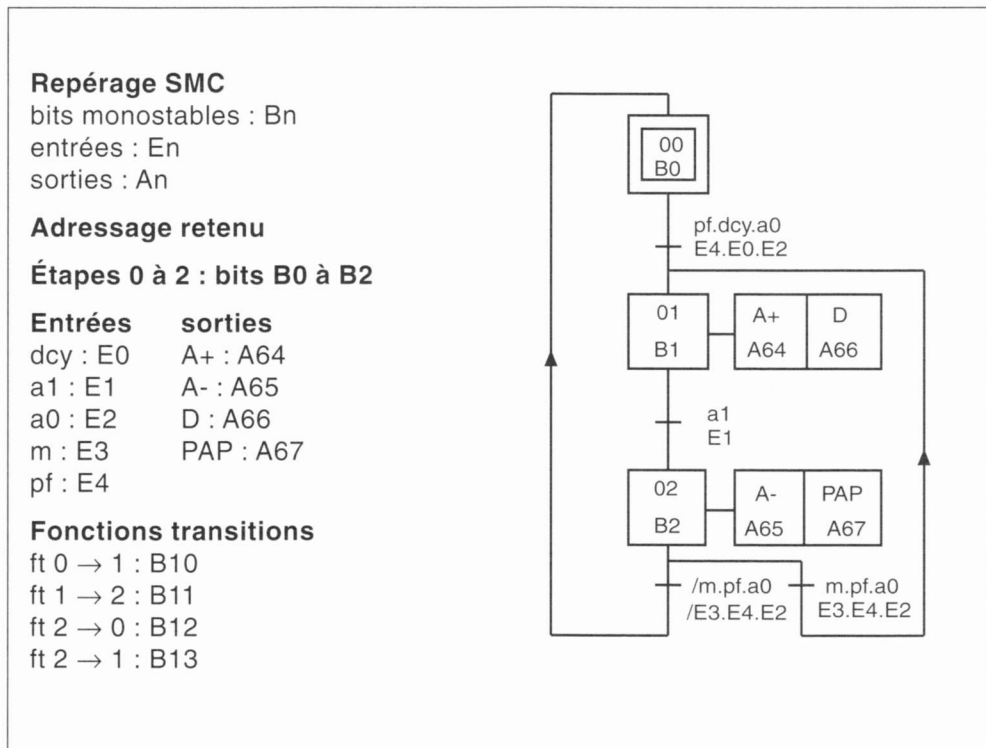


Figure 7.7
GRAFCET, niveau 3 (SMC).

Écriture en langage booléen

programme	commentaires
S0./B0=B0.	initialisation étape 0
S1.B0.E4.E0.E2=B10.	calcul de la fonction transition 0 → 1 = X0.pf.dcy.a0
S2.B1.E1=B11.	calcul de la fonction transition 1 → 2 = X1.a1
S3.B2./E3.E4.E2=B12.	calcul de la fonction transition 2 → 0 = X0./m.pf.a0
S4.B2.E3.E4.E2=B13.	calcul de la fonction transition 2 → 1 = X2.m.pf.a0
S5.B12+/B10.B0=B0.	mémoire étape 0
S6.B10+B13+/B11.B1=B1.	mémoire étape 1
S7.B11+ /B12./B13.B2=B2.	mémoire étape 2
S8.B1=A64.	si étape 1 active, mise à 1 sortie A+
S9.B2=A65.	si étape 2 active, mise à 1 sortie A-
S10.B1=A66.	si étape 1 active, mise à 1 sortie D
S11.B2=A67.	si étape 2 active, mise à 1 sortie PAP
S12.→ S1.	saut incondtionnel à la ligne S1

Écriture et/ou lecture en langage à contacts (fig. 7.8)

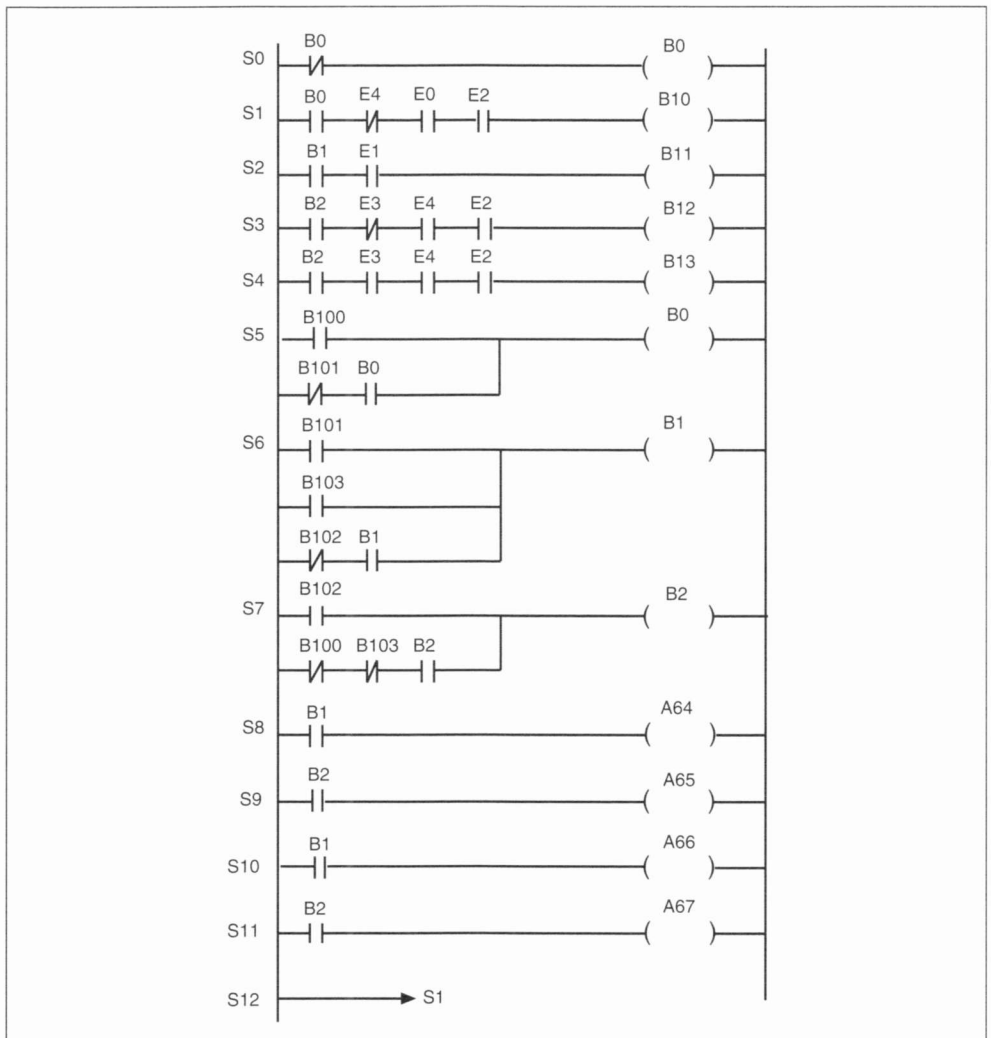


Figure 7.8
Programmation en langage
à contacts SMC.

Noter le procédé utilisé, ici, pour activer l'étape initiale lors du lancement du programme.

A la ligne S0, la bobine B0 est mise à 1 par son contact associé /B0. A la ligne S12, un saut inconditionnel est effectué à la ligne S1. Ainsi la ligne S0 ne sera exécutée que durant le premier cycle de scrutation du programme.

Ce procédé est applicable à d'autres API dans la mesure où il n'y a pas de conflit avec le chien de garde. Par exemple, il est inapplicable aux API TSX Schneider Télémécanique car leur chien de garde teste le temps écoulé entre deux passages par la première ligne du programme.

Il faut dans ce cas appliquer un autre procédé (voir application 2).



APPLICATION 2 : exploitation des bits monostables en langage à contacts PL7-2 et PL7-3 (TSX Télémécanique)

Repérage TSX

bits monostables : Bi - entrées : li,j - sorties : Oi,j

Adressage retenu

Étapes 0 à 2 : bits B0 à B2

Fonctions transitions

ft 0 → 1 : B10 ft 1 → 2 : B11

ft 2 → 0 : B12 ft 2 → 1 : B13

Entrées

dcy : I0,0

a1 : I0,1

a0 : I0,2

m : I0,3

pf : I0,4

sorties

A+ : O1,0

A- : O1,1

D : O1,2

PAP : O1,3

GRAFCET, niveau 3 (figure 7.9)

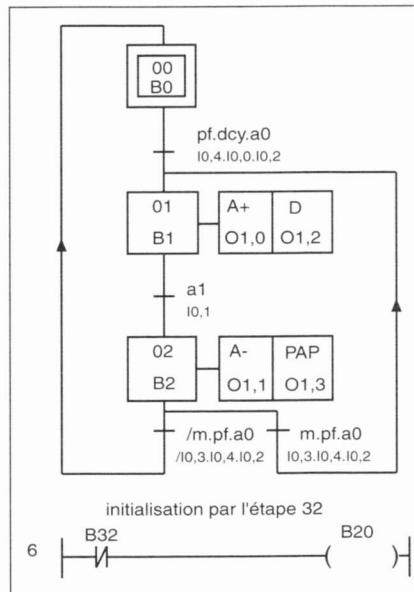


Figure 7.9 : GRAFCET, niveau 3 (TSX).

Programme en langage à contacts (figure 7.10)

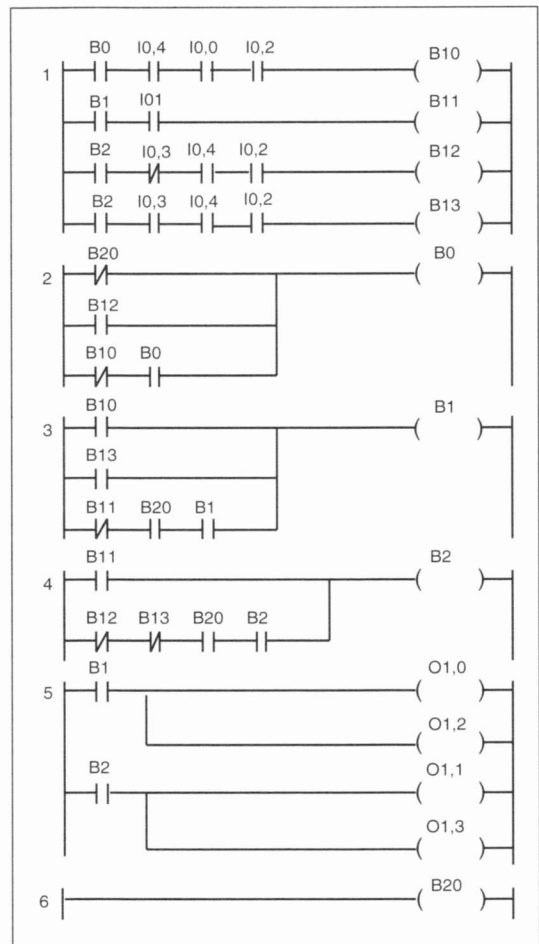


Figure 7.10 : programmation en langage à contacts (PL7-2 et PL7-3).

On notera le procédé utilisé pour initialiser le GRAFCET.

Lorsqu'on lance le programme pour la première fois (premier cycle de scrutation du programme), la mémoire-image de l'étape 0 (bit B0) est mise à 1 par l'intermédiaire du complément du bit B20 (contact /B20)(réseau 2).

A la fin du premier cycle de scrutation du programme, l'exécution du contenu du réseau 6 entraîne la mise à 1 inconditionnelle du bit B20. Il s'ensuit qu'au cycle suivant le contact B20 normalement ouvert se ferme et que le contact /B20 normalement fermé s'ouvre.

Remarquer qu'il ne semble pas indispensable pour l'instant d'autoriser la mise en mémoire des étapes 1 et 2 par le contact B20 si on n'envisage pas de forcer la situation initiale de ce GRAFCET. Par contre si ce type de forçage est souhaité il suffit de désactiver la bobine B20 du réseau 6, par une entrée manuelle (danger) ou par un bit image d'une étape d'un autre GRAFCET (exemple fig. 7.9 : étape 32).

Le contact /B20 du réseau 2 active la mémoire image de l'étape 0 et les contacts B20 des réseaux 3 et 4 désactivent respectivement les mémoires-images des étapes 1 et 2 si elles sont activées par le circuit d'auto-alimentation.

On obtient donc bien l'initialisation du GRAFCET en question. Ce sujet sera traité d'une façon plus générale dans le chapitre consacré aux structures de forçage du GRAFCET.

Remarque

Les bits systèmes SY0 et SY1 permettent également d'initialiser le GRAFCET. Il suffit de traduire en schéma à contacts le programme de l'application 3 suivante où l'initialisation est obtenue par l'intermédiaire de ces bits systèmes.



APPLICATION 3 : exploitation des bits monostables en langage littéral PL7-3 (TSX Télémécanique)

programme	commentaires
! SY1 → SY0	exploitation du bit système SY1
! SY0 → B20	exploitation du bit système SY0
! B0.I0,4.I0.0.I0,2 → B10	calcul $ft(0 \rightarrow 1) = B0.pf.dcy.a0$
! B1.I0,1 → B11	calcul $ft(1 \rightarrow 2) = B1.a1$
! B2.NOT I0,3.I0,4.I0,2 → B12	calcul $ft(2 \rightarrow 0) = B2./m.pf.a0$
! B2.I0,3.I0,4.I0,2 → B13	calcul $ft(2 \rightarrow 1) = B2.m.pf.a0$
! B20+B12+NOT B10.B0 → B0	activer étape 0
! B10+B13+NOT B11.B1 → B1	activer étape 1
! B11+NOT B12.NOT B13.B2 → B2	activer étape 2
! B1 → O1,0	action A+
! B2 → O1,1	action A-
! B1 → O1,2	action D
! B2 → O1,3	action PAP

Noter ici l'exploitation des bits systèmes SY0 et SY1, mis à 1 respectivement lors de la reprise à froid (SY0) et à chaud (SY1) pendant le premier cycle et remis à 0 au cycle suivant.

2.2. Implantation en bistable

Mémoires bistables	
avec calcul des fonctions transitions	sans calcul des fonctions transitions
étape 0 : activation $A(0) = B12$ désactivation $D(0) = B10$	$B2./m.pf.a0$ $B0.pf.dcy.a0$
étape 1 : activation $A(1) = B10 + B13$ désactivation $D(1) = B11$	$B0.pf.dcy.a0 + B2.m.pf.a0$ $= pf.a0(B0.dcy + B2.m)$ $B1.a1$
étape 2 : activation $A(2) = B11$ désactivation $D(2) = B12 + B13$	$B1.a1$ $B2./m.pf.a0 + B2.m.pf.a0$ $= B2.pf.a0$

En principe, en mode bistable, l'**activation est obtenue par l'instruction SET et la désactivation par l'instruction RESET**, mais certains constructeurs proposent encore des instructions qui leur sont propres.

Résultat d'expérimentations

En bistable, sur quelques API, cette structure ne nécessite pas de calcul des fonctions transitions si l'on prend la précaution de regrouper les activations et les désactivations dans le programme.

Ceci mérite d'être testé quand on a un doute pour un API donné.

Si le test se révèle positif, l'exploitation systématique de ce type de structure augmentera considérablement le nombre de bits internes disponibles pour les mémoires images des étapes des GRAFCET.



APPLICATION 1 : exploitation des bits bistables dans les TSX 37 et Premium (Schneider Télémécanique)

GRAFCET, niveau 3 (fig. 7.11)

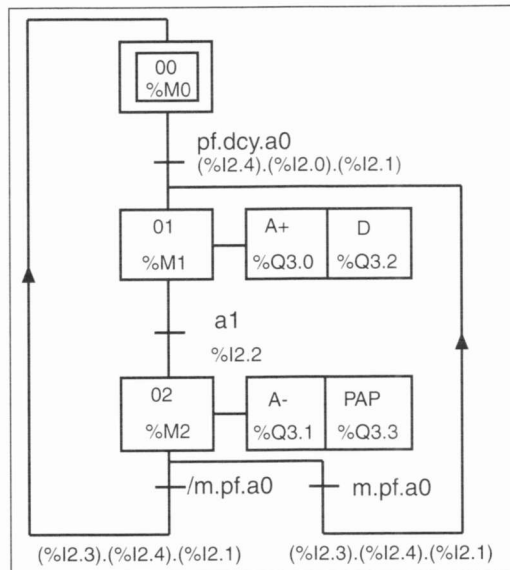


Figure 7.11
GRAFCET, niveau 3
(TSX 37 et Premium).

Repérage TSX 37

- bits bistables : %Mi (S) ou (R)
- entrées : %Ix.i
- sorties : %Qx.i

Adressage retenu

Etapes 0 à 2 : bits %M0 à %M2

Initialisation : bit %M10

Fonctions transitions

ft 0 → 1 : %M100 ft 2 → 0 : %M102

ft 1 → 2 : %M101 ft 2 → 1 : %M103

Entrées : dcy : %I2.0 m : %I2.3
a0 : %I2.1 pf : %I2.4
a1 : %I2.2

Sorties : A+ : %Q3.0 D : %Q3.2
A- : %Q3.1 PAP : %Q3.3

Le langage PL7 Pro utilise le standard de la norme CEI 1131-3 « Automates programmables - Partie 3 : Langages de programmation ». Ce standard spécifie la

syntaxe et la sémantique des éléments logiciels mis en oeuvre pour la programmation des API.

On y trouve la description :

- de deux langages textuels IL (Instruction List) et ST (Structured Text),
- de deux langages graphiques LD (Ladder Diagram) et FBD (Fonction Block Diagram),
- d'un formalisme graphique SFC (Sequential Function Chart), la forme internationale du GRAFCET.

Seuls les langages PL7 Junior et PRO permettent la programmation en littéral contrairement au langage PL7 Micro.

Programmation en langage à contacts PL7 Junior avec calcul des fonctions transitions (fig. 7.12)

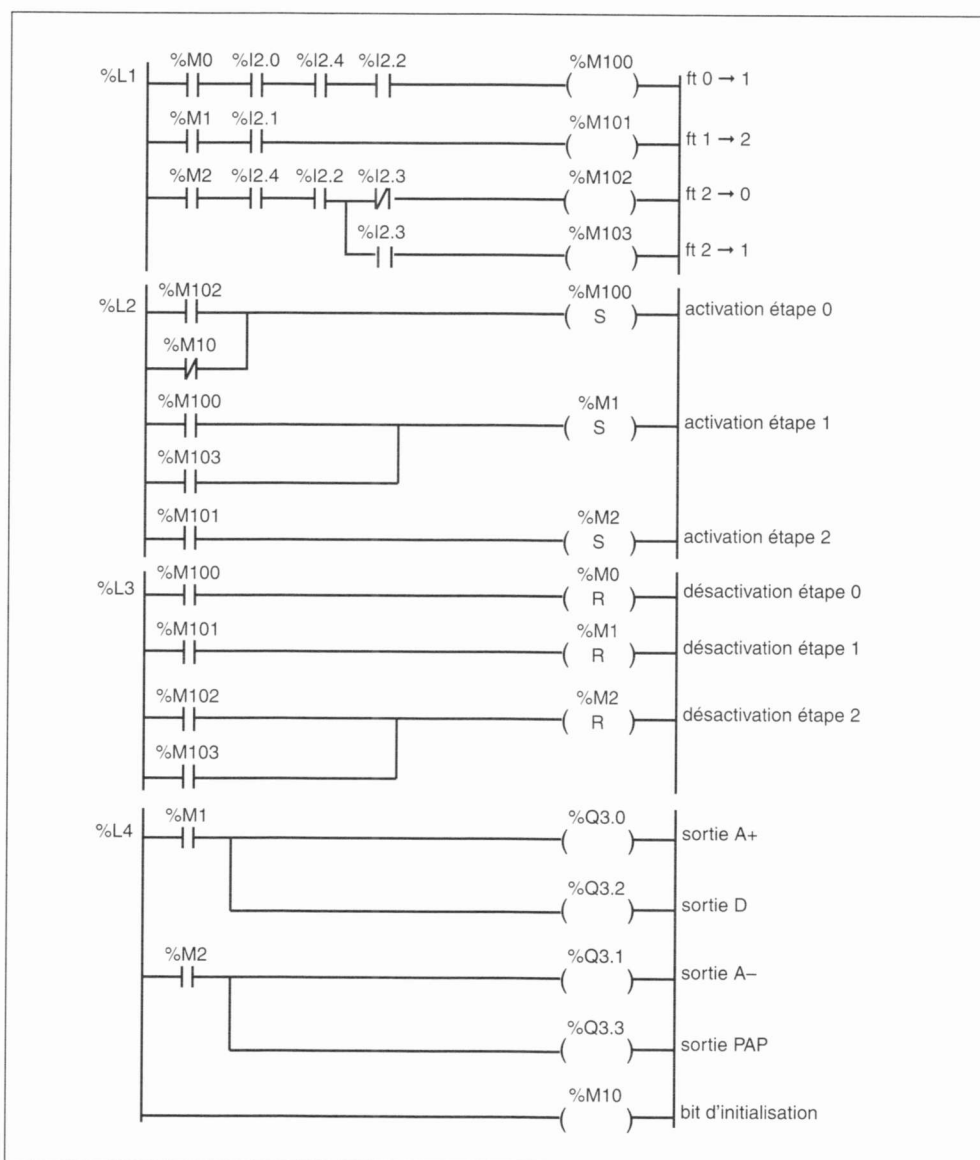


Figure 7.12
Programmation en langage à contacts PL7 Junior.

Les instructions SET (S) et RESET (R) transforment les bits %Mi monostables en bits bistables. Le bit %M10 assure l'initialisation du bit %M0, c'est-à-dire du GRAFCET, lors du premier cycle de lecture du programme.



APPLICATION 2 : exploitation des bits bistables en langage à contacts PL7-2 et PL7-3 Schneider Télémécanique sans calcul des fonctions transition (figures 7.13 et 7.14)

L'adressage est le même que pour l'application avec bits monostables.

Les instructions SET (notation S) et RESET (notation R) assurent la mémorisation des bits Bi monostables qui, de ce fait, deviennent bistables.

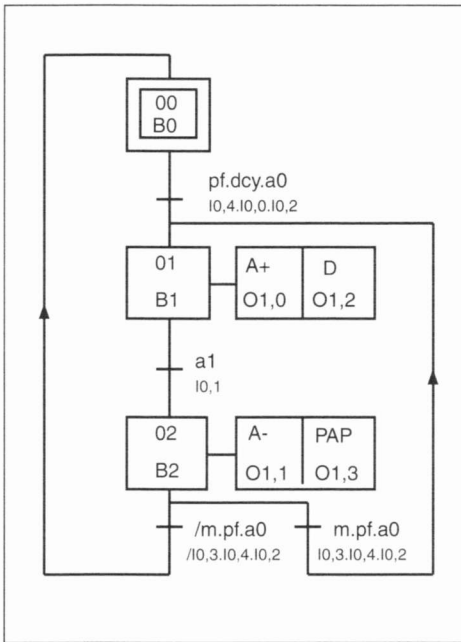


Figure 7.13
GRAFCET, niveau 3
(PL7-2 et PL7-3).

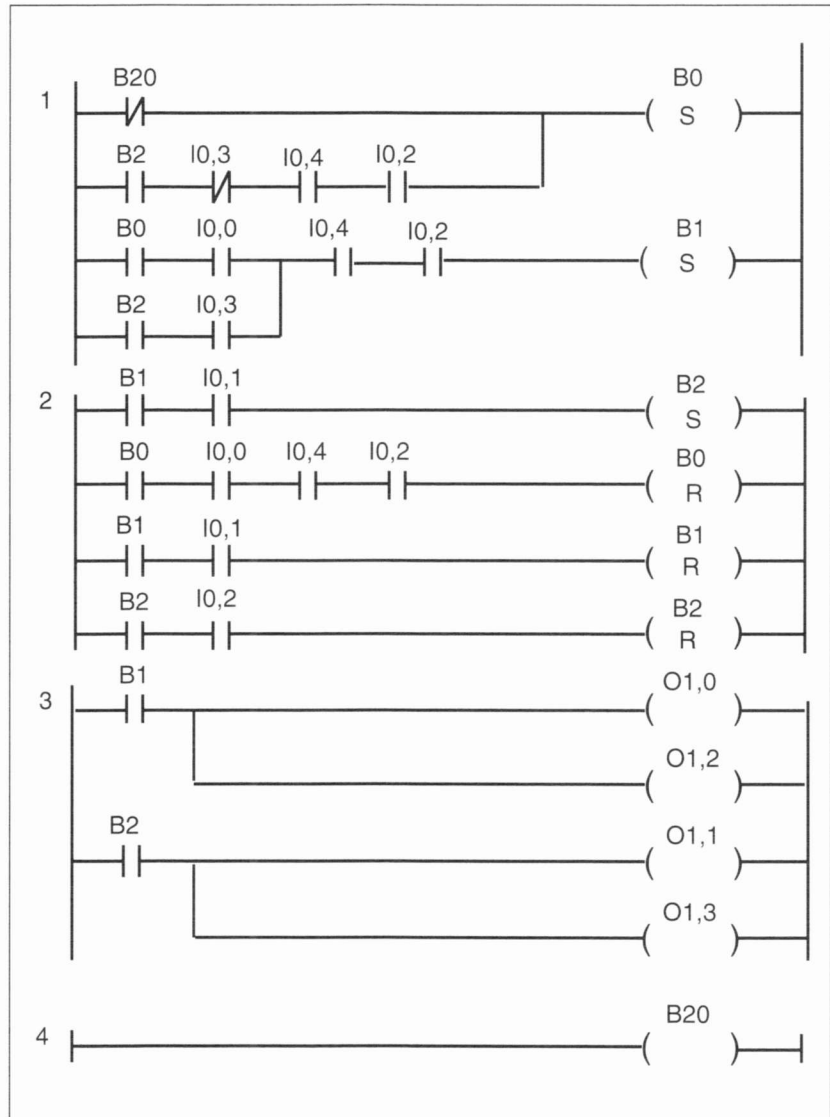


Figure 7.14
Programmation
en langage à contacts (PL7-2 et PL7-3).

Le regroupement des SET et RESET dans les réseaux 1 et 2 permet de se dispenser du calcul des fonctions transition.

L'initialisation automatique à la mise en service de l'API est réalisée de la même façon que pour la solution en monostable.

Ici aussi, l'exploitation des bits systèmes SY0 et SY1 reste possible en s'inspirant du programme en langage littéral de l'application 3 en monostable (page 130).

3. Structure informatique ou algorithmique

Cette structure s'appuie sur une **description par organigramme** des évolutions du GRAFCET. Elle exploite essentiellement la structure informatique dite **structure alternative**.

Définition de la structure alternative

algorithme : **SI (IF)** la fonction transition $ft(n \rightarrow p)$ est égale à 1
ALORS (THEN) activer l'étape p et désactiver l'étape n
 (franchissement de la transition $n \rightarrow p$)
SINON (ELSE) aller au test de la fonction suivante
FinSI (ENDIF)

Remarque

Lorsqu'à l'instruction **SINON (ELSE)** on associe un autre franchissement de transition, la structure s'apparente à la structure informatique dite **structure alternative complète**. Si l'instruction **SINON (ELSE)** est soit inexistante dans le langage de l'API soit omise volontairement, la non vérification du test de la fonction transition précédente doit assurer d'elle-même ou par une instruction spécifique de saut le passage au test de la fonction transition suivante. Dans ce cas, la structure devient une variante simplifiée de la structure informatique précédente. Par opposition elle est dite **structure alternative réduite**.

Comparaison des deux structures alternatives appliquées au GRAFCET (fig. 7.15)

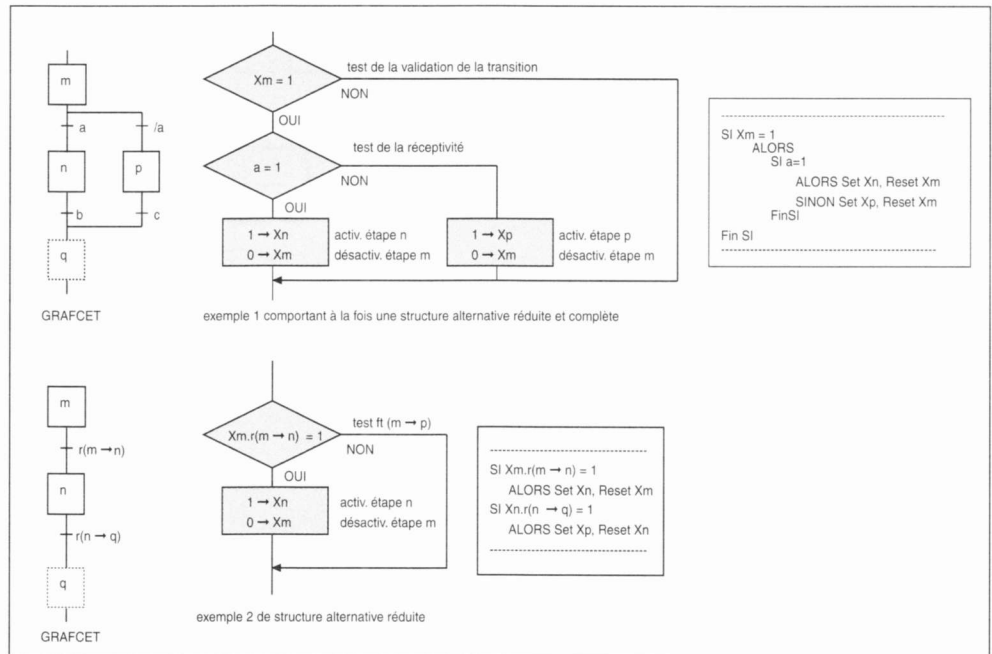


Figure 7.15
 Comparaison des structures alternatives.

Noter que le premier test ($X_m = 1$) de l'exemple 1 de la figure 7.15 appartient à une structure alternative réduite : pas de **SINON** au **ALORS** correspondant. Remarque que dans la structure alternative réduite le saut en avant est implicite.



APPLICATION 1 : exploitation des instructions de sauts des API APRIL (Série PB)

Le langage de ces API dispose d'instructions de tests et de sauts relatifs et absolus permettant de traiter la structure alternative réduite :

- saut relatif de nn lignes en avant : **SAUT Ann**,
- saut relatif de nn lignes en arrière : **SAUT Bnn**,
- saut absolu à l'adresse xxx : **SAUT xxx**.

Noter que le PB 80 a un nombre de sauts relatifs en avant limité à 10, n'a aucun saut relatif en arrière possible et possède un seul saut absolu réservé au rebouclage du programme.

Sur ce modèle, il est donc conseillé de travailler en langage GRAFCET littéral puisque celui-ci est disponible.

GRAFCET, niveau 3 (fig. 7.16)

Adressage retenu

Étapes

étape 0 : bit A00

étape 1 : bit A01

étape 2 : bit A02

Entrées

dcy : 000

a1 : 001

a0 : 002

m : 003

pf : 004

Sorties

A+ : 030

A- : 031

D : 032

PAP : 033

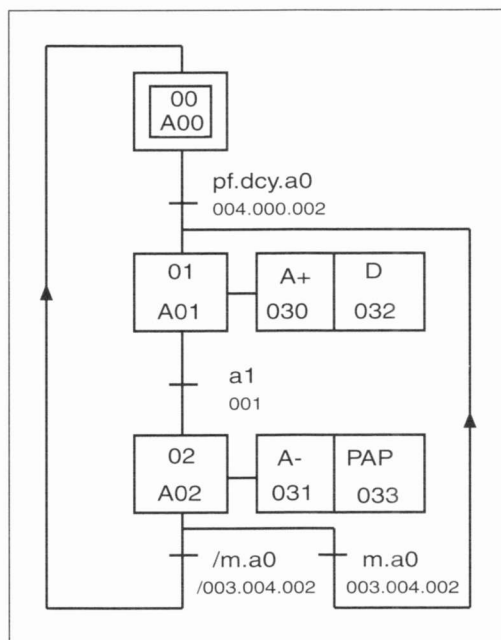


Figure 7.16 : GRAFCET, niveau 3 (PB).

Programme en langage assembleur ou mnémonique PB

0C30 SI B00	test si drapeau B00=1, si OUI saut relatif en avant de 7 lignes en C38, si NON aller en 0C32
0C31 SAUT A07	
0C32 DE A01	mise à 0 des bits A01
0C33 MZ BFF	à BFF (255 bits internes)
0C34 MU A00	activation étape initiale
0C35 DE 030	mise à 0 des sorties 030
0C36 MZ 03F	jusqu'à 03F (16 sorties carte 03)
0C37 MU B00	mise à 1 du drapeau B00
0C38 SI A00	test si X0=1
0C39 SI 004	test pf=1
0C3A SI 000	test dcy=1
0C3B SI 002	test a0=1
0C3C ET BFF	calcul fonction transition $ft(0 \rightarrow 1)=X0.pf.dcy.a0$
0C3D SI/ BFF	test si $ft(0 \rightarrow 1)=0$, si OUI saut relatif en avant de 3 lignes en 0C41, si NON aller en 0C3F
0C3E SAUT A03	
0C3F MU A01	activation étape 1
0C40 MZ A00	désactivation étape 0
0C41 SI A01	test si X1=1
0C42 SI 001	test si a1=1
0C43 ET BFE	calcul fonction transition $ft(1 \rightarrow 2)=X1.a1$
0C44 SI/ BFE	test si $ft(1 \rightarrow 2)=0$, si OUI saut relatif en avant de 3 lignes en 0C48, si NON aller en 0C46
0C45 SAUT A03	
0C46 MU A02	activation étape 2
0C47 MZ A01	désactivation étape 1
0C48 SI A02	test X2=1
0C49 SI /003	test si m=0
0C4A SI 004	test si pf=1
0C4B SI 002	test si a0=1
0C4C ET BFD	calcul fonction transition $ft(2 \rightarrow 0)=X2./m.pf.a0$
0C4D SI/ BFD	test si $ft(2 \rightarrow 0)=0$, si OUI saut relatif en avant de 3 lignes en 0C51, si NON aller en 0C4F
0C4E SAUT A03	
0C4F MU A00	activation étape 0
0C50 MZ A02	désactivation étape 2
0C51 SI A02	test X2=1
0C52 SI 003	test m=1
0C53 SI 004	test pf=1
0C54 SI 002	test a0=1
0C55 ET BFC	calcul fonction transition $ft(2 \rightarrow 1)=X2.m.pf.a0$
0C56 SI/ BFC	test si $ft(2 \rightarrow 1)=0$, si OUI saut relatif en avant de 3 lignes en 0C5A, si NON aller en 0C58
0C57 SAUT A03	
0C58 MU A01	activation étape 1
0C59 MZ A02	désactivation étape 2
0C5A SI A01	test X1=1
0C5B OU 030	mise à 1 sortie A+
0C5C SI A02	test X2=1
0C5D OU 031	mise à 1 sortie A-
0C5E SI A01	test X1=1
0C5F OU 032	mise à 1 sortie D
0C60 SI A02	test X2=1
0C61 OU 033	mise à 1 sortie PAP
0C62 SAUT C30	saut absolu en 0C30

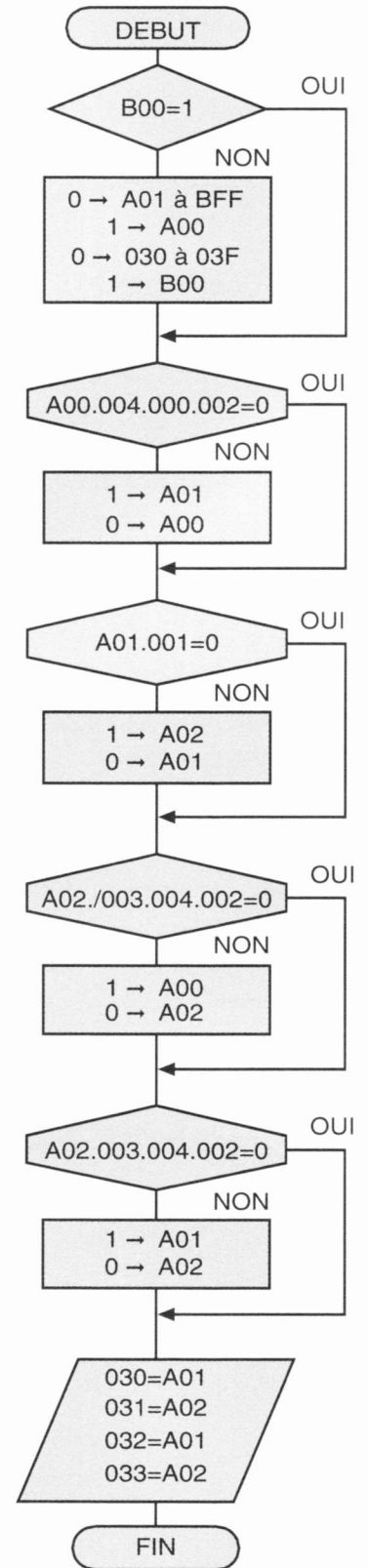


Figure 7.17 : organigramme.



APPLICATION 2 : implantation en langage littéral PL7-3 (TSX Schneider Télémécanique)

Ce langage dispose des instructions THEN (ALORS) et ELSE (SINON) permettant de traiter aussi bien la structure alternative complète que la structure alternative réduite. La structure alternative réduite n'exige pas d'instruction de saut.

Si à la suite d'un test, la réponse est OUI, les instructions situées derrière le THEN (ALORS) sont exécutées sinon le programme saute automatiquement à la ligne suivante repérée par un point d'exclamation.

Il s'agit bien dans ce cas d'une structure alternative réduite où l'instruction « sinon » est sous-entendue. La programmation d'un saut n'est pas nécessaire.

Pour le PL7-3, il s'agit d'une « **phrase conditionnelle** ».

GRAFSET, niveau 3 (fig. 7.18)

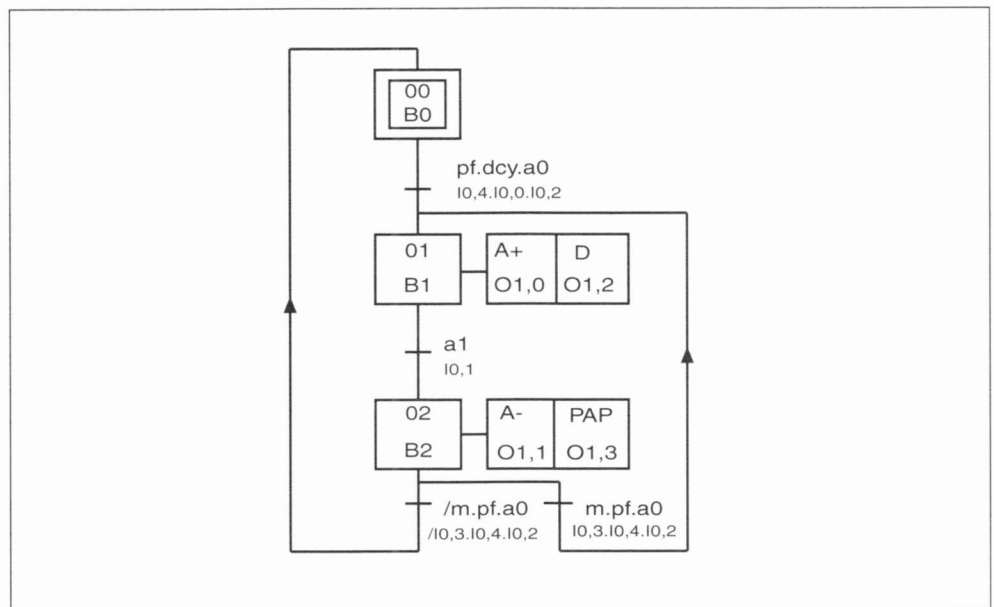


Figure 7.18
GRAFSET, niveau 3 (TSX).

Organigramme de la transition 0 → 1 en PL7-3 (fig. 7.19)

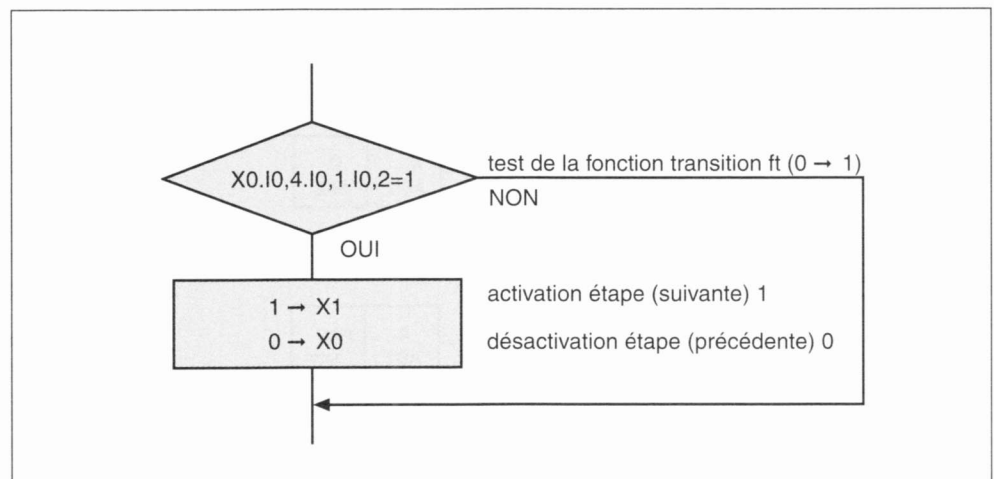


Figure 7.19
Organigramme de la transition
0 → 1 (structure alternative
réduite en PL7-3).

Programme en langage littéral PL7-3

! L1 IF NOT B20	
THEN SET B0;	initialisation du GRAFCET
RESET B1;	
RESET B2;	
SET B20	
! L2 IF B0.I0,4.I0,0.I0,2	test si B0.pf.dcy.a0=1
THEN SET B1; RESET B0	alors étapes : B1=1,B0=0
! L3 IF B1.I0,1	test si B1.a1=1
THEN SET B2; RESET B1	alors étapes : B2=1,B1=0
! L4 IF B2.NOT I0,3.I0,4.I02	test si B2./m.pf.a0=1
THEN SET B0; RESET B2	alors étapes : B0=1,B2=0
! L5 IF B2.I0,3.I0,4.I0,2	test si B2.m.pf.a0=1
THEN SET B1; RESET B2	alors étapes : B1=1,B2=0
! L6 B1 → O1,0	A+ = B1
! L7 B2 → O1,1	A- = B2
! L8 B1 → O1,2	D = B1
! L9 B2 → O1,3	PAP = B2

Noter que pour forcer la situation initiale du GRAFCET, il suffira de faire un RESET du bit B20 par l'intermédiaire d'une entrée manuelle (danger) ou par un bit interne image d'une étape d'un autre GRAFCET.

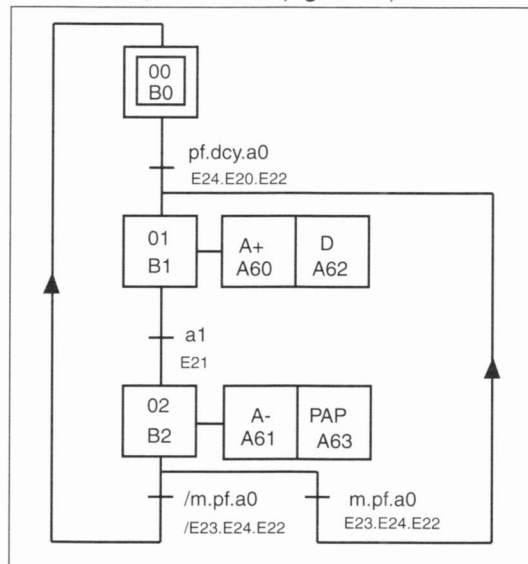
Dans ce type de formulation, il n'y a pas de saut à proprement parler. Toutes les lignes du programme sont lues successivement. Il s'ensuit que les seules actions exécutables sont celles qui sont précédées d'un test conditionné par IF ayant une réponse positive. Le repérage des lignes par des étiquettes (labels Li) n'est donc pas indispensable, le point d'exclamation suffit.

L'instruction FinSi (ENDIF) de la structure alternative complète et réduite est omise dans la plupart des langages algorithmiques des API.



APPLICATION 3 : implantation en langage informatique du C200 (ALSPA CEGELEC)

GRAFCET, niveau 3 (fig. 7.20)



Adressage retenu

Étapes

étape 0 : B0
étape 1 : B1
étape 2 : B2

Entrées

dcy : E20,
a1 : E21
a0 : E22,
m : E23
pf : E24

Sorties

A+ : A60
A- : A61
D : A62
PAP : A63

Figure 7.20
GRAFCET, niveau 3 (C200).

Description par organigramme (fig. 7.21)

Le langage du C200 présente la particularité d'avoir des instructions en français permettant la programmation des deux types de structures alternatives.

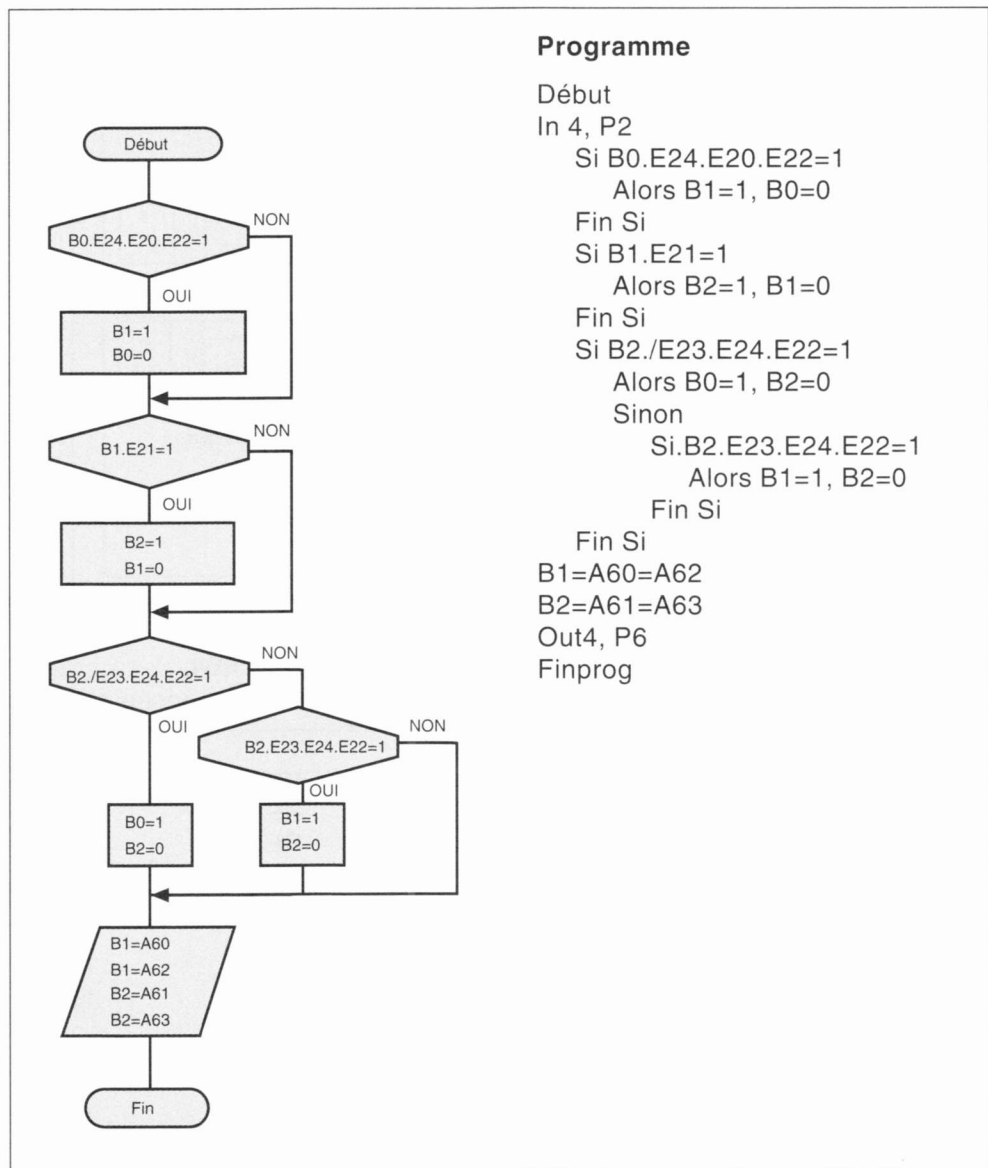


Figure 7.21
Organigramme (C200).



APPLICATION 4 : implantation en langage littéral AWL (FESTO)

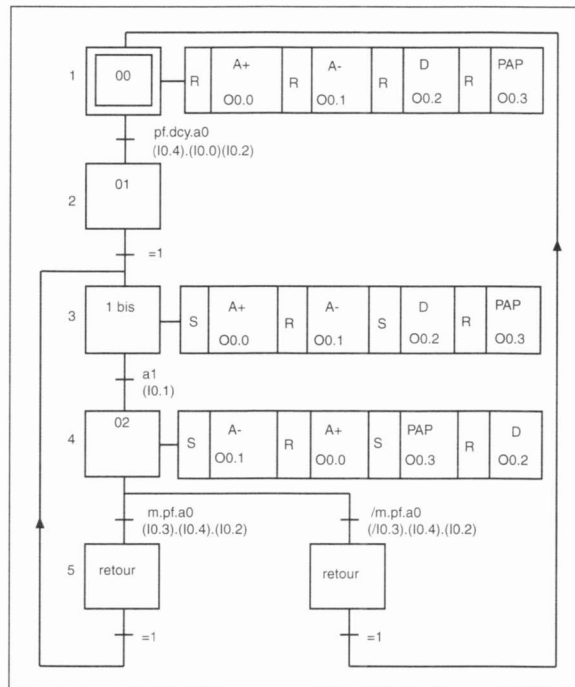
Le langage littéral AWL de Festo fait partie d'un ensemble de logiciels dit FST pour Festo Software Tool.

Ce langage est implantable sur micro-ordinateur PC ou compatible.

Pour programmer le GRAFCET, il utilise des instructions du type :

- STEP (pas) pour marquer le début d'une structure,
- IF (si), OR (ou), AND (et) et N (not, non) pour tester et calculer les réceptivités,
- THEN (alors), OTHRW (otherwise, autrement ou sinon),
- SET et RESET pour mettre à 1 ou à 0 les sorties,
- NOP (aucune opération), JMP TO n (jump to n, saut au pas n).

GRAFSET, niveau 3 (AWL FESTO)



Adressage retenu

Entrées	Sorties
dcy : I0.0	A+ : O0.0
a1 : I0.1	A- : O0.1
a0 : I0.2	D : O1,2
m : I0.3	PAP : O1,3
pf : I0,4	

Figure 7.22
GRAFSET, niveau 3
(AWL FESTO).

Programme en langage littéral AWL

(d'après la solution aimablement proposée par la société FESTO)

0001	STEP 0	(1)	'pas 1
0002	THEN	RESET	O0,0 'mise à 0 de A+
0003		RESET	O0,1 'mise à 0 de A-
0004		RESET	O0,2 'mise à 0 de D
0005		RESET	O0,3 'mise à 0 de PAP
0006	STEP 1	(2)	'pas 2
0007	IF	I0,4	'test pf=1
0008		AND	I0,0 'ET test dcy=1
0009		AND	I0,2 'ET test a0=1
0010	THEN	NOP	'aucune opération
0011	STEP 1bis	(3)	'pas 3
0012	THEN	SET	O0,0 'mise à 1 de A+
0013		RESET	O0,1 'mise à 0 de A-
0014		SET	O0,2 'mise à 1 de D
0015		RESET	O0,3 'mise à 0 de PAP
0016	STEP 2	(4)	'pas 4
0017	IF	I0,1	'test a1=1
0018	THEN	SET	O0,1 'mise à 1 de A-
0019		RESET	O0,0 'mise à 0 de A+
0020		SET	O0,3 'mise à 1 de PAP
0021	STEP RETOUR	(5)	'pas 5
0022	IF	I0,3	'test m=1
0023		AND	I0,4 'Et test pf=1
0024		AND	I0,2 'ET test a0=1
0025	THEN	JMP TO 1bis	(3) 'saut au pas 3
0026	IF	N	I0,3 'test m=0
0027		AND	I0,4 'ET test pf=1
0028		AND	I0,2 'test a0=1
0029		JMP TO 0	(1) 'saut au pas 1

Noter que les sorties sont gérées en bistable à l'intérieur de chaque pas (STEP).

Le GRAFCET de niveau 3 (niveau réalisateur) qui a été tracé (fig. 7.22) essaie de traduire le programme fourni par Festo comme solution au GRAFCET support de niveau 2 proposé au départ.

Le moins que l'on puisse dire c'est qu'il n'y a aucune transparence entre les deux GRAFCET.

En toute impartialité, il est conseillé aux utilisateurs du langage AWL de le prendre tel quel avec sa propre méthode d'analyse. Tout rapprochement avec le GRAFCET normalisé est à éviter.

4. Structure exploitant les registres pas à pas

Un registre pas à pas est caractérisé par son nombre de pas et par le fait qu'un seul pas est actif à la fois. L'évolution du registre pas à pas peut se faire dans un ordre quelconque et dans les deux sens : en avant ou en arrière.

Définition de la structure

- l'étape est associée à un pas du registre,
- l'évolution a lieu : si le pas précédent est actif (étape précédente active) et la réceptivité suivante vraie, dans ce cas le registre passe au pas suivant en désactivant le pas précédent (évolution synchrone).

Représentation symbolique d'un registre pas à pas

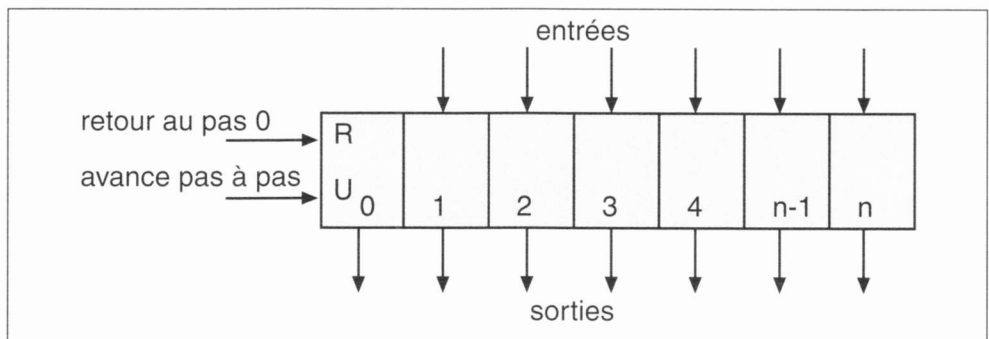


Figure 7.23
Représentation symbolique
d'un registre pas à pas.

APPLICATION 1 : exploitation du programmeur cyclique du langage PL7-2 (TSX Télémécanique)

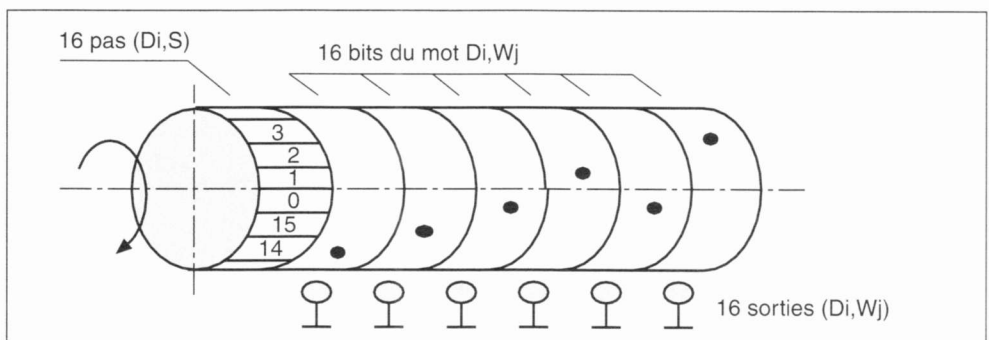


Figure 7.24
Principe du programmeur
cyclique du PL7-2 (TSX).

Le programmeur cyclique du PL7-2 (fig. 7.24) est comparable dans son principe à un programmeur à cames. Il dispose de 16 pas maximum. A chaque pas est associé un mot de 16 bits. Chacun des bits d'un mot peut gérer une sortie. Il y a donc 16 sorties possibles au total.

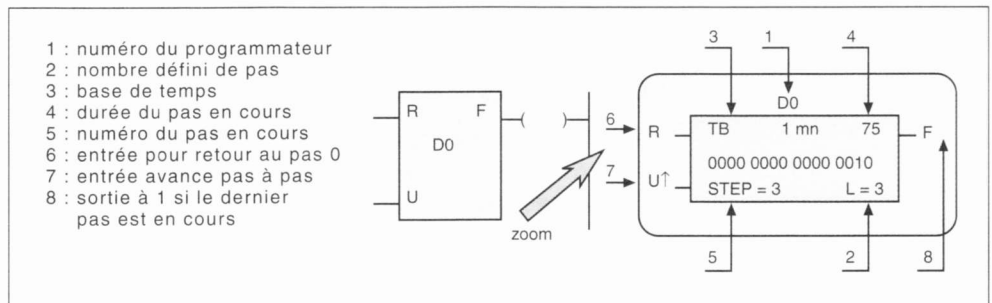
Pour programmer un programmeur cyclique il faut préciser :

- l'adresse du bloc programmeur cyclique : Di (i=0 à 7),
- le nombre de pas nécessaire (1 à 16),
- l'adresse de l'étape par rapport au pas : Di,S (S=0 à 15)
- l'adresse du mot : Di,Wj (j=1 à 15)
- l'adresse des sorties dans les mots : bits 0 à 15

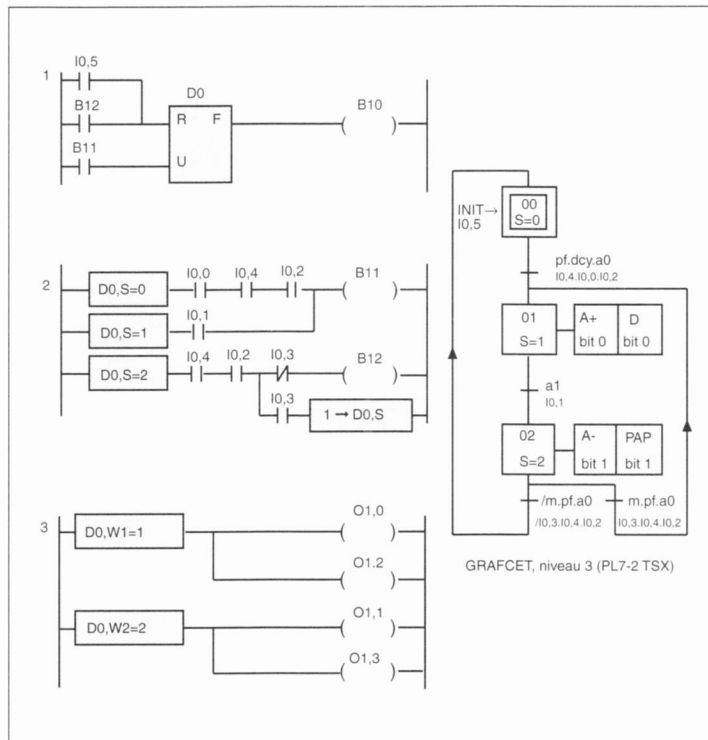
Choix pour le GRAFCET support :

- D0
- 3 (3 étapes)
- D0,S, D1,S, D2,S (étapes 0 à 2)
- D0,W1, D0,W2 (mots pas 1 et 2)
- bit 0 (A+,D), bit 1 (A-,PAP)

Figure 7.25
Bloc-fonction programmeur cyclique (PL7-2 TSX).



GRAFCET, niveau 3 et programme en langage à contacts du programmeur cyclique



Adressage retenu

Étapes 0 à 2
pas (step) : 0 à 2

Entrées
dcy : I0,0
a0 : I0,2
pf : I0,4
a1 : I0,1
m : I0,3
Init : I0,5

Sorties
A+ : O1,0
A- : O1,1
D : O1,2
PAP : O1,3

Figure 7.26
GRAFCET, niveau 3 et programme en langage à contacts (PL7-2 TSX).

Les sorties peuvent être programmées de deux manières :

- directement par la console (effectuer un zoom sur le bloc du programmeur) en affectant les sorties (Output) aux bits d'ordre 0 à F,
- ou comme il a été fait ici, en testant les bits d'ordre 0 et 1 des mots D0,W1 et D0,W2 :

- D0,W1 = (0000 0000 0000 0001) base 2 = (1) base 10 → A+ = D = 1
- D0,W2 = (0000 0000 0000 0010) base 2 = (2) base 10 → A- = PAP = 1

A savoir, le bit système SY0, sur reprise à froid, réinitialise les programmeurs cycliques sur leurs pas 0. L'initialisation peut aussi se faire, comme ici avec I0,4, manuellement par mise à 1 de l'entrée R.



APPLICATION 2 : exploitation de l'instruction AV des PB (APRIL)

Le registre pas à pas des PB est réalisé avec les bits internes d'adresses A00 à AFF ou A00 à BFF suivant le modèle de PB, en exploitant l'instruction AV. Mais l'instruction AV ne sait travailler que sur des bits d'adresses successives d'ordre croissant par exemple A10-A11-A12-A13, etc.

Cette instruction est exécutée suivant le microprogramme suivant :

- si la condition précédant AV est égale à 1 (réceptivité vraie), test du bit précédent,
- si ce bit précédent est à 1 (étape active),
 - alors activation du bit suivant
 - et désactivation du bit précédent.

Lorsque les bits ne sont pas successifs et c'est le cas lorsqu'on passe de l'étape 2 (bit A02) vers l'étape 1 (bit A01), on a recours aux instructions MU et MZ et au calcul préalable des fonctions transition correspondantes.

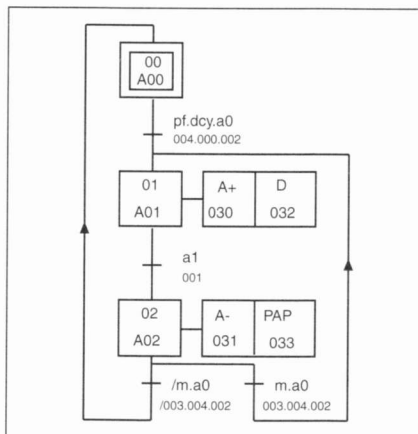


Figure 7.27
GRAF CET, niveau 3 (PB).

GRAF CET, niveau 3 (PB)

programme	commentaires
0C30 DE A01	mise à 0 du bit A01
0C31 MZ BFF	jusqu'au bit BFF
0C32 MU A00	mise à 1 du bit A00 (activation étape 0)
0C33 SI 004	test si pf=1
0C34 SI 000	test si dcy=1
0C35 SI 002	test si a0=1
0C36 AV A01	activation étape 1, désactivation étape 0
0C37 SI 001	test si a1=1
0C38 AV A02	activation étape 2, désactivation étape 1
0C39 SI A02	test si étape 2 active
0C3A SI /003	test si m=0
0C3B SI 004	test si pf=1
0C3C SI 002	test si a0=1
0C3D ET BFF	calcul fonction transition $ft(2 \rightarrow 0) = X2.a0./m$
0C3E SI BFF	test si $ft(2 \rightarrow 1) = 1$
0C3F MU A00	activation étape 0
0C40 SI BFF	test si $ft(2 \rightarrow 0) = 1$
0C41 MZ A02	désactivation étape 2
0C42 SI A02	test si étape 2 active
0C43 SI 003	test si m=1
0C44 SI 004	test si pf=1
0C45 SI 002	test si a0=1
0C46 ET BFE	calcul fonction transition $ft(2 \rightarrow 1) = X2.m.pf.a0$
0C47 SI BFE	test si $ft(2 \rightarrow 1) = 1$
0C48 MU A01	activation étape 1
0C49 SI BFE	test si $ft(2 \rightarrow 1) = 1$
0C4A MZ A02	désactivation étape 2
0C4B SI A01	test si étape 1 active
0C4C OU 030	mise à 1 sortie A+
0C4D SI A02	test si étape 2 active
0C4E OU 031	mise à 1 sortie A-
0C4F SI 001	test si étape 1 active
0C50 OU 032	mise à 1 sortie D
0C51 SI A02	test si étape 2 active
0C52 OU 033	mise à 1 sortie PAP
0C53 SAUT 0C33	saut incondtionnel en 0C33

Remarque

L'API C 200 Alspa possède des bits internes spécifiques dits bits glissants notés GS dont l'exploitation s'apparente à celle de l'instruction AV des PB April.



APPLICATION 3 : exploitation des registres du C100

Le langage du C100 dispose, pour la programmation du GRAFCET, de 64 registres d'adresses 00 à 63 de 100 pas ou étapes chacun d'adresses 00 à 99. Par suite l'adressage d'une étape se fait en associant le repère du registre et celui de son pas.

Exemple

L'étape 0015 est matérialisée par le pas 15 du registre 00.

La particularité du langage GRAFCET du C100 est qu'il associe la programmation en langage littéral (jeu d'instructions) avec un tracé automatique du GRAFCET sur l'écran de la console.

Ecran de la console du C100

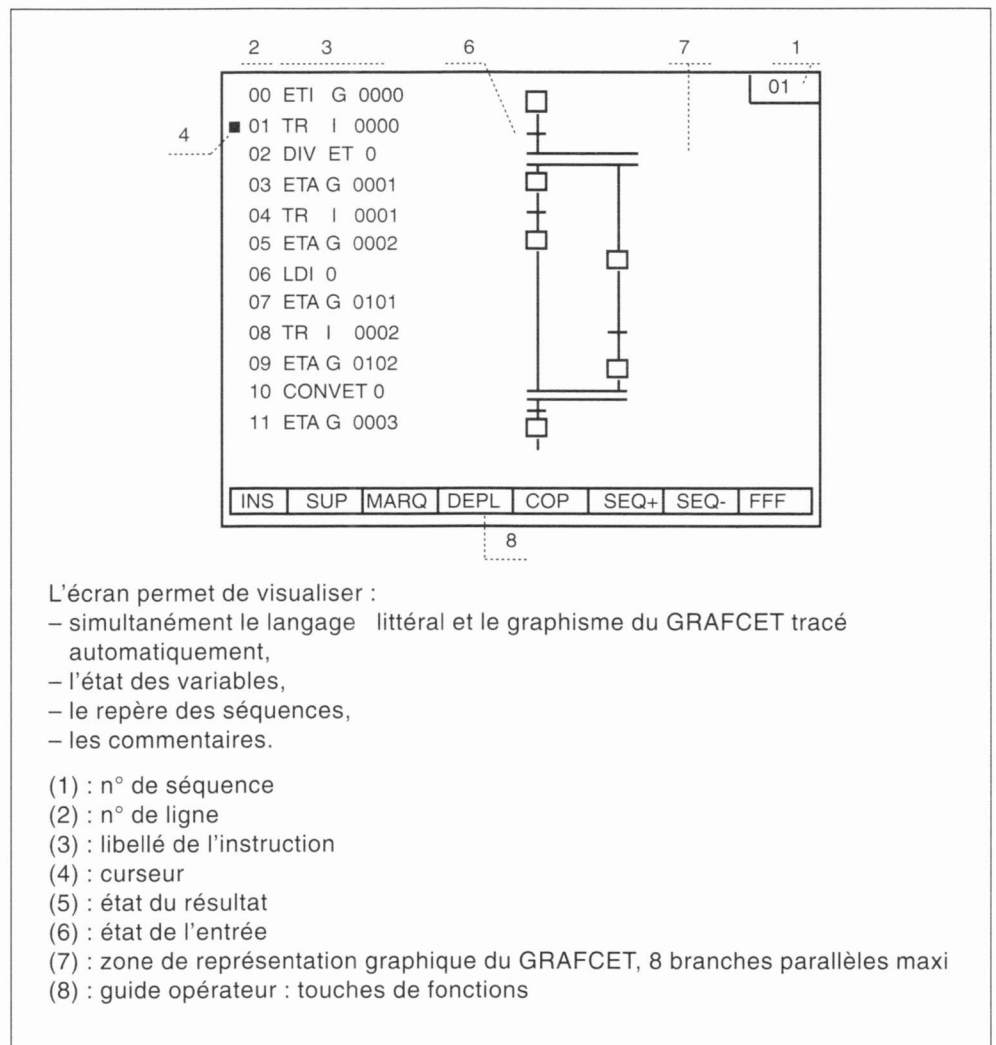


Figure 7.28
Présentation de l'écran
de la console du C100.

Jeu d'instructions propre au GRAFCET

étape initiale	ETI G 0200	choix étape 00 du registre 02 comme étape initiale
étape non initiale	ETA G 0814	étape 14 du registre 08
transition	TR I 0503	transition franchie si l'entrée 03 du module 05 est à 1
divergence en OU	DIV OU n	n est le niveau de divergence allant de 0 à 6
divergence en ET	DIV ET n	idem pour n
début branche divergente	LDI n	idem pour n
convergence en OU	CONVOU n	obligatoire si une branche comporte au moins une étape
convergence en en ET	CONVET n	obligatoire si une branche comporte au moins une étape
saut à une étape	JP G 0100	retour à l'étape 00 du registre 01
fin de séquence	FIN S	renvoie à la ligne 00 de la séquence suivante
fin de programme	FIN P	reboucle la scrutation au pas 00 de la séquence 00
insertion de commentaires	* texte	commentaires ou texte

Initialisation du GRAFCET

Utiliser la variable interne H 0707 ou une entrée externe I 0102, par exemple, et charger le numéro de l'étape initiale ou à initialiser dans le mot contenant son registre.

Les registres 00 à 63 sont contenus dans les mots respectifs W 128 à W 191.

Adressage retenu

Étapes 0 à 2 : G 0000 à G 0002 pas 00 à 02 du registre 00

Entrées : dcy : I 0100 entrée 00 du module 01
a1 : I 0101 entrée 01 du module 01
a0 : I 0102 entrée 02 du module 01
m : I 0103 entrée 03 du module 01
pf : I 0104 entrée 04 du module 01

Sorties : A+ : O 0200 sortie 00 du module 02
A- : O 0201 sortie 01 du module 02
D : O 0202 sortie 02 du module 02
PAP : O 0203 sortie 03 du module 02

GRAFCET, niveau 3 (C100)

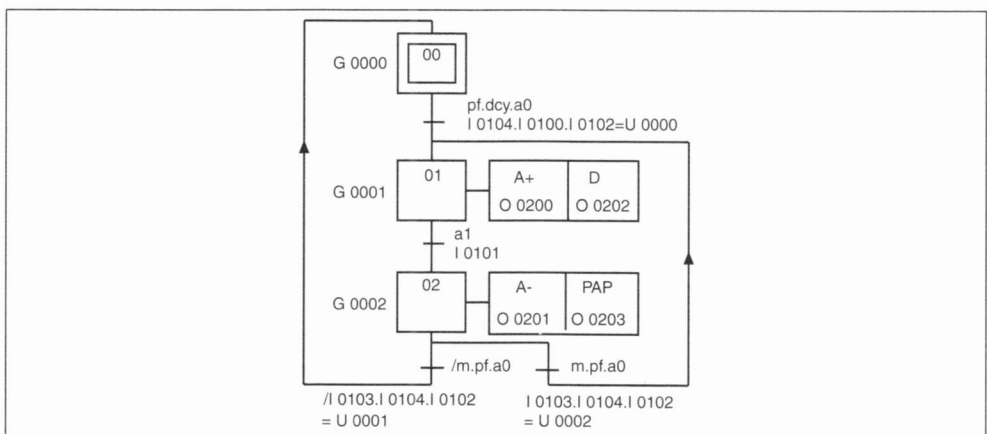


Figure 7.29
GRAFCET, niveau 3
(C100 ALSPA).

Programme en langage littéral GRAFCET C100

Séquence 00 : initialisation du GRAFCET et calcul des réceptivités

		Commentaires
00 *	INITIALISATION	
01 LD	K 0000 1	étape à initialiser
02 ET	H 0707	variable interne
03 SOR	W 128	mot contenant le registre 00
04 *	CALCUL DES RECEPTIVITES	
05 LD	I 0104	entrée pf
06 ET	I 0100	ET entrée dcy
07 ET	I 0102	ET entrée a0
08 SOR	U 0000	bit interne U 0000= pf.dcy.a0
09 LD	I 0103/	entrée /m
10 ET	I 0104	ET entrée pf
11 ET	I 0102	ET entrée a0
12 SOR	U 0001	bit interne U 0001=/m.pf.a0
13 LD	I 0103	entrée m
14 ET	I 0104	ET entrée pf
15 ET	I 0102	ET entrée a0
16 SOR	U 0002	bit interne U 0002=m.pf.a0
17 FIN	S	fin de séquence

Séquence 01 : tracé automatique du GRAFCET

		Commentaires
00 *	TRACE DU GRAFCET	
01 ETI	G 0000	définition étape initiale
02 TR	U 0000	réceptivité pf.dcy.a0
03 ETA	G 0001	activation étape 1, désactivation étape 0
04 TR	I 0101	réceptivité a1
05 ETA	G 0002	activation étape 2 et désactivation étape 1
06 DIV OU 0		début 1ère branche de la divergence en OU
07 TR	U 0001	réceptivité /m.pf.a0
08 JP	G 0000	activation étape 0 et désactivation étape 2
09 LDI	0	début 2ème branche de la divergence en OU
10 TR	U 0002	réceptivité m.pf.a0
11 JP	G 0001	activation étape 1 et désactivation étape 2
12 FIN	S	fin de séquence

Séquence 02 : commandes des sorties

		Commentaires
00 *	SORTIES	
01 LD	G 0001	test si étape 1 active
02 SOR	O 0200	mise à 1 sortie A+
03 LD	G 0002	test si étape 2 active
04 SOR	O 0201	mise à 1 sortie A-
05 LD	G 0001	test si étape 1 active
06 SOR	O 0202	mise à 1 sortie D
07 LD	G 0002	test si étape 2 active
08 SOR	O 0203	mise à 1 sortie PAP
09 FIN	P	fin de programme

Remarques

Le programme est organisé en séquences de 99 lignes maximum.
Les réceptivités comprenant plusieurs variables ou une seule variable complétée doivent être préalablement calculées et placées dans des bits internes U nnnn.

Structures d'implantation du GRAFCET dans les API

5. Structure exploitant le langage calcul sur mots

Le langage calcul permet d'effectuer des opérations sur le contenu de mots généralement de 16 bits. Il se présente comme une extension du langage à contacts ou bien fait partie intégrante des langages assembleurs ou littéraux.

Voici quelques-unes des opérations les plus courantes réalisables sur le contenu des mots :

- opérations arithmétiques : addition, soustraction, multiplication, division,
- opérations logiques bit à bit : OU, OU exclusif, ET,
- transferts : valeur dans un mot, contenu d'un mot dans un autre mot,
- transcodages : BIN-BCD, BCD-BIN, etc.

Définition de la structure

- **Associer à chaque GRAFCET (ou à chaque branche de GRAFCET) à étape active unique un mot de 16 bits dont le contenu (ou valeur) à un instant donné est égal au numéro de l'étape active à ce moment là.**

Exemple

Si le mot matérialisant le GRAFCET ou la branche de GRAFCET contient la valeur 5, l'étape 5 est obligatoirement activée. Par suite toutes les autres étapes associées à ce mot sont désactivées.

On ne peut pas trouver de moyen plus simple pour obtenir le synchronisme imposé par les règles d'évolution 2, 3 et 4.

- **Tester si la fonction transition est égale 1, alors attribuer au mot la valeur adéquate de manière à activer l'étape suivante et désactiver l'étape précédente.**

Exemple

Si le GRAFCET G0 matérialisé par le contenu du mot W0 se trouve dans la situation $S = \{5\}$ et doit atteindre après franchissement de la transition $5 \rightarrow 7$ la situation $S = \{7\}$, on effectuera les opérations suivantes :

- tester si le contenu du mot W0 est égal à 5 et si la réceptivité $r(5 \rightarrow 7)$ est vraie
- alors mettre la valeur 7 dans le mot W0, ce qui aura pour effet d'activer l'étape 7
- et de désactiver l'étape 5.

Le synchronisme imposé par les règles 2, 3 et 4 est ainsi parfaitement assuré.

- **Matérialiser les étapes en testant le contenu du mot matérialisant le GRAFCET.**

Exemple

L'étape n est identifiée au contenu n du mot associé au GRAFCET. Ainsi, si le GRAFCET G5 à étape active unique est associé au mot W5, l'étape 5 sera active quand le contenu du mot W5 sera égal à 55.

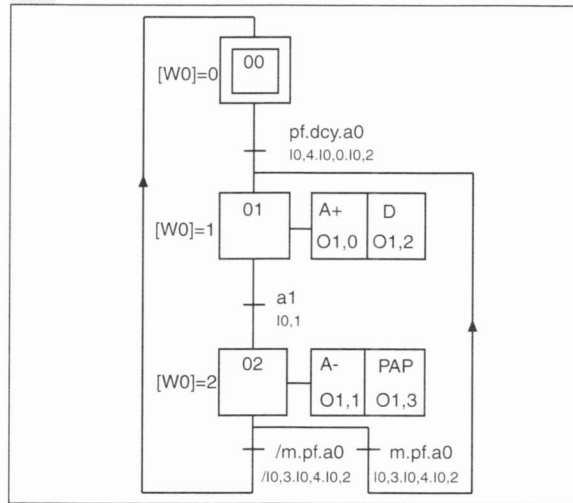
- **Initialiser le GRAFCET en mettant 0 dans le mot correspondant.**

Pour simplifier la programmation de l'initialisation des GRAFCET, on a intérêt à ce que toutes les étapes initiales réelles ou fictives soient associées au contenu 0 du mot correspondant.

Sur le plan mnémotechnique, c'est aussi à considérer.



Exemple 1 : GRAFCET à étape active unique



Le GRAFCET étant à étape active unique, un mot suffit pour le matérialiser (mot W0). Le contenu de W0, en décimal, pouvant varier de 0 à 9 999 cela revient à dire qu'on peut traiter à la limite un GRAFCET à étape active unique de 9 999 étapes. Quand on sait qu'on dispose en général d'un assez grand nombre de mots, on peut mesurer les possibilités offertes par ce type de structure. La notation [W0]=0 signifie que le contenu du mot W0 est égal à zéro.

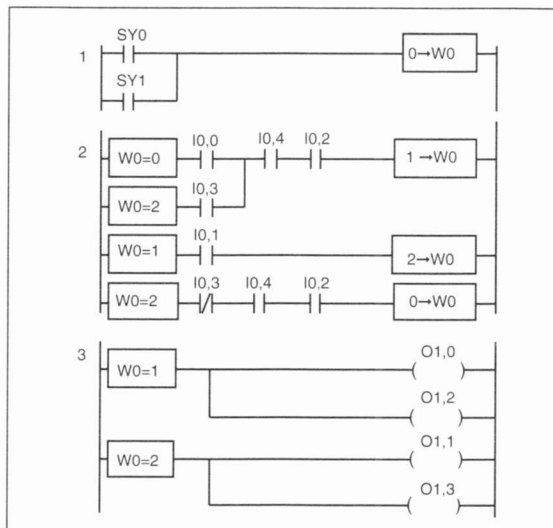
Figure 7.30
GRAFCET, niveau 3 (TSX).

Programme en langage littéral PL7-3

```

! IF SY0+SY1
THEN 0 → W0                                     initialisation
! IF IO,4.10,2([W0=0]).1,0+[W0=2].10,3)
THEN 1 → W0                                     activation étape 1
! IF [W0 = 1].10,1
THEN 2 → W0                                     activation étape 2
! IF [W0=2].NOT IO,3.10,4.10,2
THEN 0 → W0
! [W0 = 1] → O1,0                               sorties
! [W0 = 2] → O1,1
! [W0 = 1] → O1,2
! [W0 = 2] → O1,3
    
```

Programme en langage à contacts PL7-2 et PL7-3



Pour éviter de répéter les mêmes tests de pas du genre [W0 = n] de multiple fois, il semble intéressant, en début de programme, d'effectuer ces tests une fois pour toutes et d'affecter le résultat à des bits internes. Ces bits deviennent ainsi les bits-images des étapes du GRAFCET. Pour faciliter la programmation et éviter des confusions, on peut éventuellement choisir ces bits de manière à ce que leur repère soit identique au numéro des étapes. La seule limite à ce procédé restera la disponibilité en bits internes.

Figure 7.31
Programme en langage
à contacts
(PL7-2 et PL7-3 TSX).

Pour étendre le nombre de bits disponibles, on peut exploiter les bits de mots. Le contenu du mot W_i est codé globalement sur 16 bits. Mais une instruction de la forme W_i, j donne accès à chacun de ces bits. Les bits j sont numérotés en hexadécimal de 0 à F. Le bit de mot peut être lu (contact) ou écrit (bobine) en langage à contacts PL7-2 et PL7-3 et littéral PL7-3.

APPLICATION : Soit le mot W_{100} , on met chacun de ses 16 bits à 1 en activant une sortie notée $W_{100,0}$ pour le bit 0 ; $W_{100,1}$ pour le bit 1, ... , $W_{100,F}$ pour le bit 15.

Exemple 2 : GRAFCET avec parallélisme structural.

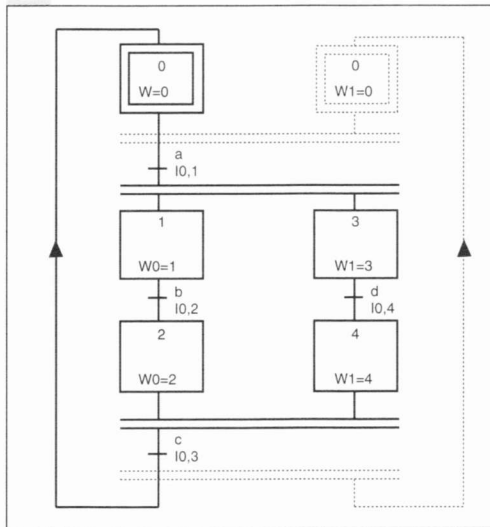


Figure 7.32
GRAFCET comportant un parallélisme structural. L'étape et les liaisons en pointillés ont été tracées pour obtenir la transparence entre le GRAFCET et le programme. Cette étape est une étape initiale fictive. Lorsqu'elle est active, la branche 3-4 est vide.

Ce GRAFCET n'étant pas à étape active unique, il faut définir les mots qui matérialiseront les branches de ce GRAFCET. Les étapes 0, 1 et 2 seront associées au contenu mot W_0 . Quant aux étapes 3 et 4, elles seront associées au contenu du mot W_1 .

Programme en langage

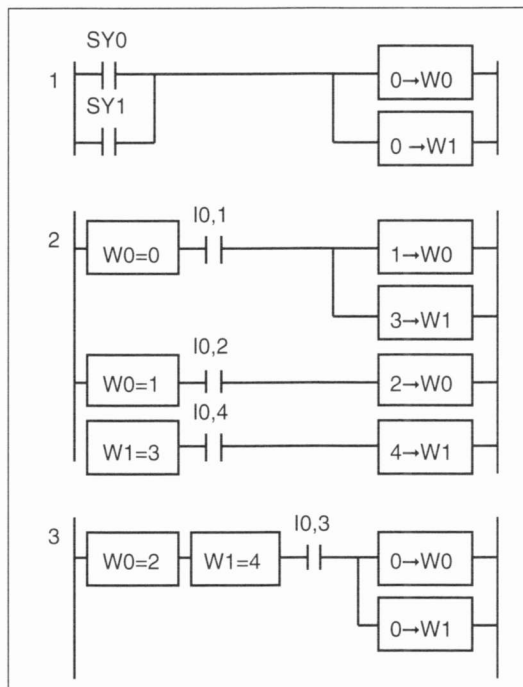


Figure 7.33
Programme en langage à contacts (PL7-2 et PL7-3 TSX).

Programme

```
! IF SY0+SY1
  THEN 0 → W0; 0 → W1
! IF [W0 = 0].IO,1
  THEN 1 → W0;3 → W1
! IF [W0 = 1].IO,2
  THEN 2 → W0
! IF [W1=3].IO,4
  THEN 4 → W1
! IF [W0=2].[W1=4].IO,3
! THEN 0 → W0;0 → W1
```

à contacts PL7-2 et PL7-3 en langage littéral PL7-3

6. Structure exploitant les langages GRAFCET littéraux

Cette structure repose sur l'implantation par le constructeur d'un langage qui, au travers du logiciel, respecte les 5 règles d'évolution du GRAFCET normalisé.

Ce langage, en général possède :

- une instruction pour tester la validation de la transition,
- une instruction pour tester si la réceptivité est vraie,
- une instruction pour franchir la transition si elle est validée et que la réceptivité associée est vraie, avec pour effet l'activation de la ou des étapes aval et la désactivation de la ou des étapes amont à cette transition.



APPLICATION 1 : langage GRAFCET littéral des PB (APRIL)

• Instructions disponibles

PRED : prédétermination (initialisation des GRAFCET)

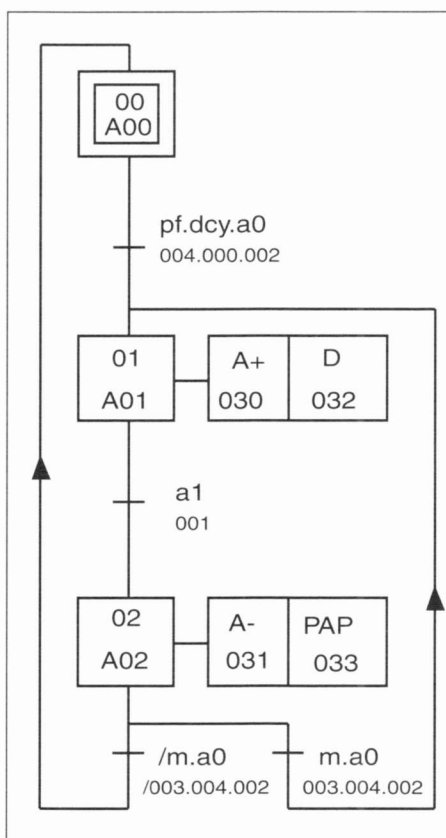
EANT : test étape antérieure à la transition n

RCEP : test réceptivité associée à la transition n

EPOS : activation étape postérieure et désactivation étape antérieure à la transition n

FORU : forçage à 1 d'une étape

FORZ : forçage à 0 d'une étape



Adressage retenu

Étapes	Entrées	Sorties
étape 0 : bit A00	dcy : 000	A+ : 030
étape 1 : bit A01	m : 003	A- : 031
étape 2 : bit A02	a1 : 001	D : 032
	a0 : 002	PAP : 033
	pf : 004	

Calculs des réceptivités

r(0 → 1) : BFF
 r(1 → 2) : BFE
 r(2 → 0) : BFD
 r(2 → 1) : BFC

Figure 7.34
GRAFCET, niveau 3 (PB).

• Programme en langage GRAFCET littéral (PB)

0C30	PRED		initialisation étape 0 ([1C00]= 8000 → 0C00)
0C31	EANT	0A00	test si étape 0 active (X0=1?)
0C33	SI	004	test si pf=1
0C34	SI	000	test si dcy=1
0C35	SI	002	test si a0=1
0C36	ET	BFF	calcul réceptivité r(0 → 1) (pf.dcy.a0=1)
0C37	RCEP	BFF	test si réceptivité r(0 → 1)=1 (pf.dcy.a0=1?)
0C39	EPOS	A01	activation étape 1 (X1=1), désactivation étape 0 (X0=0)
0C3B	EANT	A01	test si étape 1 active (X1=1?)
0C3D	RCEP	001	test si réceptivité r(1 → 2)=1 (a1=1?)
0C3F	EPOS	A02	activation étape 2 (X2=1), désactivation étape 1 (X1=0)
0C41	EANT	A02	test si étape 2 active (X2=1?)
0C43	SI/	003	test si m=0
0C44	SI	004	test si pf=1
0C45	SI	002	test si a0=1
0C46	ET	BFE	calcul réceptivité r(2 → 0)
0C47	RCEP	BFE	test si réceptivité r(2 → 0)=1 (/m.pf.a0=1?)
0C49	EPOS	A00	activation étape 0 (X0=1), désactivation étape 21 (X2=0)
0C4B	EANT	A02	test si étape 2 active (X2=1?)
0C4D	SI	003	test si m=1
0C4E	SI	004	test si pf=1
0C4F	SI	002	test a0
0C50	ET	BFD	calcul réceptivité r(2 → 1)
0C51	RCEP	BFD	test réceptivité r(2 → 1)=1 (m.pf.a0=1?)
0C53	EPOS	A01	activation étape 1 (X1=1), désactivation étape 2 (X2=0)
0C55	SI	A01	test si étape 1 active (X1=1?)
0C56	OU	030	mise à 1 sortie A+
0C57	SI	A02	test si étape 2 active (X2=1?)
0C58	OU	031	mise à 1 sortie A-
0C59	SI	A01	test si étape 1 active (X1=1?)
0C5A	OU	032	mise à 1 sortie D
0C5B	SI	A02	test si étape 2 active (X2=1?)
0C5C	OU	033	mise à 1 sortie PAP
0C5D	SAUT	0C31	saut inconditionnel en 0C31 sous l'instruction PRED

La structure de ce langage respecte bien les règles d'évolution :

- test de la validation de la transition par appel de l'étape antérieure à celle-ci : instruction EANT,
- test de la réceptivité associée à la transition : instruction RCEP, si celle-ci ne comporte qu'une seule variable non complémentée sinon calcul préalable du bit-image de cette réceptivité,
- franchissement de la transition si l'étape précédente est active et si la réceptivité est vraie : instruction EPOS, alors activation de l'étape postérieure et désactivation de l'étape antérieure à la transition.

Noter que si une même réceptivité se retrouve dans des transitions différentes, son bit-image doit être calculé extérieurement au programme GRAFCET proprement dit.

L'instruction PRED recopie la zone 1C000-1C00F dans la zone 0C00-0C0F. Dans notre exemple en écrivant 8000 soit 1000 0000 0000 0000 en 1C00, on met le bit 1A00 à 1. Sa recopie par PRED mettra le bit 0A00 à 1 de façon à initialiser le GRAFCET programmé.

Les instructions FORU et FORZ sont destinées plus spécifiquement aux forçages de situation (voir **chapitres III.13.4 et VII.10.3**).

Les instructions propres au GRAFCET contrairement aux autres instructions occupent deux lignes dans la mémoire-programme d'où la discontinuité dans la numérotation des lignes du programme.

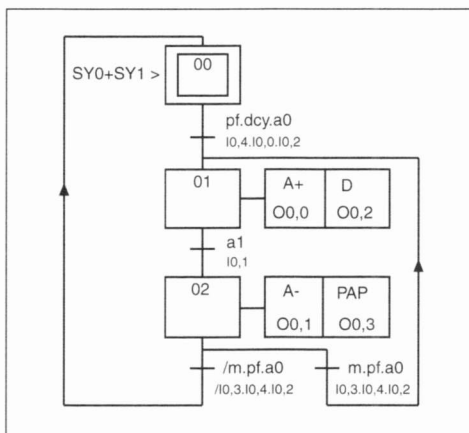


APPLICATION 2 : exploitation du langage PL7-1 des TSX 17 Schneider Télémécanique

Ce langage admet 62 étapes maximum (étapes initiales + étapes).

Le programme se divise en trois parties :

- le traitement préliminaire (initialisation, forçages, sécurités),
- le traitement séquentiel (GRAFSET),
- le traitement postérieur (sorties internes et externes).



GRAFSET, niveau 3 (TSX) (fig. 7.35)

Adressage retenu

Étapes

étape 00 : X00

étape 01 : X01

étape 02 : X02

Entrées

dcy : IO,0

a1 : IO,1

a0 : IO,2

m : IO,3

pf : IO,4

Sorties

A+ : OO,0

A- : OO,1

D : OO,2

PAP : OO,3

Figure 7.35

GRAFSET, niveau 3 (TSX).

Instructions GRAFCET du langage littéral PL7-1

L et LN : LOAD et LOAD NOT

: initialisation

A et AN : AND et AND NOT (ET et ET NON)

=* : début traitement séquentiel.

O et ON : OR et OR NOT (OU et OU NON)

-* : test étape

Xi : étape i

=* = POST : début traitement postérieur

= : activation sortie

EP : End Program (fin programme.)

Programme en langage GRAFCET littéral PL7-1

S0000 L	SY0	début traitement préliminaire, test bit système 0
S0001 O	SY1	OU test bit système 1
S0002 S	SY21	activation du bit système 21 (initialisation étape 0)
S0003 =*	00	début traitement séquentiel, test si étape initiale 0 active
S0004 L	IO,4	test si pf=1
S0005 A	IO,0	ET test si dcy=1
S0006 A	IO,2	ET test si a0=1 (calcul réceptivité)
S0007 #	01	activation étape 1, désactivation étape 0
S0008 -*	01	test si étape 1 active
S0009 L	IO,1	test si a1=1 (calcul réceptivité)
S0010 #	02	activation étape 2, désactivation étape 1
S0011 -*	02	test si étape 2 active
S0012 L	IO,4	test si pf=1
S0013 AN	IO,3	ET test si m=0
S0014 A	IO,2	ET test si a0=1 (calcul réceptivité)
S0015 #	00	activation étape 0, désactivation étape 2
S0016 -*	02	test si étape 2 active
S0017 L	IO,4	test si pf=1
S0018 A	IO,3	ET test si m=1
S0019 A	IO,2	ET test si a0=1 (calcul réceptivité)
S0020 #	01	activation étape 1, désactivation étape 2
S0021 =*	POST	début traitement postérieur
S0022 X	01	test étape 1 active
S0023 =	OO,0	mise à 1 sortie A+

S0030	EP	fin du programme

7. Structure exploitant les langages GRAFCET graphiques

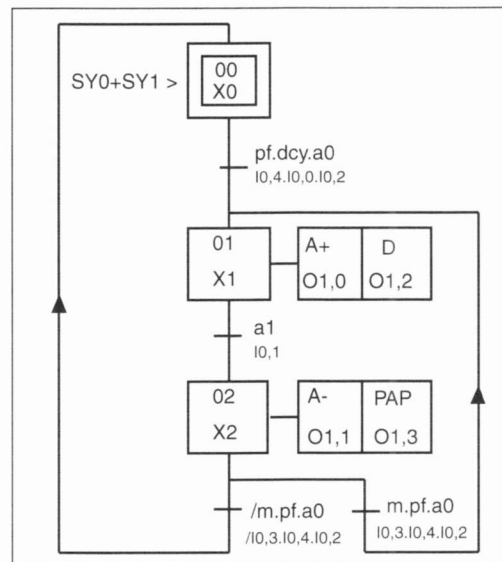
Ces langages permettent le tracé du GRAFCET sur l'écran de la console évoluée ou sur l'écran du moniteur du micro-ordinateur où est implanté le logiciel du langage GRAFCET graphique.

Les réceptivités et les actions sont programmées soit en langage à contacts, soit en langage logigramme, soit en langage booléen, assembleur ou littéral suivant le langage disponible et aussi suivant le choix du programmeur.



APPLICATION 1 : exploitation du langage GRAFCET graphique PL7-2 et PL7-3 des TSX Schneider Télémécanique

GRAFCET, niveau 3 (TSX)



Adressage retenu

Entrées

dcy : I0,0
a1 : I0,1
a0 : I0,2
m : I0,3
pf : I0,4

Sorties

A+ : O1,0
A- : O1,1
D : O1,2
PAP : O1,3

Figure 7.36
GRAFCET, niveau 3 (TSX).

Traitement séquentiel du GRAFCET et des transitions

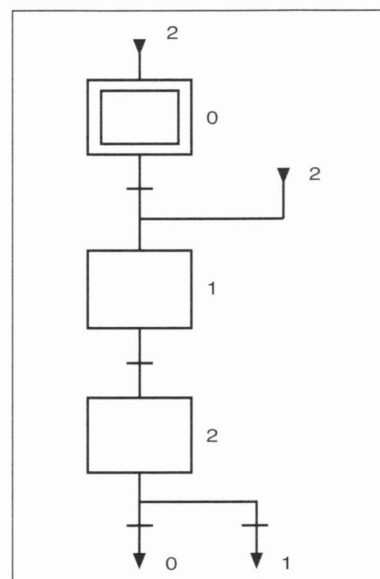


Figure 7.37
Traitement du GRAFCET.

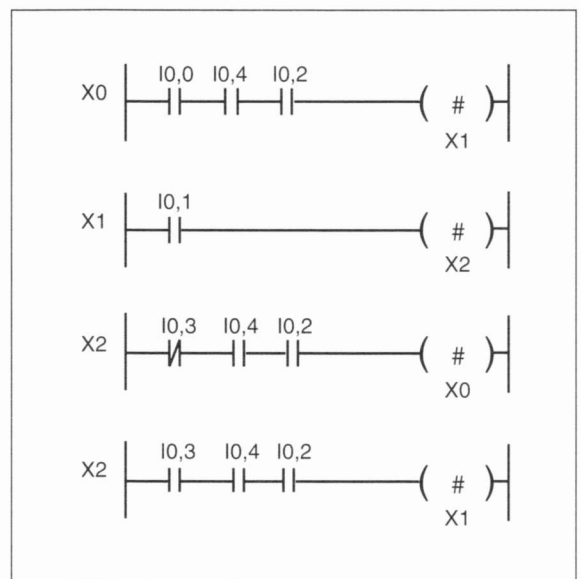
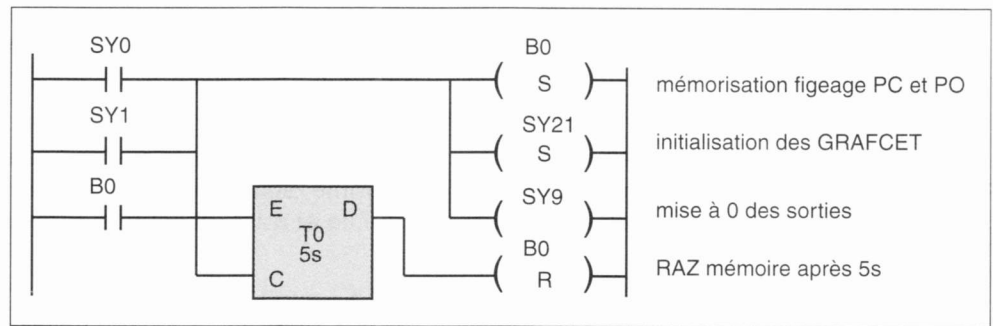
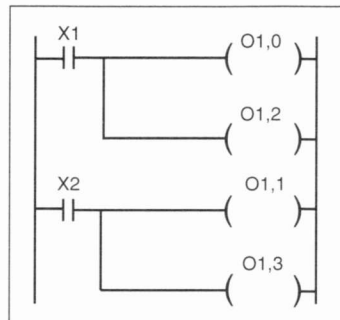


Figure 7.38
Traitement des transitions.

Traitement préliminaire



Traitement postérieur des sorties



Le programme est structuré en 3 parties, à savoir :

- **le traitement préliminaire** où l'on effectue entre autres l'initialisation de la PC (bit système SY21), la neutralisation des sorties (bit système SY9), les forçages de situations (bit système SY23);
- **le traitement séquentiel** où l'on trace le GRAFCET. Un zoom sur une transition donne alors accès au réseau où sera programmée la réceptivité correspondante soit en langage à contacts (PL7-2 et PL7-3) soit en langage littéral (PL7-3).

– **le traitement postérieur** où l'on programme les expressions logiques des sorties (actions diverses, temporisateurs, compteurs, etc.).

Noter qu'il est possible de programmer en mode symbolique plutôt qu'en adressage direct. Un tableau de correspondance entre les symboles et les adresses devra, dans ce cas, être préalablement dressé.



APPLICATION 2 : exploitation du langage GRAFCET graphique ALSPA

Ce langage est implanté dans la console évoluée P2 à écran vidéo dédiée aux API C200, C300 et C350 de la société ALSPA.

Cette console est autonome c'est-à-dire que la programmation peut se faire sans qu'elle soit connectée à l'automate.

La programmation du GRAFCET peut être réalisée en mode symbolique ou non. Les réceptivités et les sorties se programment soit en langage à contacts soit en langage booléen.

Utilisation du mode symbolique et du langage booléen

Écriture de la table des symboles

adresse	symbole	localité	str nb ref	commentaires *
E0040	DCY	G	001	Départ de cycle
E0041	A1	G	001	Fin de course aller vérin A
E0042	A0	G	001	Fin de course retour vérin A
E0043	M/A	G	001	Sélection auto, cycle par cycle

Note

* Contrairement à ce qui est présenté ici, les commentaires ne peuvent pas être incorporés à la table des symboles.

E0044	PF	G	001	Détection présence d'un lacon
A 0060	APLUS	G	001	Sortie de la tige du vérin A
A 0061	AMOIN	G	001	Rentrée de la tige du vérin A
A 0062	EVD	G	001	Électrovanne D
A 0063	PAP	G	001	Commande du pas à pas
SL 0080	ETAP0	G	002	Etape 0
SL 0081	ETAP1	G	002	Etape 1
SL 0082	ETAP2	G	002	Etape 2
SL 0100	RECEP	G	001	Calcul des réceptivités
RE 0101	R1	G	002	Réceptivité transition 0 → 1
RE 0101	R2	G	002	Réceptivité transition 1 → 2
RE 0102	R3	G	002	Réceptivité transition 2 → 0
RE 0103	R4	G	002	Réceptivité transition 2 → 1

Écriture du programme

```

1   hrd
2   1 cmt... référence API
      et programme
3   1 type $200
4   1 blo 1
5   1 dat $0
6   1 rep TEST
7   1 ver 0
8   1 nut 0
9   1 finhrd
10  1 cmt... GRAFCET 1,1
      étape active à la fois,
11  1 cmt... 1 MOT DE BITS
      SYNCHRONES D'ETAPES
      (BSE)
12  1 cmt... B0000: PREMIER BIT
      SYNCHRONE D'ETAPE
13  1 gra 0,1,1,B000
14  1
      v
      !r3
15  1 !
      [ ]etap0
      !
      !r1          v
16  1 !-----!r4
      !
      [ ]etap1
      !
17  1 !r2
      !
      [ ]etap2
18  1 !-----
      !r3          !r4
      vetap0      vetap1
19  1 fingra
20  1 cmt... PRISE EN COMPTE DES ENTREES DE LA CARTE SITUEE DANS
21  1 cmt... LE DEMI RACK 0 EN POSITION 2 (P0002) ET TRANSFERT
22  1 cmt... DANS LA ZONE DE BITS E

```

Instructions utilisées

cmt : commentaires
hrd : début configuration matériel (hardware)
fin hrd : fin configuration matériel
gra : début GRAFCET
fin gra : fin GRAFCET
sl : début sous-programme local
fin sl : fin sous-programme local
rec : réceptivité
sp : début sous-programme
fin sp : fin sous-programme
finprog : fin du programme
in : entrée (input)
out : sortie (output)
mst : mise sous tension

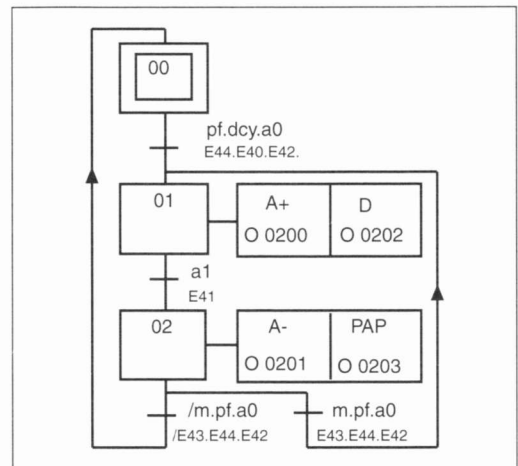


Figure 7.41 : GRAFCET, niveau 3 (ALSFA C200).

```

23 1 in 4, P0002
24 1 cmt... CALCUL DES RECEPTIVITES
25 1 SLRECEP
26 1 rec r1 , Epf . Edcy . Ea0 = 1
27 1 rec r2 , Ea1 = 1
28 1 rec r3 , / Em . Epf . Ea0 = 1
29 1 rec r4 , Em . Epf .Ea0 = 1
30 1 finsl
31 1 cmt... ACTIVATION DES BITS SYNCHRONES D'ETAPES (BSE)
32 1 SLETAP1
33 1 1 = set B0001
34 1 finsl
35 1 SLETAP2
36 1 1 = set B0002
37 1 finsl
38 1 cmt.... COMMANDE DES SORTIES DE LA ZONE A PAR LES BSE
39 1 B0001 = AAPLUS
40 1 B0002 = AAMOIN
41 1 B0003 = EVD
42 1 B0004 = PAP
43 1 cmt ... TRANSFERT DE LA ZONE DES BITS A VERS LA CARTE DE
44 1 cmt... SORTIE SITUEE DANS LE DEMI RACK 0 EN POSITION 6 (P0006)
45 1 out 4 , P 0006
46 1 finprog

```

Pour que le programme s'exécute correctement à la mise sous tension, il faut obligatoirement écrire un sous-programme d'initialisation appelé SP7F et le placer derrière le programme proprement dit.

Ce sous-programme ne sera évidemment exécuté qu'une seule fois.

```

1 1 SP7F
2 1 cmt... NEUTRALISATION DES SORTIES
3 1 1 = / AAPLUS = / AAMOIN = /EVD = /PAP
3 1 cmt... RAZ DES BITS SYSTEMES Z1, Z10, Z35 et Z40
4 1 1 = / Z0001 = / Z0010= / Z0035 = / Z0040
5 1 cmt... INITIALISATION GRAFCET 0
6 1 1 =Z0C80
7 1 cmt ... DEMASQUAGE DES INTERRUPTIONS A LA MISE SOUS TENSION
8 1 mst
9 1 finsp

```

Commentaires

L'organisation du programme se présente en général ainsi :

- 1 - définition de la configuration de l'automate,
- 2 - dessin du GRAFCET,
- 3 - définition de la configuration des cartes d'entrées et transfert en zone des bits E,
- 4 - calcul des réceptivités,
- 5 - affectation des bits synchrones d'étapes (BSE),
- 6 - affectation des BSE vers les sorties en zone A,
- 7 - transfert des bits A vers les cartes de sorties,
- 8 - sous-programme d'initialisation SP7F.

Si on utilise le mode symbolique, il faut préalablement définir la table de tous les symboles.



APPLICATION 3 : exploitation du langage GRAFCET graphique P4 version V5 ALSPA

Ce logiciel s'implante dans un micro-ordinateur IBM PC ou compatible. Son éditeur graphique bénéficie des avantages apportés par les cartes graphiques VGA ou SVGA, c'est-à-dire que la définition du graphisme est nettement supérieure à celle des écrans des anciennes consoles. L'écriture sur l'écran se fait en mode graphique alors que dans le langage étudié au paragraphe précédent, l'écriture se fait en mode texte.

Ce langage est destiné aux API C200, C300 et C350 ALSPA.

La particularité du langage P4 est d'associer les boîtes d'actions aux étapes.

Le GRAFCET tracé sur l'écran est ainsi très proche du GRAFCET de niveau 3 proposé par le concepteur de l'automatisme étudié.

Le GRAFCET est ensuite traduit en langage littéral LEA ALSPA ce qui, en principe, doit réduire la place occupée en mémoire par le programme.

A titre indicatif, voici comment se présente le GRAFCET sur l'écran du moniteur avant sa traduction en littéral (fig. 7.42).

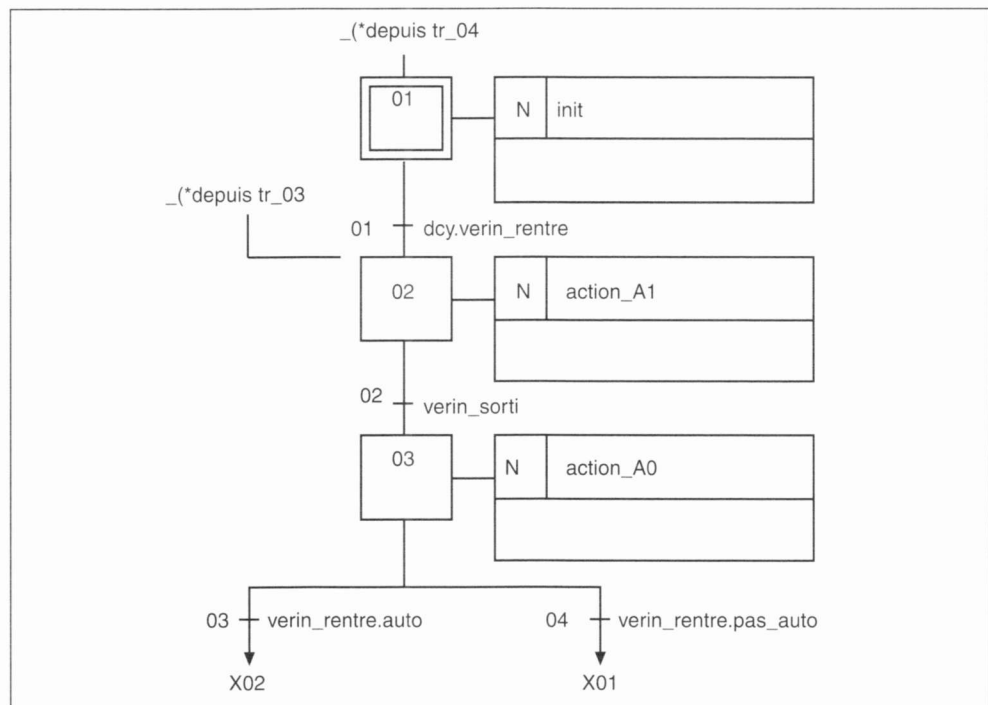


Figure 7.42
Écriture du GRAFCET
en langage P4
version V5 ALSPA.



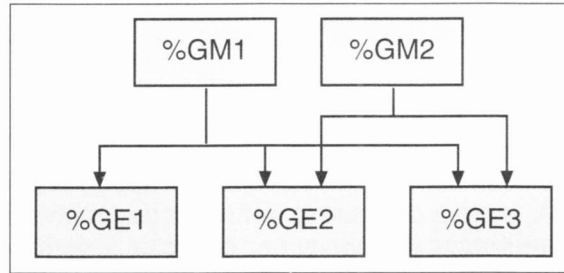
APPLICATION 4 : exploitation du logiciel ORPHEE d'APRIL

ORPHEE est un logiciel qui s'implante dans un micro-ordinateur IBM PC ou compatible.

Il permet de programmer les API APRIL 5000 et 7000 en langage à contacts ou en langage GRAFCET graphique.

Il exploite la notion de GRAFCET Maître (GRAFCET principal), la notion de GRAFCET Esclave (GRAFCET sous-programme) et la notion de macro-étape avec des particularités qu'il faut souligner.

Figure 7.43
Hiérarchie
entre les GRAFCET.



Le GRAFCET maître se note %GMn où n est un numéro-repère. De même le GRAFCET esclave se notera %GEn et la macro-étape %XEn. Un bit d'étape se note soit %GMn : Xi, soit %GEn : Xi, soit %XEn : Xi suivant le graphe d'origine avec i égal au numéro de l'étape.

La coordination entre ces GRAFCET (notion de hiérarchie) peut se faire dans les conditions suivantes :

- un %GM peut commander plusieurs %GE,
- un %GE peut être commandé par plusieurs %GM,
- par contre un %GM ne peut pas commander un autre %GM, pas plus qu'un %GE ne peut commander un autre %GE.

ORPHEE distingue quatre états différents pour la situation des graphes %GM et %GE :

- un état dit INACTIF correspondant à une situation vide,
- un état dit ACTIF où au moins une étape est active,
- un état dit GELÉ INACTIF avec sauvegarde de la situation courante et de l'état des actions associées aux étapes actives au moment du figeage,
- un état dit GELÉ ACTIF avec sauvegarde de la situation courante et rafraîchissement cyclique des sorties associées aux étapes actives au moment du figeage.

Pour mettre les graphes dans un de ces états, on utilise des Boîtes Fonctionnelles Constructeurs (BFC). Les BFC sont des sortes de blocs fonctions exploitables en langage à contacts.

Exemple : BFC pour GEL ACTIF d'un GRAFCET Esclave (fig. 7.44)

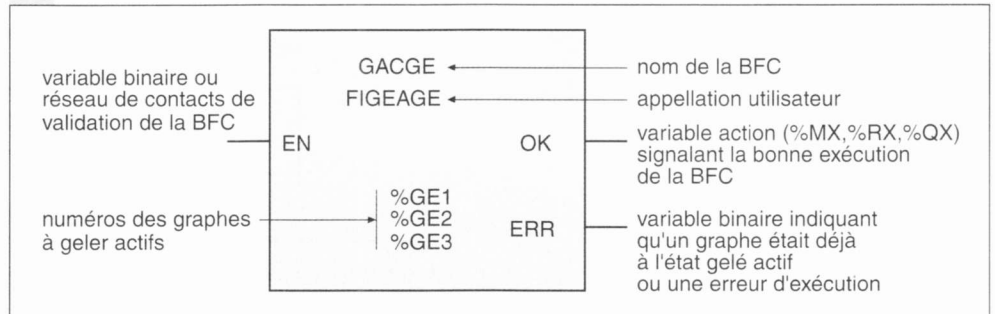


Figure 7.44
Aspect d'une boîte
fonctionnelle constructeur
(BFC).

ORPHEE dispose de 12 BFC pour activer en situation initiale ou courante sauvegardée, désactiver, geler actif ou inactif, dégeler les %GM et %GE et 3 BFC pour mémoriser la situation des %GM, %GE et %XE.

D'autres BFC permettent le forçage de situation des %GM, %GE et %XE. Par ailleurs, il existe toute une panoplie de BFC pour faire face à tous les cas de figures rencontrés en automatisme :

- calculs mathématiques arithmétiques, exponentiels et trigonométriques,
- opérations logiques,
- temporisateurs, horodateurs, compteurs, décalages, transcodages,
- traitement de tables, itérations, traitements de chaînes, tests,
- mesure et régulation, traitement d'axe numérique,
- communication sur réseau.

Le graphisme du GRAFCET respecte la norme sauf, notamment, en ce qui concerne le lancement du GRAFCET esclave et les étapes de début et fin de l'expansion d'une macro-étape.

On notera que les GRAFCET maître et esclave peuvent être des GRAFCET ouverts, c'est-à-dire que le retour en situation initiale ne pourra se faire que par l'intermédiaire d'une BFC.

Exemple : symboles GRAFCET du logiciel ORPHEE

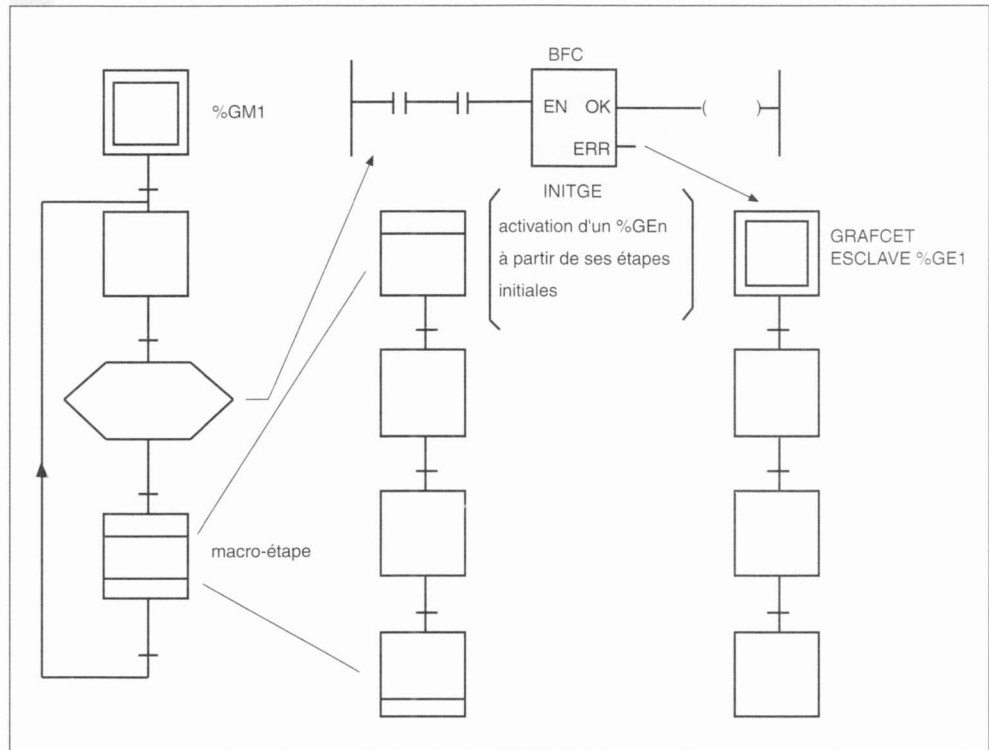


Figure 7.45
Symboles pour le GRAFCET
du langage ORPHEE.

8. Conclusion sur les structures

La structure basée sur la fonction mémoire est une structure universelle en ce sens qu'elle est programmable sur tous les types d'automates programmables industriels (API).

Elle est réalisable par l'intermédiaire de bits internes monostables ou bistables.

Avec les bits monostables, le calcul des fonctions transitions donne des structures fonctionnant sans aléas.

Comme principaux inconvénients, on peut relever le fait, d'une part, d'allonger le programme, et d'autre part, de réduire le nombre de bits internes disponibles pour matérialiser les étapes des GRAFCET.

Avec les bits bistables, si on prend la précaution de regrouper ensemble les activations puis les désactivations, on peut, en principe, se dispenser du calcul des fonctions transitions.

Si l'API est d'un type nouveau ou non testé ici, il est conseillé de faire un essai avec un GRAFCET comportant une boucle à deux étapes afin de déterminer si cet API accepte ou non ce mode de programmation.

Dans l'affirmative, outre le gain en bits internes disponibles, l'utilisation des bits bistables donne un programme plus facile à interpréter.

Les activations et les désactivations des mémoires, donc des étapes, apparaissent plus clairement.

Si le langage admet la **structure informatique** dite structure alternative réduite, les instructions de sauts (absolus ou relatifs) ne sont pas indispensables.

La structure basée sur les registres pas à pas reste d'application limitée mais son association avec le GRAFCET graphique (cas du langage du C100 ALSPA) est un plus appréciable.

La structure exploitant le langage calcul sur mot est une des plus faciles à mettre en oeuvre. Le seul reproche qu'on peut lui faire, si on n'a pas l'habitude du travail sur mot, est un manque de transparence du programme par rapport au GRAFCET.

En fait, au niveau de la console, il suffit de savoir lire le contenu des mots pour suivre en permanence l'évolution des GRAFCET du système.

Par exemple, le contenu du mot qui matérialise un GRAFCET à étape active unique indique immédiatement et sans erreur possible quelle est la situation de ce GRAFCET à l'instant considéré.

La structure exploitant le langage GRAFCET littéral suppose une parfaite connaissance des instructions du langage. Le suivi des évolutions du GRAFCET quant à lui, nécessite une bonne pratique des consoles.

La structure exploitant le langage GRAFCET graphique présente le gros avantage, pour la mise au point et la maintenance, de pouvoir suivre les évolutions du GRAFCET en mode dynamique directement sur l'écran de la console ou du moniteur.

C'est là le principal intérêt du langage GRAFCET graphique car son défaut majeur est d'être fort gourmand en capacité mémoire utilisée. Les langages non graphiques, de ce point de vue, sont nettement plus économes. Mais les capacités mémoire ont fait de tels progrès et leur prix a tellement baissé que ce n'est plus un argument de poids.

Les réceptivités et les actions sont programmées soit en langage littéral soit en langage à contacts soit en langage logigramme suivant les caractéristiques du langage de l'API.

On notera la lourdeur de la programmation dans le cas du langage GRAFCET graphique ALSPA et le manque de simplicité du logiciel ORPHEE d'APRIL.

Pour cette raison de nombreux automaticiens laissent de côté certains de ces langages GRAFCET graphiques trop sophistiqués et font appel à l'une des structures décrites précédemment.

Pour finir, il est important de noter que, quelle que soit la structure utilisée, **il faut que le programme soit bien documenté** (commentaires) et soit accompagné de l'ensemble des GRAFCET de niveau 3 (niveau réalisateur).

Au chapitre VIII, est donné un aperçu des possibilités de quelques logiciels de PGO (Programmation du GRAFCET Assistée par Ordinateur).

Certains de ces logiciels assurent la traduction en langage constructeur à partir d'une description graphique et littérale du GRAFCET par l'automaticien.

9. Tableau récapitulatif des essais faits sur divers API

Les structures étudiées ont été testées sur des API.

Les résultats sont consignés dans le tableau ci-dessous (fig. 7.46). Le numéro donne simplement l'ordre de préférence qui a été attribué à la structure.

Repérage des automates programmables industriels			
Rep	API	Sociétés	Langages
A	TSX	TELEMECANIQUE	PL7-1 (booléen littéral, GRAFCET littéral)
B	TSX	TELEMECANIQUE	PL7-2 (à contacts, programmeur cyclique, GRAFCET graphique)
C	TSX	TELEMECANIQUE	PL7-3 (à contacts, littéral, GRAFCET graphique, macro-étapes)
D	PB *	APRIL	assembleur, booléen, avance pas à pas, GRAFCET littéral
E	SMC *	APRIL	booléen, à contacts
F	C100	ALSPA CEGELEC	booléen, registre pas à pas
G	C200	ALSPA CEGELEC	booléen, à contacts, GRAFCET graphique
H	95U	SIEMENS	à contacts, logigramme
I	FPC 101	FESTO	AWL GRAFCET graphique
J	C200	ALSPA CEGELEC	P4 GRAFCET graphique
K	5000	APRIL	ORPHEE GRAFCET graphique
L	TSX	SCHNEIDER	PL7 Micro (à contacts, liste d'instructions)
M	TSX	SCHNEIDER	PL7 Junior (à contacts, liste d'instructions, littéral, GRAFCET graphique)
N	TSX	SCHNEIDER	PL7 Pro (à contacts, littéral, GRAFCET graphique)

STRUCTURES										
A P I	fonction mémoire		informatique alternative		registre pas à pas		sur mots		GRAFCET	
	monostab.	bistables	complète	réduite	programmat.	registre	littéral	à contacts	littéral	graphique
A	3	2	X	X	4				①	
B	4	3	X	X	5			①		2
C	5	4	6	3			2			①
D	5	3	4	X		2	X		①	
E	2	①	X	X		3				
F	3	4	5	X		①				2
G		2	5	①		3				4
H	2	①				3				4
I										
J										①
K										①
L	4	3						2		①
M	5	4		6			2	3		①
N	5	4		6			2	3		①

X : structure inexistante
 ① : structure recommandée
 * : ces automates programmables ne sont plus commercialisés mais ils sont encore en service dans les établissements techniques et dans de nombreuses entreprises.

Figure 7.46
Tableau récapitulatif
des tests des structures
d'implantation du GRAFCET.

Remarque

Il n'a pas été fait mention de la structure basée sur le **relais-maître** (*master-relay*).

Elle n'a pas été retenue car l'instruction relais-maître a pratiquement disparu des langages à contacts mis à part quelques-uns d'origine nord-américaine ou japonaise.

Comment se présente cette structure ?

Tant que l'expression logique de commande du relais-maître reste vraie (état 1), les instructions suivantes, écrites en langage assembleur (ou mnémonique), sont exécutables et obligatoirement exécutées.

Dans le cas contraire elles sont ignorées dans la lecture du programme.

En langage à contacts, ce sont les réseaux situés derrière le relais-maître qui sont exécutés ou ignorés selon l'état 1 ou 0 de celui-ci.

Cette structure est donc à rapprocher de la structure algorithmique. Le relais-maître joue le même rôle que le test de condition. La non-exécution des instructions ou des réseaux si le relais-maître est inactif est assimilable à un saut en avant.

Noter que la bobine Jump des langages à contacts a un comportement proche de celui du relais-maître.

10. Intégration des forçages de situation dans les structures d'implantation du GRAFCET

10.1. Rappels relatifs au forçage de situation

A. Objet du forçage de situation

Le forçage de situation (voir chapitre III.13), permet de modifier la situation d'un GRAFCET sans franchir de transition. L'ordre de forçage associé à une étape d'un GRAFCET hiérarchiquement supérieur a pour effet d'activer les étapes de la situation à atteindre et de désactiver les autres étapes du GRAFCET inférieur forcé.

B. Rappel sur la notation des forçages de situation

(figure 7.47)

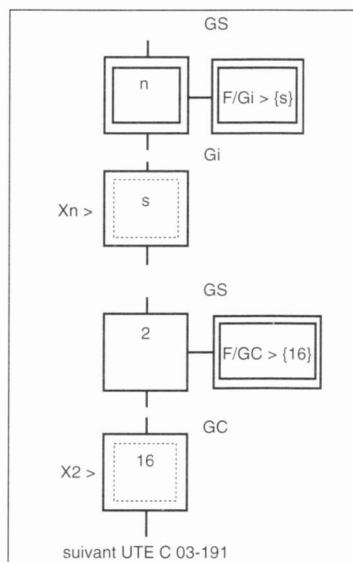


Figure 7.47
Notation UTE C 03-191
des forçages.

Si l'étape n du GRAFCET supérieur G_s est active, forçage du GRAFCET inférieur G_i sur la situation $S(G_i) = \{s\}$.

L'étape s , dans ce cas, est une étape initialisable si elle n'est pas déjà une étape initiale.

Le signal associé au symbole $>$ placé à gauche de l'étape s indique l'étape ou les étapes émettrices de l'ordre de forçage de situation.

L'ordre de forçage s'écrit à l'intérieur d'un double cadre relié à l'étape émettrice de cet ordre particulier si on retient l'écriture proposée dans la documentation UTE C 03-191. On constate que cette publication destinée à compléter la NF C 03-190 n'a pas encore été intégrée dans la réédition de 1995.

APPLICATION (fig 7.47)

Si l'étape 2 du GRAFCET GS est active, forçage du GRAFCET GC sur la situation $S(GC) = \{16\}$.

La condition X2 devient une condition supplémentaire d'activation de l'étape 16 et de désactivation des autres étapes du GRAFCET GC.

L'étape 16 est supposée être ici une étape initialisable.

C. Rappel sur les conditions liées au forçage de situation

Tant que l'ordre de forçage est émis par le GRAFCET supérieur, les évolutions du GRAFCET inférieur restent bloquées dans la situation forcée.

Pour obtenir ce résultat, il faut que l'activation par l'ordre de forçage des étapes correspondant à la situation forcée ainsi que la désactivation des autres étapes du GRAFCET forcé soient prioritaires sur toutes les autres conditions d'évolution.

Que se passe-t-il si le forçage n'est pas prioritaire ?

Les transitions associées aux étapes activées par le forçage sont validées.

Si la réceptivité associée à une transition validée est vraie, la transition devient franchissable et sera obligatoirement franchie avec pour conséquence l'oscillation des bits-images de l'étape amont et de l'étape aval à cette transition :

- forçage à 1 puis désactivation de l'étape amont,
- forçage à 0 puis activation de l'étape aval tous les deux cycles.

Pour éviter cette instabilité, il est donc indispensable, comme il a été dit, que les activations et les désactivations résultant du forçage de situation soient prioritaires sur les autres conditions d'évolution.

D. Rappel sur la variable de situation et sa notation

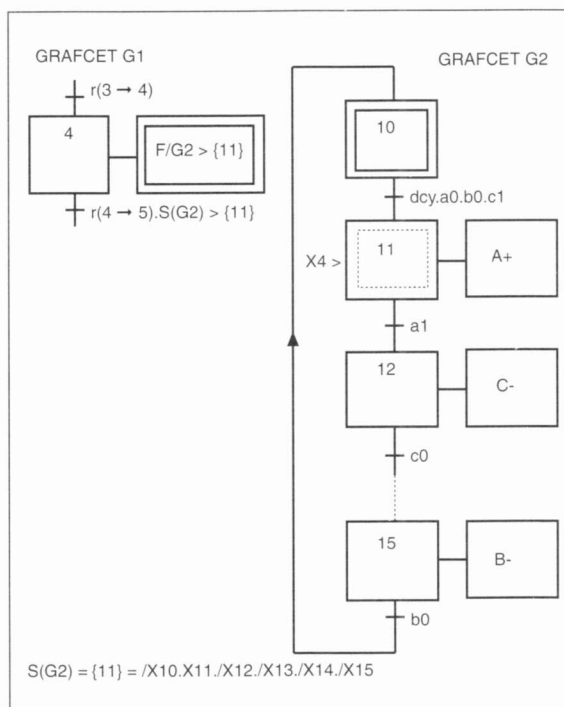


Figure 7.48
Transition franchissable
après forçage.

La variable de situation permet de vérifier si le forçage a été effectivement réalisé.

La variable de situation se note : $S(Gn) : \{\text{situation forcée}\}$. Cette variable doit être explicitée sous forme d'une expression logique.

Exemple (fig 7.48)

Le forçage de la situation $S(G2) = \{11\}$ valide la transition $11 \rightarrow 12$. Celle-ci devient franchissable et sera franchie si la réceptivité $a1$ devient vraie ($a1 = 1$).

Si on n'adopte pas une priorité à l'activation par le forçage de l'étape 11 et une priorité à la désactivation par le forçage de l'étape 12, on risque une oscillation des bits-images des étapes 11 et 12.

La variable de situation $S/G2 = \{11\}$ a pour expression logique :
 $/X10.X11./X12./X13./X14./X15$

Mise en place des priorités pour la transition litigieuse $11 \rightarrow 12$ avec une structure basée sur la fonction mémoire (sans calcul des fonctions transitions) :

$A(11) = X10.dcy.a0.b0.c1 + X4$ priorité au forçage à 1 $D(11) = X11.a1./X4$ $A(12) = X11.a1./X4$ priorité au forçage à 0 $D(12) = X12.c0 + X4$	en bistable
$X11 = X10.dcy.a0.b0.c1 + X4 + /a1.X11$ $X12 = (X11.a1 + /c0.X12)./X4$	en monostable

Il est intéressant de voir comment introduire le forçage de situation dans les différentes structures d'implantation du GRAFCET étudiées précédemment. Ce sera l'objet des paragraphes suivants.

Comme support d'étude, un ensemble de trois GRAFCET hiérarchisés comprenant un GRAFCET de Sécurité (GS), un GRAFCET de Conduite (GC) et un GRAFCET de Production Normale (GPN) a été retenu (fig. 7.49 et 7.50).

Les variables de situation ont volontairement été limitées aux signaux liés aux étapes forcées à 1 afin de ne pas alourdir la description des structures de forçage.

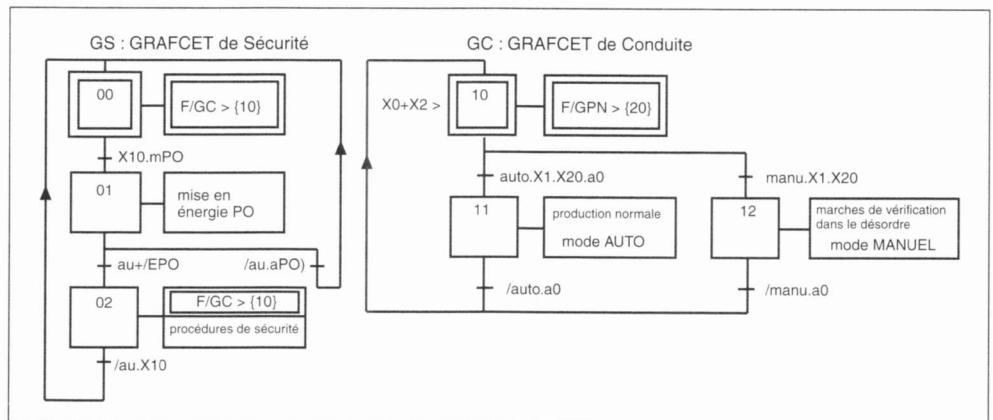


Figure 7.49
 GRAFCET
 de Sécurité GS
 et de Conduite GC, niveau 2.

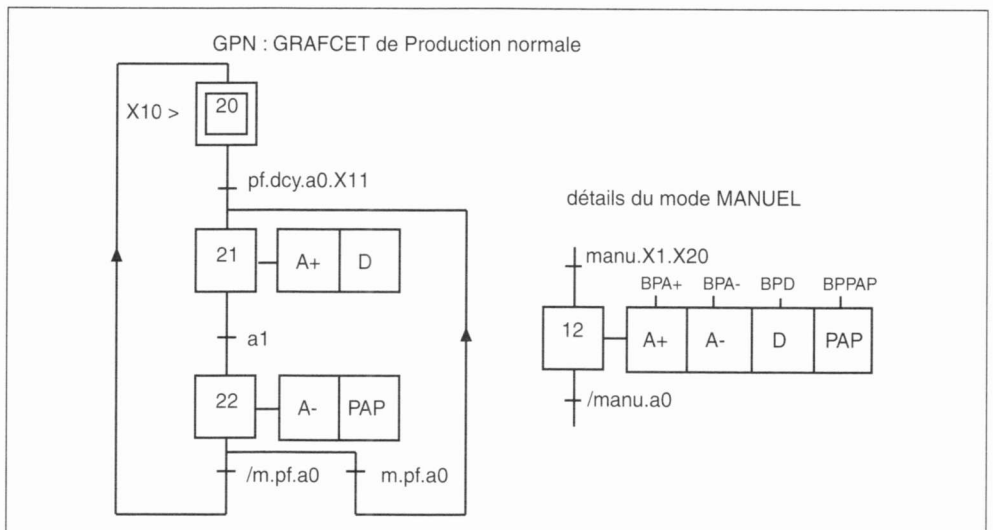


Figure 7.50
 GRAFCET
 de Production Normale GPN,
 niveau 2 et détails
 du mode manuel.

Il faut remarquer que les transitions litigieuses $10 \rightarrow 11$ et $10 \rightarrow 12$ après forçage de la situation $S(GC) = \{10\}$ par l'ordre associé à l'étape 2 ne peuvent pas être franchies puisque la condition X1 reste obligatoirement égale à 0 lorsque l'étape 2 est active.

La même remarque est valable en ce qui concerne la transition $20 \rightarrow 21$ verrouillée par la condition X11.

Le problème des priorités ne se pose donc pas pour les GRAFCET GC et GPN de l'exemple retenu.

Pour montrer comment s'effectue la mise en place des priorités si besoin est, deux cas seront étudiés : la solution avec X1 et X11 et la solution sans X1 ni X11 dans les réceptivités associées aux transitions $10 \rightarrow 11$, $10 \rightarrow 12$ et $20 \rightarrow 21$.

10.2. Cas de la structure basée sur la fonction mémoire

Calcul des fonctions transitions

$B100 = B10.auto.B1.B20.a0$	(ft ($10 \rightarrow 11$))	les étapes 10 à 12 et les étapes
$B101 = B10.manu.B1.B20$	(ft ($10 \rightarrow 12$))	20 à 22 ont pour bits-images
$B102 = B11./auto.a0$	(ft ($11 \rightarrow 10$))	respectivement les bits B10
$B103 = B12./manu.a0$	(ft ($12 \rightarrow 10$))	à B12 et les bits B20 à B22
$B104 = B20.pf.dcy.a0.B11$	(ft ($20 \rightarrow 21$))	
$B105 = B21.a1$	(ft ($21 \rightarrow 22$))	
$B106 = B22./m.pf.a0$	(ft ($22 \rightarrow 20$))	
$B107 = B22.m.pf.a0$	(ft ($22 \rightarrow 21$))	

Ordres de forçage émis par les étapes 2 et 10

$$F/GC > \{10\} = B0+B2 \qquad F/GPN > \{20\} = B10$$

Solution avec bits monostables (avec X1 et X11)

$$\begin{aligned} B10 &= \mathbf{B0+B2}+B102+B103+/B100./B101.B10 \\ B11 &= B100+/B102./\mathbf{B0./B2}.B11 \\ B12 &= B101+/B103./\mathbf{B0./B2}.B12 \quad (\text{pas de priorité au forçage de situation}) \\ B20 &= \mathbf{B10}+B106+/B104.B20 \\ B21 &= B104+B107+/B105./\mathbf{B10}.B21 \\ B22 &= B105+/B106./B107./\mathbf{B10}.B22 \end{aligned}$$

Solution avec bits bistables (avec X1 et X11)

$$\begin{aligned} A(10) &= \mathbf{B0+B2}+B102+B103 \\ D(10) &= B100+B101 \\ A(11) &= B100 \\ D(11) &= \mathbf{B0+B2}+B102 \\ A(12) &= B101 \quad (\text{pas de priorité au forçage de situation}) \\ D(12) &= \mathbf{B0+B2}+B103 \\ A(20) &= \mathbf{B10}+B106 \\ D(20) &= B104 \\ A(21) &= B104+B107 \\ D(21) &= \mathbf{B10}+B105 \\ A(22) &= B105 \\ D(22) &= \mathbf{B10}+B106+B107 \end{aligned}$$

Solution avec bits monostables (sans X1 ni X11 dans B100, B101 et B104)

$$\begin{aligned} B10 &= \mathbf{B0+B2}+B102+B103+/B100./B101.B10 \\ B11 &= (B100+/B102.B11)./ \mathbf{B0./B2} \\ B12 &= (B101+/B103.B12)./ \mathbf{B0./B2} \quad (\text{priorité au forçage de situation}) \\ B20 &= \mathbf{B10}+B106+/B104.B20 \\ B21 &= (B104+B107+/B105.B21)/ \mathbf{B10} \\ B22 &= (B105+/B106./B107.B22)/ \mathbf{B10} \end{aligned}$$

Solution avec bits bistables (sans X1 ni X11 dans B100, B101 et B104)

$A(10) = B0+B2+B102+B103$

$D(10) = (B101+B100)./B0./B2$

$A(11) = B100./B0./B2$

(priorité au forçage de situation)

$D(11) = B102+B0+B2$

$A(12) = B101./B0./B2$

$D(12) = B103+B0+B2$

$A(20) = B10+B106$

$D(20) = B104./B10$

$A(21) = (B104+B107)./B10$

$D(21) = B105+B10$

$A(22) = B105./B10$

$D(22) = B106+B107+B10$

En conclusion, ce qui importe c'est d'éviter l'instabilité des évolutions et non la manière de le faire.



APPLICATION 1 : implantation en monostable dans les PB (APRIL)

GRAF CET de Sécurité GS et de Conduite GC (fig.7.51)

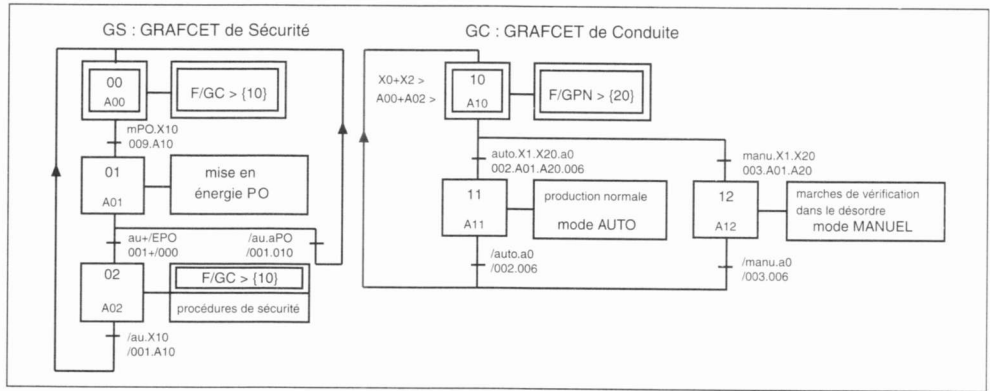


Figure 7.51
GRAF CET
de Sécurité GS
et de Conduite GC, niveau 3
(PB APRIL).

GRAF CET de Production Normale GPN (fig.7.52)

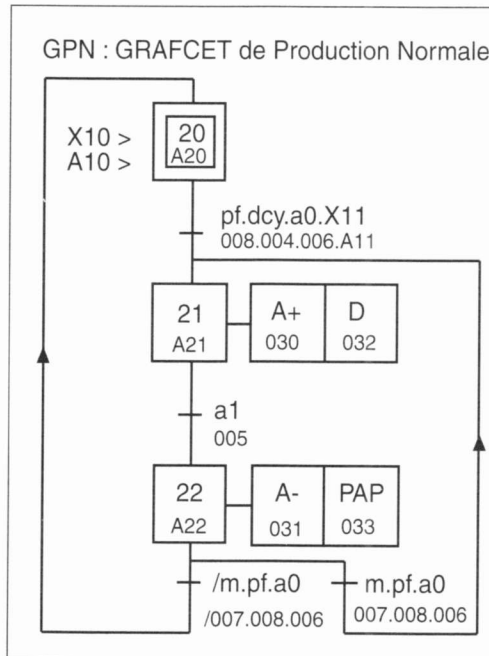


Figure 7.52
GRAF CET de Production
Normale GPN, niveau 3
(PB APRIL).

Adressage retenu

Étapes et fonctions transition

étapes 0 à 2 : bits A00 à A02

étapes 10 à 12 : bits A10 à A12

étapes 20 à 22 : bits A20 à A22

ft(10 → 11) : B10 ft(20 → 21) : B14

ft(10 → 12) : B11 ft(21 → 22) : B15

ft(11 → 10) : B12 ft(22 → 20) : B16

ft(12 → 10) : B13 ft(22 → 21) : B17

Entrées

- EPO : 000
- au : 001
- auto : 002
- manu : 003
- dcy : 004
- a1 : 005
- a0 : 006
- m : 007
- pf : 008
- mPO : 009
- aPO : 010

Sorties

- A+ : 030
- A- : 031
- D : 032
- PAP : 033

Expressions logiques à programmer

$A10 = A00 + A02 + B12 + /B10 . A10$
 $A11 = B10 + /B12 . /A00 . /A02 . A11$
 $A12 = B11 + /B13 . /A00 . /A02 . A12$
 $A20 = A10 + B16 + /B14 . A20$
 $A21 = B14 + /B15 . /A10 . A21$
 $A21 = B14 + /B15 . /A10 . A21$
 $A22 = B15 + /B16 . /B17 . /A10 . A22$



APPLICATION 2 : implantation en monostable dans les SMC (APRIL)

GRAF CET de Sécurité GS et de Conduite GC (fig.7.53)

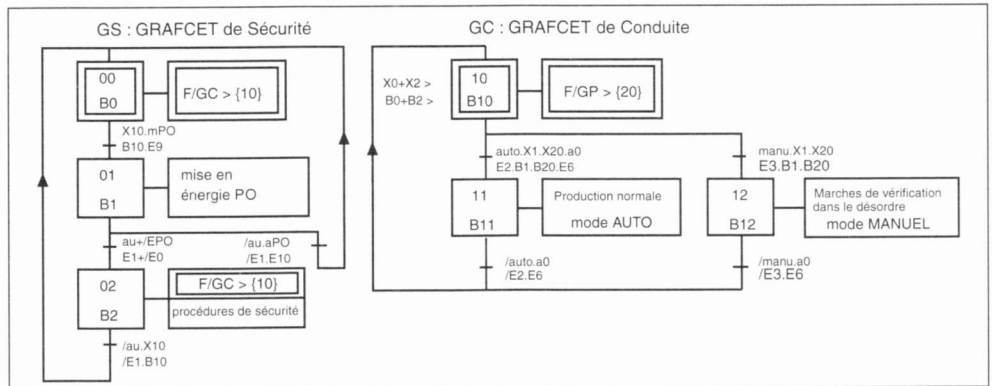


Figure 7.53
GRAF CET
de Sécurité GS
et de Conduite GC, niveau 3
(SMC APRIL).

GRAF CET de Production Normale GPN (fig.7.54)

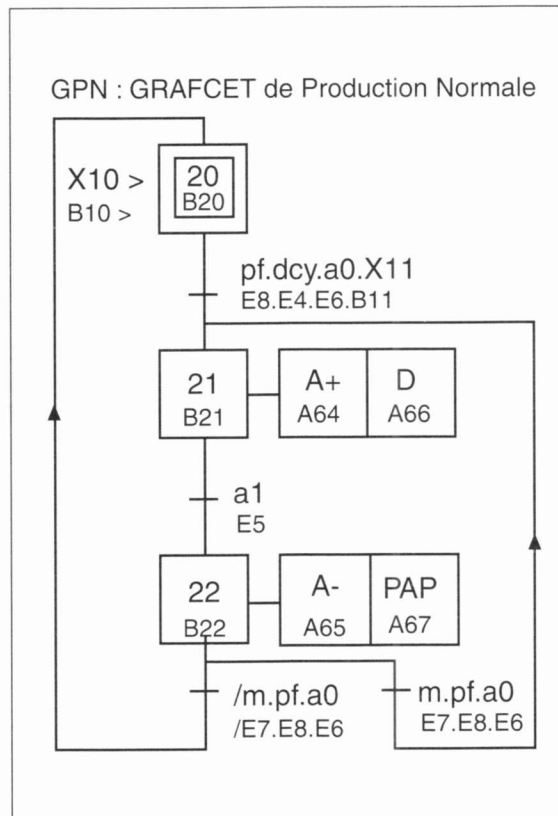


Figure 7.54
GRAF CET
de Production Normale GPN,
niveau 3 (SMC APRIL).

Adressage retenu Étapes

étapes 0 à 2 : bits B0 à B2
 étapes 10 à 12 : bits B10 à B12
 étapes 20 à 22 : bits B20 à B22

Fonctions transitions

ft(10 → 11) : bit B100
 ft(10 → 12) : bit B101
 ft(11 → 10) : bit B102
 ft(12 → 10) : bit B103
 ft(20 → 21) : bit B104
 ft(21 → 22) : bit B105
 ft(22 → 20) : bit B106
 ft(22 → 21) : bit B107

Entrées

EPO : E0
 au : E1
 auto : E2
 manu : E3
 dcy : E4
 a1 : E5
 a0 : E6
 m : E7
 pf : E8
 mPO : E9
 aPO : E10

Sorties

A+ : A64
 A- : A65
 D : A66
 PAP : A67

Expressions logiques à programmer

```

B10 = B0+B2+B102+B103+/B100.B10
B11 = B100+/B102./B0./B2.B11
B12 = B101+/B103./B0./B2.B12
B20 = B10+B106+/B104.B20
B21 = B104+B107+/B105./B10.B21
B22 = B105+/B106./B107./B10.B22
    
```

APPLICATION 3 : implantation en monostable en PL7-2 et PL7-3 (TSX Schneider Télémécanique)

GRAF CET de Sécurité GS et de Conduite GC (fig. 7.55)

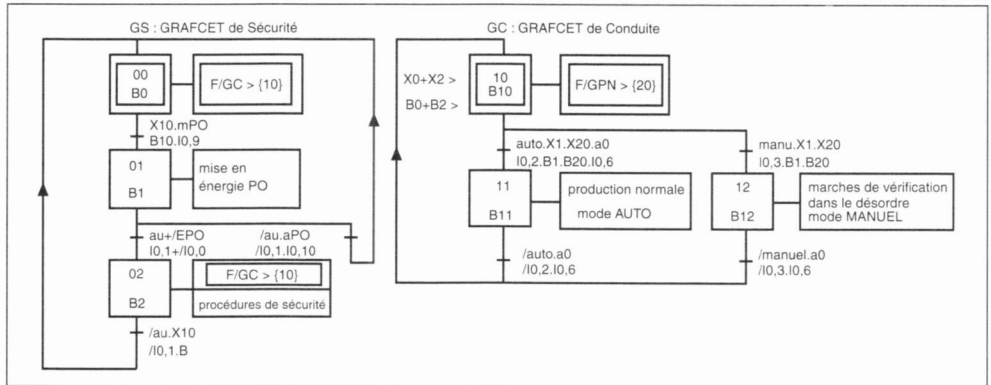


Figure 7.55
GRAF CET
de Sécurité GS
et de Conduite GC, niveau 3
(TSX).

GRAF CET de Production Normale GPN (fig. 7.56)

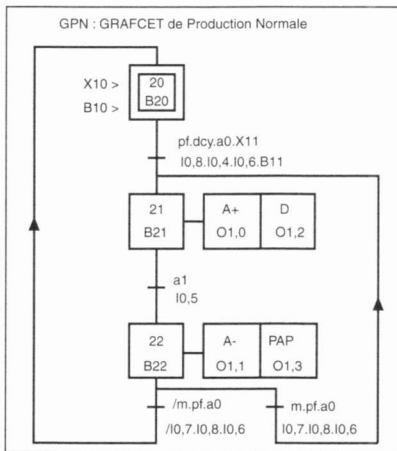


Figure 7.56
GRAF CET de Production
Normale GPN, niveau 3
(TSX).

Adressage retenu

Étapes

étapes 0 à 2 : bits B0 à B2
 étapes 10 à 12 : bits B10 à B12
 étapes 20 à 22 : bits B20 à B22

Fonctions transitions

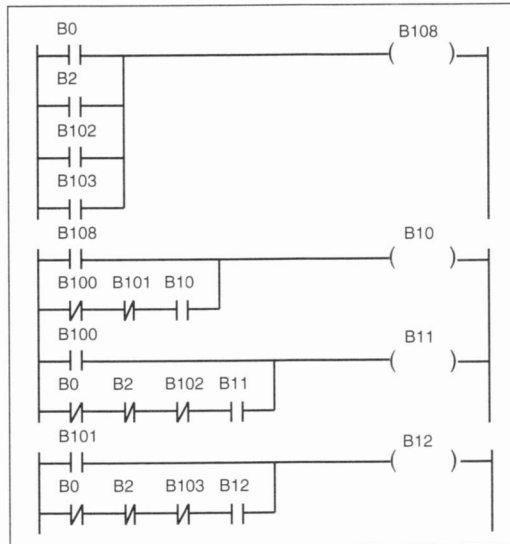
ft(10 → 11) : bit B100
 ft(10 → 12) : bit B101
 ft(11 → 10) : bit B102
 ft(12 → 10) : bit B103
 ft(20 → 21) : bit B104
 ft(21 → 22) : bit B105
 ft(22 → 20) : bit B106
 ft(22 → 21) : bit B107

Entrées

EPO : I0,0
 au : I0,1
 auto : I0,2
 manu : I0,3
 dcy : I0,4
 a1 : I0,5
 a0 : I0,6
 m : I0,7
 pf : I0,8
 mPO : I0,9
 aPO : I0,10

Sorties

A+ : O1,0
 A- : O1,1
 D : O1,2
 PAP : O1,3



Programme en langage à contacts PL7-2 et PL7-3 du GRAFCET GC (fig. 7.57)

Figure 7.57
Programme en langage à contacts limité au GRAFCET GC.



APPLICATION 4 : implantation en bistable dans les PB (APRIL)

Même adressage qu'à l'application 1 Expressions logiques à programmer

A(10) : **A00+A02+006(A11./002+A12./003)** = MU A10
D(10) : A10.A01.A20(002.006+003) = MZ A10
A(11) : A10.002.A01.A20.006 = MU A11
D(11) : **A00+A02+A11./002.006** = MZ A11
A(12) : A10.003.A01.A20 = MU A12
D(12) : **A00+A02+A12./003.006** = MZ A12
A(20) : **A10+A22./007.006** = MU A20
D(20) : A20.008.004.006.A11 = MZ A20
A(21) : A20..008.006.(A11.004+A22.007) = MU A21
D(21) : **A10+A21.005** = MZ A21
A(22) : A21.005 = MU A22
D(22) : **A10+A22.008.006** = MZ A22



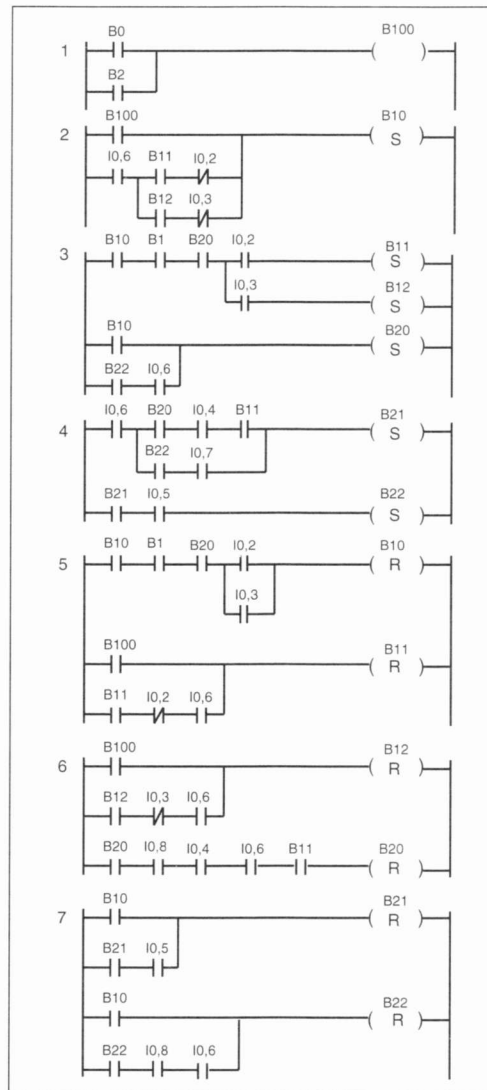
APPLICATION 5 : implantation en bistable dans les SMC (APRIL)

Même adressage qu'à l'application 2 Expressions logiques à programmer

R10 = **R0+R2+B100.(R11./E2+R12./E3)**
/R10 = R10.R1.R20 (E2.E6+E3)
R11 = R10.E2.R1.R20.E6
/R11 = **R0+R2+R11./E2.B100**
R12 = R10.E3.R1
/R12 = **R0+R2+R12.E3.B100**
R20 = **R10+R22./E7.E8.E6**
/R20 = R20.E4.E6.R11
R21 = E8.E6 (R20.E4.R11+R22.E7)
/R21 = **R10+R21.E5**
R22 = R21.E5
/R22 = **R10+R22.E6**



APPLICATION 6 : implantation en bistable en PL7-2 et PL7-3 (TSX Schneider Télémécanique)



Même adressage
qu'à l'application 3

Programme en langage à contacts
PL7-2 et PL7-3 (fig.7.58)

Figure 7.58
Programme en langage
à contacts
(TSX Télémécanique).

10.3. Cas de la structure informatique

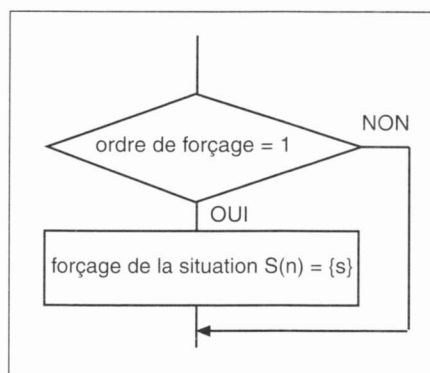


Figure 7.59
Algorithme de forçage.

Le forçage de situation fait appel à une structure indépendante du même type que celle gérant les évolutions du GRAFCET.

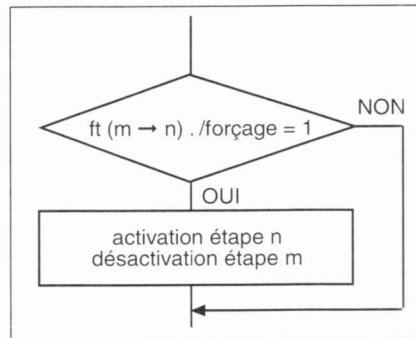
Algorithme (fig. 7.59)

SI l'ordre de forçage est vrai (état 1)
ALORS activer les étapes correspondant à la situation forcée et désactiver les autres étapes du GRAFCET à forcer
SINON aller au test suivant

Cette structure ne résout pas totalement le problème du forçage. On a vu que le forçage doit être prioritaire sur les autres ordres assurant l'évolution normale.

Ce résultat sera obtenu, si nécessaire, en verrouillant les fonctions transitions des autres structures par l'ordre de forçage.

Algorithme avec verrouillage de la transition (fig. 7.60)



SI la fonction transition est égale à 1
ET SI l'ordre de forçage n'est pas émis
ALORS activer l'étape suivante (étape n) et désactiver l'étape précédente (étape m)
SINON aller au test suivant

Noter qu'il ne sert à rien de verrouiller systématiquement toutes les fonctions transitions si cela n'est pas nécessaire. Retour sur le GRAFCET support GC à titre d'exemple (fig. 7.61).

L'étude est limitée à la transition 10 → 11. Que se passe-t-il si la condition X1 n'est pas introduite dans l'expression logique de la fonction transition 10 → 11 ?

Si un ordre de forçage est émis par l'étape 0 ou l'étape 2, après forçage, la réponse aux deux tests sera OUI si auto.X20.a0 = 1.

L'étape 10 sera alternativement activée puis désactivée tant que l'ordre de forçage sera maintenu. De même, l'étape 11 sera alternativement désactivée puis activée.

Figure 7.60
Verrouillage de la transition.

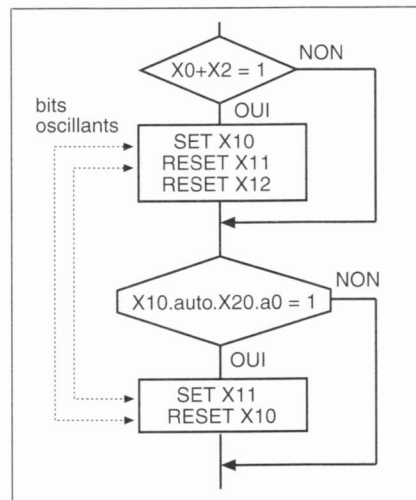


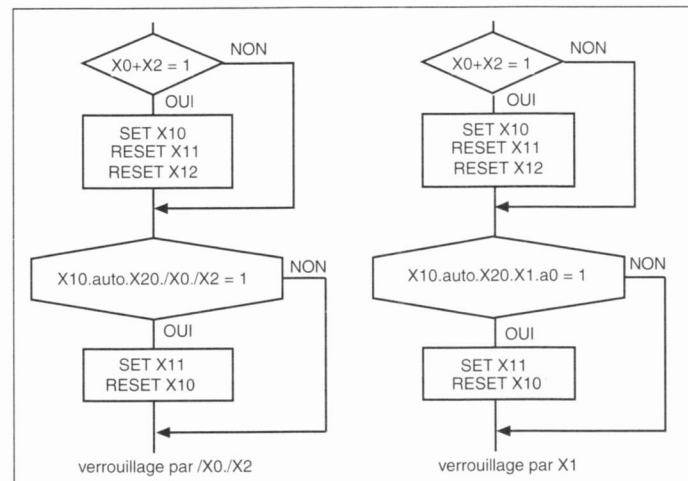
Figure 7.61
Instabilité sans verrouillage.

Il en résultera une instabilité des bits internes correspondants.

Pour éviter ce phénomène, la solution est de verrouiller la fonction transition litigieuse par le complément de l'ordre de forçage pour qu'elle ne soit plus vraie lorsque cet ordre est émis.

Dans le cas présent, cela donne : $X10.auto.X20./X0./X2$.

Comparaison entre les deux façons de verrouiller la transition litigieuse (fig. 7.62)



Comme les situations $S(GC) = \{0\}$ et $S(GC) = \{1\}$ ne peuvent coexister, le verrouillage de la transition 10 → 11 par X1 résout ici le problème.

Il n'est pas nécessaire ici d'avoir recours au verrouillage par le complément de l'ordre de forçage.

Figure 7.62
Verrouillage de la transition litigieuse.

Conclusion

Vérifier si les réceptivités associées aux transitions validées après forçage sont vraies. Si c'est le cas, il faut procéder à un verrouillage des fonctions transitions correspondantes par le complément de l'ordre de forçage.



APPLICATION 1 : implantation en langage assembleur dans les PB (APRIL)

$\neg(A00+A02) = \neg A00.\neg A02$

ce qui entraîne :

SI/ A00
SI/ A02
SAUT.....
MU A10
MZ A11
MZ A12.....
.....

Algorithme du forçage (fig. 7.63)

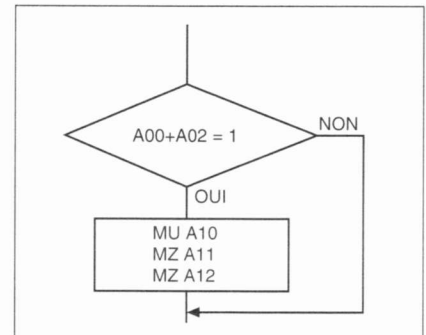


Figure 7.63

Algorithme de forçage (PB APRIL).

Programme en langage assembleur (PB APRIL)

Même adressage que précédemment.

0C30 DE	A00	mise à 0 des bits A00
0C31 MZ	BFF	à BFF (256 bits)
0C32 DE	030	mise à 0 des sorties 030
0C33 MZ	03F	à 03F (16 sorties, carte 03)
0C34 MU	A00	activation étape 0 (X0 = 1)
0C35 MU	A10	activation étape 10 (X10 = 1)
0C36 MU	A20	activation étape 20 (X20 = 1)
.....		
..... SI	A10	test si étape 10 active (X10 = 1)
..... SI	002	test si auto = 1
..... SI	A01	test si étape 1 active (X1 = 1)
..... SI	A20	test si étape 20 active (X20 = 1)
..... SI	006	test si a0 = 1
..... ET	BFF	calcul $ft(10 \rightarrow 11)$ (X10.auto.X1.X20.a0)
..... SI/	BFF	test si X10.auto.X20.X1.a0 = 0, SI OUI saut relatif
..... SAUT	A03	en avant de 3 lignes, SI NON aller à la ligne suivante
..... MU	A11	activation étape 11 (X11 = 1)
..... MZ	A10	désactivation étape 10 (X10 = 0)
.....		etc.
module de forçage		
..... SI/	A00	test si l'étape 0 est inactive (pas d'ordre de forçage)
..... SI/	A02	test si l'étape 2 est inactive (pas d'ordre de forçage)
..... SAUT	A04	SI OUI saut en avant de 4 lignes, SI NON effectuer le forçage
..... MU	A10	activation étape 10 (X10 = 1)
..... MZ	A11	désactivation étape 11 (X11 = 0)
..... MZ	A12	désactivation étape 12 (X12 = 0)
.....	
..... SAUT	C37	saut inconditionnel en 0C37

APPLICATION 2 : implantation en langage littéral PL7-3
(TSX Schneider Télémécanique)

Exploitation de la structure alternative réduite du langage PL7-3 (fig. 7.64) :

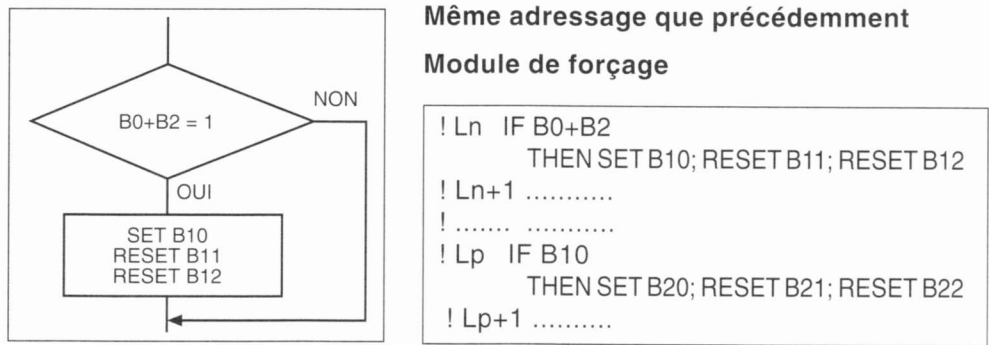


Figure 7.64
Algorithme de forçage
(TSX).

10.4. Cas de la structure exploitant les registres pas à pas

Les GRAFCET GS, GC et GPN du GRAFCET support sont respectivement matérialisés par les programmeurs cycliques D0, D1 et D2.

Si l'ordre de forçage est émis par le pas 0 (X0) ou par le pas 2 (X2) de D0, le programmeur D1 est ramené sur le pas 0 (X10) correspondant à la situation forcée $S(GC) = \{10\}$.

Si l'ordre de forçage est émis par le pas 0 (X10) de D1, le programmeur D2 est ramené sur le pas 0 (X20) correspondant à la situation forcée $S(GPN) = \{20\}$.

APPLICATION 1 : cas du programmeur cyclique du PL7-2
(TSX Schneider Télémécanique)

Réseau de forçage (fig. 7.65)

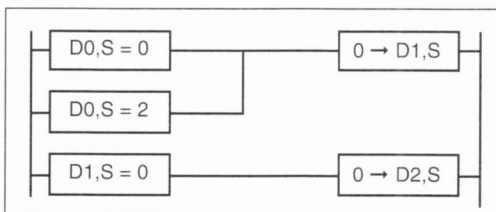


Figure 7.65
Réseau de forçage.

SI le programmeur D0 (GRAFCET GS) est sur le pas 0 (test $D0, S = 0$)($X0 = 1$) ou sur le pas 2 (test $D0, S = 2$)($X2 = 1$),

ALORS mettre le programmeur D1 sur le pas 0 (transfert $0 \rightarrow D1, S$) ($X10 = 1$).

SI le programmeur D1 (GRAFCET GC) est sur le pas 0 (test $D1, S = 0$)($X10 = 1$),

ALORS mettre le programmeur D2 sur le pas 0 (transfert $0 \rightarrow D2, S$) ($X20 = 1$).

APPLICATION 2 : implantation dans les registres du C100
(ALSPA CEGELEC)

Les GRAFCET GS, GC et GPN seront respectivement matérialisés par les registres G00, G01 et G02.

Le forçage des GRAFCET GC et GPN se fera en testant respectivement les pas 00 et 02 du registre G00 et le pas 00 du registre G01.

Il faudra donc programmer les expressions logiques suivantes compte tenu du langage du C100 :

G0000+G0002+G0101./auto.a0	
+G0102./manu.a0=G0100	forçage et activation étape 10
G0100.auto.G0001.G0200.a0=G0101	activation étape 11
G0100.manu.G0001.G0002=G0102	activation étape 12
G0100+G0202./m.pf.a0=G0200	forçage et activation étape 20
pf.a0(G0200.dcy.G0101+G0202.m)=G0201	activation étape 21
G0201.a1=G0202	activation étape 22

10.5. Cas de la structure exploitant le langage calcul sur mots

Les GRAFCET GS, GC et GPN sont matérialisés respectivement par les mots W0, W1 et W2.

Réseau de forçage (fig. 7.66)

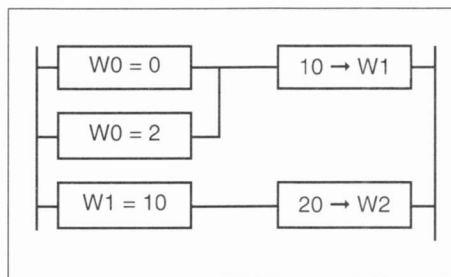


Figure 7.66
Réseau de forçage.

Algorithme

```

SI [W0] = 0 OU [W0] = 2
(X0+X2 = 1)
ALORS mettre 10 dans le mot W1
(X10 = 1, X11 = 0, X12 = 0)
SI [W1] = 10
(X10 = 1)
ALORS mettre 20 dans le mot W2
(X20 = 1, X21 = 0, X22 = 0)

```

10.6. Cas des langages GRAFCET littéraux

Pour que le forçage de situation soit possible, il faut que le langage dispose d'instructions appropriées.



APPLICATION 1 : implantation dans le langage GRAFCET littéral des PB (APRIL)

Ce langage possède des instructions spécifiques aux forçages, à savoir :

- FORU pour le forçage de bit à 1,
- FORZ pour le forçage à 0 de bits successifs.

Exemple

Forçage du GRAFCET 4 (étapes 40 à 48)
sur la situation S(G4) = {40} par l'étape 10

S	A10	test si l'étape 10 de forçage est active
FORU	A40	activation de l'étape 40
SI	A10	test si l'étape 10 de forçage est active
FORZ	A41	désactivation de l'étape 41
	A48	jusqu'à l'étape 48

Module de forçage du GRAFCET GC de l'application :

(F/GC > {10} par X0 ou X2)

0Cn	SI	A00	test si étape 0 active (X0 = 1?)
0Cn+1	SI	A02	test si étape 2 active (X2 = 1?)
0Cn+2	OU	B00	calcul A00+A02
0Cn+3	SI	B00	test si A00+A02 = 1? (ordre de forçage)
0Cn+4	FORU	0A10	forçage à 1 de l'étape 10 (X10 = 1)
0Cn+6	SI	B00	test si A00+A02 = 1? (ordre de forçage)
0Cn+7	FORZ	0A11	forçage à 0 étape 11 (X11 = 0)
0Cn+9		0A12	et étape 12 (X12 = 0)

Remarque

Les instructions de forçage FORU et FORZ sont logées dans deux mots de 16 bits et occupent ainsi deux lignes dans la mémoire du programme.



APPLICATION 2 : implantation en langage PL7-1 (TSX Schneider Télémécanique)

Programme de forçage en traitement préliminaire et commentaires

S0000 L	SY0	test bit système de démarrage à froid et début du traitement préliminaire
S0001 O	SY1	OU test bit système de démarrage à chaud
S0002 O	B0	OU bit assurant le maintien de SY9 pendant la durée de la temporisation T0
S0003 S	B0	mise à 1 du bit B0
S0004 S	SY21	initialisation des GRAFCET et figeage en situation initiale
S0005 =	SY9	mise à 0 des sorties
S0006 =	T0	lancement d'une temporisation de 5 secondes
S0007 L	T0	test si temporisation terminée
S0008 R	B0	mise à 0 du bit B0 → fin figeage situation initiale et fin blocage des sorties
S0009 L	X0	si étape 0 active
S0010 O	X2	OU si étape 2 active
S0011 AN	X10	ET si étape 10 inactive soit (X0+X2)/X10 (déverrouillage de SY23)
S0012 R	X11	désactivation étape 11
S0013 R	X12	désactivation étape 12
S0014 S	X10	activation étape 10
S0015 S	SY23	validation du forçage du GRAFCET GC
S0016 L	X10	si étape 10 active
S0017 AN	X20	ET si étape 20 inactive soit X10./X20 (déverrouillage de SY23)
S0018 R	X21	désactivation étape 21
S0019 R	X22	désactivation étape 22
S0020 S	X20	activation étape 20
S0021 S	SY23	validation du forçage du GRAFCET GPN
S0022 =*=	00	fin du traitement préliminaire et début du traitement séquentiel, test étape initiale 0
S0023	

La programmation du forçage se fait en traitement préliminaire (tête de programme). Dans les langages PL7-1, PL7-2 et PL7-3, le forçage est appelé pré-positionnement. Il s'opère par l'intermédiaire des instructions SET et RESET des bits Xi des étapes et du bit système SY23 qui assure leur validation.

La remise à 0 de SY23 est réalisée automatiquement par le système au cycle suivant à condition que le SET de SY23 ne soit plus commandé. Si le bit SY23 est maintenu à 1 par un ordre interne (étape) ou externe (capteur) l'ensemble des GRAFCET est figé sur la situation courante.

Noter comment le bit SY23 est déverrouillé respectivement aux lignes S0011 et S0017 par le complément de l'étape forcée à 1.

Sans cette précaution, le bit système SY23 est maintenu à 1 tant que l'ordre de forçage est émis avec pour conséquence le figeage de l'ensemble des GRAFCET.

Le système est bloqué puisqu'on ne peut plus quitter l'étape émettrice de l'un ou l'autre des ordres de forçage.

10.7. Cas des langages GRAFCET graphiques

Ces langages respectent les concepts du GRAFCET mais ignorent encore le concept de forçage de situation pour sa part non encore normalisé. Par contre, ils comportent généralement des instructions (bits systèmes, blocs fonctions, etc.) qui permettent d'appliquer le concept de forçage.

L'intégration du concept de forçage est maintenant réalisée dans quelques logiciels propriétaires (langage associé à un type d'API) et dans des logiciels de PGAO.



APPLICATION 1 : implantation en langage PL7-2 et PL7-3 V4 (TSX Schneider Télémécanique)

Ces deux langages possèdent trois bits systèmes pour effectuer différents cas de forçage :

– **bit système SY21 :**

sa mise à 1 (SET) assure le forçage en situation initiale de l'ensemble des GRAFCET,

– **bit système SY22 :**

sa mise à 1 (SET) assure le forçage en situation vide de l'ensemble des GRAFCET,

– **bit système SY23 :**

sa mise à 1 (SET) valide le forçage d'un GRAFCET par les instructions SET et RESET. Son maintien à 1 fige l'ensemble des GRAFCET sur la situation courante.

Ces trois bits systèmes doivent être exploités uniquement en traitement préliminaire.

Les bits SY21, SY22 et SY23 sont remis automatiquement à 0 par le système en fin de traitement séquentiel à condition qu'aucune variable ne les maintienne à 1.

Il faut étudier de près le cas du bit SY23. Si ce bit est maintenu à 1, l'évolution de l'ensemble des GRAFCET est bloquée. Il est donc indispensable d'assurer la remise à 0 de SY23 par le programme lui-même.

Application aux GRAFCET support GC et GPN

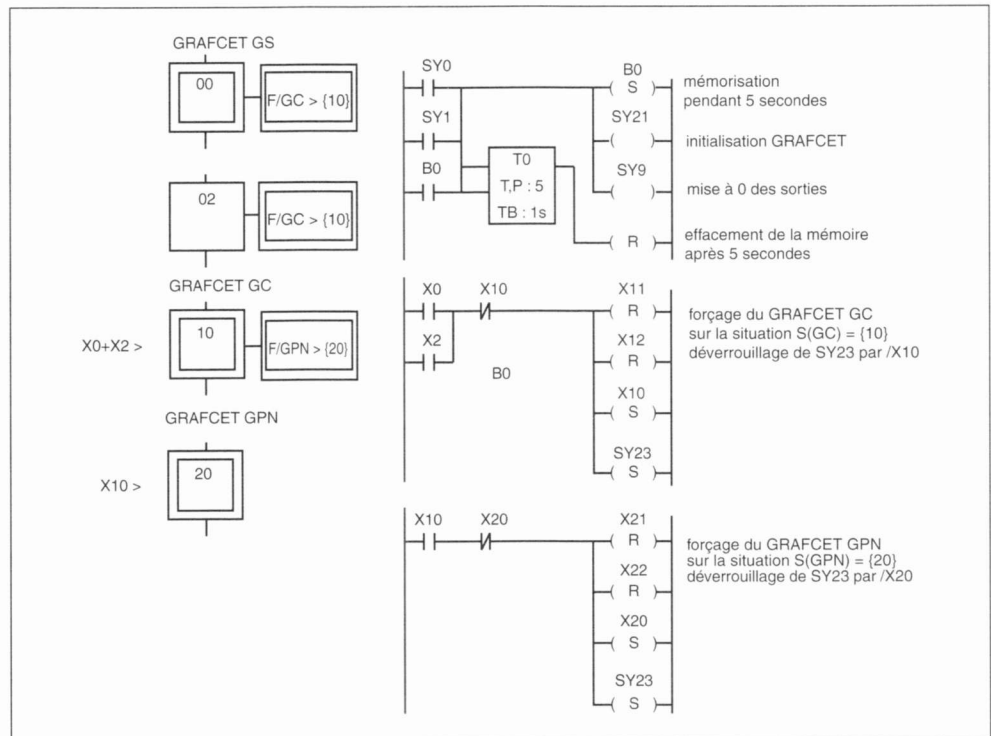


Figure 7.67
Traitement préliminaire
du forçage de situation.

Le déverrouillage de $SY23$ est obtenu par le complément des étapes forcées à 1 : $/X10$ et $/X20$.



APPLICATION 2 : implantation en langage PL7-3 V5 (TSX Télémécanique)

Cette nouvelle version V5 du langage PL7-3 dispose, sous forme d'extension, de Blocs Fonctions Optionnels (OFB) pour résoudre certains des problèmes rencontrés en automatisation :

- OFB de diagnostic du process,
- OFB de communication,
- OFB de commande d'axes,
- OFB de régulation,
- **OFB pour le GRAFCET (OFB MSIT).**

L'OFB MSIT de mise en situation du GRAFCET permet de réaliser le forçage ou le figeage d'un GRAFCET, la sauvegarde et la restitution d'une situation.

Paramètres de l'OFB MSIT

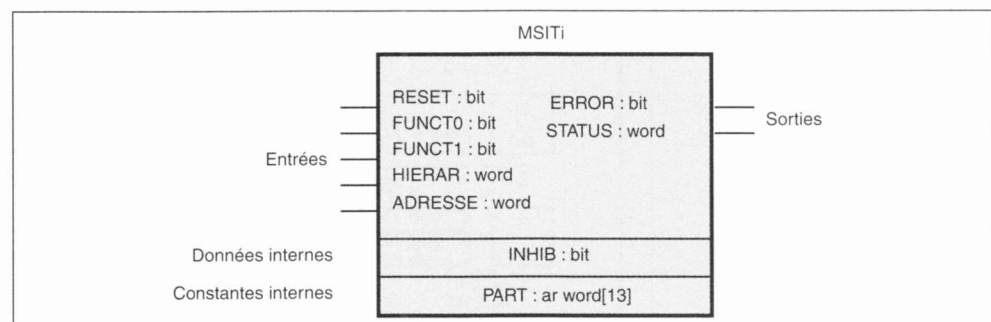


Figure 7.68
Paramètres de l'OFB MSIT.

Renseignements sur les paramètres bits et mots d'entrées-sorties de l'OFB MSIT

RESET : ce bit d'entrée permet de réinitialiser le flag "forçage en cours" de l'OFB

FUNCT0 : ce bit d'entrée permet de choisir le type de fonction à réaliser :

FUNCT0=0 : forçage ou libération du GRAFCET,

FUNCT0=1 : sauvegarde ou restitution d'une situation.

FUNCT1 : ce bit d'entrée permet de choisir la fonction à réaliser parmi le type défini par FUNCT0 :

si FUNCT0=0 et FUNCT1=0 alors libération du GRAFCET,

si FUNCT0=0 et FUNCT1=1 alors forçage du GRAFCET,

si FUNCT0=1 et FUNCT1=0 alors sauvegarde de situation,

si FUNCT0=1 et FUNCT1=1 alors restitution de la situation sauvegardée.

HIERAR : ce mot d'entrée permet de communiquer à l'OFB le niveau de hiérarchie du GRAFCET émettant l'ordre de forçage (niveau le plus élevé : 0, le plus bas 255).

ADRESS : ce mot d'entrée indique à l'OFB l'adresse à partir de laquelle sera mémorisée :

soit la situation de forçage (mots constants CWi),

soit la situation sauvegardée (mots internes Wi).

ERROR : ce bit de sortie signale si la fonction choisie s'est déroulée correctement :

0 : aucune erreur,

1 : un ou plusieurs défauts constatés.

STATUS : les bits de ce mot de sortie indiquent le type de défaut si ERROR=1.

La programmation de l'OFB MSIT s'effectue soit en langage à contacts soit en langage littéral.

Les OFB MSIT sont repérés par un numéro : MSIT0, MSIT1,...

Programmation de l'OFB MSIT en langage à contacts (fig. 7.69)

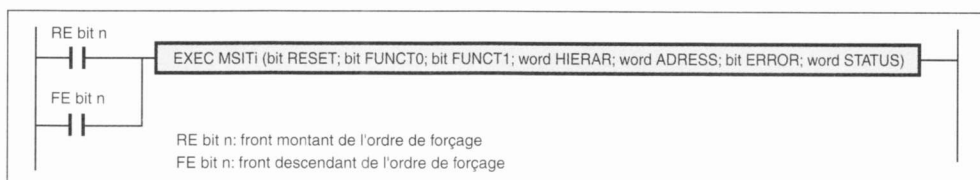


Figure 7.69
Programmation de l'OFB MSIT en langage à contacts (PL7-3 V5 Télémécanique).

Programmation en langage littéral

! IF RE(bit n) + FE(bit n) THEN EXEC MSITi (bit RESET; bit FUNCT0; bit FUNCT1; word HIERAR; word ADRESS; bit ERROR; word STATUS)

Application à l'ensemble des GRAFCET supports GS, GC et GPN (fig. 7.70 et 7.71)

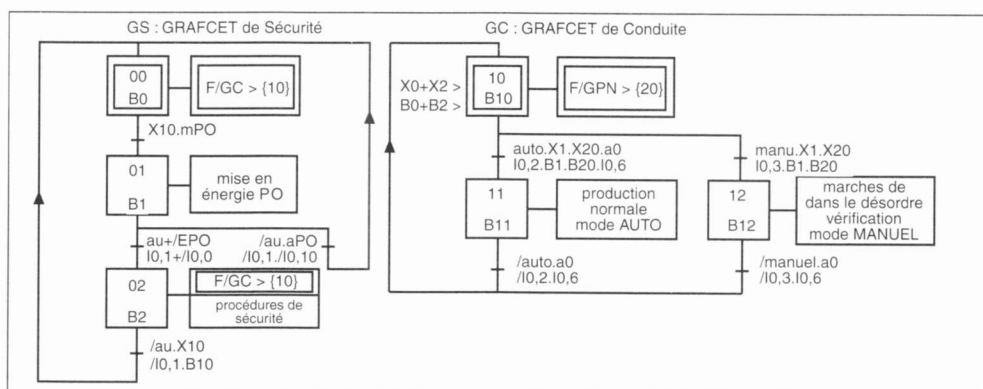
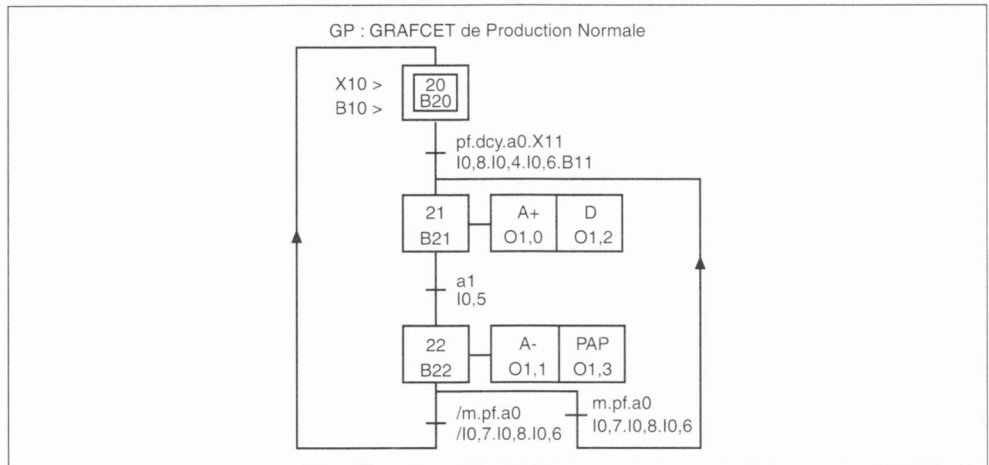


Figure 7.70
GRAFCET de Sécurité GS et de Conduite GC, niveau 3 (TSX Télémécanique).

Figure 7.71
 GRAFCET de Production
 Normale GPN, niveau 3
 (TSX).



Programme du forçage en langage littéral

Deux OFB MSIT sont nécessaires :

- MSIT0 pour forcer le GRAFCET GC à partir du GRAFCET GS (ordre X0+X2) (hiérarchie 0),
- MSIT1 pour forcer le GRAFCET GPN à partir de GC (ordre X10) (hiérarchie 1).

```

< Calcul des bits internes pour exécution des RE et FE
! X0+X2 → B20
! X10 → B21
< Forçage du GRAFCET GC
! IF RE(B20) + FE(B20) THEN EXEC MSIT0 (B0; B1; B2; 0; CW20; B3; W20)
< Libération du GRAFCET GC
! IF RE(/B20) + FE(/B20) THEN EXEC MSIT0 (B0; B1; B2; 0; CW20; B3; W20)
< Forçage du GRAFCET GPN
! IF RE(B21) + FE(B21) THEN EXEC MSIT1 (B4; B5; B6; 1; CW23; B7; W21)
< Libération du GRAFCET GPN
! IF RE(/B21) + FE(/B21) THEN EXEC MSIT1 (B4; B5; B6; 1; CW23; B7; W21)
  
```

Programme de la sauvegarde et de la restitution des situations en langage littéral

Deux OFB MSIT sont nécessaires :

- MSIT2 pour sauvegarder et restituer la situation du GRAFCET GC à partir du GRAFCET GS (ordre X0+X2) (hiérarchie 0),
- MSIT3 pour sauvegarder et restituer la situation du GRAFCET GPN à partir du GRAFCET GC (ordre X10) (hiérarchie 1).

```

< Calcul des bits internes pour exécution des RE et FE
! X0+X2 → B22
! X10 → B23
< Sauvegarde de la situation de GC
! IF RE(B22) + FE(B22) THEN EXEC MSIT2 (B8; B9; B10; 0; W30; B11; W22)
< Restitution de la situation de GC
! IF RE(/B22) + FE(/B22) THEN EXEC MSIT2 (B8; B9; B10; 0; W30; B11; W22)
< Sauvegarde de la situation de GPN
! IF RE(B23) + FE(B23) THEN EXEC MSIT3 (B12; B13; B14; 1; W33; B15; W23)
< Restitution de la situation du GRAFCET GPN
! IF RE(/B23) + FE(/B23) THEN EXEC MSIT3 (B12; B13; B14; 1; W33; B15; W23)
  
```

L'état des bits d'entrée Funct0 et Funct1 est à déterminer en fonction du mode désiré :

- si $X0 + X2 = 1$
mettre Funct0 à 0 et Funct1 à 1 forcera le GRAFCET GC sur la situation de forçage mémorisée de CW20 à CW22,
- si $X0 + X2 = 0$
mettre Funct0 et Funct1 à 0 assurera la libération de GC.

De même :

- si $X0 + X2 = 1$
mettre Funct0 à 1 et Funct1 à 0 sauvegardera la situation courante de GC dans les mots W30 à W32,
- si $X0 + X2 = 0$
mettre Funct0 et Funct1 à 1 assurera la restitution de la situation mémorisée dans les mots W30 à W32,

Le même raisonnement est applicable au GRAFCET GPN.



APPLICATION 3 : cas du langage ORPHEE (APRIL)

Ce logiciel dispose d'un ensemble de Blocs Fonctionnels Constructeurs (BFC) comportant entre autres :

- des BFC pour forcer la situation des GRAFCET %GM, %GE et %XE,
- des BFC pour désactiver un GRAFCET avec ou sans sauvegarde de la situation courante,
- des BFC pour geler un GRAFCET dans l'état actif ou inactif.

Ensemble des BFC disponibles

désignation de la fonction	BFC
activer un graphe maître à partir d'étapes désignées	ACTIGM
activer un graphe esclave à partir d'étapes désignées	ACTIGE
dégeler des graphes maîtres	DEGELGM
dégeler des graphes esclaves	DEGELGE
désactiver des graphes maîtres	DESACGM
désactiver des graphes esclaves	DESACGE
geler à l'état actif des graphes maîtres	GACGM
geler à l'état actif des graphes esclaves	GACGE
geler à l'état inactif des graphes maîtres	GINGM
geler à l'état inactif des graphes esclaves	GINGE
activer un graphe maître à partir de ses étapes initiales	INITGM
activer un graphe esclave à partir de ses étapes initiales	INITGE
mémoriser le contexte d'une entité graphe maître	MEMGM
mémoriser le contexte d'une entité graphe esclave	MEMGE
mémoriser le contexte d'une entité macro-étape	MEMXE
forçage des étapes à l'état actif dans un graphe maître	FORCGM
forçage des étapes à l'état actif dans un graphe esclave	FORCGE
forçage des étapes à l'état actif dans une macro-étape	FORCXE

Figure 7.72
Ensemble des BFC
disponibles (ORPHEE).

Le forçage des graphes est réservé pour la phase de mise au point.

L'ordre de forçage émis vers l'entrée EN doit être de nature impulsionnelle (front montant) afin de lever les contraintes sur le GRAFCET forcé.

Les numéros des étapes à activer sont contenus dans une table TXA.

Exemple de BFC de forçage

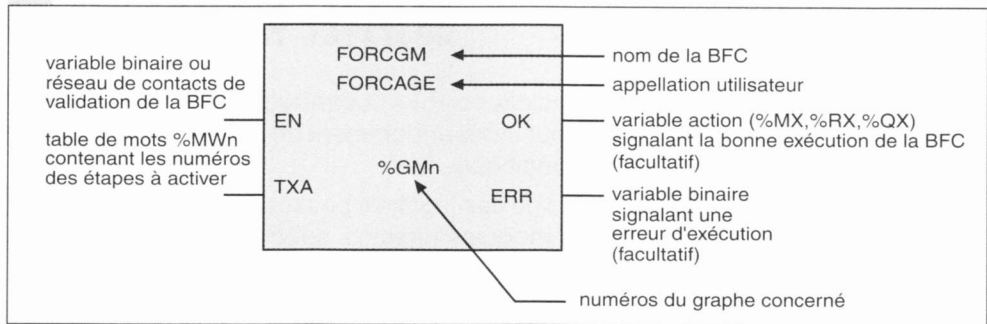


Figure 7.73
BFC de forçage (ORPHEE).

Le forçage du type F/GRAFCE T_n > {situation s} est réalisable moyennant l'utilisation de deux BFC dont les entrées EN seront validées par l'ordre de forçage émis par l'étape du GRAFCET supérieur à laquelle cet ordre est associé :

- la première BFC du type DESAC.. désactive l'ensemble des étapes du GRAFCET n à forcer,
- la seconde BFC du type FORC.. active les étapes correspondant à la situation à forcer.

Exemple : forçage du GRAFCET Esclave %GE1 par l'étape 5 du GRAFCET %GM1

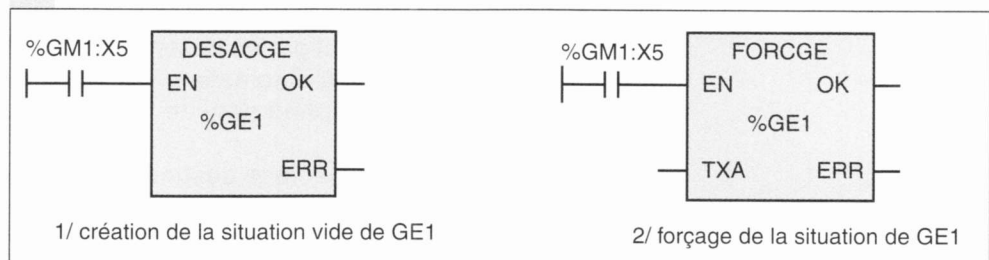


Figure 7.74
Forçage du GRAFCET
Esclave %GE1 par l'étape 5
du GRAFCET %GM1.

VIII. LOGICIELS DE PROGRAMMATION DU GRAFCET ASSISTÉE PAR ORDINATEUR (P_{GAO})

Les logiciels de P_{GAO} permettent de programmer le GRAFCET sur un micro-ordinateur indépendamment de l'automate programmable industriel et de son langage spécifique.

Certains de ces logiciels possèdent des traducteurs de GRAFCET et pour ce faire ont des modules logiciels spécialisés.

La traduction peut se faire :

- soit en langage constructeur (cas de CADEPA),
- soit en langage propriétaire en exploitant le processeur de l'automate (cas d'AUTOMGEN).

Évidemment une liaison série adéquate doit être établie entre le micro-ordinateur et l'API.

1. Logiciel CADEPA 6.20 de la société FAMIC-EURILOR[®]

1.1. Renseignements sur ce logiciel

CADEPA est un logiciel pour la Conception et la réalisation Assistées D'Équipements de Production Automatisés d'où son nom.

Il permet :

- de décrire graphiquement par un ensemble de GRAFCET hiérarchisés le fonctionnement du système à automatiser,
- de traduire cette description dans le langage de l'automate programmable adopté,
- d'imprimer un dossier machine destiné au service de maintenance pour faciliter la compréhension et obtenir un meilleur suivi de l'installation en service.

CADEPA ne peut intervenir qu'au niveau 2 de la description lorsque la PO et la PR sont entièrement définies. L'analyse préparatoire doit être faite par un autre moyen.

L'implantation de CADEPA nécessite un PC Pentium équipé de Windows 3.1, 95, 98 ou NT disposant :

- d'une mémoire vive de 32 Mo minimum,
- de 30 Mo libres au moins sur le disque dur,
- d'une carte et d'un écran graphique VGA (SVGA 17 pouces recommandé),
- d'une sortie série ou parallèle,
- d'un câble PC/API du constructeur ou d'un boîtier de conversion RS232/boucle suivant le type d'API.

La fourniture d'une licence inclut : un manuel utilisateur, un système de protection, un CD-Rom supportant le logiciel muni d'un simulateur GRAFCET, d'un module de configuration de menus et d'un ou plusieurs post-processeurs au choix.

Post-processeurs disponibles

AEG MODICON série 984	MITSUBISHI A1S et FX
ALLEN BRADLEY SLC 500 et PLC-5	SIEMENS 90U à 155U
APRIL P15	TELEMECANIQUE TSX 17-20, 27-20, 47-20
CEGELEC ALSPA C50 et C100	TELEMECANIQUE TSX 07 (nano)
CEGELEC 80-25, 80-35, 80-75	TELEMECANIQUE TSX 37 (micro) avec PL7-Junior
FESTO FPC 101B et 103	TELEMECANIQUE TSX 57 (Premium)
FESTO FEC Dos	Toutes cibles programmables en langage C
GE FANUC 90-30 et 90-70	

1.2. Aperçu des possibilités de CADEPA

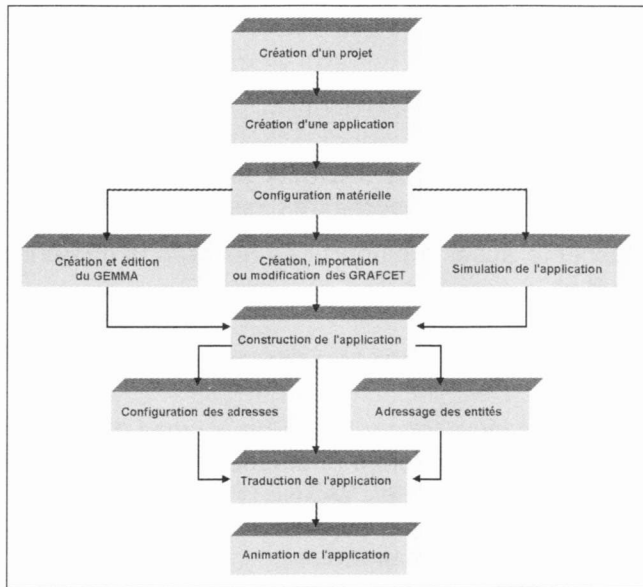


Figure 8.1
Etapes pour réaliser une application avec CADEPA.

Pour concevoir et réaliser une application opérationnelle sur un API, le logiciel CADEPA impose de passer par une suite ordonnée d'étapes (fig. 8.1).

1.3. Éditeur GRAFCET de CADEPA

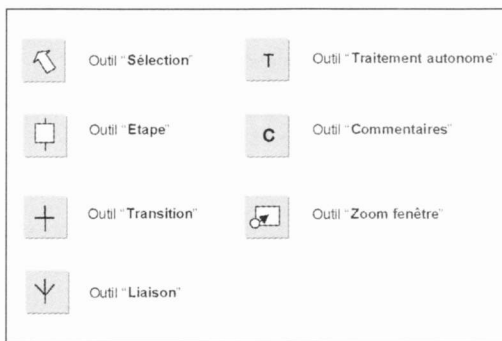


Figure 8.2
Possibilités de l'éditeur CADEPA.

L'éditeur GRAFCET permet la saisie des éléments de base du GRAFCET, des réceptivités associées aux transitions, des actions associées aux étapes et des commentaires.

A cet effet, la barre du menu de l'éditeur donne, en cliquant sur le bouton approprié, accès aux éléments recherchés du GRAFCET à construire (fig. 8.2).

L'éditeur CADEPA accepte tous les concepts du GRAFCET dont celui de macro-étape (UTE C 03-191) (fig. 8.3).

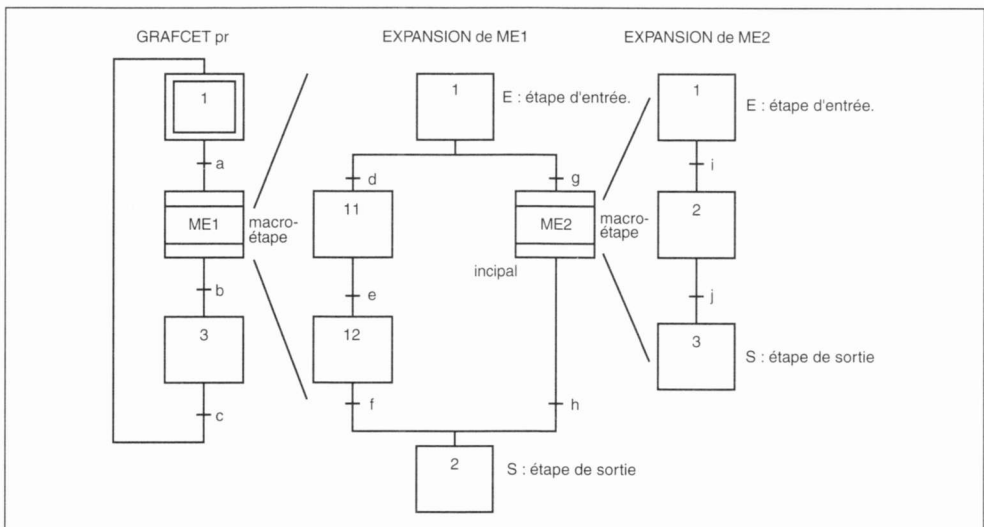
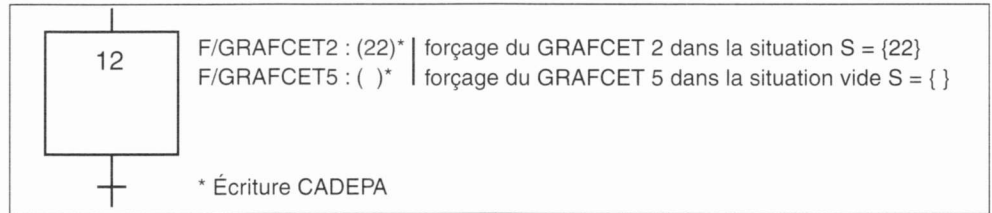


Figure 8.3
Concepts du GRAFCET acceptés par le logiciel CADEPA.

Figure 8.4
Codes opératoires
de forçage de CADEPA.



Il faut noter aussi que CADEPA exploite le concept de forçage de situation sous la forme de codes opératoires associés à l'étape (fig. 8.4).

Il accepte aussi la plupart des opérations numériques courantes (voir tableau fig. 8.5).

1.4. Codes opératoires reconnus pour les actions et les équations (fig. 8.5)

Ces codes interviennent dans deux types d'équations :

- avec sortie maintenue (mémorisation) :
 - mise à 1 (exemple : VANNE1=1 SI NIVEAU1H),
 - mise à 0 (exemple : VANNE1=0 SI/NIVEAU1H),
- avec sortie non maintenue :
 - combinaison d'entrées (exemple : THERMIQ=THPOMPE1+THPOMPE2),
 - affectation de sortie (exemple : VANNE1 SI /NIVEAU1H).

CODES OPÉRATOIRES	FONCTIONS
Variables Tout ou Rien a=b a=/b a=b+c a=b++c a=b.c a=(b+c).d a=1 a=0 a=1 if b >a <a T/a/durée F/GRAF CETi : (s)* F/GRAF CETi : () * a,b if d a;b if d	Affectation Complément OU inclusif OU exclusif ET Mise en facteur Mise en mémoire Effacement de la mémoire Condition de mise en mémoire Front montant Front descendant Lancement de la temporisation T si a = 1 Forçage du GRAF CET i dans la situation S = {s} Forçage du GRAF CET i dans la situation vide S = { } Virgule (actions soumises à la même condition d) Point virgule (actions indépendantes)
Variables numériques [] [a=b] [a='n'] [a=b.OR.c] [a=b.XOR.c] [a=b.AND.c] [a=b+c] [a=b-c] [a=b*c] [a=b/c] [a.INC.'n'] ou [a.INC] [a.DEC.'n'] ou [a.DEC] [a:DLG.'n'] ou a.DLG] [a.DLD.'n'] ou [a.DLD] [a=(b+c)*d] [a>b] [a<b] [a>=b] [a<=b] [a<>b] [a=?b]	Début et fin d'une expression numérique Affectation du contenu de b dans a Affectation de la valeur immédiate n à a OU inclusif bit à bit entre b et c dans a OU exclusif bit à bit entre b et c dans a ET bit à bit entre b et c dans a Addition de b + c dans a Soustraction de b-c dans a Multiplication de b par c dans a Division de b par c dans a Incrémentation de la valeur n ou 1 par défaut Décrémententation de la valeur n ou 1 par défaut Décalage logique de n bits ou de 1 par défaut Décalage logique à droite de n bits ou de 1 par défaut Produit de la somme b+c par d dans a Tests : supérieur, inférieur, supérieur ou égal, inférieur ou égal, différent, égal

Figure 8.5
Codes opératoires
reconnus pour les actions
et les équations.

1.5. Codes opératoires reconnus pour les réceptivités (fig. 8.6)

CODES OPERATOIRES	FONCTIONS
Variables logiques /b b+c b++c b.c (b+c).d =1 >a <a T/a/durée	Complément OU inclusif OU exclusif ET Mise en facteur Réceptivité toujours vraie Front montant de a Front descendant de a Fin de temporisation
Variables numériques ['n'] [a>b] [a<b] [a>=b] [a<=b] [a<>b] [a=?b]	Valeur immédiate n Tests : supérieur, inférieur, supérieur ou égal inférieur ou égal ,différent, égal

Figure 8.6
Codes opératoires reconnus
pour les réceptivités.

Autres notations CADEPA

^I, ^E	Variable globale interne et externe
^L, ^I^L	Variable locale et locale interne
^M	Module langage constructeur
Xi : GRAFCETn	Etape i du GRAFCET n

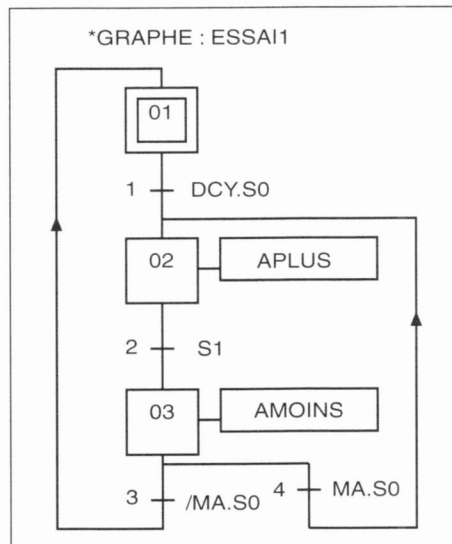


Figure 8.7
GRAFCET CADEPA,
niveau 3.

1.6. Exemples de traductions en langages d'API

Écriture du GRAFCET avec l'éditeur de CADEPA (fig. 8.7)



APPLICATION 1 : traduction en langage PL7-2 et PL7-3 (TSX Schneider Télémécanique) (fig. 8.8)

On notera que CADEPA effectue la traduction du GRAFCET en langage à contacts en utilisant la fonction mémoire en monostable.

Les fonctions transitions sont calculées préalablement et mises en bits internes (bits B104 à B107).

Les étapes sont matérialisées par les bits internes B101 à B103.

Les fonctions transitions sont notées Yi et les étapes Xi.

L'initialisation est assurée par le bit B100 à 0 puis mis à 1 inconditionnellement en fin du premier cycle de scrutation (réseau *S10).

Le listing de sortie du programme (fig. 8.8) fournit à la fois les symboles et les adresses des entrées, des sorties et des bits internes.

Cela suppose qu'une table des symboles et de leurs adresses respectives a été préalablement établie.

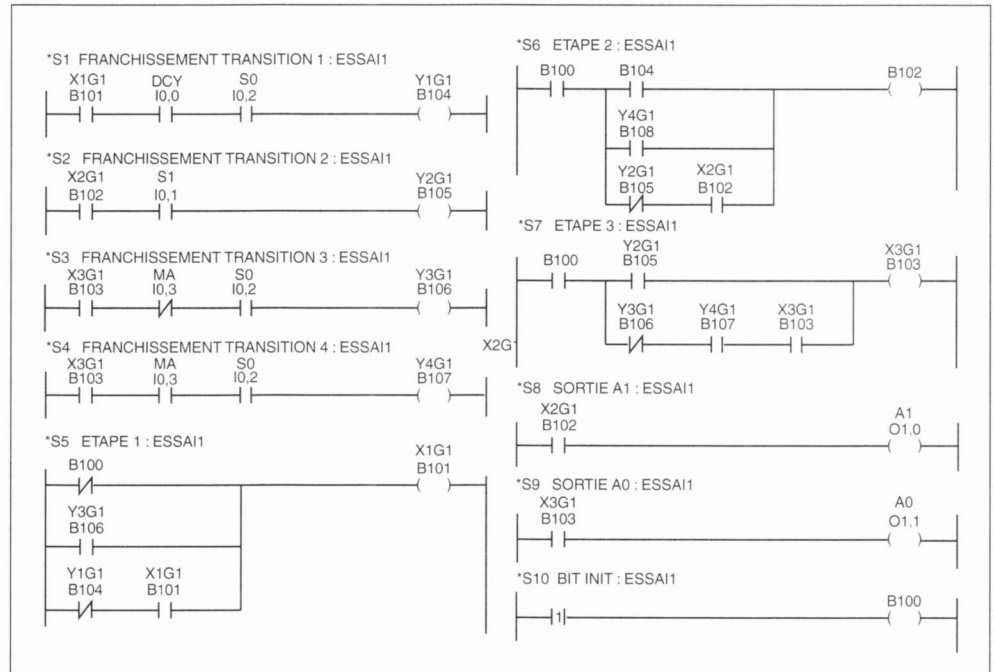


Figure 8.8
Traduction en langage
à contacts PL7-2 et PL7-3
(TSX Schneider
Télemécanique).

Remarque

Vu le nombre relativement limité de bits internes, une option est proposée pour travailler avec des bits de mot. Le bit de mot est noté W_i , j où i est le numéro du mot en décimal et j le numéro du bit en hexadécimal (0 à F).



APPLICATION 2 : traduction en langage SIEMENS

Écriture du GRAFCET avec l'éditeur de CADEPA (fig. 8.9)

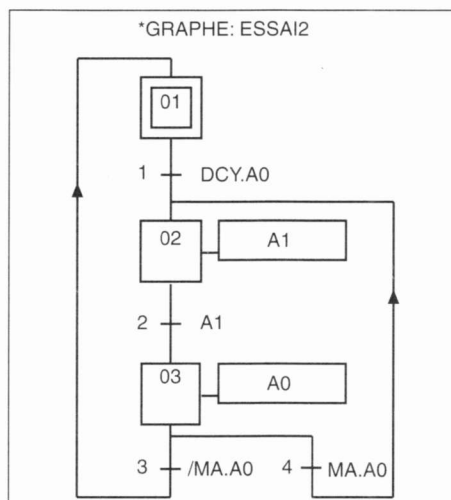


Figure 8.9
GRAFCET CADEPA,
niveau 3.

Traduction en langage à contacts SIEMENS (fig. 8.10)

La traduction de CADEPA en langage à contacts SIEMENS se fait par l'intermédiaire de la fonction mémoire en bistable.

Les fonctions transitions sont notés Y_i et sont d'abord calculées (bit M21.4 à M21.7).

Les étapes sont notées X_i (bits M21.1 à M21.3).

L'initialisation a lieu par l'intermédiaire du bit M21.0 et du module OB001.

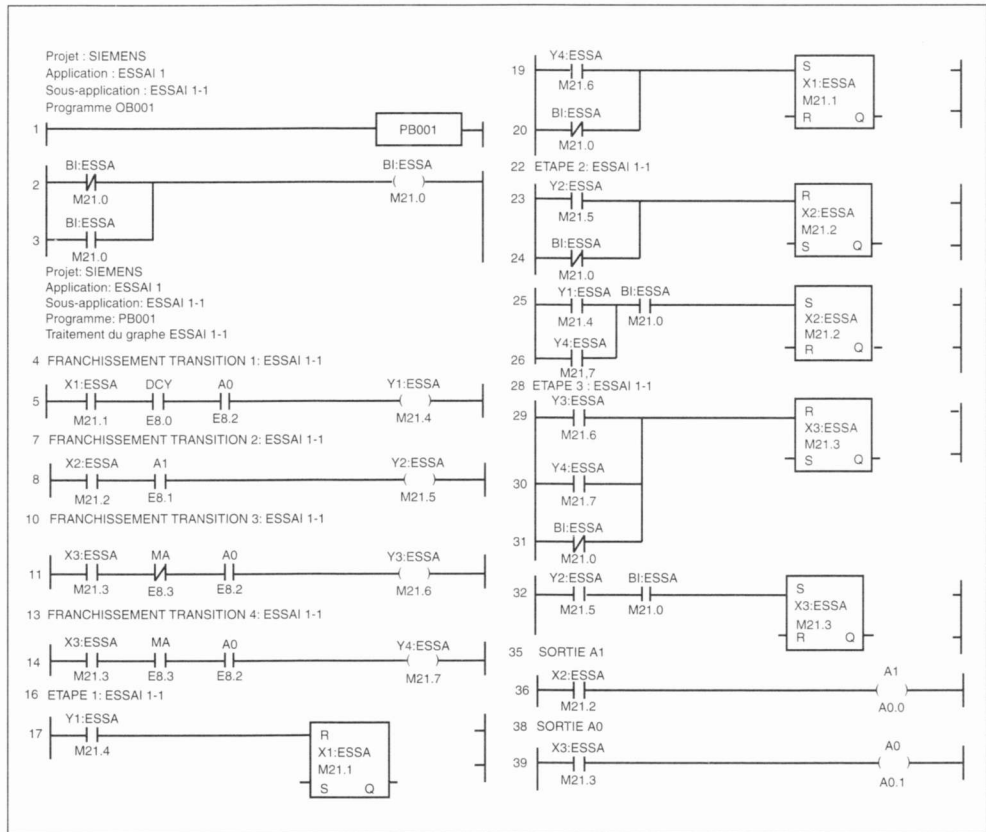


Figure 8.10
 Traduction en langage
 à contacts (SIEMENS).

2. Atelier d'automatique ALOGRAF de la société UXP

2.1. Renseignements sur ALOGRAF, version 4.01

Le package d'Alograf comprend :

- un CD-ROM contenant :
 - l'atelier logiciel ALOGRAF complet,
 - la documentation détaillée en ligne,
 - des utilitaires
- une clé de protection électronique pour port parallèle ou USB
- une documentation générale ALOGRAF
- un additif sur le nouvel éditeur GRAFCET d'ALOGRAF V4.01

L'utilisation en évaluation, sans clé de protection, est limitée au maximum à 16 entrées-sorties et 8 GRAFCET, sur des cibles virtuelles sous Windows, WINPC 16 bits et WINPC32 32 bits.

L'implantation d'Alograf nécessite un PC équipé de Windows 98, NT4, 2000 ou plus disposant :

- d'une mémoire vive de 32 Mo minimum,
- d'un lecteur de CD-ROM pour installer ALOGRAF,
- de 20 Mo libres au moins sur le disque dur,
- d'une carte et d'un écran graphique VGA,
- d'une sortie parallèle ou d'un port USB.

2.2. Réalisation d'une application

Le développement et l'exécution d'une application s'opèrent en 9 étapes.

Après chaque saisie une sauvegarde est nécessaire pour ne pas perdre les données entrées au clavier.

Etape 1 : création de l'application et sélection de la cible

Une fois qu'un nom a été choisi pour l'application et une cible sélectionnée par l'utilisateur, ALOGRAF crée un répertoire pour sauvegarder le travail.

Une cible est un matériel d'automatisme qui exécute le programme d'application généré par le poste de développement Alograf. Une cible est donc un calculateur d'automatisme (sic).

Voici quelques-unes des cibles actuelles les plus courantes :

- PCGRAF (coprocesseur pour PC),
- PC sous MS-DOS ou LINUX,
- POWER PC (Motorola) sous Hard Hat Linux,
- système VME,
- station UNIX,
- microcontrôleur AMD 188/186, carte électronique dédiée, etc.

La cible Alograf sous LINUX semble prometteuse. Un seul PC (ou PC/VME) sous système multitâche supporte l'application de contrôle-commande et l'application de supervision. Ces deux applications coopèrent tout en respectant les contraintes temps réel garantissant le déterminisme et les performances de l'ensemble.

Etape 2 : édition de la nomenclature des variables

L'éditeur de nomenclature NOMEN permet de saisir les divers types de variables de l'application :

- les entrées-sorties externes Tout Ou Rien (TOR) et analogiques (ANA) associées aux capteurs et aux actionneurs de la partie opérative,
- les variables numériques internes sous divers formats : caractères (1 octet), nombres entiers courts sur 2 octets (de - 32 768 à +32 767), nombres entiers longs sur 4 octets (de - 2 147 483 648 à +2 147 483 647), nombres entiers positifs ou négatifs, en virgule flottante simple précision (de -3,4 E⁺³⁸ à +3,4 E⁺³⁸) ou double précision (de -1,79 E⁺³⁰⁸ à +1,79 E⁺³⁰⁸).
- des tableaux regroupant plusieurs variables du même type peuvent aussi être déclarés.

Un symbole jouant le rôle de mnémotechnique doit être attribué à chacune des variables de l'application et les propriétés particulières des variables précisées (voir fig. 8.11 et 8.12)

(1*) Nom de l'entrée	(2*)	(3*)	(4*)	(5*)	(6*) Commentaires
BAU	0	0	0	0001	coup de poing d'arrêt d'urgence
BMARCHE	0	0	1	0002	impulsion de mise en marche

(1*) colonne des symboles mnémotechniques.
(2*) colonne filtrage : si la variable d'entrée est filtrée (1), son état doit rester inchangé, soit un certain temps, soit un certain nombre de cycles de programme pour être prise en compte. Si elle n'est pas filtrée (0), elle est directement prise en compte.
(3*) colonne d'inversion ou non du sens de l'activité de la variable : (0) non inversé, (1) inversé.
(4*) colonne de traitement des impulsions : (1) prise en compte impulsion montante ou descendante, (0) pas d'impulsion.
(5*) colonne de l'index mémoire : compris entre 1 et 9999.
(6*) colonne commentaires sur les variables d'entrée.

Figure 8.11
Saisie des variables
d'entrées

La saisie des variables de sortie est relativement plus simple (fig. 8.12)

(1*) Nom de la sortie	(2*)	(3*)	(4*) Commentaires
VERIN_AP (5*)	0	1	Commande aller du vérin A
VERIN_AM (5*)	0	2	Commande retour du vérin A

(1*) colonne des symboles mnémotechniques des variables de sortie.
 (2*) colonne sens d'activité de la variable de sortie : (0) normale, (1) inverse.
 (3*) colonne de l'index en mémoire.
 (4*) colonne commentaires sur les variables de sortie.
 (5*) l'écriture A+ et A- n'est pas acceptée par l'éditeur NOMEN, ni par l'éditeur GRAFCET.

Figure 8.12
Saisie des variables de sortie.

Etape 3 : édition des GRAFCET et chaînage des GRAFCET

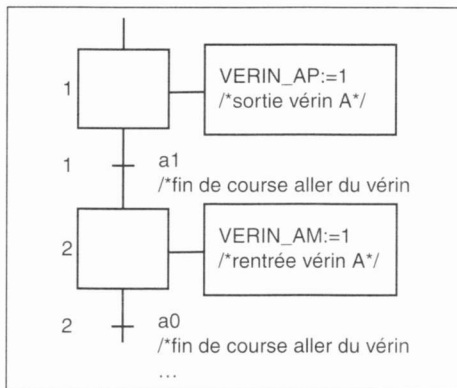


Figure 8.13
Saisie du GRAFCET.

Tout GRAFCET conforme à la norme CEI 1131-3 peut être tracé dans la page de l'éditeur graphique. Les symboles graphiques suivants sont mis en place automatiquement par simple clic de souris :

- étape initiale et étape normale,
- réceptivité associée à la transition,
- action associée à l'étape,
- divergences et convergences en OU et en ET.

Les mnémoniques et les commentaires sont intégrés au graphisme (fig. 8.13).

Le chaînage permet de déterminer explicitement (déterminisme) l'ordre dans lequel les GRAFCET seront traités (scrutés). Le chaînage est défini par l'utilisateur. Lors de l'exécution le chaînage peut être modifié en fonction des modes de marche et de l'état du procédé. C'est ainsi que les fonctions de hiérarchie, forçage, figeage, reprise, sécurité, appel de sous-programmes et ressources, etc. sont mises en œuvre dans ALOGRAF.

Etape 4 : intégration des blocs fonctionnels dans les GRAFCET (UBF)

Le module UBF d'ALOGRAF fait office d'interface entre l'informaticien (programmation en C et C++, voire en Turbo Pascal ou JAVA) et l'automaticien (utilisateur d'ALOGRAF).

L'utilisation de blocs fonctionnels permet d'étendre les possibilités de l'automatisme. Tout module de programmation spéciale dont la description par GRAFCET se révèle délicate peut être introduit sous forme de blocs fonctionnels.

Les blocs fonctionnels peuvent être standards (compteurs, temporisateurs) ou spécifiques à une application : modules de calculs arithmétiques et trigonométriques, régulation PID et RST, asservissement d'axes, etc.

Les blocs fonctionnels permettent à l'utilisateur de tirer ainsi le meilleur parti du calculateur cible (pilotes E/S, noyau multitâche, interruption temps réel, etc.).

Le bloc fonctionnel peut être appelé soit comme un sous-programme (ou fonction) associé à une étape ou à une transition d'un GRAFCET, soit être considéré comme une partie opérative interne traitée dans ce cas à la fin du cycle de traitement des GRAFCET.

Etape 5 : configuration des entrées-sorties

Cette opération a pour but d'établir un lien entre l'adressage des entrées-sorties d'ALOGRAF et celles physiques de la cible.

Etape 6 : compilation et exécution de l'application

La compilation s'effectue en deux temps :

- traduction préalable en langage C ou C++ des GRAFCET décrits avec l'éditeur (contrairement aux API dont la plupart traduisent les GRAFCET en mode interprété),
- compilation proprement dite dans le code exécutable de la cible.

L'exécution sera donc optimisée car directement liée aux performances du micro-processeur de la cible. L'algorithme avec recherche de stabilité (ARS) propre à ALOGRAF (*) débouche sur des applications au comportement déterministe conformément aux recommandations de la norme CEI 848.

Etape 7 : téléchargement et exécution de l'application sur la cible

La communication entre la plate-forme de développement contenant ALOGRAF et la cible peut être assurée par une liaison série RS 232 ou par un réseau Ethernet TCP/IP. Après introduction et exécution sur la cible d'un programme spécifique (MONPC.EXE, par exemple sur un PC cible), l'application pourra être transférée entre les deux postes.

L'application peut alors être lancée sur la cible qui devient indépendante.

Etape 8 : mise au point de l'application

L'éditeur GRAFCET peut se mettre en mode animation afin de visualiser l'évolution dynamique des GRAFCET de la cible (plusieurs simultanément).

L'utilitaire MISOP permet :

- de visualiser l'état des variables associées à ces GRAFCET,
- de forcer éventuellement les valeurs des variables externes et internes.

Etape 9 : impression de la documentation complète de l'application

La documentation est imprimée en format A3 ou A4 avec un cartouche renseigné par l'utilisateur. Une prévisualisation des GRAFCET est proposée sur l'écran. L'utilisateur peut aussi exporter ses GRAFCET dans des documents Microsoft Word.

(*) Voir article de J.L. Plagnol « Le GRAFCET pas déterministe... ? Ah bon ! » paru dans la revue " Mesures " de juin 1995.

3. Logiciel AUTOMGEN de la société IRAI (versions 6 et 7)

3.1. Renseignements sur ce logiciel

Ce logiciel s'implante dans un micro-ordinateur PC équipé :

- de WINDOWS 95, 98, ME, NT4, 2000 ou XP.
- d'une mémoire vive (RAM) de 32 Mo minimum,
- d'un disque dur avec 20 Mo libres après mise en place d'AUTOMGEN.

Il est fourni avec une clé de protection EASY-LOCK (dongle) empilable se montant sur le port parallèle.

Notations des variables

entrées	i<numéro>
sorties	o<numéro>
bits internes	u<numéro>
bits images d'étape	x<numéro>
temporisateurs	t<numéro>
compteurs	c<numéro>
mots entiers de 16 bits	m<numéro>
mots entiers de 32 bits	l<numéro>
nombres réels	f<numéro>

3.2. Possibilités d'AUTOMGEN

Automgen dispose d'un éditeur graphique acceptant le GRAFCET, les schémas à contacts (ladder), les logigrammes, l'organigramme, les blocs fonctionnels et un éditeur de texte pour le langage littéral. Il est conforme à la norme CEI 1131-3.

Les variables i, o, u, x sont du type booléen (binaire ou Tout ou Rien).

Les fronts se notent $\uparrow i<\text{numéro}>$ ou $\uparrow u<\text{numéro}>$ (front montant) et $\downarrow i<\text{numéro}>$ ou $\downarrow u<\text{numéro}>$ (front descendant).

Les variables t, c, m sont du type numérique et codées sur 16 bits dont un bit de signe (-32 728 à +32 727).

Elles peuvent faire l'objet de calculs arithmétiques (+, -, *, /), d'opérations logiques bit à bit (complément, ET, OU inclusif, OU exclusif), de décalages à droite ou à gauche, de transferts entre mots ou vers des bits, être incrémentées ou décrémenteés, de comparaisons (=, <, >, <>, =>, =<). Une réceptivité toujours vraie se note =1.

Les variables f sont du type numérique réel et sont codées pour la mantisse et l'exposant sur 16 bits dont un bit de signe ($-32\,727 \cdot E^{2 \cdot 16}$ à $32\,728 \cdot E^{2 \cdot 16}$).

Les actions ou ordres peuvent être du type Tout ou Rien ou numérique.

L'ordre est placé dans un rectangle d'action, associé à l'étape dans le cas du GRAFCET.

L'action peut être égale, complémentaire (N), mémorisée (S et R pour Set et Reset), impulsionnelle à l'activation (P1) ou à la désactivation (P0), incrémentée ou décrémenteée dans le cas d'un mot.

La notation normalisée, avec le symbole % en tête des variables, est admise.

3.3. Organisation générale (fig. 8.14)

Après édition du GRAFCET (ou du logigramme ou du programme en littéral) on procède à la compilation.

Ensuite, après avoir relié correctement le micro-ordinateur à l'API, on lance la traduction à travers le post-processeur approprié.

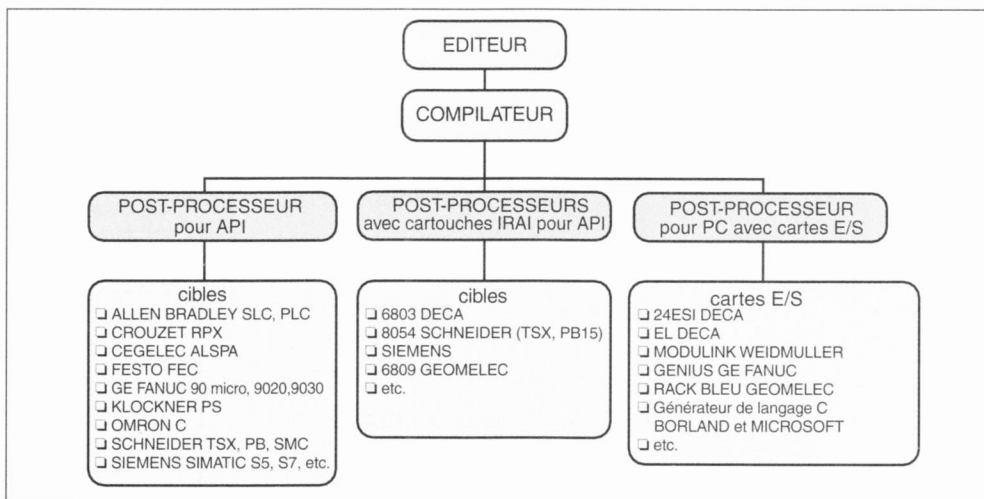


Figure 8.14
Organisation générale.

3.4. Éditeur GRAFCET

Cet éditeur permet de tracer tous les éléments graphiques du GRAFCET normalisé, notamment les cadres d'actions avec choix de la longueur.

Il accepte la notion de macro-étape :

Syntaxe

```
# {options}<N° d'étape>"Nom de folio (extension GR7 par défaut)"
```

Exemple

La macro-étape M1 a pour expansion le GRAFCET MAC.GR7.

Les options possibles pour la macro-étape :
 D : le forçage à 0 de la macro-étape entraîne la mise à 0 de l'ensemble des étapes de l'expansion associée,
 E : met fin à l'exécution de l'expansion même si l'étape de fin n'est pas atteinte dès que la macro-étape n'est plus activée.

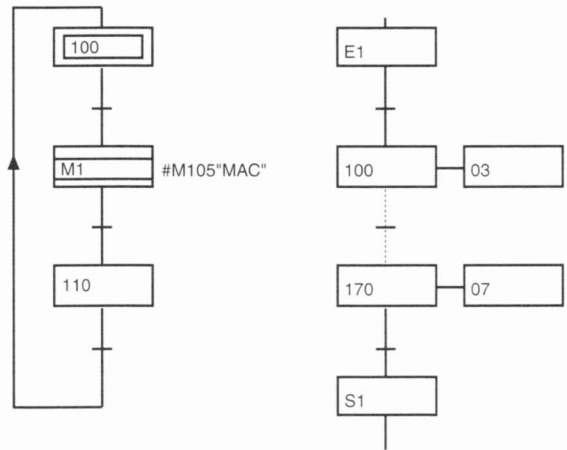


Figure 8.15
Macro-étape et expansion.

Un GRAFCET est identifié par le numéro d'une de ses étapes, de préférence une étape initiale.

AUTOMGEN accepte la notion de forçage dans le cadre d'action sous la forme générale suivante (non conforme avec la documentation UTE C 03-191) :

Syntaxe

F<n° GRAFCET> : (situation à forcer)

Exemple (fig. 8.16)

Forçage du GRAFCET 200 dans la situation S = {203}

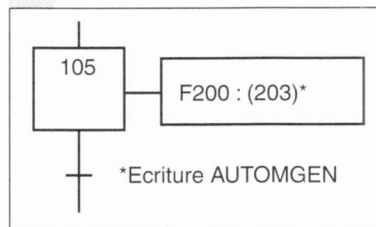


Figure 8.16
Forçage de situation.

On peut aussi forcer à 1 ou 0 une étape d'un GRAFCET ouvert (non rebouclé) si nécessaire pour faire évoluer des GRAFCET s'apparentant à des GRAFCET sous-programmes ou à des GRAFCET de tâche.

Exemple (fig. 8.17)

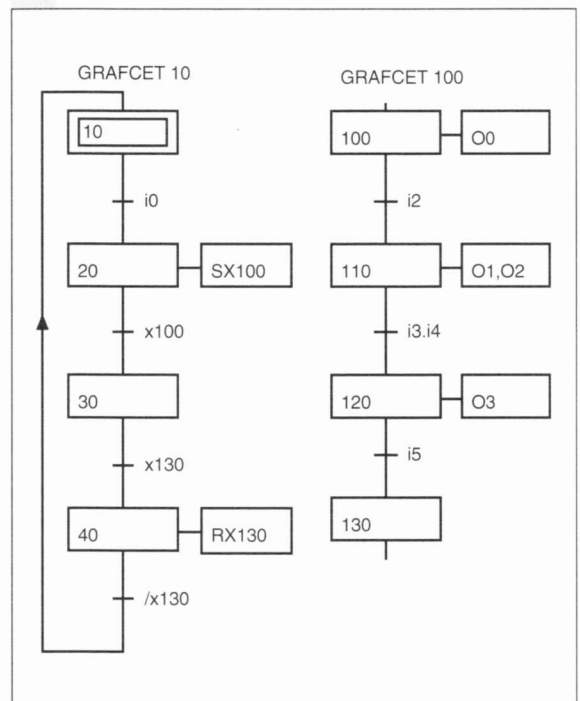


Figure 8.17
Forçage d'un GRAFCET non rebouclé.

Commentaires de la figure 8.17

L'étape 20 force à 1 l'étape 100 du GRAFCET 100 préalablement en situation vide. L'activation de l'étape 100 permet de franchir la transition 20 → 30. En fin d'évolution du GRAFCET 100, l'activation de l'étape 130 permet de franchir la transition 30 → 40 du GRAFCET 10.

L'étape 40 force à 0 l'étape 130 du GRAFCET 100. Le GRAFCET 100 passe en situation vide ce qui assure la réinitialisation du GRAFCET 10 par franchissement de la transition 40 → 10. Le GRAFCET G100 se retrouve dans une situation vide $S(G100) = \{ \}$.

De plus AUTOMGEN possède des instructions pour mémoriser la situation d'un GRAFCET dans différents cas.

1/ ÉTAT ACTUEL D'UN GRAFCET

Syntaxe

G<n° GRAFCET> : N° de bit

Cette instruction mémorise l'état du GRAFCET n, c'est-à-dire sa situation courante dans une série de bits successifs d'un espace réservé par la commande #B.

Exemple

G30 : 140 mémorise l'état du GRAFCET 30 (situation courante) à partir du bit B140.

2/ ÉTAT PARTICULIER D'UN GRAFCET

Syntaxe

G<n° GRAFCET> : <N° de bits> (liste d'étapes actives)

Cette instruction mémorise l'état défini par la liste des étapes actives du GRAFCET n, à partir du bit indiqué.

Exemples

G0 : 100 mémorise l'état actif actuel du GRAFCET 0 à partir du bit B100.

G0 : 200 () mémorise l'état vide (situation vide) du GRAFCET 0 à partir du bit B200.

G10 : 150 (1, 2) mémorise l'état du GRAFCET 10 dans lequel seules les étapes 1 et 2 sont actives, à partir du bit B150.

3/ FORÇAGE D'UN GRAFCET À PARTIR D'UN ÉTAT (SITUATION) MÉMORISÉ

Syntaxe

F<n° GRAFCET> : <N° de bit>

Cette instruction force le GRAFCET n dans l'état mémorisé à partir du bit précisé.

Exemples

G0 : 100 sauvegarde la situation courante du GRAFCET 0,

F0 : 100 restaure la situation courante précédemment mémorisée.

4/ FIGEAGE D'UN GRAFCET

Syntaxe

F<n° GRAFCET>

Cette instruction fige le GRAFCET n et bloque le système dans la situation courante.

Exemple

F100 fige le GRAFCET 100.

3.5. Application : dosage de produit liquide (chapitre VII, § 1, Fig. 7.1))

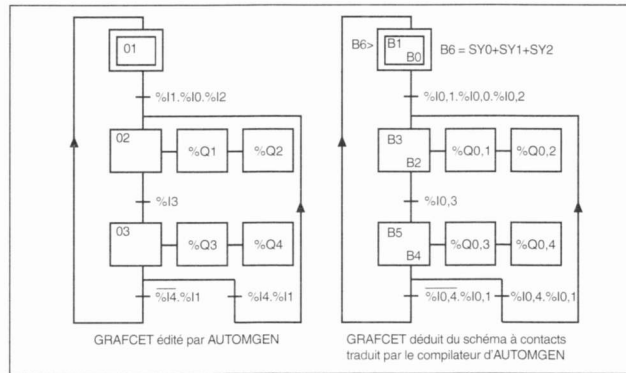


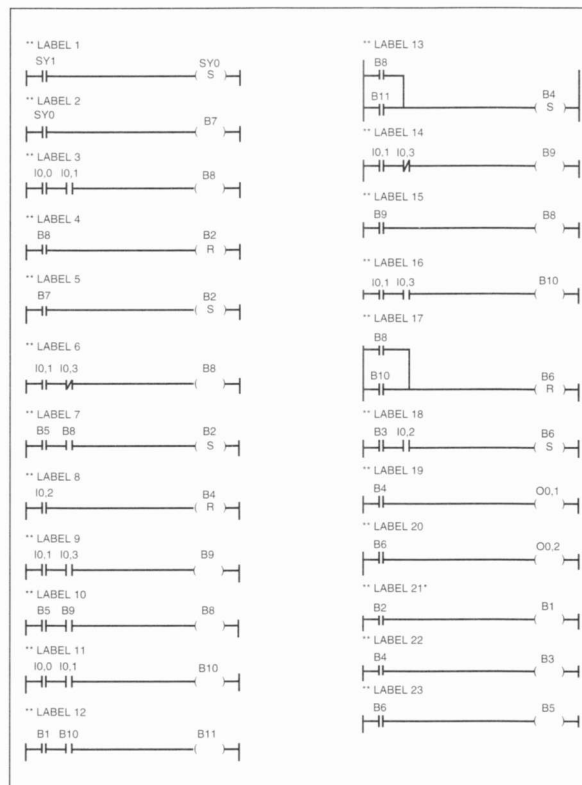
Figure 8.18
GRAFFCET édité sur PC
par AUTOMGEN

Edition sur PC, compilation
et transfert sur TSX 17
Schneider Télémécanique
(fig. 8.18 et 8.19)

Remarque

On notera que la traduction se fait en langage à contacts propre au langage PL7-2 du TSX 17. La structure utilisée est difficile à mettre en évidence. Les réseaux 1 et 2 initialisent le GRAFCET par l'intermédiaire du bit B6 dans le réseau 3. Les réseaux 9 et 10 affectent les sorties en fonction des bits B3 et B5 images des étapes 2 et 3. Le réseau 11 crée les bits B0, B2, B4, images des bits B1, B3, B5, eux-mêmes images des étapes 1, 2, 3. Les bits intermédiaires B7 et B8 après utilisation pour un calcul sont réutilisés pour le calcul suivant. L'objectif est de minimiser le nombre de bits mobilisés pour résoudre le problème. AUTOMGEN, quand c'est possible, utilise des bits de mots, ce qui permet d'étendre la capacité du langage en bits disponibles.

AUTOMGEN exploite à sa manière les règles d'exécution des réseaux de contacts du PL7-2.



Rappel de la page 75,
§ Remarque :

- le programme est exécuté réseau par réseau selon l'ordre des réseaux écrits en mémoire.
- une séquence étant délimitée par une connexion verticale, l'exécution d'un réseau s'effectue séquence par séquence, de haut en bas et de gauche à droite.
- les bobines sont traitées en dernier lieu de haut en bas.

Le manque de transparence entre la solution en réseaux de contacts et le GRAFCET initial est évident.

Pour procéder aux modifications et au dépannage sur site, il est pratiquement obligatoire de repasser par l'éditeur et le traducteur d'AUTOMGEN. Ce n'est pas un gros problème si ce logiciel est implanté, par exemple, dans un PC portable connecté à l'API.

Figure 8.19
Traduction en langage PL7-2
pour TSX 17 Schneider
Télémécanique.

IX. APPLICATION

1. Présentation du poste à automatiser

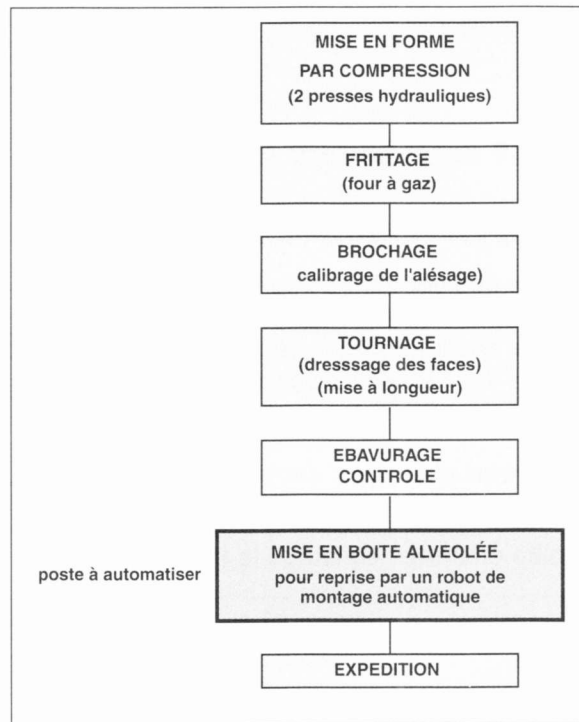


Figure 9.1
Ligne de produit « PIGNON ».

Un constructeur de véhicules automobiles fait fabriquer le pignon d'une pompe à huile par une société spécialisée dans le frittage des métaux.

Ce procédé appliqué aux aciers s'est révélé la solution la plus productive (meilleur indice qualité/prix) pour fabriquer ce pignon de faible dimension :

- \varnothing extérieur : 30 mm,
- épaisseur : 18 mm,
- \varnothing alésage : 16 mm.

La ligne de fabrication de ce produit presque entièrement automatisée est présentée ci-contre (fig. 9.1). La situation du poste à automatiser dans l'atelier de frittage est mise en évidence.

Il s'agit du conditionnement des pignons avant expédition.

• Particularité des boîtes de rangement (fig. 9.2)

Les boîtes de rangement alvéolées comportent une structure en nid d'abeilles de 5 lignes et 8 rangées, ce qui donne au total 40 cases. La hauteur de la boîte, et par suite de chaque alvéole, est suffisante pour contenir 3 pignons verticalement. En définitive, chaque boîte peut recevoir 120 pignons.

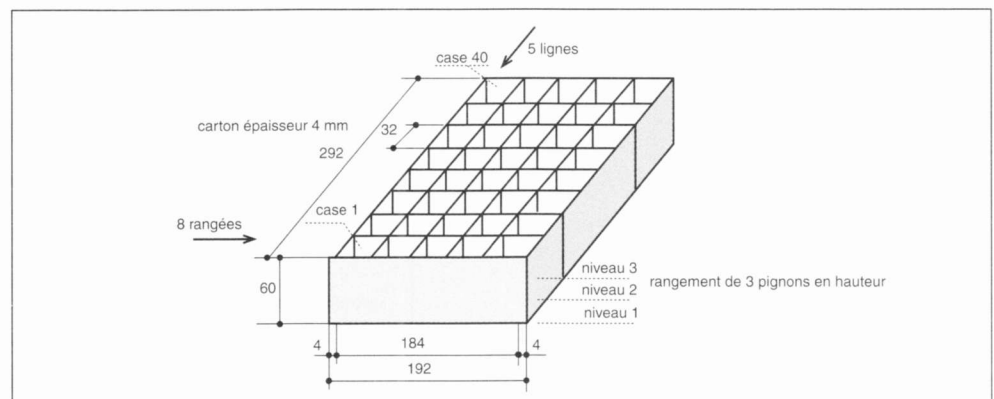


Figure 9.2
Boîte de rangement alvéolée.

• Pourquoi ce type de conditionnement ?

Chez le constructeur, la chaîne d'assemblage de la boîte de vitesses utilise des robots de manutention. Grâce à cette boîte, on obtient un positionnement parfaitement déterminé et toujours identique des pignons. La programmation du robot de préhension des pignons suivant les 3 axes de l'espace X, Y, Z s'en trouve facilitée.

• **Quel est le problème au niveau du poste de conditionnement ?**

Les pignons arrivent du poste précédent par l'intermédiaire d'une table vibrante rectiligne. Il faut saisir 3 pignons successifs et les placer dans la case 1. La même opération est à exécuter pour les cases suivantes jusqu'à la case 40.

Pour éviter une réinitialisation du support de boîte il est souhaitable, pour le carton suivant, de commencer par remplir la case 40 et de finir à la case 1 et ainsi alternativement.

2. Analyse descendante du problème

2.1. Point de vue processus (niveau 1)

- Inventaire des contraintes de la fonction globale,
- inventaire des fonctions principales,
- inventaire des contraintes liées aux fonctions principales,
- inventaire des tâches résultantes,
- tracé d'un GRAFCET de coordination des tâches.

A. Inventaire des contraintes de la fonction globale

Le conditionnement, c'est-à-dire le rangement des pignons dans une boîte alvéolée en carton constitue la fonction globale.

fonction globale	contraintes et paramètres connus ou déduits ou imposés par le client
conditionnement (rangement de 120 pignons de pompe en boîte alvéolée en carton)	<ul style="list-style-type: none"> - dimensions du pignon : <ul style="list-style-type: none"> diamètre extérieur: 30 mm épaisseur: 18 mm alésage diamètre: 16 mm - poids du pignon : 0,07 kg <ul style="list-style-type: none"> poids de la boîte vide : négligeable poids de la boîte pleine : 8,5 kg - boîte de 5 x 8 = 40 cases → rangement de 40 x 3 = 120 pignons <ul style="list-style-type: none"> dimensions de la boîte hors tout : 192 x 292 mm x 60 mm épaisseur du carton de la boîte : 4 mm - déplacement ou pas entre 2 cases : 36 mm - cadence de production des presses : <ul style="list-style-type: none"> 100 à 300 pièces/heure → durée de remplissage d'une boîte comprise entre 72 et 24 minutes - autonomie du poste : > 1 heure → 2 boîtes sur le poste → transfert <p>1^{re} boîte :</p> <ul style="list-style-type: none"> - mise en place de 3 pignons dans une case (système de chargement des pignons (SCP)) - déplacement relatif boîte/SCP, case par case, de la case 1 à la case 5 <ul style="list-style-type: none"> idem pour les cases 11 à 15, 21 à 25, 31 à 35 (axe Xav (sens vers l'avant)) - déplacement relatif boîte/SCP de la case 5 à la case 6, idem pour les cases 10 à 11, 15 à 16, 20 à 21, 25 à 26, 30 à 31, 35 à 36 (axe Yd (sens vers la droite)) - déplacement relatif boîte/SCP, case par case, de la case 6 à la case 10 <ul style="list-style-type: none"> idem pour les cases 16 à 20, 26 à 30, 36 à 40 (axe Xar (sens vers l'arrière)) - permutation automatique de la boîte pleine et de la boîte vide <ul style="list-style-type: none"> 2 positions pour la boîte (aller-retour) : <ul style="list-style-type: none"> - pour la mise en place manuelle d'une boîte vide et l'évacuation de la boîte pleine - pour le chargement automatique des 120 pignons dans la boîte <p>2^e boîte :</p> <ul style="list-style-type: none"> - déplacement relatif boîte/SCP inverse en partant de la case 40 (axes Xar et Xav) - déplacement relatif boîte/SCP de la case 36 à la case 35, idem pour cases 31 à 30, 26 à 25, 21 à 20, 16 à 15, 11 à 10, 6 à 5 (axe Yg (sens vers la gauche)) <p>boîtes suivantes : idem alternativement</p>

Figure 9.3
Inventaire des contraintes de la fonction globale.

Cette liste non exhaustive des contraintes permet d'orienter les choix mécaniques et technologiques du bureau d'étude. Elle permet aussi de faire apparaître les fonctions principales donc les tâches à prendre en compte par l'automaticien.

Les mouvements relatifs boîte/système de chargement et la définition des axes méritent une analyse plus détaillée (fig. 9.4).

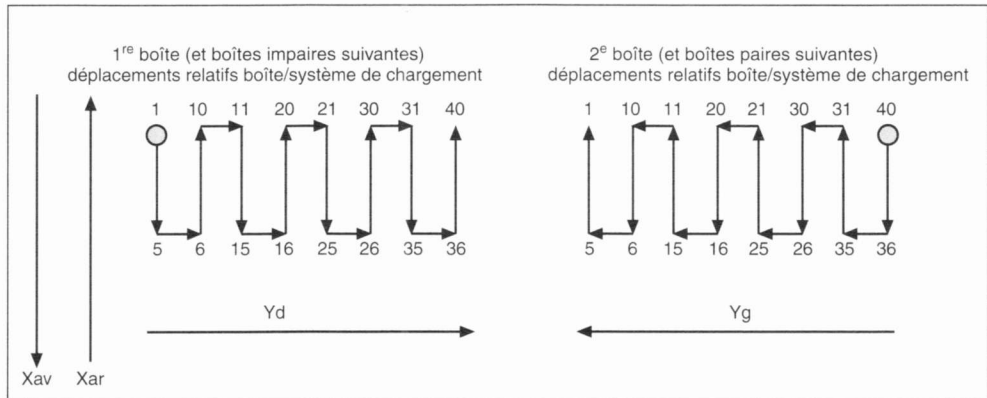


Figure 9.4
Mouvements relatifs boîte-système de rangement et définition des axes X et Y.

B. Inventaire des fonctions principales

Cet inventaire découle de l'analyse précédente.

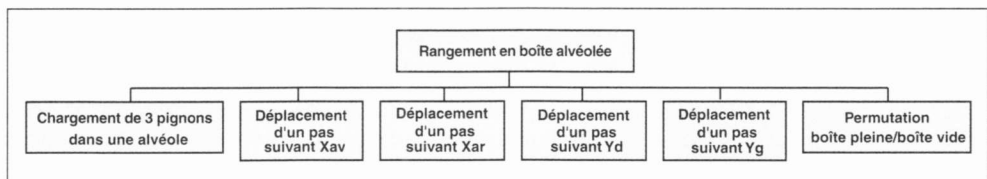


Figure 9.5
Inventaire des fonctions principales ou tâches.

C. Inventaire des contraintes liées aux fonctions principales

fonctions principales	contraintes et paramètres connus ou déduits ou à calculer
système de chargement de 3 pignons dans une case de la boîte	prise avec centrage du pignon utilisation de la gravité si possible anti-retour mécanique à prévoir (trappe à ressort) course à déterminer vitesse réglable maxi-mini à déterminer comptage de 1 à 3 cadence : temps par pièce compris entre 36 et 12 secondes
déplacements suivant Xav et Xar	déplacement linéaire course d'un pas : 36 mm course totale : $36 \times 4 = 144$ vitesse réglable maxi-mini à déterminer
déplacement suivant Yd et Yg	déplacement linéaire course d'un pas : 36 mm course totale : $36 \times 7 = 252$ vitesse réglable max-mini à déterminer
permutation boîte pleine et boîte vide	déplacement circulaire continu ou alternatif rotation de 180° vitesse réglable maxi-mini à déterminer

Figure 9.6
Inventaire des contraintes liées aux fonctions principales.

D. Inventaire des tâches résultantes

Il a été décidé, à ce stade de l'analyse, de retenir les tâches suivantes :

- la tâche T1 « chargement de 3 pignons »,
- la tâche T2 « avance d'un pas suivant axe Xav »,
- la tâche T3 « avance d'un pas suivant axe Xar »,
- la tâche T4 « avance d'un pas suivant axe Yd »,
- la tâche T5 « avance d'un pas suivant axe Yg »,
- la tâche T6 « permutation aller boîte pleine-boîte vide »,
- la tâche T7 « permutation retour boîte pleine-boîte vide ».

E. Étude du GRAFCET de Coordination des Tâches de Production normale (GCTP)

a. Analyse algorithmique

Avant d'aborder le tracé de ce GRAFCET, il est nécessaire de bien comprendre comment va s'effectuer le chargement de la 1^{re} boîte. Pour la compréhension, la 1^{re} boîte est considérée comme impaire, la 2^e boîte comme paire et ainsi de suite.

- 00 Mettre manuellement une boîte vide (boîte impaire) face au système de chargement des pignons
- 01 SI PO en référence (conditions initiales) et présence d'une boîte et départ de cycle
- 02 ALORS si présence pignon, charger 3 pignons dans la case 1 (à 4, 11 à 14, 21 à 24, 31 à 34) de la boîte
- 03 SI fin chargement de 3 pignons case 1 (à 4, 11 à 14, 21 à 24, 31 à 34)
- 04 ALORS avancer d'un pas suivant Xav,
- 05 SI fin avance pas suivant Xav et pas fin rangée Xav
- 06 ALORS aller en 02
- 07 SINON si présence pignon, charger 3 pignons dans la case 5 (15, 25, 35)
- 08 SI fin chargement de 3 pignons case 5 (15, 25, 35)
- 09 ALORS avancer d'un pas suivant axe Yd
- 10 SI fin avance pas suivant Yd
- 11 ALORS si présence pignon, charger 3 pignons dans la case 6 (à 9, 16 à 19, 26 à 29, 36 à 39)
- 12 SI fin chargement de 3 pignons case 6 (à 9, 16 à 19, 26 à 29, 36 à 39)
- 13 ALORS avancer d'un pas suivant axe Xar
- 14 SI fin avance pas suivant Xar et pas fin rangée Xar
- 15 ALORS aller en 11
- 16 SINON si présence pignon, charger 3 pignons dans la case 10 (20, 30, 40)
- 17 SI fin chargement de 3 pignons case 10 (20, 30, 40) et pas fin ligne Yd
- 18 ALORS avancer d'un pas suivant Yd
- 19 SINON aller en 22
- 20 SI fin pas suivant Yd
- 21 ALORS aller en 02
- 22 ALORS permuter (permutation aller) boîte pleine (boîte impaire) et boîte vide (boîte paire)
- 23 SI fin permutation aller
- 24 ALORS si présence pignon, charger 3 pignons dans la case 40 (à 37, 30 à 27, 20 à 17, 10 à 7) de la boîte
- 25 SI fin chargement de 3 pignons case 40 (à 37, 30 à 27, 20 à 17, 10 à 7)
- 26 ALORS avancer d'un pas suivant Xav
- 27 SI fin pas suivant Xav et pas fin rangée Xav
- 28 ALORS aller en 24
- 29 SINON si présence pignon, charger 3 pignons dans la case 36 (26, 16, 6)
- 30 SI fin chargement de 3 pignons case 36 (26, 16, 6)
- 31 ALORS avancer d'un pas suivant Yg
- 32 SI fin pas suivant Yg
- 33 ALORS si présence pignon, charger 3 pignons dans la case 35 (à 32, 25 à 22, 15 à 12, 5 à 2)
- 34 SI fin chargement de 3 pignons case 35 (à 32, 25 à 22, 15 à 12, 5 à 2)
- 35 ALORS avancer d'un pas suivant Xar
- 36 SI fin pas suivant Xar et pas fin rangée Xar
- 37 ALORS aller en 33
- 38 SINON si présence pignon, charger 3 pignons dans la case 31 (21, 11, 1)
- 39 SI fin chargement case 31 (21, 11, 1) et pas fin ligne Yg
- 40 ALORS avancer d'un pas suivant Yg
- 41 SINON aller en 44
- 42 SI fin pas suivant Yg
- 43 ALORS aller en 24
- 44 ALORS permuter boîte pleine et boîte vide (permutation retour)
- 45 SI fin permutation retour et présence boîte et cycle automatique
- 46 ALORS aller en 02

Cette description algorithmique se transpose, sans difficulté majeure, en un GRAFCET pratiquement linéaire.

De plus, il faut noter que le lancement répétitif des mêmes tâches aboutit à une structure typique de GRAFCET avec lancement de sous-programmes ou de tâches.

b. Version linéaire découlant de l'algorithme (fig. 9.7)

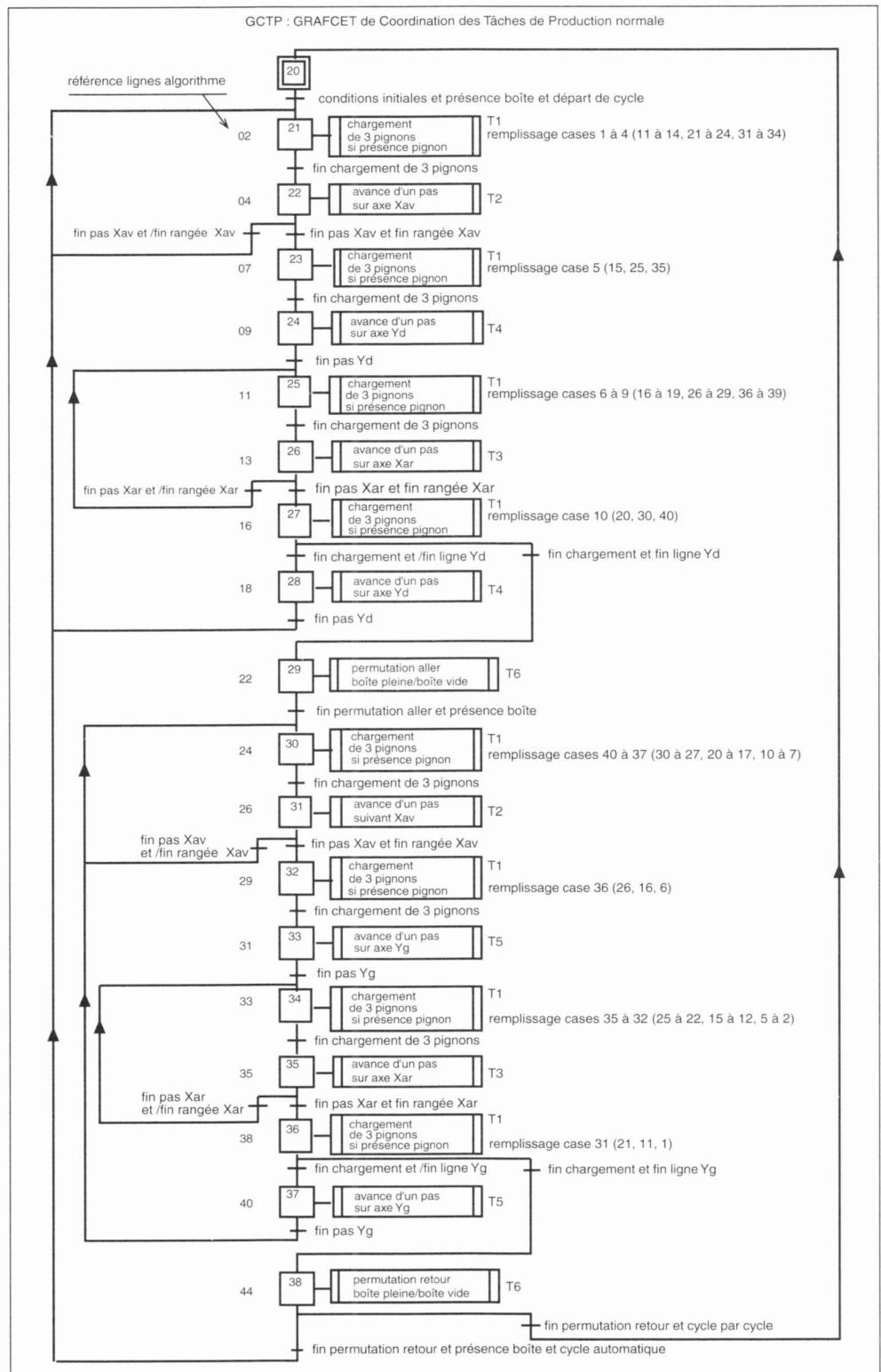


Figure 9.7
GRAFCET de Coordination des Tâches de Production normale (GCTP)(version linéaire).
Il est sous-entendu que l'opérateur, entre chaque remplissage, enlève la boîte pleine et la remplace par une boîte vide.

- Deux autres solutions approchantes sont également possibles :
- l'une en utilisant une sélection par les capteurs fin de course des tables X et Y,
 - l'autre en comptant le nombre de chargements et de déplacements suivant les axes X et Y.

c. Sélection par les capteurs fin de course des tables X et Y (fig. 9.8)

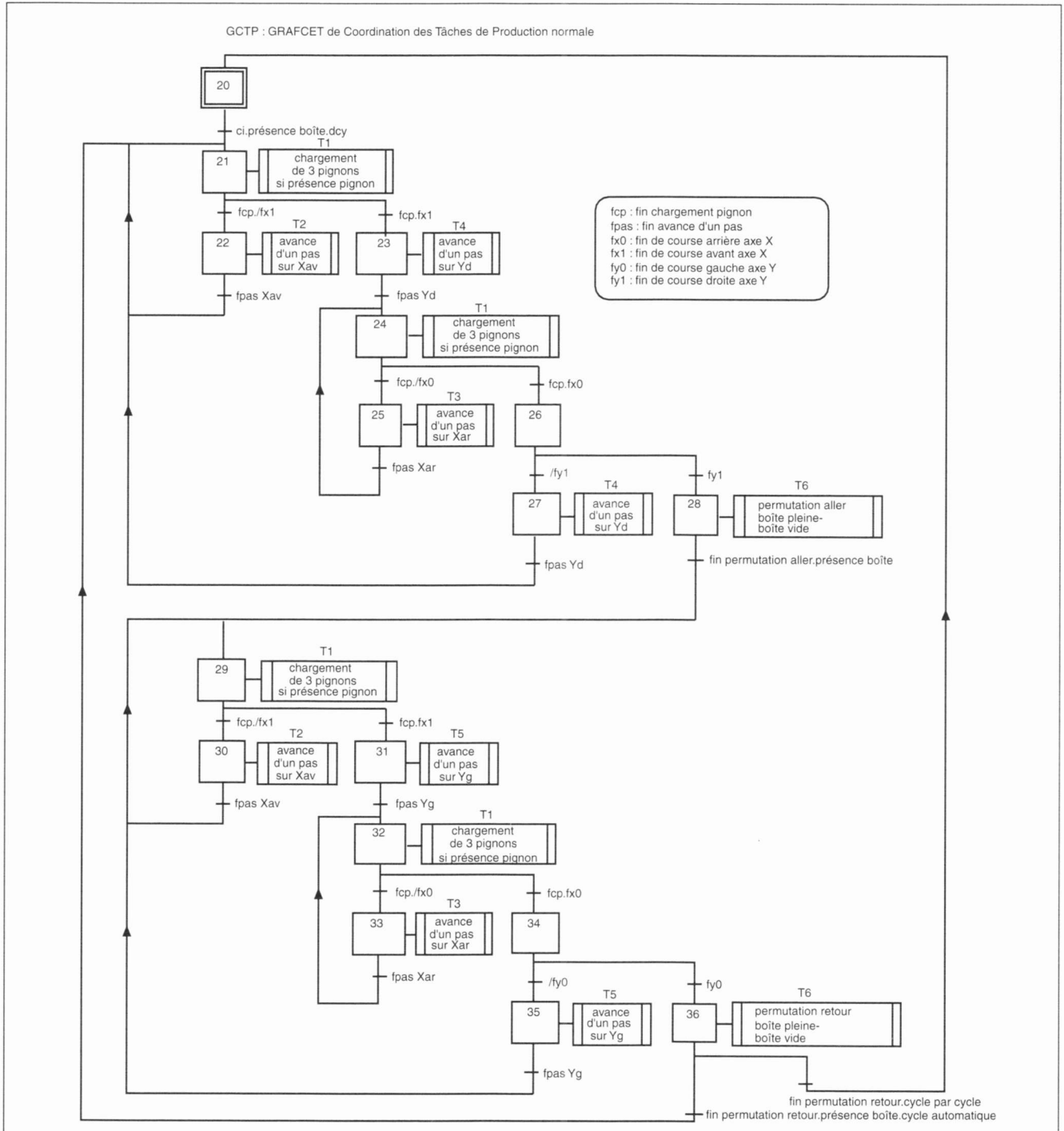


Figure 9.8 : GRAFCET de Coordination des Tâches de Production normale (GCTP), (sélection par les capteurs fin de course des tables X et Y).

d. Sélection par comptage sur les axes X et Y (fig. 9.9)

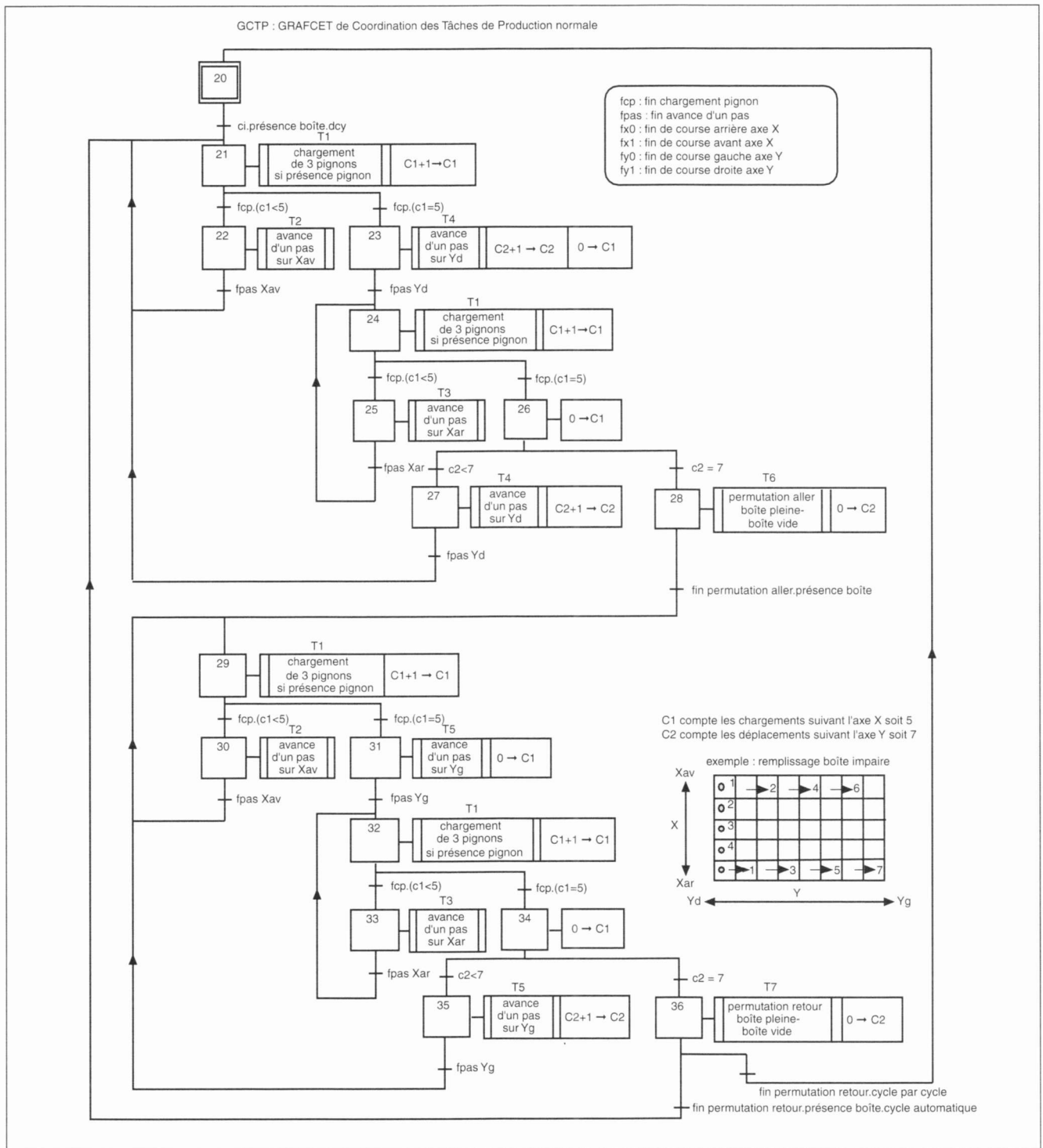


Figure 9.9 :
GRAFCET de Coordination
des Tâches de Production
normale (GCTP), (sélection
par comptage des charge-
ments suivant l'axe X et des
déplacements suivant l'axe Y).

Il est aussi possible de regrouper les tâches identiques au niveau de la même étape moyennant une recherche plus laborieuse. C'est ce qui va être fait en exploitant la notion de bits d'aiguillage dits aussi bits drapeaux étudiée au chapitre III.10.3.

e. Sélection par bits-drapeaux (fig. 9.10)

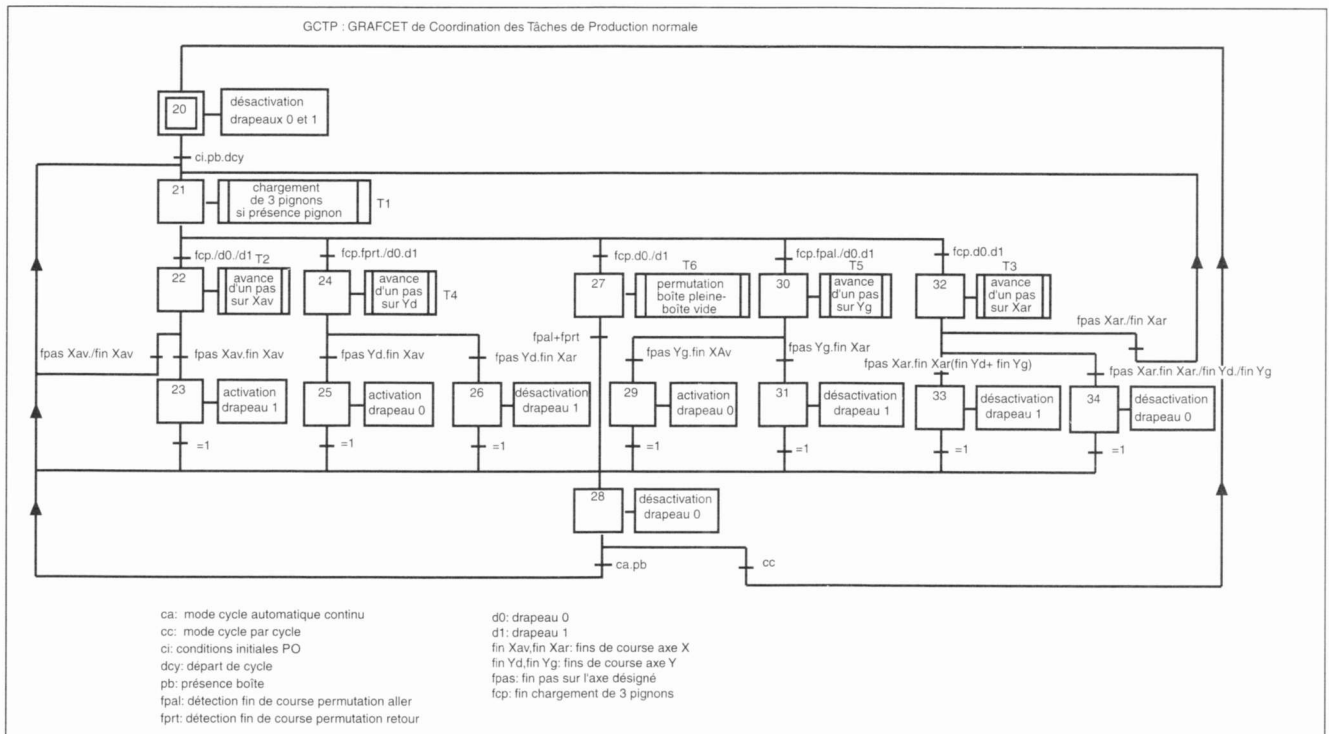


Figure 9.10
GRAFCET de Coordination
des Tâches de Production
normale (GCTP), sélection
par bits-drapeaux.

Remarque

Chacune des tâches n'étant associée qu'à une seule et unique étape, l'exploitation du concept de macro-étape est envisageable (voir fig. 9.25 et 9.26). Les tâches T6 et T7 ont, ici, été regroupées en une seule tâche T6.

2.2. Point de vue PO-PR (niveau 2)

- Choix technologique des mécanismes de la PO,
- choix des actionneurs et des préactionneurs et de leur technologie,
- choix des capteurs-machines et de leur technologie,
- étude des modes de marche et d'arrêt à l'aide du GEMMA,
- ensemble des GRAFCET et hiérarchie,
- tracé des GRAFCET de niveau 2,
- définition du pupitre de commande.

A. Choix technologique de la PO

Les déplacements pas à pas peuvent être obtenus de diverses manières, entre autres :

- par commande d'axe avec moteur à courant continu et une commande numérique,
- par commande d'axe avec moteur pas à pas,
- par commande d'axe avec déplacement par vérin.

La permutation de la boîte pleine et de la boîte vide peut être obtenue notamment :

- par un moteur rotatif continu (moteur électrique à un seul sens de marche),
- par un moteur rotatif alternatif (moteur à deux sens de marche),
- par un moteur rotatif pneumatique (moteur oscillant).

Pour des raisons d'ordre économique, **la solution retenue par le bureau d'étude fait appel à une table rotative et à translation rectiligne X-Y réalisée à partir d'éléments modulaires standard en technologie pneumatique.**

Ses caractéristiques essentielles sont les suivantes :

- plateau rotatif pneumatique oscillant à 2 positions à 180°,
- mouvements rectilignes orthogonaux X et Y obtenus par avance pneumatique,
- avance pas à pas réglable à l'aide de 2 vérins assurant l'avance et l'indexage,
- indexage mécanique de chaque pas sur réglettes interchangeables, évitant l'addition des jeux,
- blocage de la table par vérin,
- vérins à piston magnétique et capteurs fin de course ILS (Interrupteur à Lame Souple),
- guidage en double colonne à roulements axiaux de haute précision,
- courses X et Y disponibles en standard : 300, 400, 500, 600 et 700 mm.

Le système de chargement retenu utilise un vérin pneumatique avec un embout autocentreur. Le pignon arrive par la goulotte de la table vibrante et se positionne en butée sur un vé de centrage.

B. Disposition du poste de conditionnement des pignons (fig. 9.11)

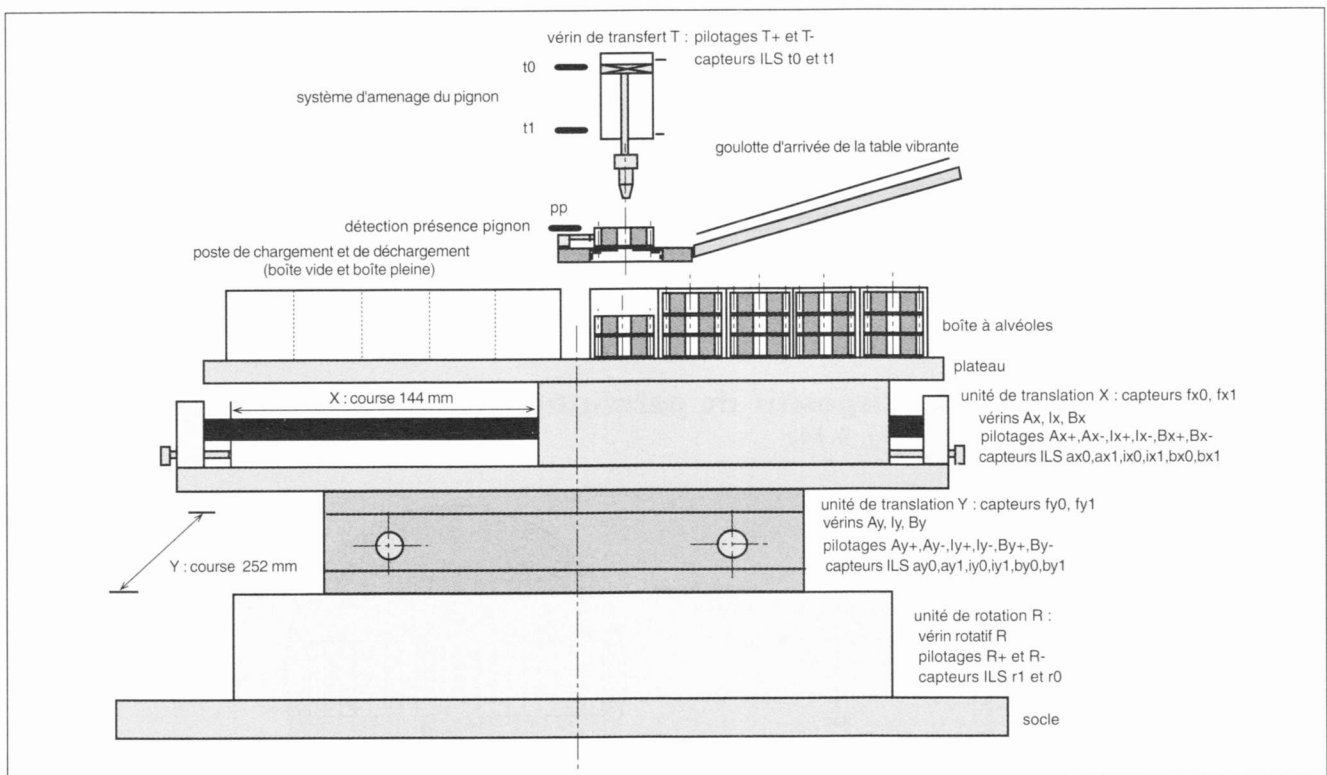


Figure 9.11
Poste de conditionnement
des pignons.

C. Détails du système de chargement (amenage) d'un pignon (figure 9.12)

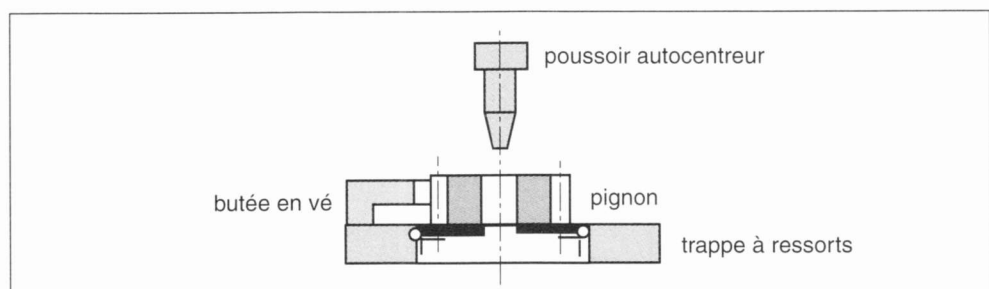


Figure 9.12
Système de chargement
d'un pignon (amenage).

D. Principe de l'avance pas à pas par indexage (fig. 9.13)

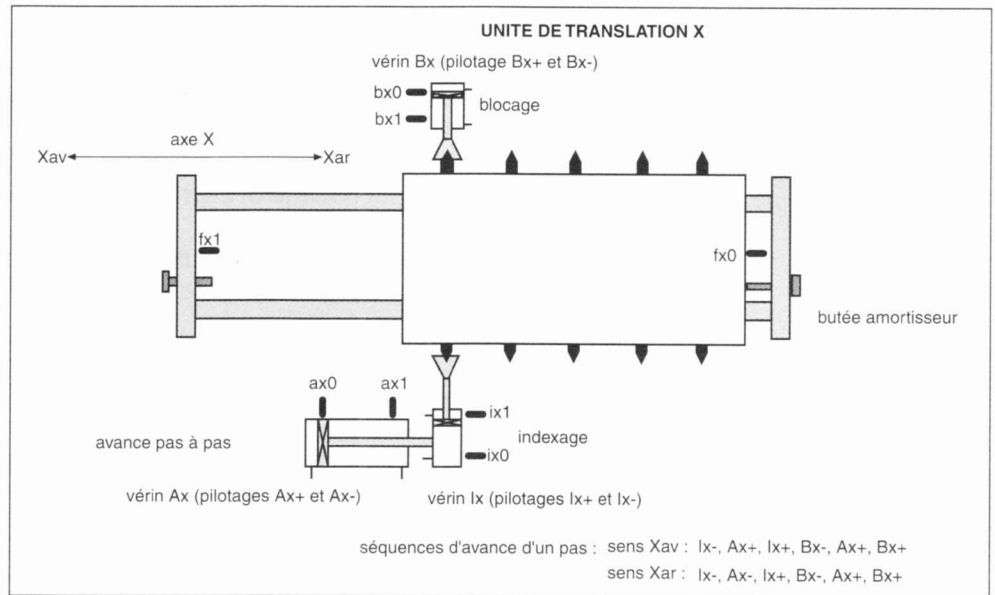


Figure 9.13
Principe de l'avance
pas à pas par indexage.

L'unité de translation Y est conçue sur le même principe, le nombre de plots d'indexage sur la barrette étant de 8 au lieu de 5 pour l'axe des X.

Pour le repérage, il suffit de remplacer la lettre X par la lettre Y :

- vérin d'avance pas à pas Ay (pilotages Ay+ et Ay-, capteurs ay1 et ay0),
- vérin d'indexage Iy (pilotages Iy+ et Iy-, capteurs iy1 et iy0),
- vérin de blocage By (pilotages By+ et By-, capteurs by1 et by0).

E. Dispositif de permutation boîte pleine-boîte vide (fig. 9.14)

(fig. 9.14)

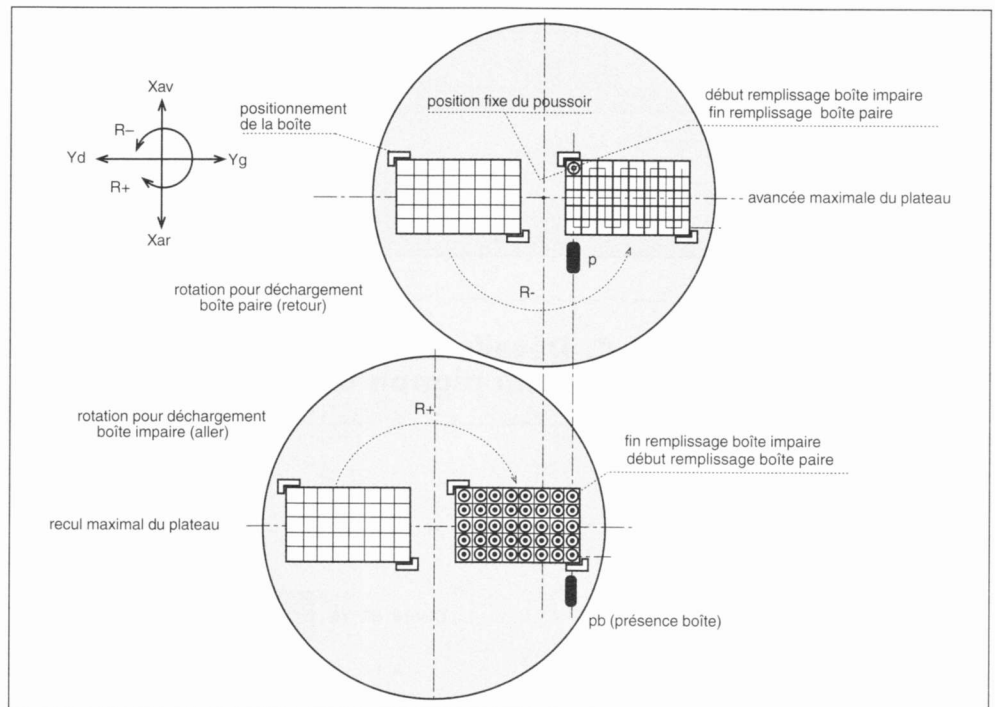


Figure 9.14
Dispositif de permutation
boîte pleine-boîte vide.

F. Choix des actionneurs et des préactionneurs

Comme il été vu, le système d'avance pas à pas des tables X et Y est obtenu avec des vérins pneumatiques à double effet.

Le système de chargement, par souci d'unification, fait appel lui aussi à un vérin pneumatique à double effet. Les préactionneurs dans l'optique d'une PC réalisée avec un automate programmable seront du type à pilotage électropneumatique.

Pour des raisons de sécurité (maintien en position des vérins en cas de coupure de courant notamment), la préférence a été donnée à des distributeurs à double pilotage électropneumatique.

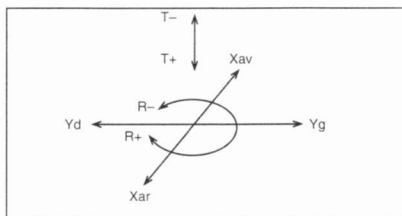
G. Choix des capteurs-machines

Les capteurs fin de course des actionneurs retenus sont des interrupteurs à lame souple dits ILS commandés par le piston magnétique du vérin.

Les capteurs fin de course des tables X et Y et de présence d'un pignon pour des raisons de fiabilité sont du type inductif.

La détection de la boîte est obtenue avec une cellule photoélectrique infrarouge.

H. Repérage des tâches et des mouvements (fig. 9.15)



T1 : aménagement de 3 pignons dans la boîte (T+ et T-),

T2 : déplacement table X vers l'avant (Xav),

T3 : déplacement table X vers l'arrière (Xar),

T4 : déplacement table Y vers la droite (Yd),

T5 : déplacement table Y vers la gauche (Yg),

T6 : rotation du plateau aller-retour (R+ et R-),

Figure 9.15
Repérage des mouvements.

Les tâches permutation aller T6 et retour T7 ont été regroupées en une tâche unique T6' compte tenu de la solution technique adoptée.

I. Recherche des modes de marche et d'arrêt à l'aide du GEMMA

a. Notions sur le GEMMA

Le **Guide d'Etude des Modes de Marche et d'Arrêt (GEMMA)** est un outil-méthode développé par l'Agence nationale pour le **DE**veloppement de la **P**roduction **A**utomatisée (**ADEPA**).

Son objectif est la prise en compte des spécifications opérationnelles relatives aux modes de marche et d'arrêt et aux procédures de sécurité. La grille du GEMMA s'utilise dans l'esprit « check-list ». Elle permet :

- de passer en revue systématiquement les différents modes de marche envisageables,
- de voir quels sont ceux susceptibles d'être retenus pour le système en cours d'étude, compte tenu du cahier des charges (souhaits du client),
- de chercher les conditions d'évolutions entre eux.

Il en va de même en ce qui concerne les procédures de sécurité.

Les concepts du GEMMA ne seront pas étudiés ici (*). Pour le moment, on se contentera de présenter la grille du GEMMA et de commenter son exploitation pour cette application.

Note

(*) Consulter l'ouvrage « LE GEMMA » édité dans la même collection.

b. Disposition générale de la grille du GEMMA (fig. 9.16)

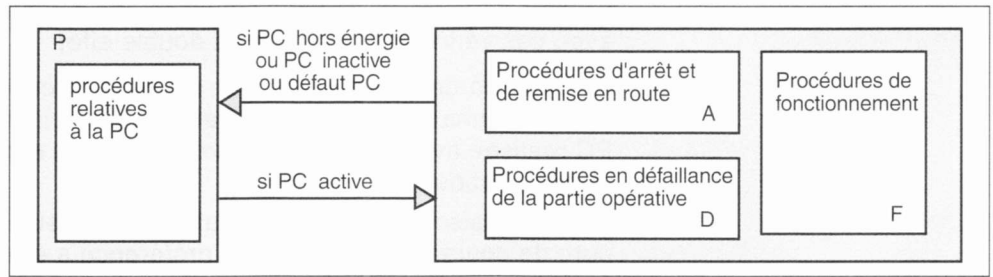


Figure 9.16
Disposition générale
de la grille du GEMMA.

On distingue trois grandes familles de procédures repérées par une lettre :

- A : les procédures d'arrêt et de remise en route,
- F : les procédures de fonctionnement,
- D : les procédures en défaillance de la partie opérative.

Toutes ces procédures concernent essentiellement la PO et ont lieu avec la PC active, c'est-à-dire en état de marche. Sinon, on revient en PZ, zone où sont regroupées toutes les procédures relatives à la PC.

c. Grille complète du GEMMA (fig.9.17)

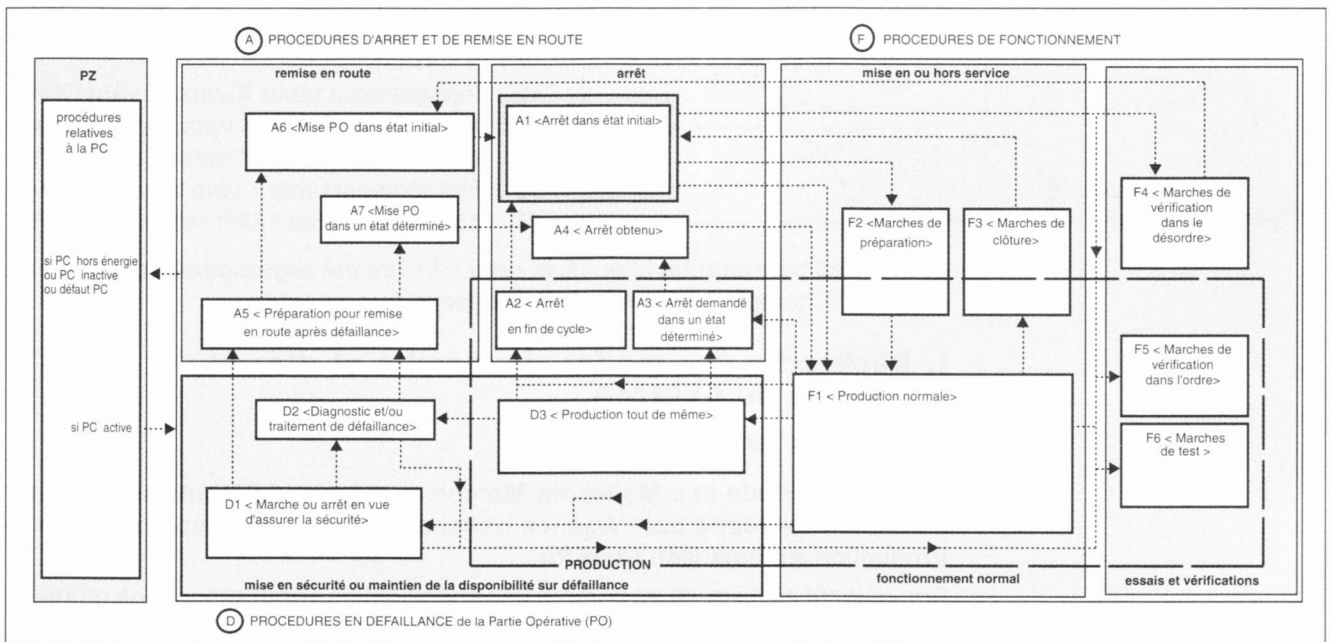


Figure 9.17
Grille complète du GEMMA.

Chacune des cases repérée par une lettre (A, D, F) suivie d'un numéro s'appelle un rectangle-état. La lettre rappelle à quelle famille de procédure appartient le rectangle-état.

Exemple

F1 est le rectangle-état où sera décrite la procédure de production normale.

Une liaison est à tracer entre deux rectangles-états (ou plus simplement entre deux états) lorsque ceux-ci ont retenu l'attention du concepteur. L'évolution suivant cette liaison sera liée ou non à une condition (expression logique) à déterminer. Cette condition sera écrite près d'un tiret comparable au tiret de la transition du GRAFCET.

La grille du GEMMA suggère seulement les principales liaisons entre les états. D'autres liaisons sont réalisables et justifiables suivant les besoins.

E *exemple* : évolutions entre états sur le GEMMA

Partant de l'état A1 <Arrêt dans l'état initial>, il faut aboutir à l'état F1 <Production normale> en passant ou non par l'état F2 < Marches de préparation>.

En fin d'exécution de F1, il faut revenir en A1 en passant ou non par l'état F3 < Marches de clôture>.

Pour la mise au point ou les réglages, partant de A1 ou de F1 possibilité d'aller dans les états F4 <Marches de vérification dans le désordre>, F5 <Marches de vérification dans l'ordre> ou F6 < Marches de test>.

Étant en F1, en cas de demande d'arrêt d'urgence, évolution vers l'état D1 <Marche ou arrêt en vue d'assurer la sécurité>.

Ensuite de nombreuses solutions sont envisageables soit pour revenir en A1 soit pour reprendre la production normale en F1.

Il faut absolument que l'automatisme se trouve dans un seul état à la fois ce qui signifie, en clair, qu'un seul mode doit être actif au même moment. On pourra alors affirmer qu'il y a « unicité de mode » au niveau du fonctionnement.

Par exemple, il faut que les conditions d'évolutions entre l'état A1 et les états F1 et F4 soient contradictoires (verrouillage logique et/ou technologique par sélecteur) pour que l'automatisme ne puisse se trouver à la fois dans l'état F1 <Production normale> ou Mode Auto et dans l'état F4 <Marches de vérification dans le désordre> ou Mode Manuel.

Si le GEMMA n'apporte pas directement la solution quant au choix des modes de marche et d'arrêt, il permet par contre de réfléchir aux différentes procédures envisageables et de retenir celles qui conviennent le mieux au poste à automatiser et au respect de son cahier des charges. Parallèlement, on détermine les conditions d'évolution entre les états pour que l'unicité de mode soit respectée.

Même avec une grille au format A4 ou A3, la place disponible dans les rectangles-états est relativement restreinte pour y décrire la procédure liée à chaque état. Il est plutôt conseillé de travailler hors de la grille du GEMMA en utilisant la méthode dite dite « par boucles fonctionnelles séparées » (*).

C'est cette méthode qui va être exploitée ici pour l'analyse des modes de marche et d'arrêt du poste de conditionnement des pignons.

J. Analyse des modes de marche et d'arrêt

a. Boucle de production avec arrêt en fin de cycle (fig. 9.18)

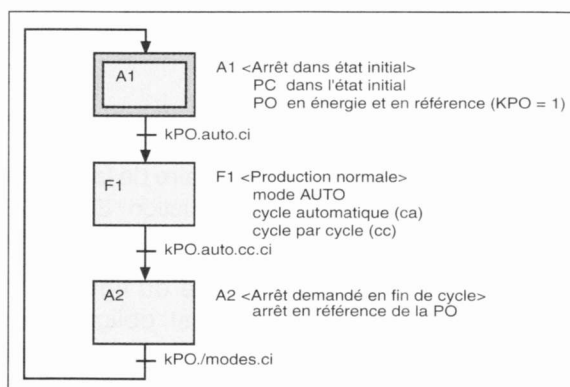


Figure 9.18
Boucle de production normale.

Note

(*) Consulter l'ouvrage consacré au GEMMA et à la conception de SAP sûrs, édité dans la même collection.

b. Boucle des marches de vérification dans le désordre

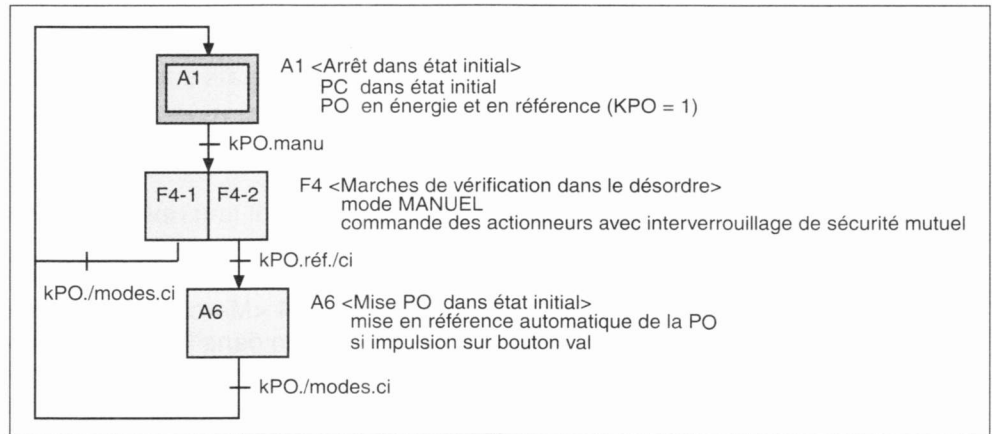


Figure 9.19
Boucle des marches
de vérification
dans le désordre.

La sortie du mode manuel (état F4) envisagée ici, suppose que la PO est ou non en référence (vérification des conditions initiales : ci = 1 ou 0).

La mise en référence en mode manuel risque de poser un problème au technicien de maintenance s'il ne connaît pas parfaitement ce poste de travail.

La mise en référence automatique (état A6) élimine ce problème. Le déclenchement de la mise en référence est obligatoirement dépendante, après sélection de ce mode, d'une impulsion sur le bouton val (validation) (directive 89/392/CEE concernant la sécurité, article 1.2.3).

c. Boucle figeage des GRAFCET de Production et mise en référence de la PO (fig. 9.20)

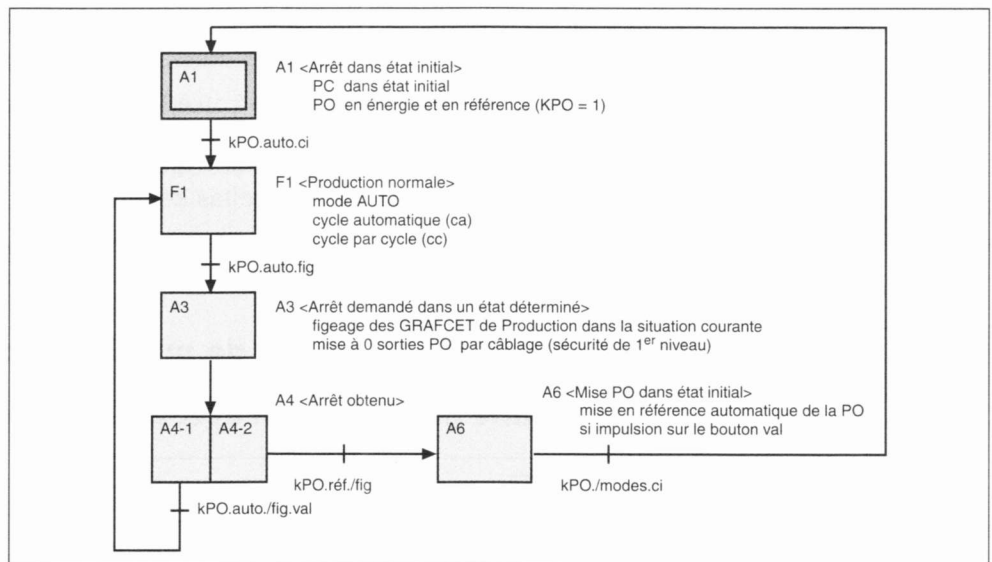


Figure 9.20
Boucle figeage PC et mise en
référence PO.

Le figeage demandé par l'intermédiaire de la variable fig (sélecteur du pupitre) ne concerne que les GRAFCET de Production : GRAFCET de Coordination des Tâches de Production normale GCTP et GRAFCET des tâches GT1 à GT6 (voir plus loin).

La reprise de la production (sortie du figeage) ne peut être effective après désélection de la variable fig que si, obligatoirement, une impulsion est donnée sur le bouton val (validation) situé sur le pupitre de commande (directive 89/392/CEE concernant la sécurité, article 1.2.3).

d. Boucle de défaillance (fig. 9.21)

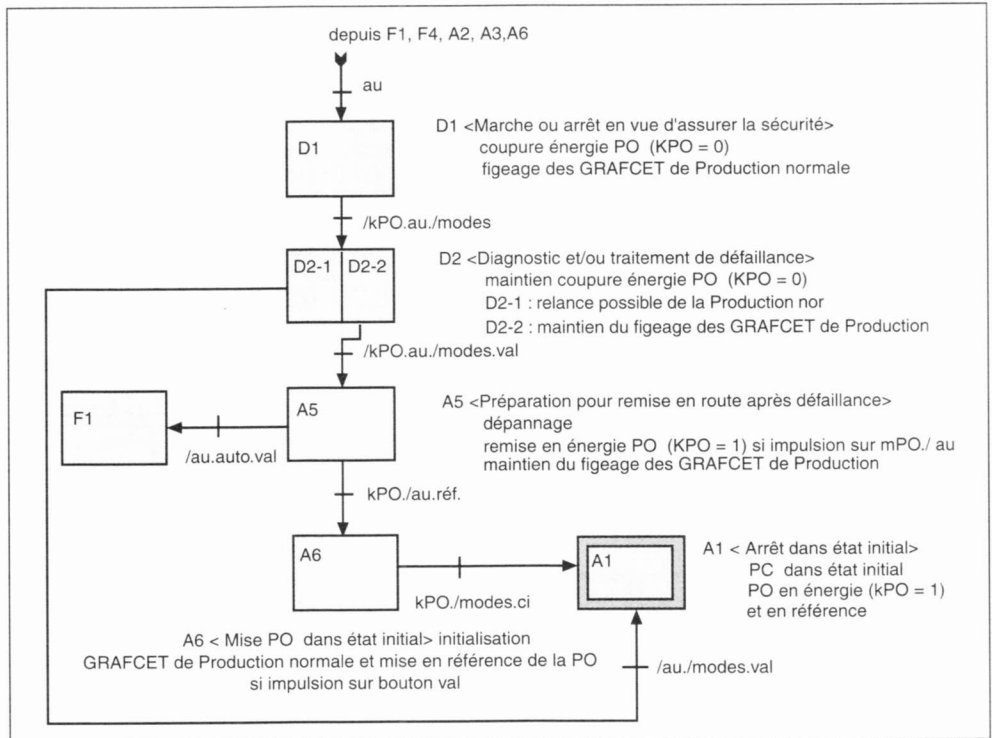


Figure 9.21
Boucle de défaillance.

Le dédoublement des états F4, A4 et D2 est justifié par le fait que deux évolutions ont été admises à partir de ces états.

K. Hiérarchie des GRAFCET (fig. 9.22)

Le GRAFCET de Sécurité (GS) se déduit de la boucle en défaillance.

Le GRAFCET de Conduite (GC) se déduit des autres boucles précédentes.

Le GRAFCET de Coordination des Tâches de Production normale (GCTP) se déduit du GRAFCET de niveau 1.

Les autres GRAFCET sont à étudier en fonction du processus du conditionnement.

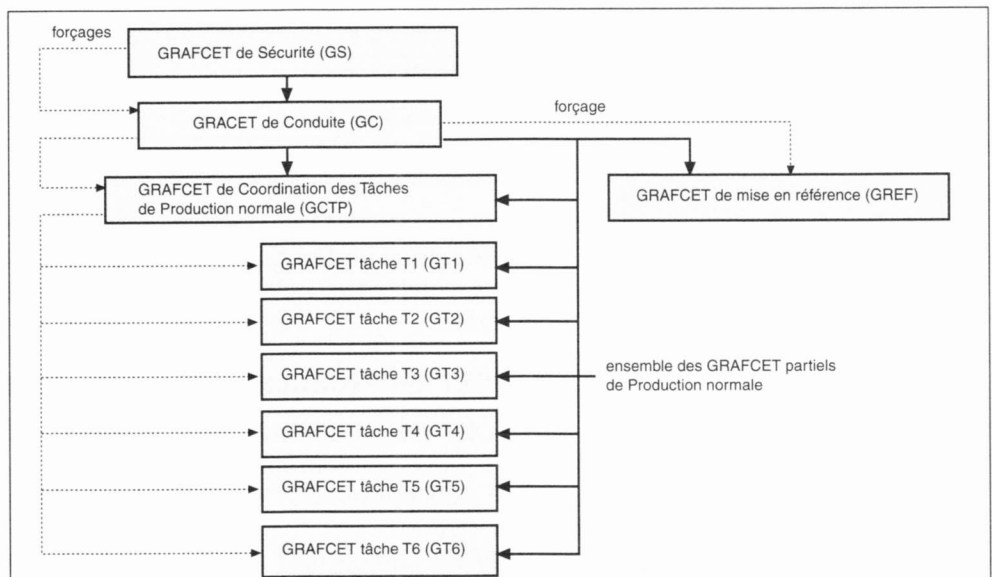


Figure 9.22
Hiérarchie des GRAFCET.

L. Tracé des GRAFCET

a. GRAFCET de Sécurité (GS) et de Conduite (GC) (fig. 9.23)

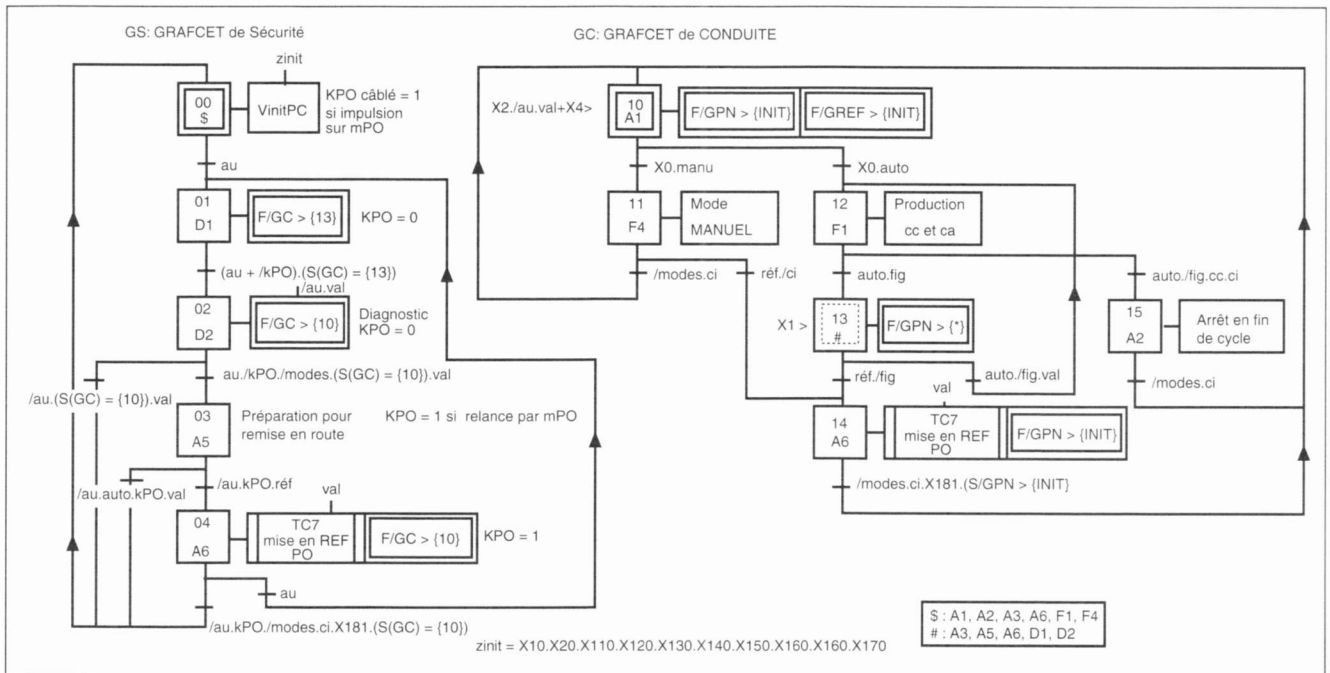


Figure 9.23
GRAFCET de Sécurité (GS)
et GRAFCET de Conduite
(GC), niveau 2.

La méthode utilisée pour le figeage des GRAFCET de Production normale demandée dans l'état D1 (situation $S(GPN) = \{1, 13\}$) dépendra du constituant retenu pour réaliser la P.C. ou du procédé préféré de l'automaticien.

Dans le cas d'une PC réalisée avec un API, différentes possibilités peuvent être offertes en fonction du langage disponible (voir chapitre III.13.3).

Vérification de l'unicité de mode

états A	situations	états F	situations	états D	situations
A1	$S(GS, GC) = \{0, 10\}$	F1	$S(GS, GC) = \{0, 12\}$	D1	$S(GS, GC) = \{1, 13\}$
A2	$S(GS, GC) = \{0, 15\}$	F4	$S(GS, GC) = \{0, 11\}$	D2	$S(GS, GC) = \{2, 13\}$
A3 = A4	$S(GS, GC) = \{0, 13\}$				
A5	$S(GS, GC) = \{3, 13\}$				
A6	$S(GS, GC) = \{4, 13\} + \{0, 14\}$				

À chaque état du GEMMA retenu ici correspond une situation unique des GRAFCET de Sécurité et de Conduite. On constate également que toutes les évolutions sont exclusives au niveau des réceptivités et par suite l'unicité de mode est confirmée.

b. GRAFCET de Production Normale (GPN)

Le GRAFCET global de Production Normale correspond à un ensemble de GRAFCET partiels comprenant :

- un GRAFCET partiel GCTP de coordination des tâches de Production normale T1 à T6',
- six GRAFCET partiels GT1 à GT6' gérant respectivement les tâches T1 à T6'.

1/ GRAFCET de Coordination des Tâches de Production normale (GCTP)

Ce GRAFCET se déduit de l'un des GRAFCET tracés au niveau 1.

Quatre solutions ont été proposées :

- une première solution dite version linéaire,
- une deuxième solution dite avec sélection par les capteurs fins de course des tables X et Y,
- une troisième solution, dite avec décomptage des déplacements sur les axes X et Y,
- une quatrième solution dite avec sélection par bits drapeaux.

A titre comparatif, seules les solutions 1 et 4 ont été retenues sans trancher pour l'une ou pour l'autre. La solution 1 permet d'exploiter le concept de sous-programme tandis que la solution 2 est exploitable avec le concept de tâche ou celui de macro-étape.

• GRAFCET de Coordination des Tâches de Production normale (GCTP), version linéaire, solution avec séquences répétitives (GRAFCET sous-programmes) (fig. 9.24)

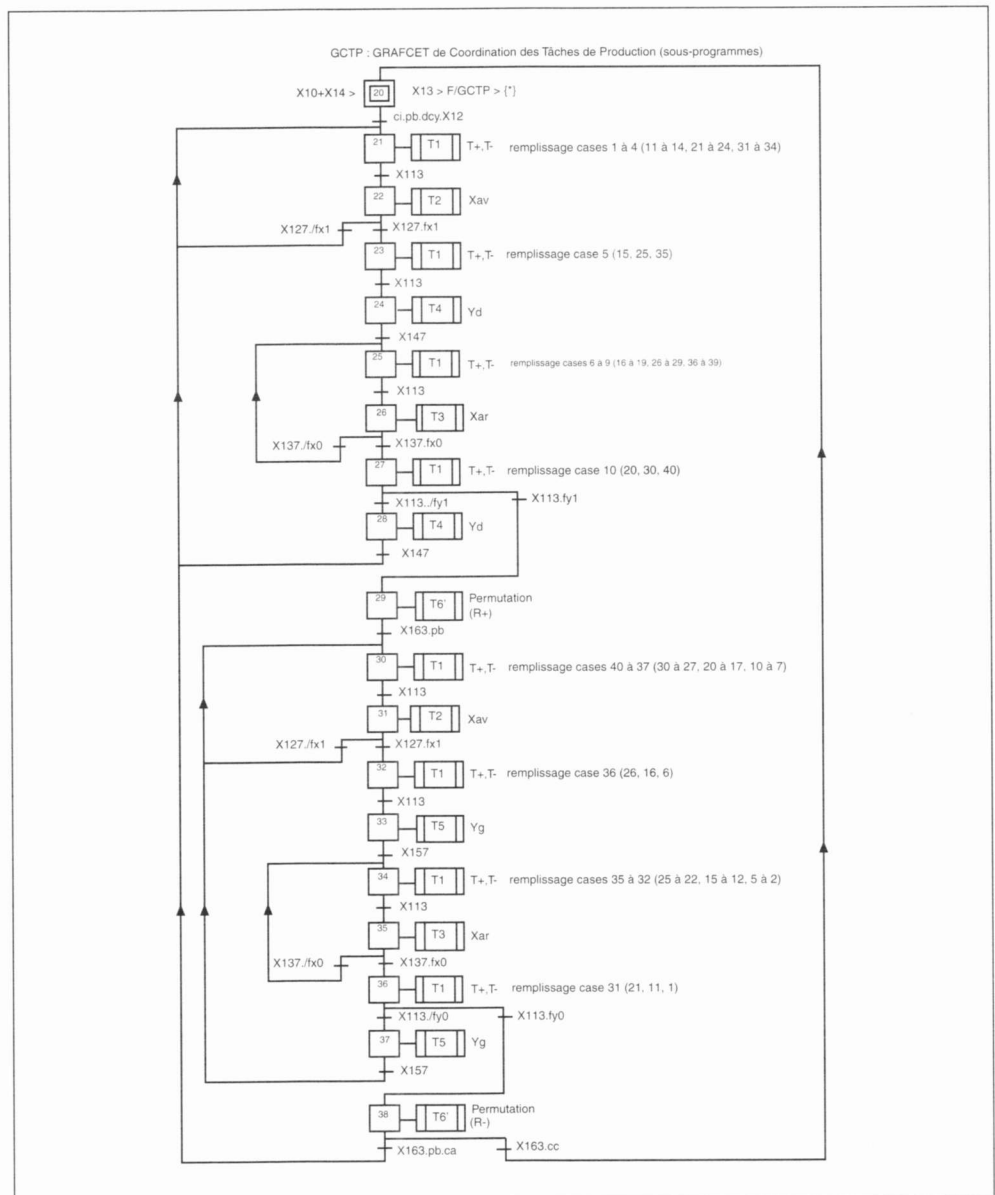


Figure 9.24
GRAFCET de Coordination
des Tâches de Production
normale (GCTP),
version linéaire, solution
avec séquences répétitives.

• **GRAF CET de Coordination des Tâches de Production normale (GCTP), sélection par bits drapeaux, solution avec GRAFCET de tâches** (fig. 9.25)

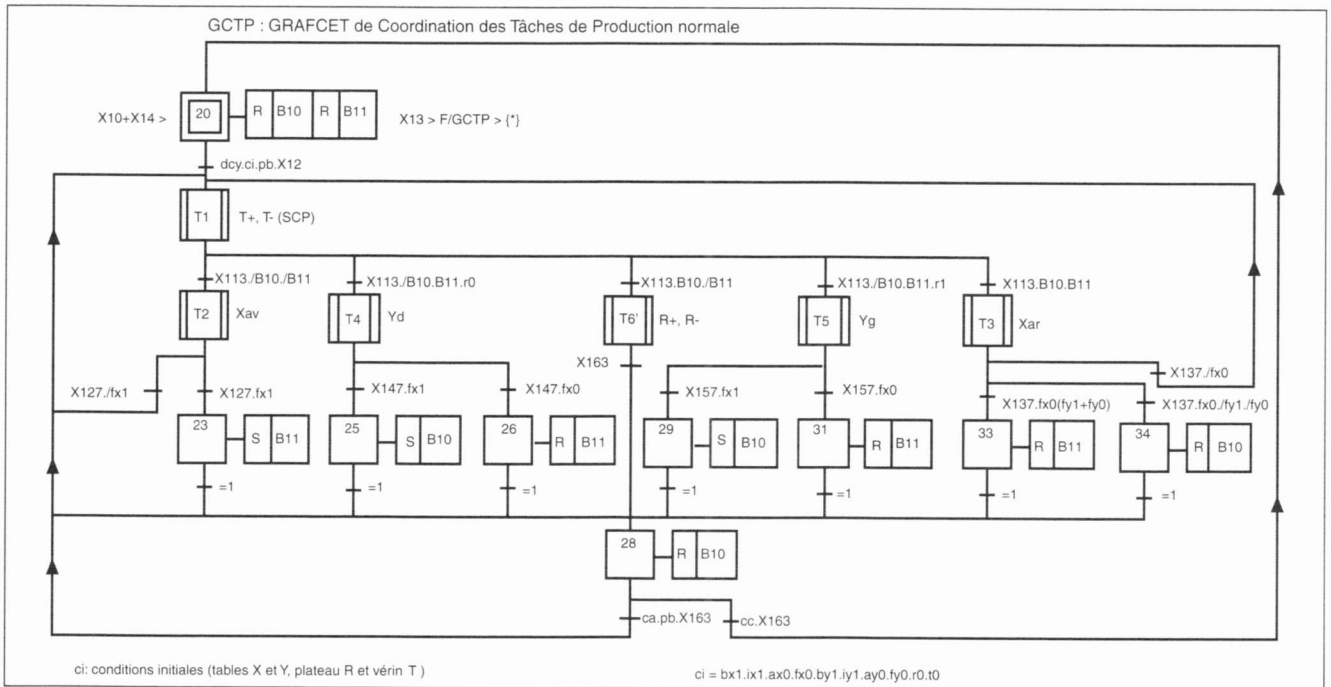


Figure 9.25
 GRAFCET de Coordination des Tâches de Production normale (GCTP), sélection par bits-drapeaux, solution avec GRAFCET de tâches

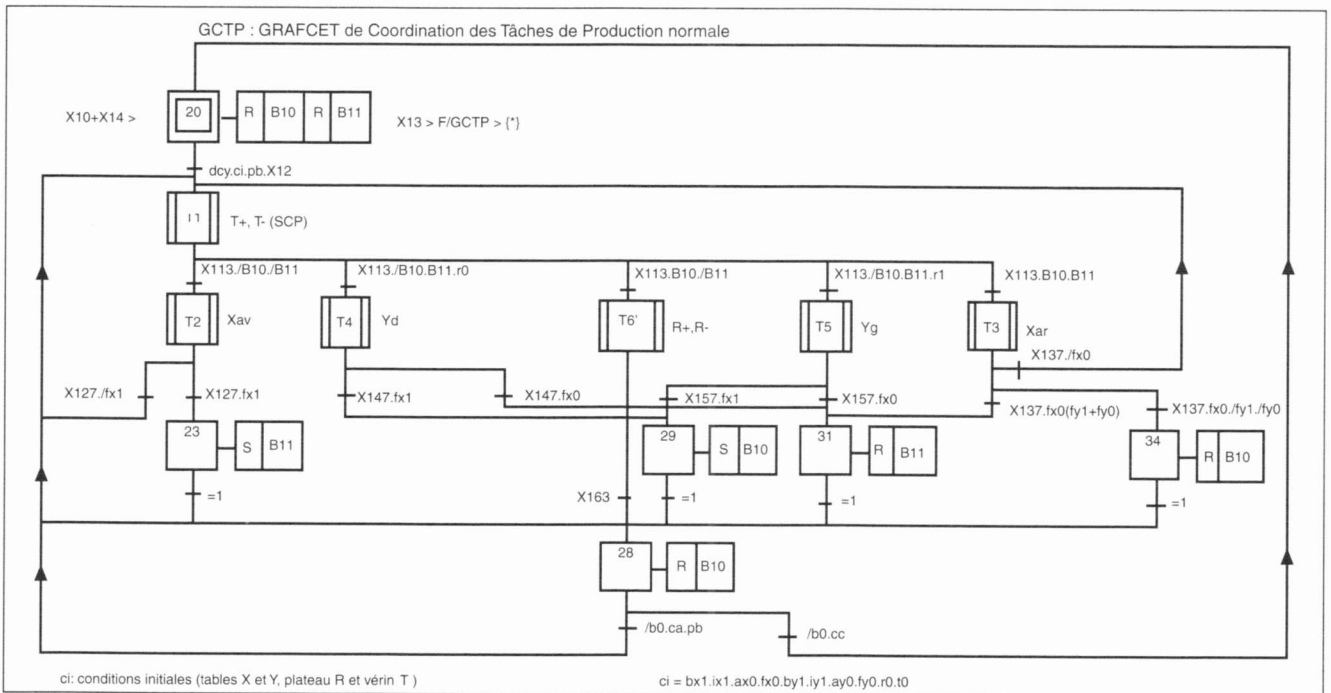


Figure 9.26
 GRAFCET de Coordination des Tâches de Production normale (GCTP), solution avec GRAFCET de tâches après fusionnement d'étapes.

Remarquer qu'il est possible de fusionner les étapes 25 et 29 puisqu'elles sont associées à la même action Set de B10 et ont la même transition en sortie d'étape : même réceptivité toujours vraie (=1) et même destination (étape 21). Il en est de même des étapes 26, 31 et 33 : même action Reset de B11 et même transition en sortie d'étape : même réceptivité (=1) et même destination (étape 21). Après fusionnement des étapes 25 et 29, des étapes 26, 31 et 33, des étapes 27 et 28 seules subsistent respectivement les étapes 29, 31 et 27 (fig. 9.26). Les tâches étant non répétitives, on peut substituer aux étapes 21, 22, 24, 27, 30 et 32 les macro-étapes respectives M1 à M6 auxquelles sont systématiquement associées les expansions XM1 à XM6. Noter que le forçage d'un GRAFCET de tâche voire d'un GRAFCET sous-programme pose moins de problèmes que le forçage de situation d'une macro-étape et de son expansion. Le langage PL3 V5 des TSX Schneider, par exemple, dispose d'un bit système SY24, qui mis à l'état 1, désactive la macro-étape si le bit correspondant du mot-système (SW8 à SW11) est lui-même mis à 0. La situation de l'expansion correspondante devient vide. Pour le figeage, une solution existe avec le verrouillage des transitions des expansions par l'information /X13 (voir exemple page 94, fig. 3.119).

2/ GRAFCET des tâches T1 à T6 (fig. 9.27 à 9.32)

Figure 9.27
GRAFCET GT1 de la tâche
T1 : chargement
de 3 pignons.

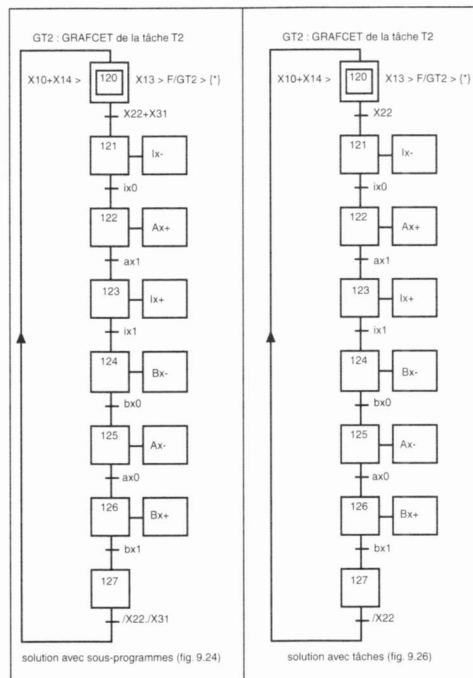
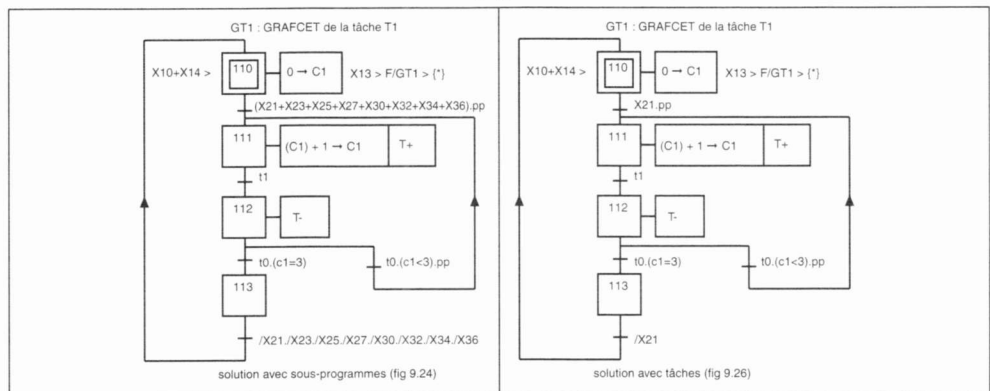


Figure 9.28 : GRAFCET GT2 de la tâche T2,
déplacement sur l'axe Xar.

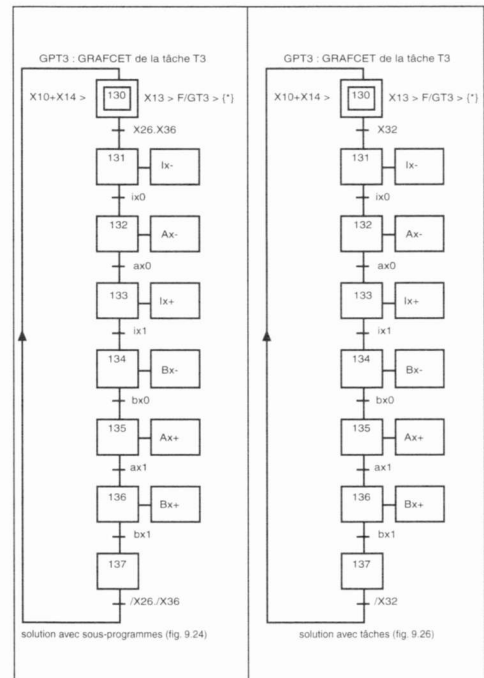


Figure 9.29 : GRAFCET GT3 de la tâche T3,
déplacement sur l'axe Xav.

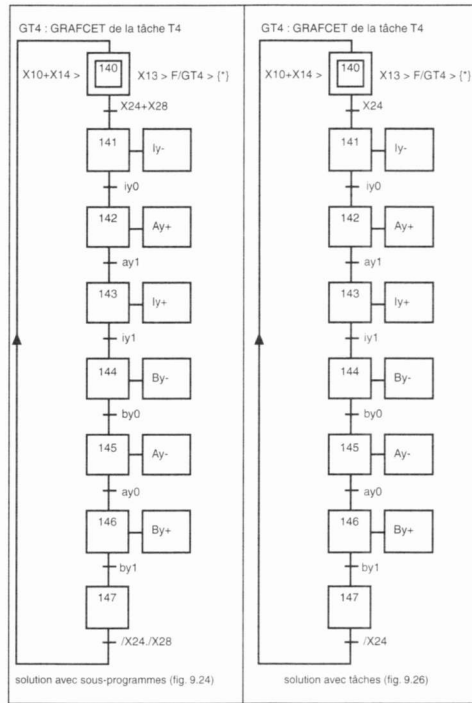


Figure 9.30 : GRAFCET GT4 de la tâche T4, déplacement sur l'axe Yd.

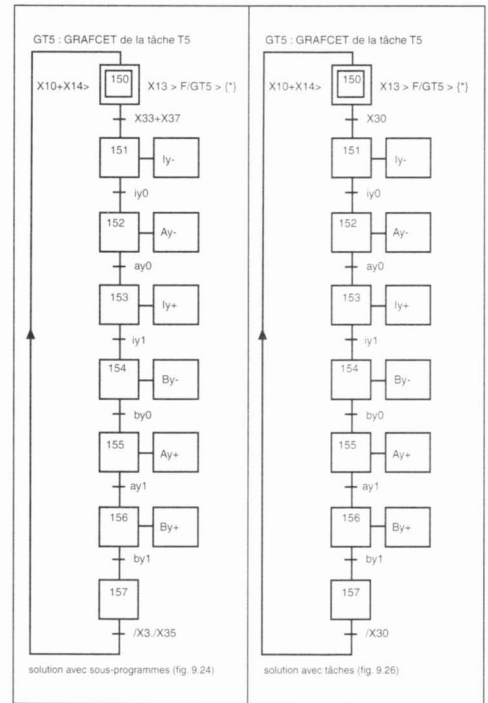


Figure 9.31 : GRAFCET GT5 de la tâche T5, déplacement sur l'axe Yg.

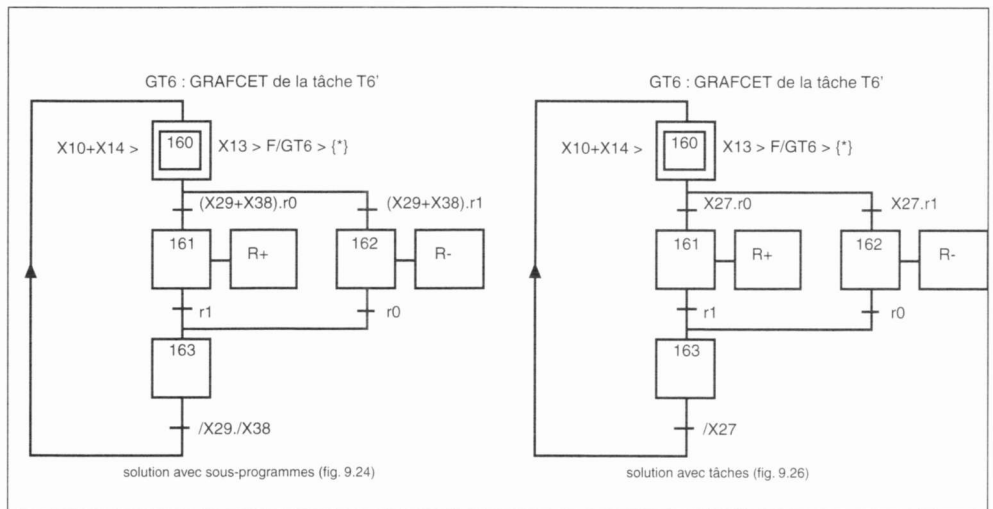


Figure 9.32
GRAFCET GT6 de la tâche
T6' : permutation
aller-retour.

Comme dans le GRAFCET de coordination des tâches, le figeage, c'est-à-dire l'arrêt des évolutions dans la situation courante, est obtenu si $X13 = 1$.

c. GRAFCET de mise en référence (fig. 9.33)

La mise en référence adoptée autorise la commande simultanée des deux axes X et Y. Cette solution a pour but d'abrégier la durée d'exécution de l'ensemble de cette opération.

Noter que l'étape 181 peut être supprimée. Dans ce cas, la réceptivité de la nouvelle transition 180, $189 \rightarrow 170$ sera : /X4./X14.

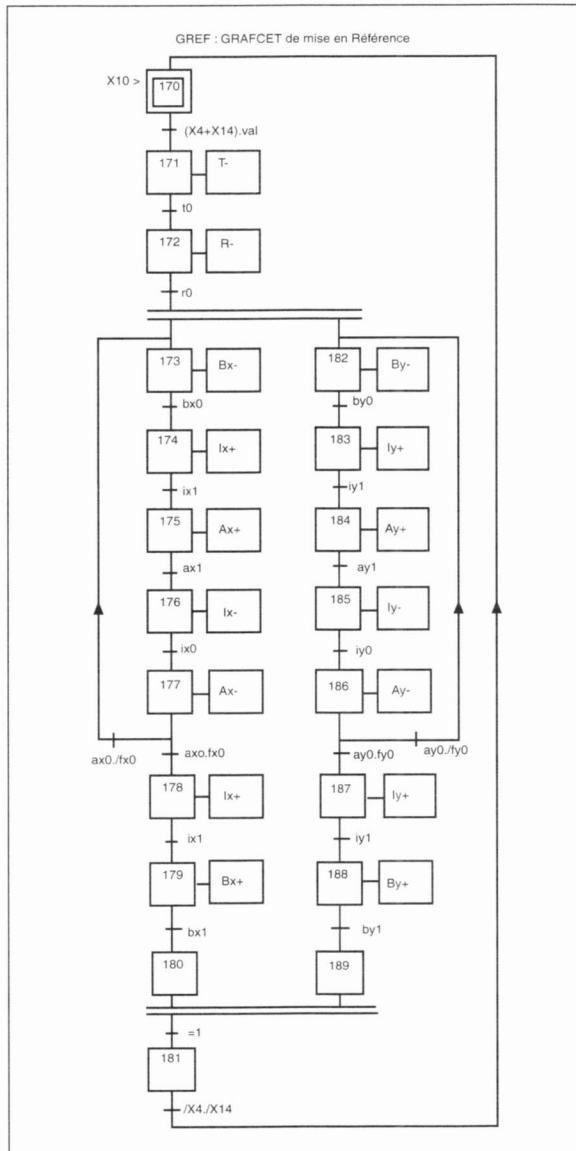


Figure 9.33
GRAFCET GREF de mise
en référence de la PO.

M. Détails du mode manuel (fig. 9.34)

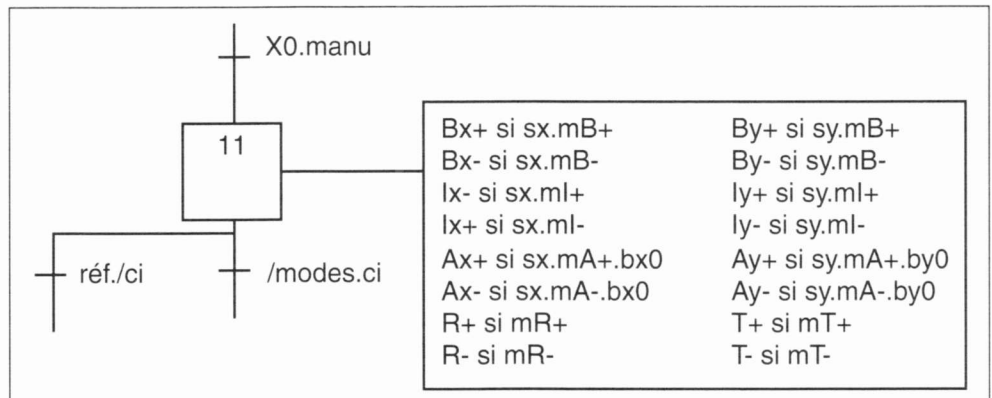


Figure 9.34
Détail des commandes
du mode manuel.

Le sélecteur d'axes SX, SY permet de réduire le nombre de boutons poussoirs du mode manuel, donc du pupitre de commande, et le nombre d'entrées du constituant de la PC.

N. Définition et organisation du pupitre de commande

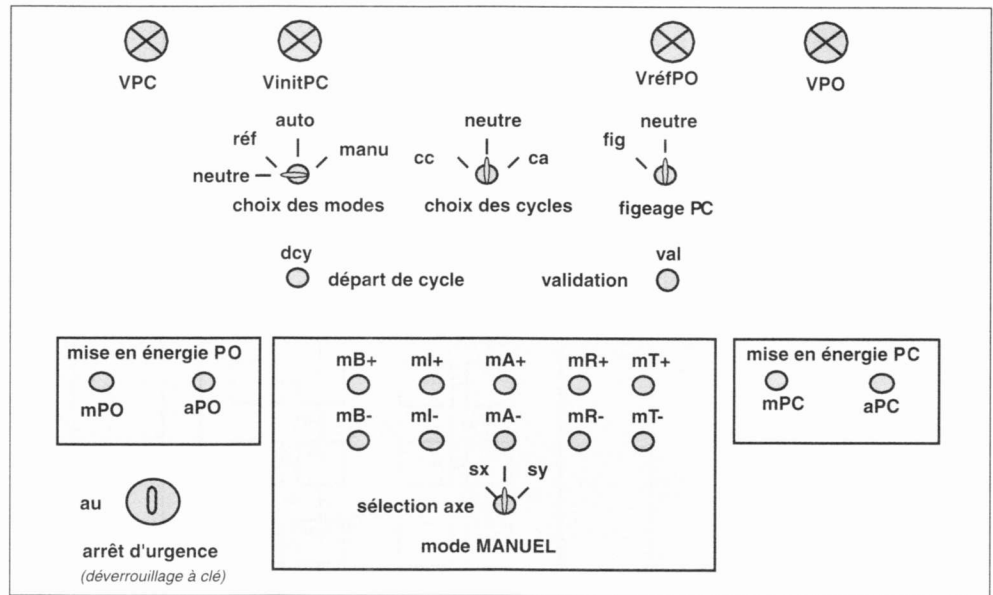


Figure 9.35
Organisation du pupitre.

2.3. Point de vue réalisateur (niveau 3)

- inventaire des entrées-sorties,
- choix du constituant de la PC,
- expressions logiques des sorties,
- sécurité câblée,
- gestions des bits d'aiguillage (éventuellement).

A. Inventaire des entrées-sorties à gérer par la PC

48 entrées

Sélecteurs

auto, manu, réf, cc, ca, fig, sx, sy.

Boutons poussoirs

lancement : dcy, val,

mise en énergie : mPO, aPO, mPC, aPC

mode manuel : mB+, mB-, ml+, ml-, mA+, mA-, mR+, mR-, mT+, mT-,

coup de poing à clé : au.

Capteurs ILS fin de course

axe X : bx0, bx1, ix0, ix1, ax0, ax1,

axe Y : by0, by1, iy0, iy1, ay0, ay1,

rotation : r0, r1,

amenage pignon : t0, t1,

Capteurs inductifs

fin de course axes X et Y : fx0, fx1, fy0, fy1,

présence pignon : pp,

Capteur à cellule photoélectrique : présence boîte : pb.

Contact de relais électromagnétique : surveillance énergie sur la PO : kPO

18 sorties

Pilotages des distributeurs

Bx+, Bx-, lx+, lx-, Ax+, Ax-

By+, By-, ly+, ly-, Ay+, Ay-,

R+, R-, T+, T-.

Voyants : VréfPO, VinitPC

Les voyants VPC et VPO sont seulement liés à la mise en énergie et non comptabilisés ici.

Étapes

90, 86, 82 ou 81 étapes suivant solution retenue pour les GRAFCET GPCT et GREF :

GS : 5, GC : 6, GPCT : 19 ou 15 ou 12, GPT1 : 4, GPT2 : 8, GPT3 : 8, GPT4 : 8, GPT5 : 8, GPT6 : 4, GREF : 20 ou 19.

B. Choix du constituant de la PC

Si on veut se donner une possibilité d'évolution dans le futur, par exemple l'ajout de l'automatisation de l'alimentation en boîtes vides et de l'évacuation des boîtes pleines, le choix ne peut guère s'orienter que vers un constituant programmable comme un automate programmable industriel.

Vu le nombre d'entrées et de sorties dénombrées précédemment et respectivement de 48 et 18, tout automate programmable industriel de moyenne gamme est susceptible de convenir.

Sa configuration comprendra au moins 3 cartes de 16 entrées dont une carte compatible avec les capteurs inductifs et 2 cartes de 16 sorties. Pour garder la possibilité d'améliorer plus tard l'automatisation du poste de conditionnement des pignons, il devra disposer d'emplacements pour au moins une carte d'entrée et une carte de sortie supplémentaires.

Le choix de la marque et du type d'automate ainsi que de son langage sera à discuter avec le client pour tenir compte des autres API déjà en place chez cet industriel. Les connaissances en programmation et en maintenance du personnel seront également à prendre en compte.

C. Expressions logiques des sorties (solution avec sous-programmes (fig. 9.24) ou avec tâches (fig. 9.26))

mode manuel	mode auto	mode réf
Bx+ = X11.sx.mB+	+ X126 + X136	+ X179
Bx- = X11.sx.mB-	+ X124 + X134	+ X173
lx+ = X11.sx.ml+	+ X123 + X133	+ X174 + X178
lx- = X11.sx.ml-	+ X121 + X131	+ X176
Ax+ = X11.sx.mA+.bx0	+ X122 + X135	+ X175
Ax- = X11.sx.mA-.bx0	+ X125 + X132	+ X177
By+ = X11.sy.mB+	+ X146 + X156	+ X188
By- = X11.sy.mB-	+ X144 + X154	+ X182
ly+ = X11.sy.ml+	+ X143 + X153	+ X183 + X187
ly- = X11.sy.ml-	+ X141 + X151	+ X185
Ay+ = X11.sy.mA+.by0	+ X142 + X155	+ X184
Ay- = X11.sy.mA-.by0	+ X145 + X152	+ X186
R+ = X11.mR+	+ X161	
R- = X11.mR-	+ X162	+ X172
T+ = X11.mT+	+ X111	
T- = X11.mT-	+ X112	+ X171
VinitPC = X0.X10.X20.X110.X120.X130.X140.X150.X160.X170		
VréfPO = bx1.ix1.ax0.fx0.by1.iy1.ay0.fy0.r0.t0		

Figure 9.36
Expressions logiques
des sorties (solution avec
sous-programmes et avec
tâches fig. 9.24 et 9.26).

Le voyant VréfPO s'allume si les tables et les autres vérins sont en référence. Son expression logique n'est pas liée aux GRAFCET.

D. Sécurité câblée (fig. 9.37)

Ce schéma est à compléter des protections conseillées par le constructeur de l'automate programmable adopté.

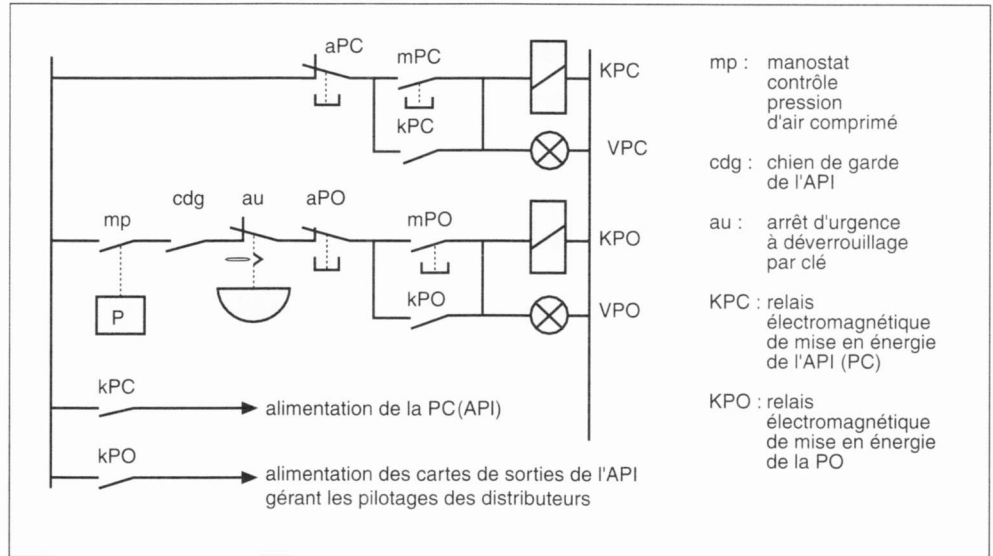


Figure 9.37
Sécurité câblée.

E. Expressions logiques des bits-drapeaux

Si l'on retient la solution du GRAFCET GPCT avec aiguillages par bits drapeaux (fig. 9.26), il faut envisager la gestion de ces bits.

a. Solution par GRAFCET particuliers (fig. 9.38)

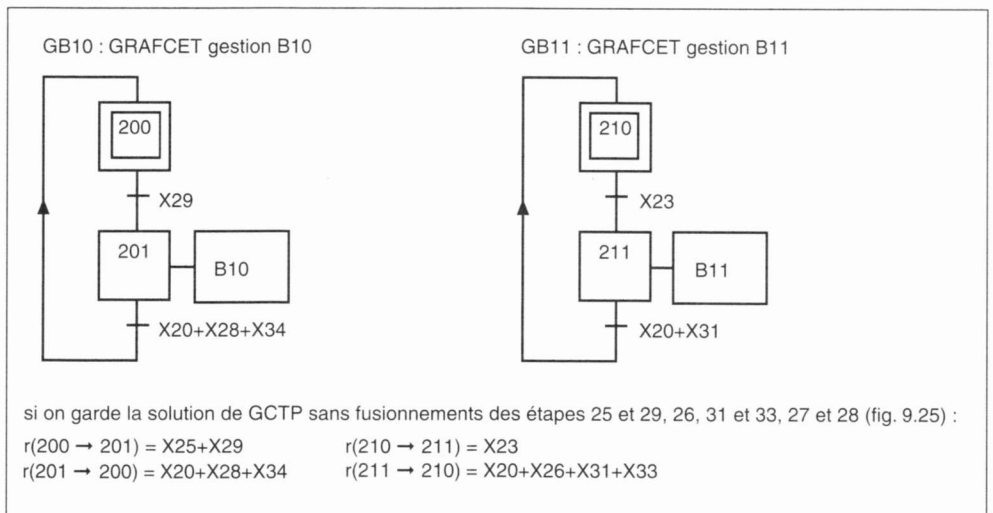


Figure 9.38
GRAFCET de gestion
des bits drapeaux.

$$B10 = X201$$

$$B11 = X211$$

b. Solution avec bits internes bistables

Avec fusionnement des étapes (fig. 9.26)

$$S(B10) = X29$$

$$R(B10) = X20 + X28 + X34$$

$$S(B11) = X23$$

$$R(B11) = X20 + X31$$

Sans fusionnement des étapes (fig. 9.25)

$$S(B10) = X25 + X29$$

$$D(B10) = X20 + X28 + X34$$

$$S(B11) = X23$$

$$D(B11) = X20 + X26 + X31 + X33$$

c. Solution avec des bits monostables (fonction mémoire à inscription prioritaire)

Avec fusionnement des étapes (fig. 9.26)

$$B10 = X29 + /X20./X27./X34.B10$$

$$B11 = X23 + /X20./X31.B11$$

Sans fusionnement des étapes (fig. 9.25)

$$B10 = X25 + X29 + /X20./X28./X34.B10$$

$$B11 = X23 + /X20./X26./X31./X33.B11$$

ADDENDA

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. A chacune des parties est applicable une échelle de temps interne et externe sans commune mesure.

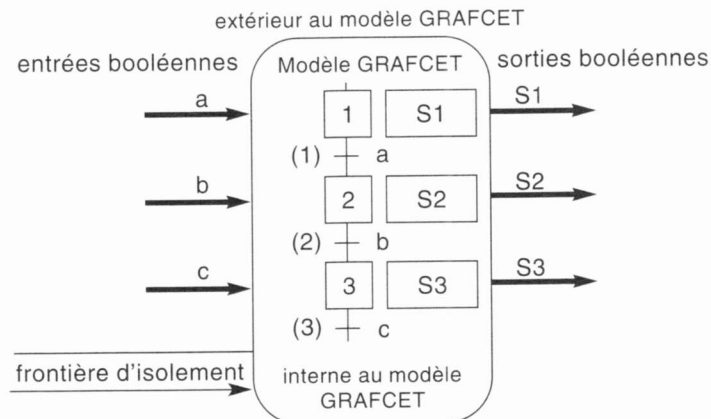
Postulat 1 : A l'échelle de temps externe, tout changement d'état des entrées est pris en compte par le modèle dès son apparition. La totalité des conséquences de cet événement sur le modèle est déterminée à temps nul. Depuis l'extérieur du modèle, les

événements d'entrée et les états de sorties qui en résultent sont vus à la même date.

Postulat 2 : A 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.

A 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.

Un algorithme de recherche de stabilité (ARS) est intégré à certains logiciels de Programmation du GRAFCET Assistée par Ordinateur (PGA). C'est le cas d'ALOGRAF de la société UXP, par exemple.

Cas des situations instables

Une situation $S = \{n\}$ est dite instable 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. 3.31).

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 (voir fig. 3.31). Cette donnée est à prendre en compte d'un point de vue réalisation.

X. SYMBOLIQUE PROPRE AU GRAFCET, À L'ORGANIGRAMME ET AU LOGIGRAMME

1. Structures graphiques du GRAFCET (fig. 10.1)

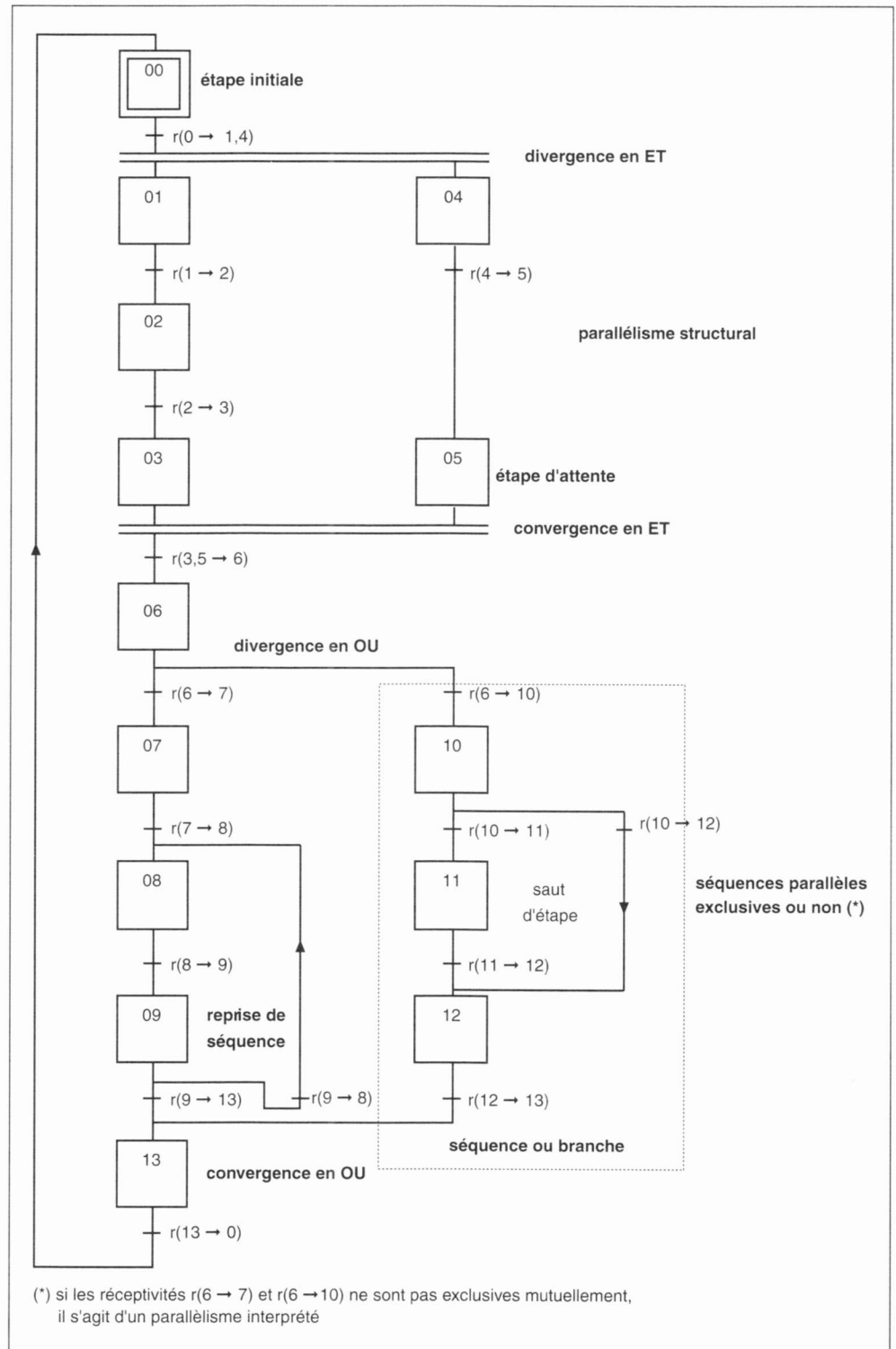


Figure 10.1
Structures graphiques
du GRAFCET.

2. Étapes (fig. 10.2)

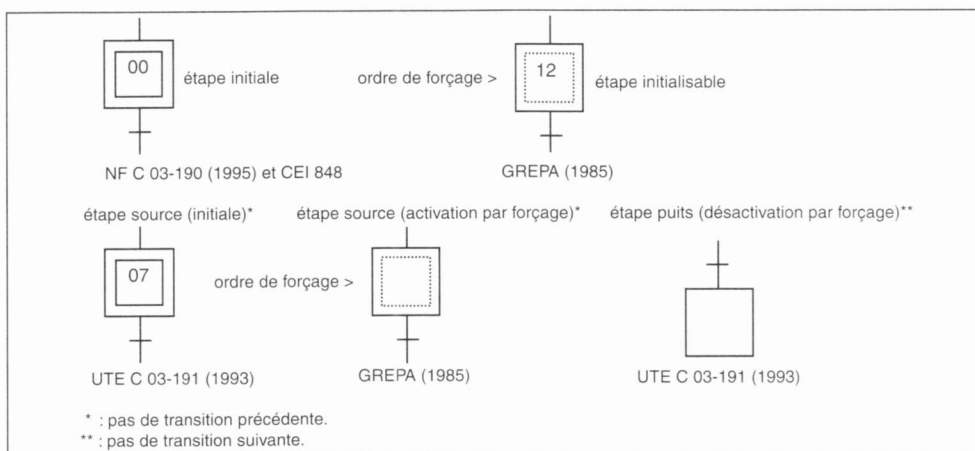


Figure 10.2
Symboles des étapes.

3. Macro-représentations (fig. 10.3)

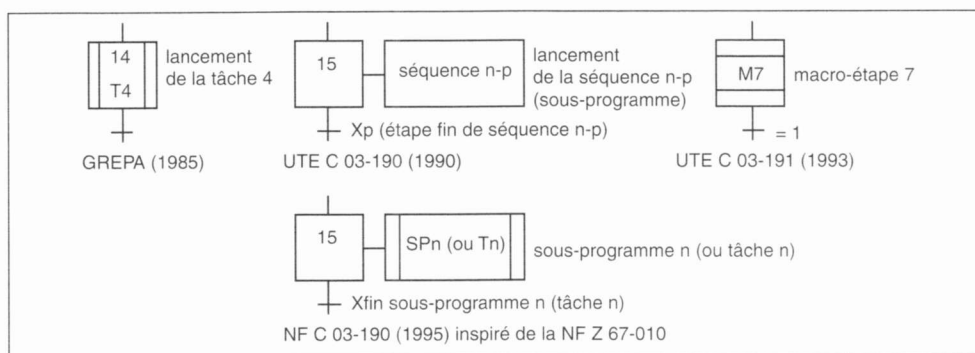


Figure 10.3
Les macro-représentations.

4. Ordres ou actions associés à l'étape (fig. 10.4 et 10.5)

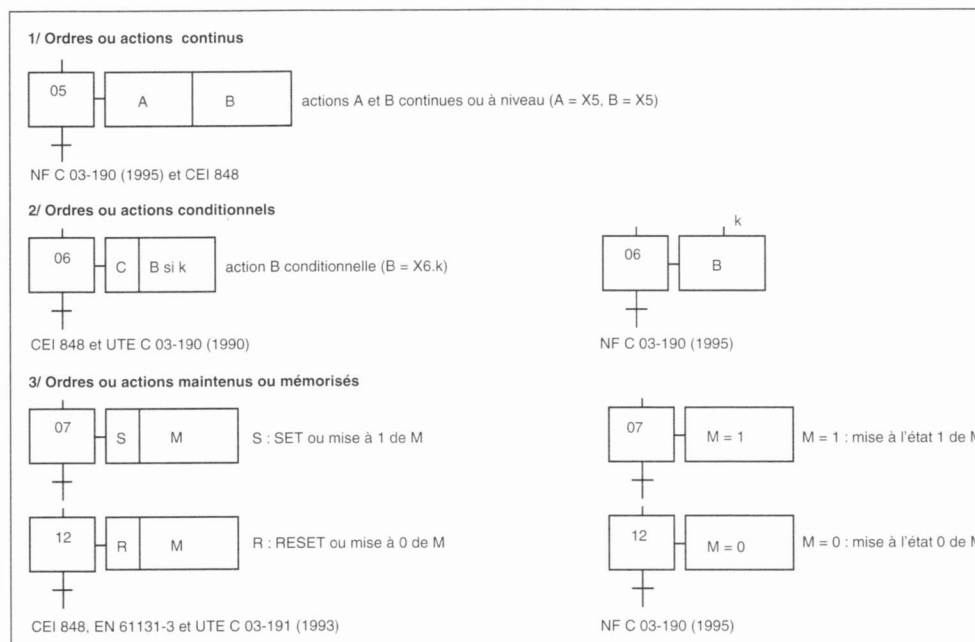


Figure 10.4
Ordres ou actions
associés à l'étape.

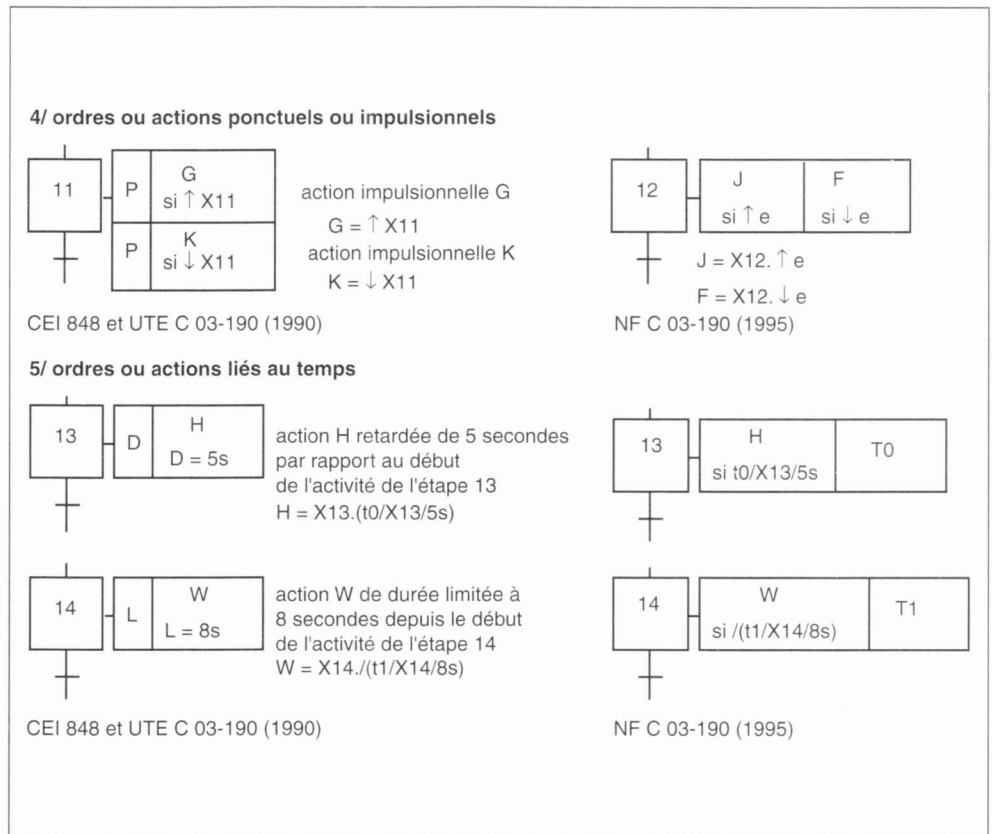


Figure 10.5
 Ordres ou actions associées à l'étape.

5. Transitions et réceptivités associées (fig. 10.6)

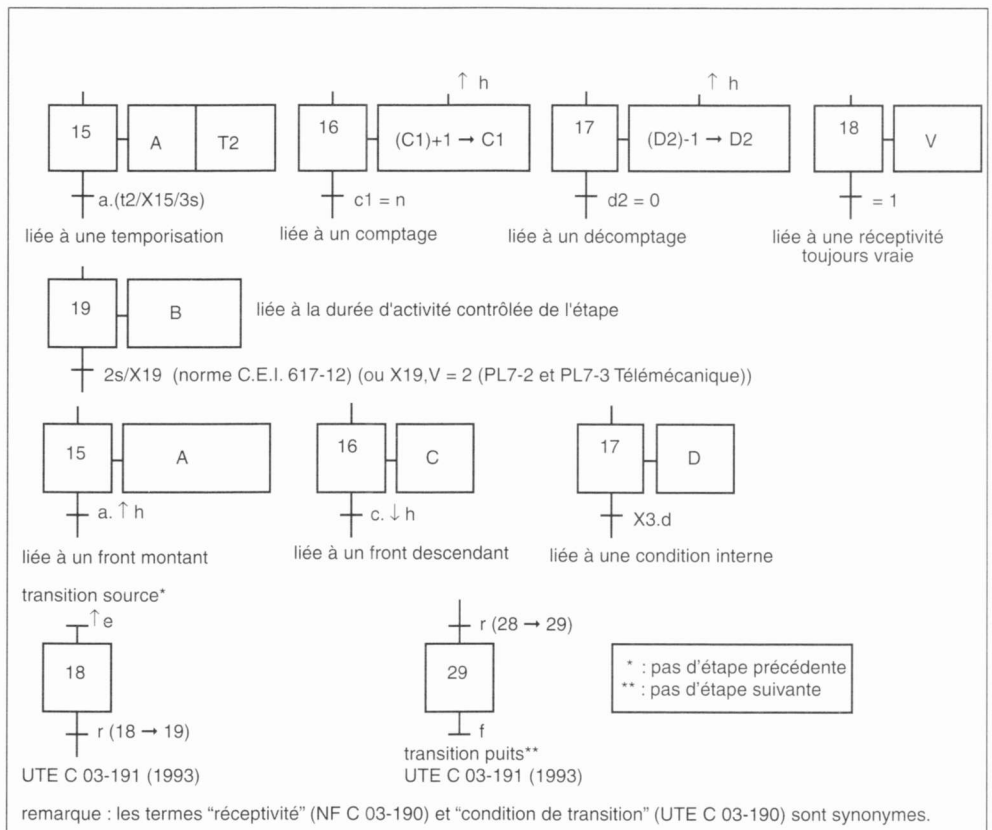


Figure 10.6
 Transitions et réceptivités associées.

6. Forçages de situation (fig. 10.7)

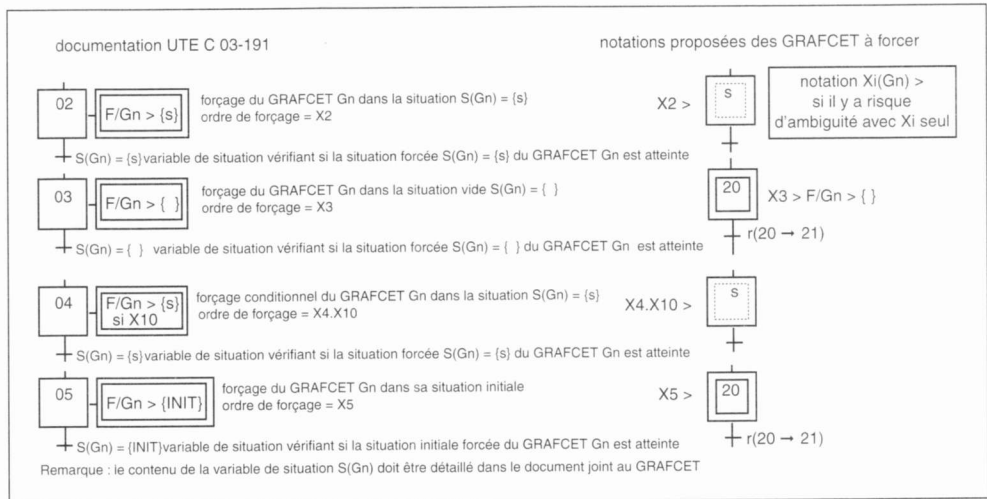


Figure 10.7
Forçages de situation.

7. Figeages de situation (fig. 10.8)

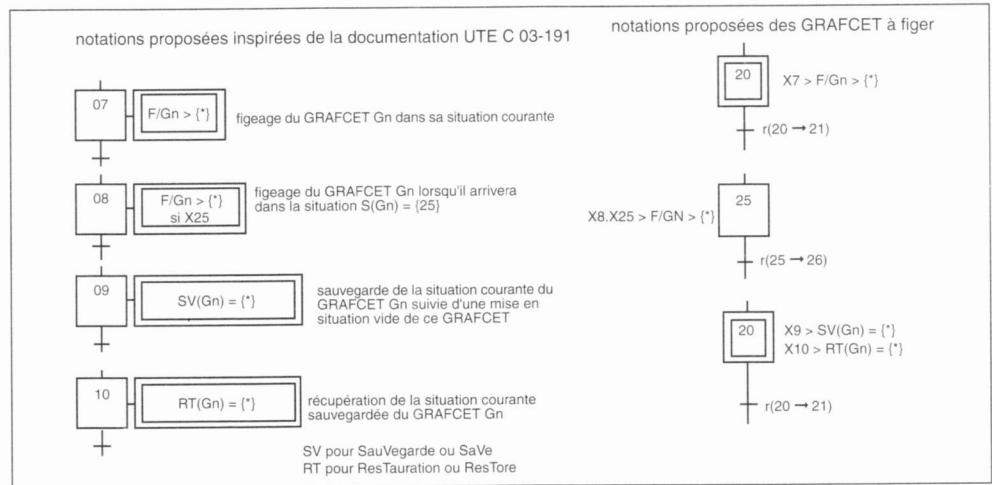


Figure 10.8
Figeages de situation.

8. Forçage des sorties de la PO et de la PR (fig. 10.9)

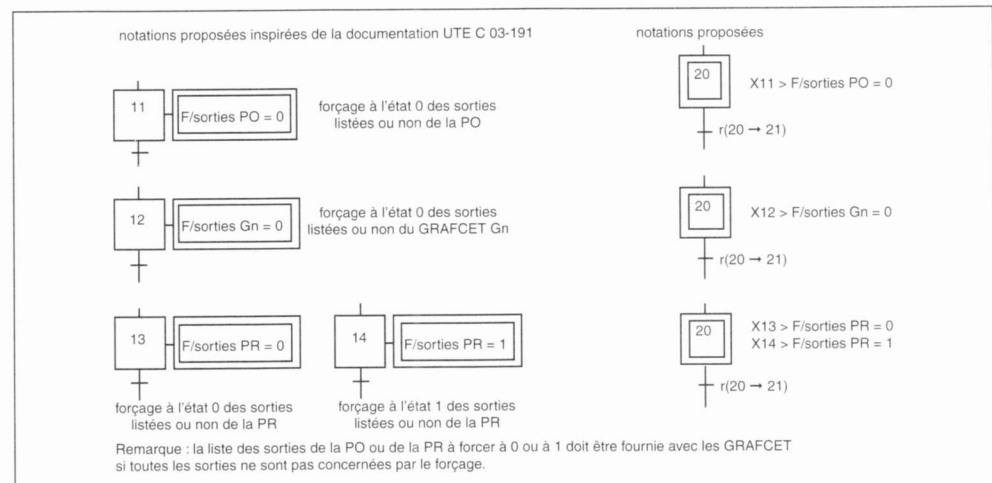


Figure 10.9
Forçage des sorties de la PO et de la PR.

9. Symboles pour organigrammes (extrait NF Z 67-010) (fig. 10.10)

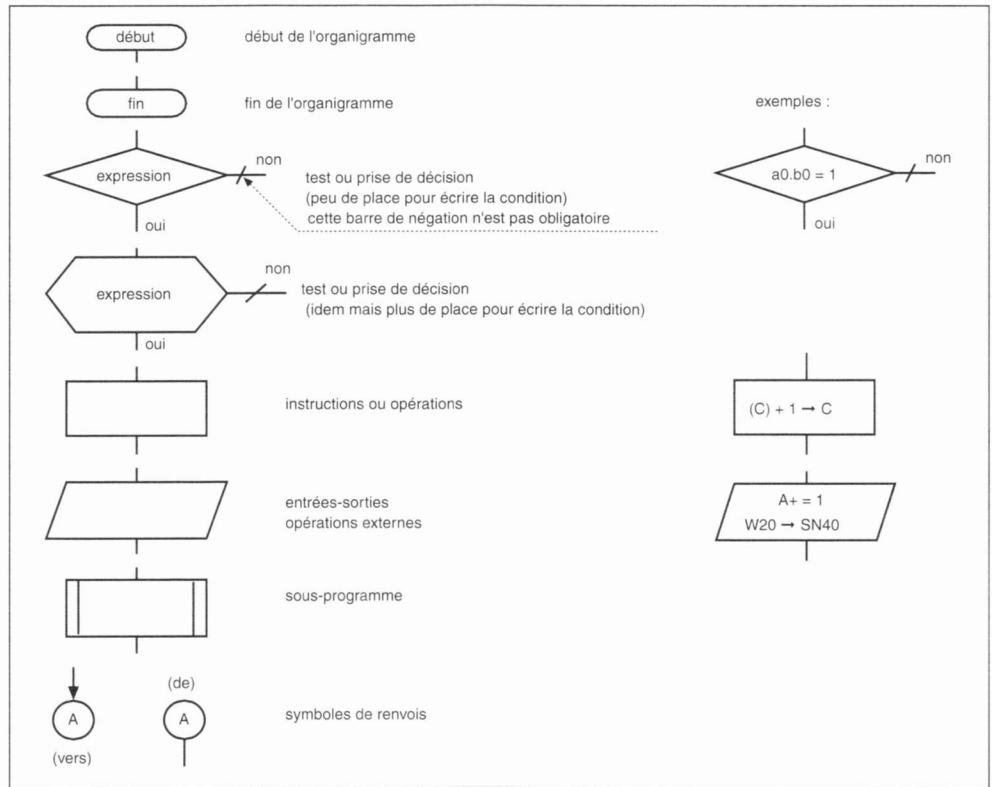


Figure 10.10
Symboles pour organigrammes (NF Z 67-010).

10. Structures d'implantation algorithmique du GRAFCET (fig. 10.11)

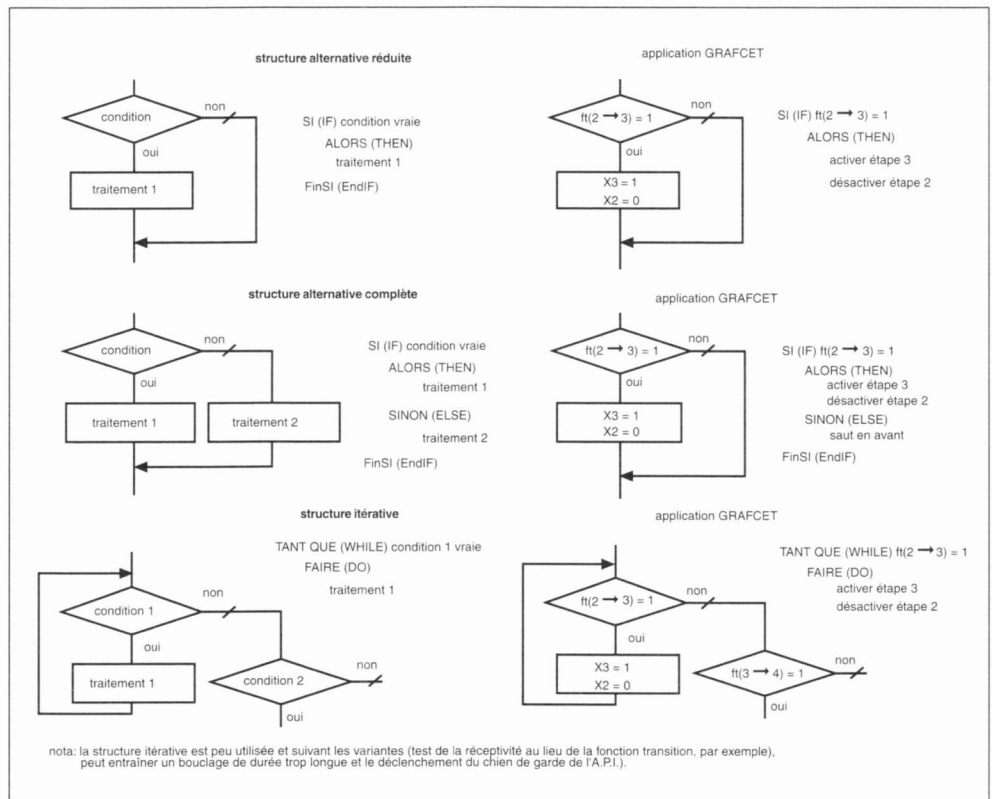


Figure 10.11
Structures d'implantation du GRAFCET.

11. Symboles pour logigrammes (fig. 10.12)

(extrait des normes françaises C 03-212, E 49-600 et E 49-602)

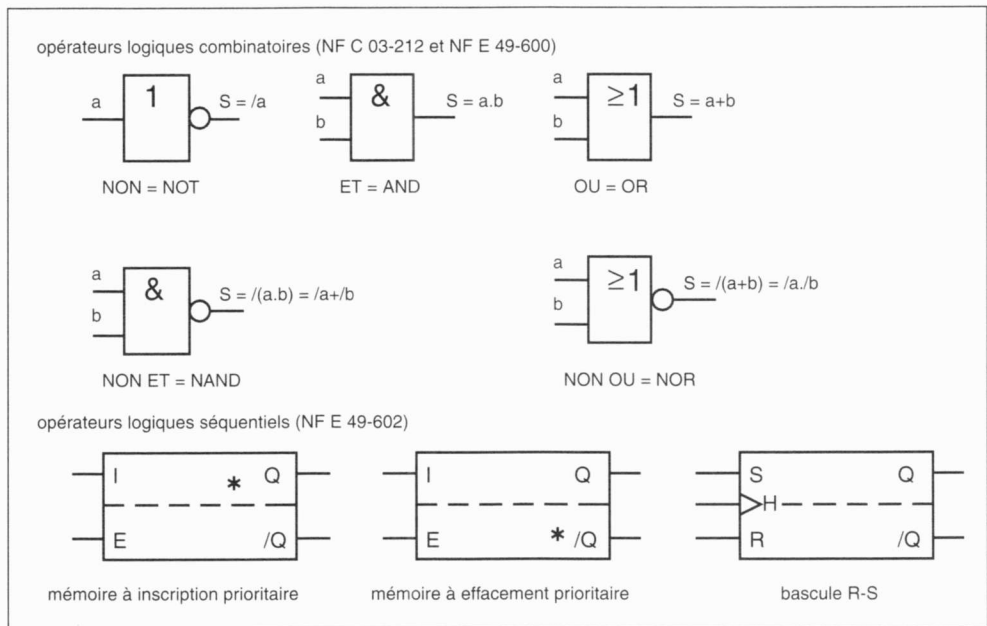
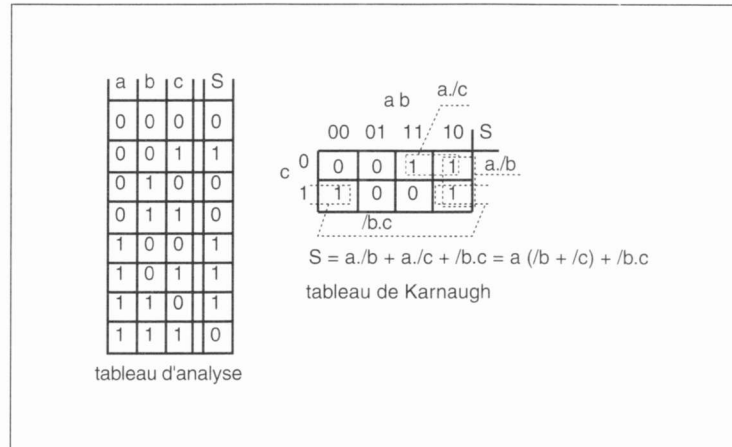


Figure 10.12
Symboles pour logigrammes
(NF C 03-212, NF E 49-600
et 49-602).

XI. AUTRES MÉTHODES D'ANALYSE ET DE SYNTHÈSE DES AUTOMATISMES

1. Analyse combinatoire (principe)

Figure 11.1
Analyse combinatoire.
L'état de la sortie S est analysé en fonction de l'état des 3 entrées a, b, c.
L'état de la sortie S est reporté dans le tableau de Karnaugh à 3 variables.
Les côtés du tableau sont codés en binaire réfléchi (changement d'état d'une seule variable entre deux combinaisons adjacentes).
Le regroupement des cases adjacentes permet de sortir les termes simplifiés de l'expression logique de la sortie S.



2. Méthode matricielle (méthode d'Huffman)

APPLICATION

Commande d'un moteur électrique R à partir de 2 boutons poussoirs m (marche) et a (arrêt).

Donner la priorité à l'arrêt sur la marche.

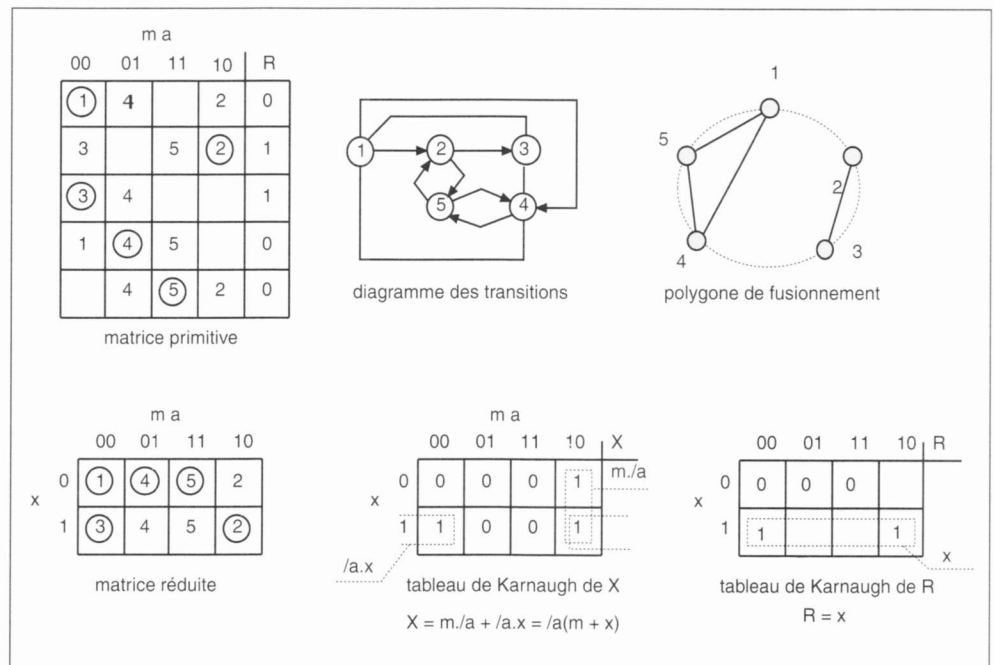


Figure 11.2
Méthode matricielle.

L'analyse se conduit en plusieurs temps.

1. Dresser le tableau de la matrice primitive : 4 colonnes d'entrée (2 variables d'entrée (m et a) → 2 puissance 2 combinaisons) et 1 colonne de sortie. Nombre de lignes inconnu.
2. Remplir la matrice primitive en analysant le problème. Les cercles correspondent à des états stables. On atteint un état stable en passant par un état instable (même chiffre non cerclé).
3. Le diagramme des transitions, non indispensable, permet de vérifier l'ensemble des transitions.
4. Étudier les fusionnements possibles des lignes de la matrice primitive. Le fusionnement n'est possible que si on superpose des cases ayant même numéro, états stables et états instables confondus.
5. Constaté que le fusionnement à 2 lignes est possible : groupage des états stables 1, 4, 5 et 2, 3.
6. Dresser la matrice réduite à 4 colonnes et 2 lignes. Les 2 lignes sont décodées par l'intermédiaire de la variable auxiliaire X.
7. Remplir la matrice réduite en donnant la priorité à l'état stable sur l'état instable.
8. Remplir le tableau de Karnaugh de la variable secondaire X suivant le principe suivant : la variable secondaire X prend le même état dans l'état stable et l'état instable de même numéro.
Si X est un relais : la bobine X s'excite, le contact x n'est pas encore fermé : $X = 1, x = 0$ (état instable).
Puis après un délai égal au temps de réponse du relais, le contact x se ferme : $X = 1, x = 1$ (état stable).
Vérification : (1) \longrightarrow 2 \longrightarrow (2)
 $X = 0$ $X = 1$ $X = 1$
9. Sortir l'expression logique de X : $X = /a(m + x)$
10. Remplir le tableau de Karnaugh de R d'après la matrice primitive en ne tenant compte que des états stables : $R = x$.

3. Description par chronogramme



APPLICATION : feux tricolores de carrefour à 2 voies A et B.

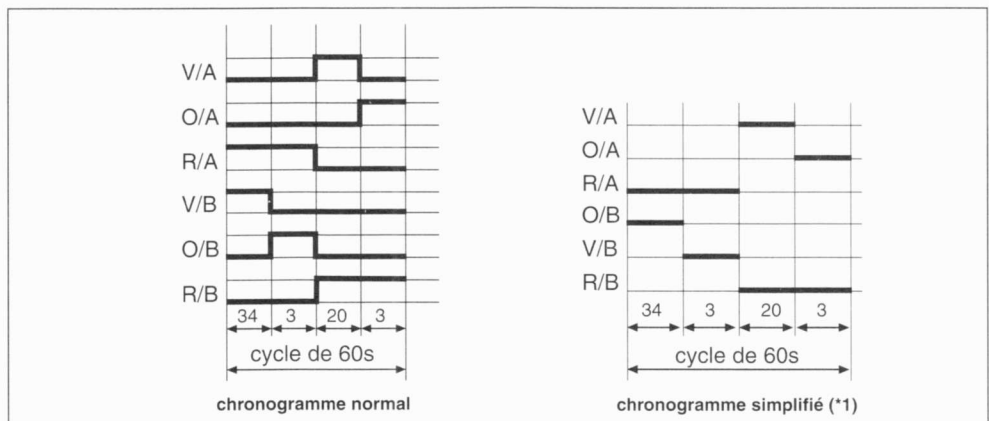


Figure 11.3
Description
par chronogramme.

*1 Le diagramme de GANTT s'apparente, dans sa représentation, au chronogramme simplifié. Chaque trait représente une tâche. Il permet, par exemple de planifier le travail de chaque corps de métier intervenant dans la construction d'un immeuble pour fixer le début et la fin de chaque tâche et déterminer ainsi la date de réception du bâtiment achevé.

Dans un chronogramme le temps s'écoule de gauche à droite. La largeur des colonnes n'est pas nécessairement proportionnelle au temps. Horizontalement les niveaux 0 et 1 sont soit représentés par un échelon (chronogramme normal) soit seul le niveau 1 est figuré par un trait renforcé (chronogramme simplifié).

4. Diagramme des phases

APPLICATION

Plateau tournant entraîné par un moteur R avec contact fin de course f actionné par une came. Départ par impulsion sur un bouton m (marche).

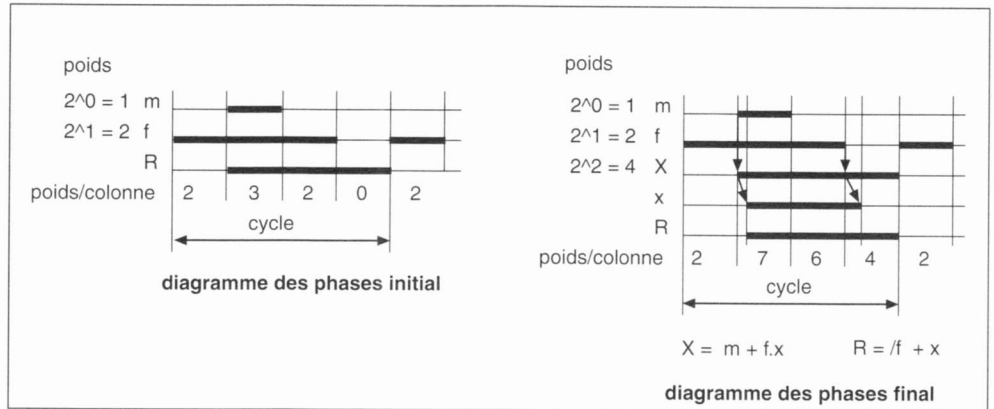


Figure 11.4
Analyse par le diagramme
des phases.

Le diagramme des phases initial s'inspire du chronogramme (fig. 11.4).

Des poids sont attribués aux variables d'entrée ce qui permet de calculer le poids des colonnes. On s'aperçoit que 2 colonnes ont le même poids : 2. Il faut donc introduire une troisième variable X (mémoire) pour différencier ces 2 colonnes. Sur le diagramme des phases final, ces colonnes ont pour poids respectifs 2 et 6.

L'expression de X s'obtient en observant comment s'inscrit et s'efface la mémoire X, $I(X) = m$, $E(X) = /f \rightarrow X = m + f.x$ (inscription prioritaire) ou $X = f(m + x)$ (effacement prioritaire).

L'expression de R s'obtient en observant que $R = 1$ si $f = 0$ ou $x = 1$ d'où $R = /f + x$.

La mise en place des variables secondaires et la recherche des expressions logiques restent le point faible de cette méthode.

5. Méthode en cascade

Cette méthode repose sur l'emploi systématique du bloc-opératif étendu « vérin + distributeur à double pilotage + capteurs fin de course ».

Notations utilisées

Vérin :	N
Mouvement sortie de tige :	N+
Mouvement rentrée de tige :	N-
Pilotage déclenchant la sortie de tige :	N+
Pilotage déclenchant la rentrée de tige :	N-
Capteur fin de course aller :	n1
Capteur fin de course retour :	n0

Description du cycle à partir des mouvements (idem aux pilotages) (vérins A, B, C)

Exemple : séquences de vérins et appellations (fig. 11.5)

cycle pendulaire : A+ A-
 cycle carré : A+ B+ A- B-
 cycle en L : A+ B+ B- A-
 cycle en U : A+ B+ A- A+ B- A-

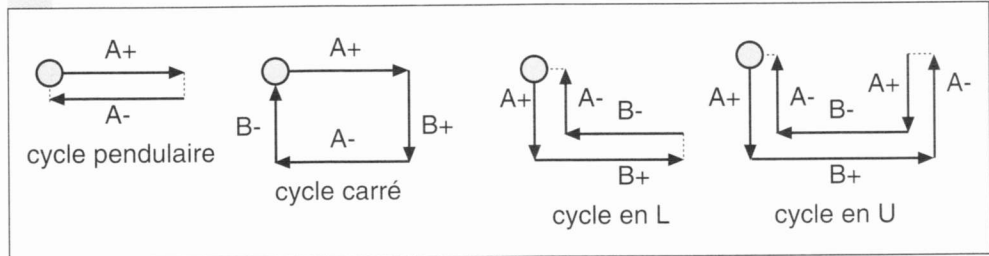


Figure 11.5
Schéma de cycles
avec vérins.

Principe de base de la méthode en cascade

Le pilotage suivant est dépendant du capteur actif précédent c'est-à-dire du capteur fin de course passant de l'état 0 à l'état 1 (commande en cascade).

Exemple 1

Cycle pendulaire : A+ A-, départ par impulsion sur le bouton dcy.
 impulsion sur dcy \rightarrow A+ = 1 \rightarrow a1 = 1 \rightarrow A- = 1 \rightarrow a0 = 1 \rightarrow A+ = 1 si dcy
 On en déduit : A+ = dcy.a0
 A- = a1

Vérification de la non-contradiction d'effet

Il y a contradiction d'effet si les deux pilotages A+ et A- du distributeur du vérin A sont commandés en même temps. Le tiroir du distributeur conserve alors la position qu'il avait précédemment. Le distributeur à double pilotage se comporte comme une mémoire à entrées simultanées passives.

Ici, les pilotages A+ et A- sont tributaires des capteurs fin de course a1 et a0 technologiquement exclusifs. Il est évident que par leur position relative au vérin A, ils ne peuvent être simultanément à l'état 1.

Exemple 2

cycle en L : A+ B+ B- A-, départ par dcy.
 Si on applique le même raisonnement que pour le cycle pendulaire, on obtient : impulsion sur dcy \rightarrow A+ = 1 \rightarrow a1 = 1 \rightarrow B+ = 1 \rightarrow b1 = 1 \rightarrow B- = 1 \rightarrow b0 = 1 \rightarrow A- = 1 \rightarrow a0 = 1 \rightarrow A+ si dcy.
 On en déduit : A+ = dcy.a0 B+ = a1
 A- = b0 B- = b1

Les capteurs a0 et b0 étant actionnés en même temps, il y a contradiction d'effet entre les pilotages A+ et A-.

Il en est de même pour les capteurs a1 et b1 ce qui fait que les pilotages B+ et B- sont également en contradiction d'effet.

Le cycle se bloquera soit en fin du mouvement A- soit en fin du mouvement B+.

Solution levant les contradictions d'effet

On alimente les capteurs responsables des contradictions d'effet par les voies complémentaires /x et x d'une variable secondaire X de type bistable : distributeur à double pilotage pour la version pneumatique, relais électromagnétique bistable pour la version électropneumatique. Les contradictions d'effet sont ainsi levées.

On peut le montrer graphiquement.

Description graphique (fig. 11.6)

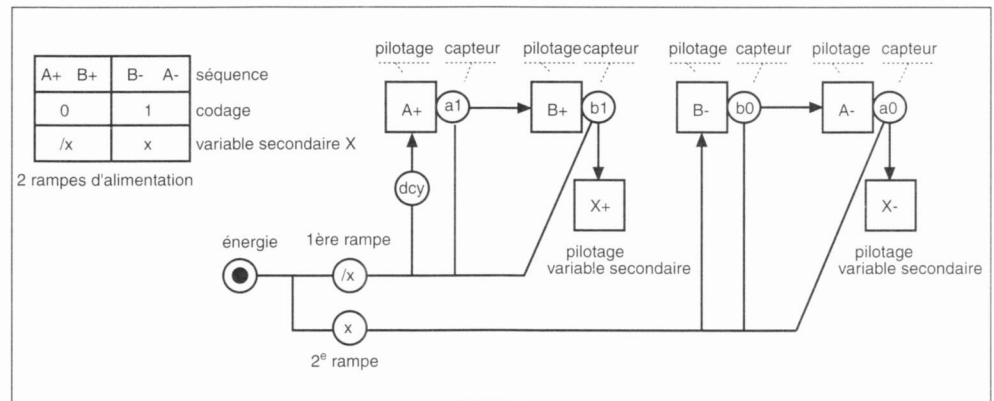


Figure 11.6
Cycle en L.

De la figure 11.6, on en déduit : $A+ = /x.dcy$ $B+ = /x.a1$ $X+ = /x.b1$
 $A- = x.b0$ $B- = x$ $X- = x.a0$

Les contradictions d'effet sont bien levées par les rampes /x et x.

Découpage en zones levant les contradictions d'effet

Il faut examiner la séquence des mouvements qui est identique à celle des pilotages.

Il faut séparer les deux pilotages du même distributeur et créer des zones auxquelles on associe les rampes d'alimentation des capteurs.

Le découpage donne alors 2 zones, 4 zones, 8 zones, etc. qu'il faut coder de manière appropriée avec 1, 2, 3, etc. variables secondaires (mémoires auxiliaires). Pour éviter les aléas de séquence, on utilise le code binaire réfléchi dérivé du code binaire naturel. La particularité du code binaire réfléchi est qu'une seule variable change d'état à la fois lorsqu'on passe d'une combinaison à la combinaison adjacente.

Exemple : cycle en U

La séparation des pilotages des distributeurs A et B fait apparaître 4 zones. Pour coder ces zones en binaire réfléchi, il faut deux variables X et Y (fig. 11.7).

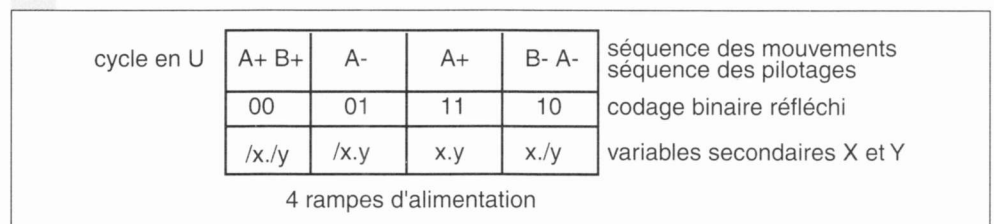


Figure 11.7
Découpage en zones
et codage des rampes
du cycle en U.

Solution

La description graphique n'est pas obligatoire, les expressions logiques se déduisant avec un peu d'habitude directement du tableau de la figure 11.7.

$$\begin{array}{llll} A+ = dcy./x./y + x.y & B+ = a1./x./y & X+ = a0./x.y & Y+ = b1./x./y \\ A- = /x.y + b0.x./y & B- = x./y & X- = a0.x./y & Y- = a1.x.y \end{array}$$

On notera que les entrées des mémoires auxiliaires sont commandées par impulsion.

L'impulsion est provoquée par l'autorupture due à l'alimentation au travers de sa rampe complémentaire. Par exemple, X+ est alimenté au travers de la rampe /x pour mettre en service la rampe x.

Remarque

1. Si le découpage donne par exemple 3 zones, on crée une zone supplémentaire pour faciliter le codage en binaire réfléchi (figure 11.8).

Figure 11.8
Passage de 3 zones
à 4 zones.

cycle	A+ B+	B- C+	C-	A-	séquence des mouvements séquence des pilotages
	00	01	11	10	codage binaire réfléchi
	/x./y	/x.y	x.y	x./y	variables secondaires X et Y

On a séparé C- de A- alors que cela n'était pas nécessaire afin de créer 4 zones codées en binaire réfléchi.

$$\begin{array}{lllll} A+ = /x./y.dcy & B+ = /x./y.a1 & C+ = /x.y.b0 & X+ = /x.y.c1 & Y+ = /x./y.b1 \\ A- = x./y & B- = /x.y & C- = x.y & X- = x./y.a0 & Y- = x.y.c0 \end{array}$$

2. La méthode en cascade peut intégrer les blocs opératifs «moteur électrique-contacteur-capteurs fin de course ou à défaut variable du contacteur».

exemple : séquence marche moteur M, A+, A-, arrêt moteur M.

$$dcy = 1 \rightarrow M+ = 1 \rightarrow km = 1 \rightarrow A+ = 1 \rightarrow a1 = 1 \rightarrow A- = 1 \rightarrow a0 = 1 \rightarrow M- = 1 \rightarrow km = 0$$

3. Les expressions logiques résultant de l'analyse par la méthode en cascade sont programmables, par exemple, en langage littéral ou en langage à contacts dans les API.

6. Séquenceurs

Le module d'étape est constitué d'une mémoire bistable associée à un opérateur logique ET pour le traitement de la fonction transition d'activation et un opérateur OU pour le traitement de la désactivation par le module suivant ou par la RAZ générale.

En associant plusieurs modules d'étapes, on constitue un séquenceur. Le nombre de modules d'étapes dépend du nombre d'étapes du GRAFCET à matérialiser.

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

L'effacement de la mémoire du module 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.

Exemple choisi : le séquenceur pneumatique Parker-Pneumatic (fig. 11.9).

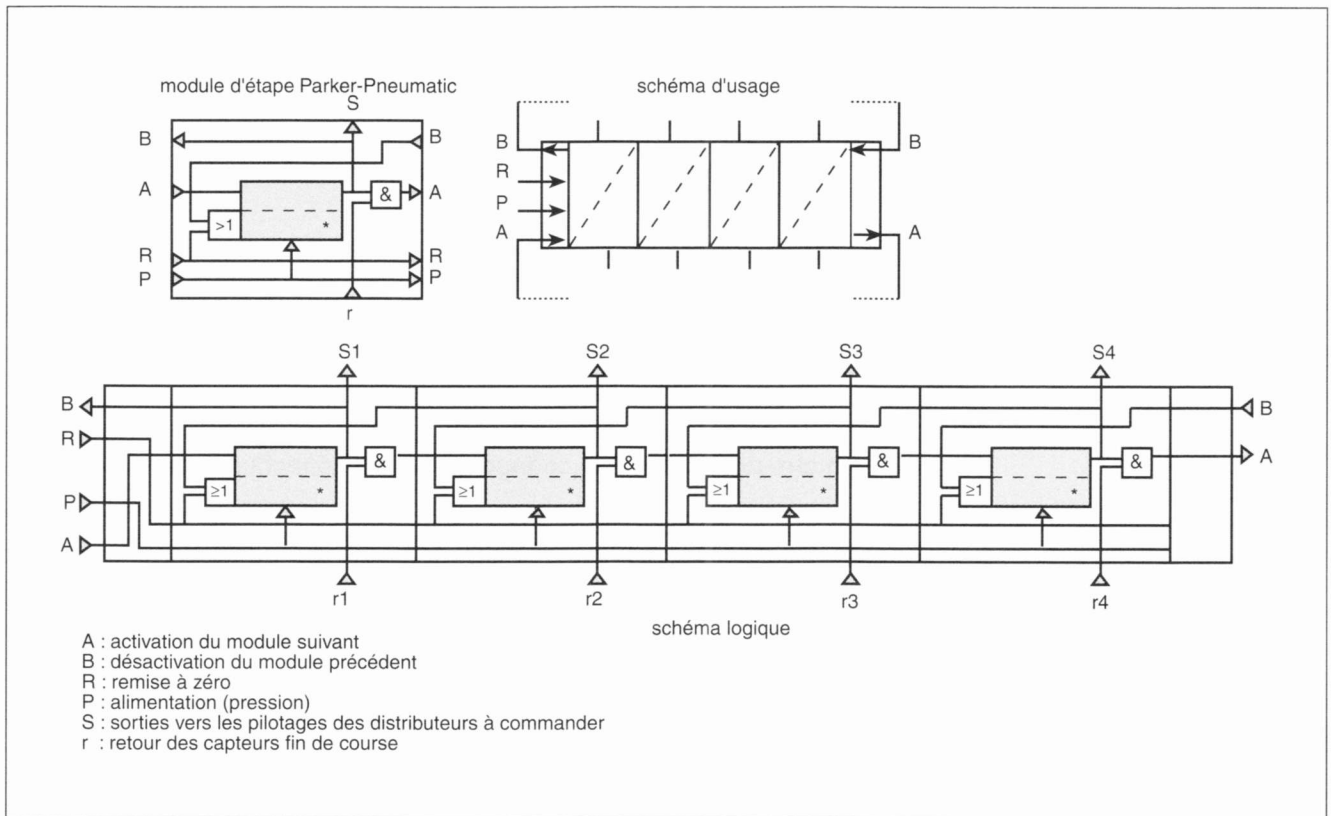


Figure 11.9
Séquenceur pneumatique
Parker-Pneumatic.

 **APPLICATION** au poste de dosage (voir chapitre VII, fig. 7.1)

GRAFSET original modifié pour séquenceur (fig. 11.10)

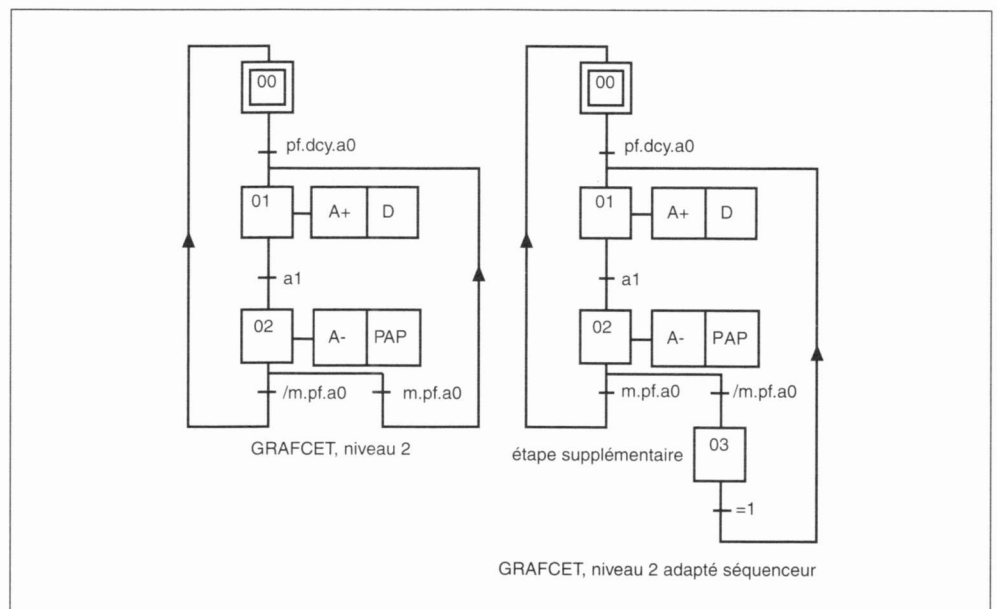


Figure 11.10
Poste de dosage
d'un produit liquide.

Schéma du séquenceur du poste de dosage (fig. 11.11)

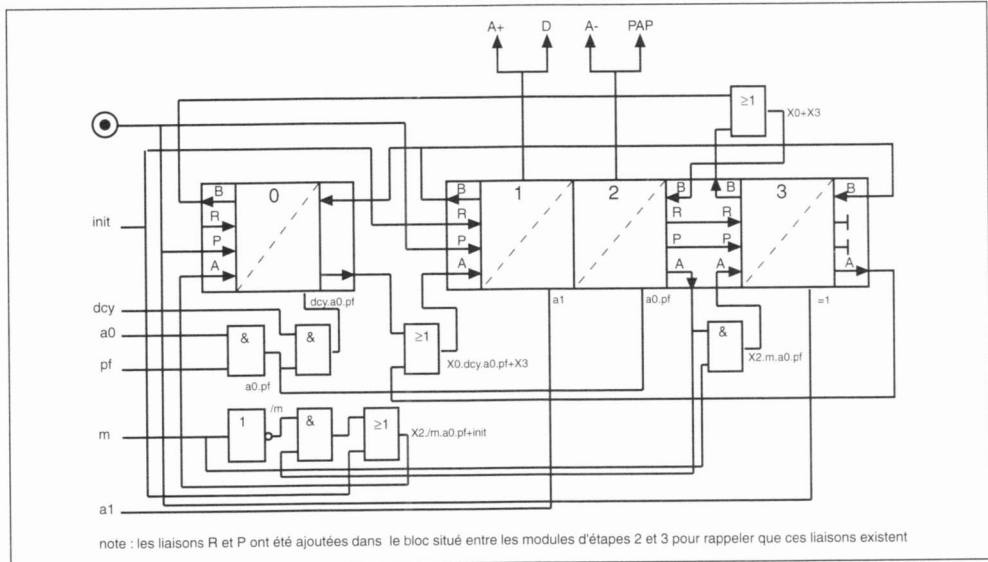


Figure 11.11
Séquenceur Parker-
Pneumatic pour le poste
de dosage.

7. Organiphase de la Télémécanique

Ce moyen de description précurseur du GRAFCET s'appuyait, pour la réalisation de l'automatisme, sur un séquenceur obtenu par association de modules de phases.

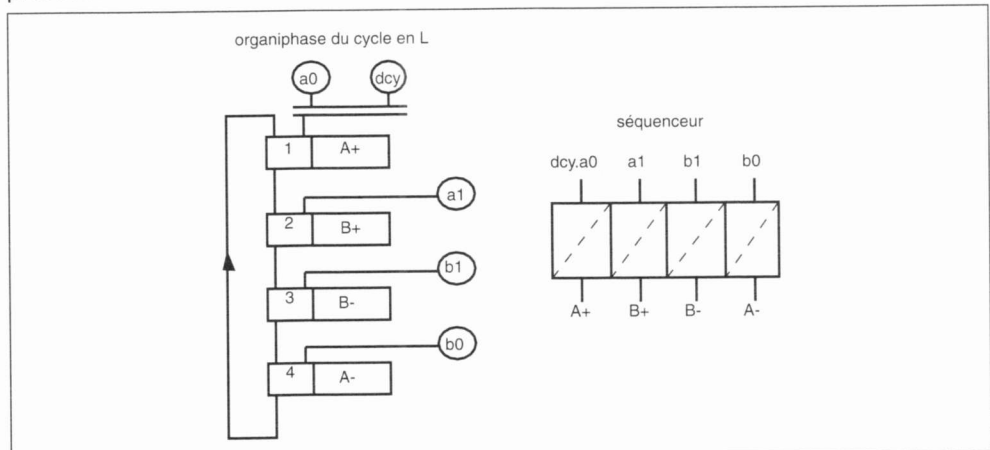


Figure 11.12
Organiphase Télémécanique.

Note

- Les séquenceurs pneumatiques suivants sont encore commercialisés :
- Séquenceur pneumatique Parker Pneumatic (origine Télémécanique).
 - Séquenceur pneumatique Joucomatic (origine Crouzet).
 - Séquenceur pneumatique Siebe Pneumatics (origine Climax).
 - Séquenceur pneumatique VE Atlas Copco.
 - Séquenceur Quickstepper Festo.

Pour plus de détails sur ces séquenceurs pneumatiques, consulter l'ouvrage *La PNEUMATIQUE dans les systèmes automatisés de production* même collection, Editions Casteilla.

TERMINOLOGIE PROPRE AU GRAFCET ET AUX STRUCTURES D'IMPLANTATION DANS LES AUTOMATES PROGRAMMABLES INDUSTRIELS

A

- Action** : ce terme désigne aussi bien une fonction, une tâche ou un ordre suivant le niveau de la description. L'action est obligatoirement associée à une ou plusieurs étapes.
- Action continue** : action exécutée tant que l'étape, à laquelle elle est associée, reste active. (Synonyme : action à niveau).
- Action conditionnelle** : action exécutable si l'étape, à laquelle elle est associée, est active et la condition associée vraie (symbole C pour Conditioned).
- Action conditionnelle à durée limitée** : action à durée limitée par rapport à la durée d'activité de l'étape à laquelle elle est associée (symbole L pour Limited).
- Action conditionnelle retardée** : action retardée par rapport au début d'activité de l'étape à laquelle elle est associée (symbole : D pour Delayed).
- Action impulsionnelle** : action exécutable si l'étape, à laquelle elle est associée, est active et qu'un événement se produit : front montant ou descendant d'une variable externe ou interne (symbole P pour Pulsed).
- Action mémorisée** : action maintenue par mémorisation pendant l'activité de plusieurs étapes successives (symbole : S pour Stored utilisé pour la mise EN et HORS mémoire d'où certains risques d'ambiguïté), la tendance actuelle est d'indiquer la mise en mémoire (inscription) par S pour Set et la mise hors mémoire (effacement) par R pour Reset (norme EN 61131-3) (synonyme : action maintenue).
- Activation** : mise à 1 d'une étape (état actif).
- Actionneur** : organe de commande d'un mécanisme.
- Adressage** : définit la position d'une entrée ou d'une sortie sur la carte correspondante de l'API. Concerne également la position des bits internes dans l'espace réservé de la mémoire de l'API.
- Aiguillage** : structure de dérivation sélective du GRAFCET (synonyme : divergence en OU exclusif).
- Alphanumérique** : qui concerne à la fois des lettres et des chiffres et par extension tous les caractères de l'informatique. Le code ASCII (*American Standard Code for Information Interchange*) attribue un code à chacun des caractères associés aux touches des claviers des consoles des API ou des ordinateurs.
- Algorithme** : description précise et complète de la suite des opérations élémentaires nécessaires pour résoudre un problème donné. L'organigramme est un des moyens utilisés pour décrire graphiquement un algorithme (norme française Z 67-010).
- Analyse descendante** : méthode d'analyse basée sur la décomposition progressive du problème en partant de la fonction globale pour aboutir aux détails (synonyme : décomposition arborescente).
- Arborescence** : procédé de décomposition graphique d'un problème inspiré de l'arbre : tronc, branches, rameaux, feuilles, nervures. Ce procédé graphique est utilisé pour illustrer certaines phases de l'analyse descendante.
- ASCII** : acronyme de *American Standard Code for Information Interchange*. Codage sur 8 bits autorisant $2^8 = 256$ combinaisons soit le codage de 256 caractères utilisés en informatique : lettres, chiffres, symboles arithmétiques, caractères semi-graphiques, etc.
- Assembleur** : logiciel permettant la traduction en langage machine (langage binaire du microprocesseur) d'un programme écrit en langage mnémonique (langage assembleur).
- Asservissement** : automatisme où la sortie est assujettie à suivre l'évolution de la valeur d'entrée ou consigne. Exemples : tour à copier hydraulique. Entrée : palpeur sur gabarit, sortie : vérin hydraulique asservi.
Autre exemple : servocommandes des ailerons d'un avion. Entrée : manche à balai, sortie vérin hydraulique ou vérin électrique asservi.
- Automate programmable industriel (API ou A.P.I.)** : (norme NF C 63-850) constituant de traitement de l'information destiné à la commande des machines et des installations dans leur environnement industriel.
La constitution d'un API est comparable de celle d'un micro-ordinateur.

L'architecture de l'API comprend pour l'essentiel :

- un bloc d'alimentation,
- une unité centrale à base de microprocesseur,
- une mémoire contenant le moniteur,
- une mémoire contenant le langage de l'API,
- une mémoire contenant le programme de l'application, et plus spécifique, des cartes d'entrées tout ou rien et/ou des cartes d'entrées analogiques, des cartes de sorties tout ou rien et/ou des cartes de sorties analogiques.

Synonyme : contrôleur programmable.

Autres actions ou ordres : ordre de lancement d'un GRAFCET sous-programme ou de tâche, ordre de forçage de situation d'un GRAFCET, ordre de figeage de situation d'un GRAFCET, ordre de forçage à 0 des sorties, ordre de sélection des modes de marche et d'arrêt, ordre de mise EN ou HORS énergie de la PO, ordre de demande de calcul, etc.

AZERTY : caractérise un clavier dont les lettres associées aux six premières touches forment ce mot. Ce type de clavier est surtout utilisé dans les pays non anglo-saxons, car il dispose de touches pour les voyelles accentuées. Le clavier utilisé par les anglo-saxons est du type QWERTY.

B

Base de temps : horloge interne dont les tops (impulsions) sont exploités par exemple dans les temporisateurs.

Bit (binary digit) : unité élémentaire d'information de nature binaire ayant pour valeur 0 ou 1.

Bit bistable : bit interne à deux états stables (0 ou 1) jouant le rôle de mémoire. Doit être inscrit (Set) puis effacé (Reset).

Bit interne : donnée de un bit adressée dans la mémoire de l'API.

Bit monostable : bit interne jouant le rôle de mémoire par l'intermédiaire d'une expression logique programmée (mémoire à inscription prioritaire ou à effacement prioritaire au choix).

Bit système : bit interne ayant un rôle déterminé à jouer dans le programme de l'API.

Binaire naturel : suite naturelle obtenue par comptage en base 2.

Exemple : 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111.

Binaire réfléchi : codage établi de manière à ce qu'un seul bit change d'état à la fois lorsqu'on passe d'une combinaison à la suivante.

Exemple : 0000 0001 0011 0010 0110 0111 0101 0100 1100 1101 1111 1110 1010 1011 1001 1000.

Bloc-opératif : bloc symbolique regroupant l'actionneur et son préactionneur.

Exemples : moteur électrique et son contacteur, vérin pneumatique et son distributeur, etc.

Bloc-opératif étendu : bloc symbolique regroupant l'actionneur, son préactionneur et les capteurs associés.

Exemple : vérin pneumatique, son distributeur et ses capteurs fin de course.

Boucle : dans un programme, retour en arrière déclenché par une instruction de saut.

Boucle par boucle : méthode utilisée pour traiter plus aisément certains modes de marche et d'arrêt en dehors de la grille du GEMMA.

Byte : mot de 8 bits (synonyme : octet).

C

Capacité mémoire : nombre d'octets de la mémoire. S'exprime en Ko (Ko = 1 024 octets).

Exemple : une mémoire de 4 Ko contient $4 \times 1\,024 = 4\,096$ octets soit $4 \times 1\,024 \times 8 = 32\,768$ bits.

Capteur : composant qui fournit, à la partie commande, des informations binaires, numériques ou analogiques en provenance de la partie relation et de la partie opérative.

Exemple : consigne de l'opérateur, position d'un actionneur, présence d'un objet, phénomène physique, etc.

Chien de garde : dispositif de sécurité interne d'un API chargé de détecter un défaut dans l'exécution du programme. Le chien de garde teste généralement la durée d'un cycle de scrutation du programme. L'information du chien de garde est disponible soit sur un contact de relais électromagnétique, soit par l'intermédiaire d'un bit interne spécifique. En anglais : watch-dog.

Chronogramme : moyen de description graphique des évolutions d'un automatisme séquentiel où le temps constitue la grandeur essentielle.

Exemples : cycle de machine à laver, cycle de feux de carrefour, etc.

Clavier : périphérique servant d'interface de communication entre le programmeur et l'unité centrale de l'API ou de l'ordinateur. Il supporte les touches alphanumériques et de fonction. Suivant la disposition des premières touches, les claviers seront du type AZERTY ou QWERTY.

Codage : transcription d'un langage connu dans un langage conventionnel.

Exemple rencontré en automatique : codage du décimal en décimal codé binaire (DCB).

Composant

de signalisation : composants de la PR ramenant des informations binaires traitées par la PC vers l'opérateur ou les utilisateurs.

Exemples : voyants lumineux, dispositifs sonores, écrans, etc.

Composant

de visualisation : composants de la PR ramenant des informations alphanumériques de la PC vers l'opérateur.

Exemples : écrans à cristaux liquides, écrans à plasmas, écrans cathodiques.

Compteur : le compteur i est noté C_i et son contenu (C_i). La variable de comptage est notée $c_i = n$ où n est la valeur à atteindre. Un compteur doit être mis ou remis à 0 avant utilisation. Le contenu du compteur évolue de la valeur 0 à la valeur n par incrémentation.

Notation : $(C_i) + \text{incrément} \rightarrow C_i$, par exemple $(C_2) + 1 \rightarrow C_2$ ou $C_2 + 1 \rightarrow C_2$.

La variable de comptage $c_i = n$ prend la valeur 1 lorsque la valeur courante C_i , V du compteur est égale à la valeur présélectionnée C_i , $P = n$. Un compteur peut fournir plusieurs variables de comptage : $c_i = n$, $c_i = p$, $c_i = q$, etc. si le compteur dispose d'autant de présélections C_i , $P = n$, C_i , $P = p$, C_i , $P = q$, etc.

Condition : information ou groupe d'informations prenant en compte l'état 0 ou 1 de la ou des variables d'entrée et leur relation logique.

Consigne : ordre émis par l'opérateur. Peut être de nature binaire, numérique ou analogique.

Exemples : ordre départ de cycle (binaire), présélection d'un compteur (numérique), consigne en tension du réglage d'un régulateur de vitesse (analogique).

Console

de programmation : périphérique ou terminal d'un API ou d'un ordinateur comportant un clavier et un écran de visualisation (synonyme : terminal de programmation).

Console de

programmation évoluée : console d'API comportant un clavier et un moniteur à écran cathodique. Ces consoles ont pratiquement disparu au profit des micro-ordinateurs émulés par un logiciel approprié leur permettant de communiquer avec l'API.

Contact électrique : lamelle métallique assurant la fermeture ou l'ouverture d'un circuit électrique. Il s'agit aussi des points d'appui de cette lamelle.

Contacteur : relais électromagnétique de puissance de commande des actionneurs électriques.

Exemple : contacteur pour moteur électrique asynchrone triphasé.

Contenu d'un mot,

d'un compteur : se note entre crochets ou entre parenthèses.

Exemples : [W15] : contenu du mot W15, (C1) : contenu du compteur C1.

Contradiction d'effet : effet s'annulant de signaux simultanés sur les deux pilotages d'un distributeur.

Contraintes : tout ce dont il faut tenir compte puis solutionner pour obtenir un résultat conforme au cahier des charges de l'automatisme.

Contrôle de position : réalisé par un capteur-machine.

Exemple : interrupteur fin de course d'une table commandée par un vérin hydraulique.

Contrôle de présence : réalisé par un capteur-machine.

Exemple : détection d'une grille de protection par une cellule photoélectrique.

Contrôle de résultat : réalisé par un capteur-machine.

Exemple : vérification de la température atteinte avec un thermostat.

Conversationalnel : mode de programmation ou d'exploitation d'un logiciel. Le système pose des questions auxquelles l'utilisateur doit répondre (synonyme : mode interactif). Le système réagit alors en fonction de la réponse donnée.

Coordination

entre GRAFCET : possibilité de faire évoluer un GRAFCET à partir d'informations liées à la situation d'autres GRAFCET.

Corrélation d'événements : relation logique réciproque entre deux événements.

CPU : (acronyme de Central Processor Unit) : processeur ou microprocesseur.

Curseur : repère mobile sur l'écran du moniteur dans les traitements de texte. Ne pas confondre avec le pointeur commandé par la souris.

Cycle automatique : mode de fonctionnement qui permet d'enchaîner de façon continue le cycle de l'automatisme (abréviation : ac ou c/a pour automatique continu).

Cycle de lecture : période correspondant à une exécution unique (lecture) du programme.

Cycle par cycle : mode de fonctionnement discontinu qui n'autorise qu'un seul cycle à chaque intervention de l'opérateur (abréviation : cc ou c/c pour cycle par cycle).

D

DCB : (acronyme de Décimal Codé Binaire) : codage binaire sur 4 bits des chiffres décimaux (0 à 9) n'utilisant que 10 combinaisons sur les 16 possibles.

En anglais : BCD acronyme de *Binary Coded Decimal*.

Décodage : transcription d'un langage conventionnel dans un langage connu.

Exemple rencontré en automatique : décodage du langage binaire en langage décimal codé binaire (DCB).

Décompteur : le décompteur i est noté D_i et son contenu (D_i). La variable de décomptage est notée $d_i = 0$. Un décompteur doit être chargé à la valeur de présélection D_i , $P = n$. Le contenu du décompteur évolue de la valeur n à la valeur 0 par décrémentation.

Notation : $(D_i) - \text{décrément} \rightarrow D_i$, par exemple $(D_5) - 1 \rightarrow D_5$ ou $D_5 - 1 \rightarrow D_5$.

La variable de décomptage $d_i = 0$ prend la valeur 1 lorsque la valeur courante D_i , V du décompteur est égale à 0.

Décrémenter : retrancher un incrément (valeur 1 par défaut) au contenu d'un décompteur.

Délais (hypothèses) : la durée de franchissement d'une transition ainsi que la durée d'activité d'une étape peut être considérée aussi petite que l'on veut mais non nulle. L'activation et la désactivation des étapes aval et amont d'une transition se font simultanément (évolution synchrone). Pratiquement dans les API activation et désactivation se font séquentiellement mais dans le même cycle.

Démultiplexeur : opérateur logique assurant la sélection d'une sortie parmi n sorties.

Désactivation : mise à 0 d'une étape (état inactif).

Diagramme des phases : procédé de description graphique des automatismes séquentiels dérivé du chronogramme.

Dialogue

homme-machine : communication à double sens entre l'opérateur et l'automatisme. Elle est obtenue par l'intermédiaire des composants de la PR : claviers, pupitre de commande, écrans de terminaux, signalisation lumineuse et sonore, imprimante (journal de bord), etc.

Digit : synonyme de chiffre. En binaire le digit est équivalent à un bit (*binary digit*).

Distributeur : préactionneur d'un actionneur pneumatique ou hydraulique.

Exemple : distributeur 5/2 (5 orifices, 2 positions) à simple pilotage électropneumatique alimentant un vérin pneumatique à double effet.

Un distributeur comporte un tiroir plan ou cylindrique à 2 ou 3 positions. Le déplacement du tiroir est commandé par un pilotage avec retour par ressort (type monostable) ou par deux pilotages opposés (type bistable).

Donnée : bit ou groupe de bits (mot) matérialisant une information.

Drapeau : bit d'orientation dans une sélection de séquence. En anglais : flag.

E

EAROM : acronyme d'*Electrically Alterable ROM*. Mémoire morte à écriture et effacement électrique. Synonyme : EEPROM.

Échelle à relais : traduction française de «ladder diagram». Correspond en fait à un schéma électrique à contacts. Le langage graphique à contacts permet la programmation directe des échelles à relais. Les symboles des contacts et des sorties du langage sont différents des symboles des contacts électriques et des bobines de relais.

Écran à cristaux liquides : dispositif d'affichage exploitant la propriété des cristaux liquides de réfléchir la lumière lorsqu'ils sont convenablement orientés sous l'action d'un champ électrique.

- Éditeur** : logiciel facilitant l'entrée des textes en clair ou des programmes écrits en langage mnémotechnique ou littéral.
- EEPROM** : acronyme d'*Electrically Erasable Programmable ROM*. Synonyme : EAROM. Mémoire morte à écriture et effacement électrique.
- Encodage** : synonyme de codage. Ce terme est employé dans le cas du codage des claviers alphanumériques (encodeur de clavier).
- Entrée analogique** : entrée recevant une information non binaire (entrée ANA).
Exemple : tension 0-10 V, courant 0-20 mA.
- Entrée numérique** : entrée multiple regroupant plusieurs entrées tout ou rien dont les signaux sont traités globalement sous forme d'un mot (entrée NUM).
Exemple : traitement en un mot de 16 bits des signaux DCB de 4 roues codeuses branchées sur une entrée numérique.
- Entrée tout ou rien** : entrée recevant une information ou un signal binaire (entrée TOR).
- EPROM** : acronyme de *Erasable PROM*. Mémoire morte programmable et effaçable seulement par exposition aux rayons ultraviolets (UV).
- E/S** : acronyme d'Entrée/Sortie. En Anglais : I/O pour Input/Output.
- Étape** : l'étape permet de définir la situation de la PC.
L'étape est soit active soit inactive. L'étape *i* émet un signal noté X_i . $X_i = 1$ si l'étape *i* est active, $X_i = 0$ si l'étape *i* est inactive. Une étape peut être réactivée sans avoir été désactivée. Une étape à la fois activée et désactivée reste activée (règle 5).
- Étape d'attente** : étape de fin de séquence dans un parallélisme structural.
- Étape initiale** : étape active lorsque la PC est initialisée.
- Étape initialisable** : étape active à la suite d'un forçage.
- Événement** : information prenant en compte le changement d'état ou de niveau d'une variable d'entrée. (Synonyme : front).
Notations : front montant : $0 \rightarrow 1$ (notation \uparrow),
front descendant : $1 \rightarrow 0$ (notation \downarrow).
- Évolution asynchrone** : évolution non conforme aux règles d'évolution 2, 3 et 4 du GRAFCET normalisé (NF C 03-190). La désactivation de l'étape précédente est fonction de l'activation de l'étape suivante (séquenceurs).
- Évolution synchrone** : évolution simultanée s'effectuant entre deux tops d'horloge. Dans le cas du GRAFCET normalisé, évolution conforme aux règles 2, 3 et 4 (NF C 03-190). L'activation de la ou des étapes aval et la désactivation de la ou des étapes amont de la même transition sont simultanées. L'activation et la désactivation sont fonction de la même condition dite fonction transition.
- Expansion** : branche de GRAFCET associée à la macro-étape.

F

- Figeage de la PO** : blocage sur place de tous les actionneurs de la PO par désactivation des entrées des préactionneurs.
Ce résultat est rendu possible si un choix technologique approprié des actionneurs et de leurs préactionneurs a été préalablement fait.
- Figeage de situation** : blocage des évolutions du GRAFCET global ou d'un GRAFCET partiel dans sa situation courante.
Notation : $F/G > \{ * \}$ (cas du GRAFCET Global) ou $F/G_n > \{ * \}$ (cas du GRAFCET G_n).
L'annulation du figeage entraîne la libération du GRAFCET global ou du GRAFCET G_n , c'est-à-dire qu'elle autorise la reprise des évolutions à partir de la situation figée.
- Flag** : drapeau en anglais.
- Fonction mémoire** : en logique séquentielle, fonction assurant la mémorisation d'un signal binaire.
- Fonction opérative d'acquisition (FOA)** : fonction assurée par un composant de la PR traitant les signaux d'entrée.
Exemple : roue codeuse traduisant un chiffre de 0 à 9 en 4 signaux binaires.
- Fonction opérative de commande (FOC)** : fonction assurée par un composant de la PO traitant des signaux de sortie.
Exemple : distributeur pneumatique à double pilotage faisant office de mémoire.
- Fonction opérative de traitement (FOT)** : fonction assurée par un composant de la PO commandé par la PC et renvoyant les signaux traités vers la PC.
Exemple : minuterie lancée par la PC avec retour du signal temporisé vers celle-ci.

Fonction transition : condition qui, lorsqu'elle est vraie, entraîne obligatoirement un changement de situation du GRAFCET. Cette condition vérifie que la transition est validée et que la réceptivité associée est vraie. Si la fonction transition est vraie (égale à 1), la transition est franchissable et obligatoirement franchie, ce qui entraîne simultanément : l'activation de la ou des étapes immédiatement suivantes et la désactivation de la ou des étapes immédiatement précédentes.

Forçage à 0 ou 1 par la console de programmation : mise prioritaire à 0 ou à 1 d'une entrée, d'une sortie ou d'un bit interne. Ce type de forçage permet d'effectuer des tests lors de la mise au point ou pour assurer la maintenance.

Forçage à 0 des actions (sorties) : ordre émis par un GRAFCET supérieur vers un ou des GRAFCET inférieurs ayant pour effet d'annuler toutes les actions de la PO ou seulement celles qui sont associées aux étapes du ou des GRAFCET partiels désignés.
Notations : F/sorties PO = 0 (mise à 0 des sorties listées de la PO),
F/sorties Gn = 0 (mise à 0 des sorties listées du GRAFCET Gn).

Forçage de situation : ordre émis par un GRAFCET supérieur vers un GRAFCET inférieur pour qu'il passe immédiatement de sa situation courante dans une situation imposée sans franchissement de transition.
Notations : forçage du GRAFCET Gn dans la situation S : F/Gn > { s },
forçage du GRAFCET Gn dans une situation vide : F/Gn > { }.
L'ordre de forçage est toujours prioritaire sur les autres conditions d'évolution.

Franchissement de transition : une transition est franchissable et obligatoirement franchie lorsqu'elle est validée et que la réceptivité associée est vraie (fonction transition = 1).

Front descendant : signal pris en compte seulement lors de son évolution de l'état 1 vers l'état 0.
Notation : ↓

Front montant : signal pris en compte seulement lors de son évolution de l'état 0 vers l'état 1.
Notation : ↑

G

GEMMA : acronyme de Guide d'Etude des Modes de Marche et d'Arrêt. Le GEMMA est un moyen graphique (grille) pour rechercher les modes de marche et d'arrêt convenant au problème d'automatisme posé et les conditions pour évoluer d'un mode à un autre en respectant l'unicité de mode. Cette démarche débouche sur le tracé des GRAFCET de gestion de l'énergie, de sécurité et de conduite.

GRAFCET : acronyme de GRAPhe Fonctionnel de Commande, Etape, Transition.

GRAFCET hiérarchisés : ensemble de GRAFCET comprenant en général et pour l'essentiel :
– un GRAFCET de Sécurité noté GS gérant l'énergie sur la PO et les procédures de sécurités,
– un GRAFCET de Conduite noté GC gérant les modes de marche et d'arrêt,
– un GRAFCET de Production normale noté GPN gérant la production normale.
Le GRAFCET GS est hiérarchiquement supérieur au GRAFCET GC et le GRAFCET GC est hiérarchiquement supérieur au GRAFCET GPN. Des ordres de forçages de situation peuvent être émis d'un GRAFCET supérieur vers un GRAFCET inférieur et non l'inverse. GC peut superviser des GRAFCET particuliers à chacun des modes de marche retenus. De même GPN peut correspondre à un ensemble de GRAFCET comprenant un GRAFCET de coordination des tâches et les GRAFCET partiels de chacune des tâches de production. Dans les cas complexes, un GRAFCET de gestion de l'énergie sur la PO noté GEPO est ajouté. Il est hiérarchiquement supérieur au GRAFCET GS.

Graphique : se dit d'un langage permettant la transcription directe du dessin du GRAFCET ou du dessin d'un schéma à contacts sur l'écran de la console de programmation.
Exemples : langage GRAFCET graphique PL7-3 Télémécanique, langage graphique à contacts PL7-2 Télémécanique.

H

Hexadécimal : base de numération à 16 symboles : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Hiérarchisation d'une description par GRAFCET : décomposition de la description en plusieurs GRAFCET du niveau supérieur au niveau inférieur. Hiérarchie couramment utilisée :
– GRAFCET de gestion de l'Energie sur la PO (GEPO),
– GRAFCET de Sécurité (GS),

- GRAFCET de Conduite (GC),
- GRAFCET de coordination des Tâches (GCT) ou GRAFCET de Production normale (GPN),
- GRAFCET des tâches 1 à n (GT1 à GTn).

Horloge : dispositif fournissant des impulsions à une fréquence déterminée.
Exemple : une horloge interne est indispensable à un microprocesseur pour cadencer le traitement des instructions du langage machine.
Le microprocesseur PENTIUM haut de gamme de Intel a une fréquence d'horloge supérieure à 133 MHz.

I

ILS : acronyme d'Interrupteur à Lame Souple (appellation anglaise : REED). Capteur à contact électrique sous ampoule de verre commandable de l'extérieur par un aimant.

Imparité : code de contrôle vérifiant si le nombre de bits à l'état 1 est impair.

Imprimante : périphérique de la console de programmation d'un API ou d'un ordinateur assurant l'impression du listage d'un programme ou du suivi de l'exécution du programme (journal de bord) (synonyme : télétype (TTY)).

Les imprimantes sont caractérisées par leur mode d'impression :

- imprimantes matricielles à aiguilles,
- imprimantes à marguerite (qualité courrier),
- imprimantes à jet ou à bulles d'encre noir et couleur,
- imprimantes à rayon laser noir et couleur (haut de gamme).

Incrément : la plus petite valeur considérée. Synonyme : pas.
Exemple : incrément de comptage.

Incrémenter : ajouter un incrément (valeur 1 par défaut) au contenu d'un compteur.

Initialisation de la PC : mise en condition préalable de la partie commande (PC) d'un système automatisé autorisant le démarrage : RAZ de certaines mémoires, RAZ des compteurs, présélection des décompteurs, désactivation des temporisateurs, activation des étapes initiales, désactivation des autres étapes, etc.

Instruction : code opération du langage machine d'un microprocesseur. L'instruction s'écrit soit en mnémotechnique soit directement en hexadécimal mais de toute façon est traduite en binaire par le programme-moniteur pour pouvoir être transférée dans la mémoire-programme.

Interface : élément se trouvant au point de jonction de l'API avec ses entrées-sorties.
Exemple : coupleurs optoélectroniques assurant l'isolation des cartes d'entrées par rapport aux capteurs de la machine.

Interruption prioritaire : signal pris en compte immédiatement par l'API ou l'ordinateur. Il a pour effet d'arrêter le traitement en cours et de lancer le programme associé à l'interruption.

J

Jump : traduction de « saut » en anglais.

K

K : préfixe, énoncé Kilo, dont la valeur est 2 puissance 10 soit 1 024. Est utilisé pour préciser la capacité mémoire d'un API ou d'un ordinateur.

Exemple : mémoire de 16 K mots de 16 bits.

Ko : abréviation de kilo-octet. 1 Ko = 1 024 octets = 8 192 bits.

L

Langage : ensemble des caractères, des symboles et des règles de syntaxe qui permettent de communiquer avec une machine programmable : micro-ordinateur, automate programmable industriel, manipulateur, robot, etc.

Langage assembleur : langage littéral utilisant le mnémotechnique du microprocesseur plus quelques instructions spécifiques à l'assemblage. Le programme écrit en langage assembleur sera traduit en langage machine (binaire) par l'assembleur proprement dit.

Langage GRAFCET alpha-

numérique ou littéral : langage possédant un jeu d'instructions pour programmer les étapes, les transitions, les réceptivités et les actions.

Langage GRAFCET

graphique : langage disposant des symboles graphiques pour tracer la structure du GRAFCET sur l'écran de la console ou du moniteur. Les réceptivités et les actions sont programmées soit en langage littéral soit en langage graphique à contacts soit en logigramme.

Langage informatique : langage graphique pour API disposant des symboles pour tracer directement l'organigramme sur l'écran de la console ou du moniteur.

Langage logique : langage alphanumérique ou littéral disposant uniquement des instructions pour programmer les opérateurs logiques combinatoires (NON, ET, OU, etc.) et séquentiels (mémoires).

Langage logigramme : langage graphique disposant des symboles des opérateurs logiques pour tracer un logigramme sur l'écran de la console ou du moniteur.

Langage machine : langage binaire du microprocesseur qu'on visualise généralement en hexadécimal.

LCD : acronyme de *Liquid Crystal Display*. Composant d'affichage à cristaux liquides.

LED : acronyme de *Light Emitting Diode*. Composant d'affichage à diode électroluminescentes. En français DEL : Diode ElectroLuminescente.

Liaisons orientées : les liaisons indiquent les voies d'évolutions possibles de la situation du GRAFCET. Elles relient les étapes aux transitions et les transitions aux étapes sous forme de lignes verticales et horizontales. Sauf indication contraire (flèche) l'évolution se fait toujours de haut en bas.

Logiciel : programme ou ensemble de programmes pour effectuer un traitement sur ordinateur. On distingue le logiciel de base ou système d'exploitation et les logiciels d'application : éditeurs, tableurs, etc.

Logigramme : schéma représenté à l'aide des symboles des opérateurs logiques combinatoires (NON, ET, OU, NOR, NAND, comparateurs, etc.) et séquentiels (mémoires, bascules, compteurs, etc.).

M

Macro-représentation : terme générique regroupant les concepts de sous-programmes ou de tâches et de macro-étapes.

Macro-étape : représentation unique d'un ensemble d'étapes et de transitions décrit par ailleurs sous forme d'expansion de la macro-étape.

Marche automatique : mode de marche autorisant la production normale en continu ou en cycle par cycle (état F1 du GEMMA).

Marche étape par étape : mode de marche permettant la vérification dans l'ordre du cycle de production (état F5 du GEMMA).

Marche manuelle : mode de marche permettant la vérification dans le désordre du fonctionnement des actionneurs (état F4 du GEMMA).

Master-relay : relais de tête d'un schéma à contacts. En français : relais-maître.

Mécanisme : ensemble d'organes mécaniques assurant une fonction déterminée pouvant se substituer à une opération manuelle.

Exemple : plateau rotatif composé pour l'essentiel d'un vérin pneumatique avec pignon et crémaillère entraînant le plateau.

Mémoire : dispositif capable de stocker une ou plusieurs quantité(s) d'informations binaires.

Mémoire à effacement

prioritaire : mémoire d'un bit câblée ou programmée où le signal d'effacement est prioritaire sur le signal d'inscription.

Expression logique : $X = /Ex (Ix +x)$

Mémoire à inscription

prioritaire : mémoire d'un bit câblée ou programmée où le signal d'inscription est prioritaire sur le signal d'effacement.

Expression logique : $X = Ix + /Ex.x$

C'est ce type de mémoire qui permet de matérialiser la règle 5 du GRAFCET.

Mémoire programme : mémoire vive sauvegardée (batterie) ou sauvegardable (EEPROM) où est placé le programme de l'automatisme étudié.

Mémoire vive : mémoire s'effaçant dès qu'elle n'est plus alimentée (RAM). Elle contient le programme de l'automatisme. Elle peut être sauvegardée par batterie interne.

Mémoire morte : mémoire non effaçable (ROM) ou effaçable sur commande (EPROM, EEPROM).

- Méthode matricielle** : méthode d'analyse des automatismes séquentiels où le retard entre la mise à 1 ou à 0 de l'entrée et de la sortie d'une variable auxiliaire est pris en compte (synonyme : méthode d'Huffman).
Exemple : prise en compte du retard entre l'excitation de la bobine d'un relais électromagnétique et la fermeture de son contact.
- Microprocesseur** : processeur intégré sur une puce de silicium.
- Minuterie** : temporisateur électromécanique ou électronique gérant des temps importants (jusqu'à 48 heures, par exemple). Les minuterie électromécaniques sont construites autour d'un moteur électrique synchrone associé à une boîte de vitesses (cas des minuterie multi-gammes). Un doigt fixé sur un plateau agit en fin de rotation sur un interrupteur fin de course (ou sur deux décalés). Le signal de sortie est disponible soit sous forme d'impulsion soit sous forme continue jusqu'au réarmement de la minuterie.
- Mise en référence** : procédure pour mettre les actionneurs dans la position correspondant au début du cycle de l'installation automatisée (état A6 du GEMMA).
- Mnémonique** : (de mémoire) écriture simplifiée des codes opérations à l'aide des lettres significatives pour un microprocesseur donné.
Exemple : mnémonique *LDA pour LoaD register A from memory*.
- Modes d'arrêt** : procédures envisagées pour arrêter une installation automatisée soit dans les conditions normales de fonctionnement soit en cas d'urgence afin d'assurer la sécurité des personnes et la protection du matériel.
- Modes de marches** : définition des divers fonctionnements possibles de l'installation automatisée.
- Moniteur** : composant comportant un écran cathodique noir ou couleur. Ne pas confondre avec le programme-moniteur.
- Mot** : ensemble de bits traités globalement. En anglais : word.
Exemples : mot de 8 bits, mot de 16 bits.
- Multiplexeur** : opérateur logique assurant la sélection d'une entrée parmi n entrées.

N

- Niveau 1** : premier niveau de description par GRAFCET, indépendamment de tout choix technologique. Correspond à une description faite d'un point de vue processus : inventaire des tâches et recherche de leur coordination. Cette analyse se traduit par le tracé d'un GRAFCET de coordination des tâches.
- Niveau 2** : deuxième niveau de description par GRAFCET après choix technologique des composants de la PO et de la PR : actionneurs, préactionneurs et capteurs. Correspond à une description faite d'un point de vue PC Cette analyse débouche sur un ensemble de GRAFCET hiérarchisés.
- Niveau 3** : troisième niveau de description par GRAFCET après choix technologique du constituant de la PC Correspond à une description d'un point de vue réalisateur.

O

- Octet** : ensemble de 8 bits (en anglais : byte).
- Occurrence d'événements** : simultanéité de présence de deux événements.
- Opérateur** : conducteur de la machine ou de l'installation automatisée.
- Opérateurs logiques combinatoires** : opérateurs dont l'état de la sortie ne dépend que de la combinaison de l'état de ses entrées.
Exemples : opérateurs NON, ET, OU, transcodeur, etc.
Ces opérateurs sont représentés par des symboles normalisés (NF C 03.212 et E 49-600). Ces opérateurs sont matérialisés sous forme modulaire en technologie électronique et pneumatique.
- Opérateurs logiques séquentiels** : opérateurs dont l'état de la sortie dépend de l'ordre dans lequel l'état de ses entrées a évolué.
Exemples : mémoires, bascules, etc.
Ces opérateurs sont représentés par des symboles normalisés (NF E 49-602). Ces opérateurs sont matérialisés sous forme modulaire en technologie électronique ou pneumatique.
- Organigramme** : représentation graphique d'un algorithme utilisant les symboles de la norme NF Z 67-010.

- P** **Parallélisme interprété** : structure particulière d'un GRAFCET où plusieurs branches parallèles peuvent être dans certains cas exclusives et dans d'autres cas inclusives c'est-à-dire à évolutions simultanées.
- Parallélisme structural** : structure particulière d'un GRAFCET comportant des branches à évolutions simultanées.
- Parité** : code de contrôle vérifiant si le nombre de bits à l'état 1 est pair.
- Partie commande**
- (PC ou P.C.)** : partie de l'automatisme qui regroupe les composants et les constituants assurant le traitement des informations d'entrée en vue de commander séquentiellement les sorties.
- Partie opérative**
- (PO ou P.O.)** : partie de l'automatisme regroupant les actionneurs, les préactionneurs et les capteurs-machines.
- Partie relation (PR ou P.R.)** : partie de l'automatisme regroupant tous les composants (éléments de signalisation, capteurs-opérateurs, terminaux, etc.) assurant la communication entre le ou les opérateurs et la ou les PC ou entre les PC elles-mêmes.
- Pas à pas** : registre dont une seule sortie est à l'état 1 à la fois. L'activation du pas suivant entraîne la désactivation du pas précédent (évolution synchrone ou asynchrone suivant les dispositifs pas à pas).
- Périphérique** : dispositif extérieur à l'API ou à l'ordinateur et nécessaire à sa mise en œuvre (clavier, moniteur) ou à l'accomplissement d'une tâche particulière (imprimante, lecteur-enregistreur de disque magnétique souple (floppy), table traçante, etc.).
- Pilotage de distributeur** : interface entre la PC et le tiroir du distributeur. Le pilotage peut être du type électrique (électro-aimant), pneumatique ou hydraulique ou électropneumatique.
- Pointeur** : repère lumineux sur l'écran d'une console ou d'un moniteur et déplaçable avec la souris. Le pointeur permet de désigner sur l'écran l'élément du menu qui intéresse l'utilisateur. Un clic sur le bouton gauche de la souris valide ce choix (ne pas confondre avec le curseur).
- Position de référence** : position initiale des actionneurs choisie par le concepteur de l'automatisme. La mise en référence de la PO a pour objet justement de placer les actionneurs dans cette position particulière.
- Préactionneur** : composant d'interface entre l'API et l'actionneur.
Exemples : contacteur électromagnétique alimentant un moteur électrique, distributeur pneumatique alimentant un vérin.
- Présélection** : chargement préalable d'un registre dont le contenu sera comparé à celui du compteur ou du décompteur (synonymes : prépositionnement, prédétermination).
- Procédé** : méthode à suivre pour obtenir un résultat (fabrication, conditionnement, etc.).
- Procédure** : souvent synonyme de sous-programme.
- Procédure d'arrêt d'urgence** : ensemble des dispositions prévues en cas de demande d'arrêt d'urgence.
- Processeur** : unité centrale de traitement (CPU) composé de l'unité arithmétique et logique, d'un séquenceur associé à une horloge, de registres divers. Le CPU contrôle et exécute les instructions du programme.
- Processus** : ensemble des opérations d'élaboration d'un produit selon un procédé déterminé au moyen d'unités de traitement et de transformation.
- Programmeur cyclique** : composant permettant la commande pas à pas des sorties.
- Programme** : suite des instructions à faire exécuter par le processeur. Un programme écrit en langage graphique sera de toutes façons traduit en une suite d'instructions elles-mêmes traduites en binaire.
- Programme-moniteur** : programme en mémoire morte (ROM ou EPROM) chargé de gérer les fonctions de base de l'API et de sa console de programmation : contrôle du clavier, liaison clavier écran, affichage sur l'écran, interprétation du langage de programmation, etc.
En abrégé : moniteur (à éviter, risque de confusion avec l'écran cathodique du terminal).
- Programmeur** : personne chargée de l'étude et du chargement des programmes.
- PROM** : acronyme de *Programmable ROM*. Mémoire morte programmable par l'utilisateur.
- Pupitre de commande** : partie de l'installation automatisée regroupant les capteurs-opérateurs, les dispositifs de visualisation (écrans) et de signalisation lumineuse et sonore.

Q

QWERTY : caractérise un clavier d'ordinateur ou de console de programmation d'un API dont les premières touches forment ce mot. Autre clavier : clavier AZERTY.

R

RAM : acronyme de *Random Access Memory*. Mémoire vive à accès aléatoire.

RAU : acronyme de Remise A Un.

RAZ : acronyme de Remise A Zéro.

Réceptivité (condition de transition) : proposition logique associée à une transition. La réceptivité peut être vraie (état 1) ou fausse (état 0). La réceptivité est une expression logique combinatoire de une ou plusieurs variables (conditions et/ou événements). Les variables peuvent être :

- des variables d'entrées ou de sorties,
- des variables internes (bits internes B_i , bits d'étapes X_i),
- des variables de comptage, de décomptage, de temporisation, etc.

Récupération de la situation courante sauvegardée

avant vidage: possibilité offerte par certains langages d'API et par certains logiciels de PGOA. Notation de l'ordre de Restauration (ResToration en anglais) de la situation courante associée à l'étape j d'un GRAFCET supérieur G_s : $RT(G_n) > \{ * \}$
Notation du GRAFCET inférieur G_n dont la situation courante est restaurée : $X_j > RT(G_n) > \{ * \}$

Registre : mémoire destinée au stockage temporaire des informations (données, adresses, etc.)

Registre à décalage : registre où les bits peuvent subir une translation à droite ou à gauche avec ou sans rebouclage. Aucun ou un ou plusieurs bits peuvent être à l'état 1 à la fois (synonyme : pile). Si le premier bit entré est le premier sorti on parle de pile FIFO (First In, First Out). Si le dernier bit entré est le premier sorti, on parle de pile LIFO (Last In, First Out).

Registre à désempilage : registre qui se remplit en une seule opération et se vide progressivement.

Registre à empilage : registre qui se remplit progressivement puis se vide en une seule opération..

Registre pas à pas : dispositif dont un seul pas est actif à la fois.

Règles d'évolution

du GRAFCET : règle 1 : relative à la situation initiale,
règle 2 : relative au franchissement d'une transition,
règle 3 : relative à l'évolution de la situation,
règle 4 : relative aux évolutions simultanées,
règle 5 : relative à l'activation et à la désactivation simultanée de la même étape.

Règles de syntaxe : ces règles appliquées au GRAFCET assurent le respect de la règle d'alternance étape-transition et transition-étape.

Régulation : automatisme où la sortie est maintenue à une valeur fixe déterminée par la valeur d'entrée ou consigne.

Exemple : régulation en température d'une salle. Entrée : tension-image de la température, sortie : commande du compresseur et d'un ventilateur d'un ensemble de climatisation, contrôle de la température par thermostat.

La comparaison entre la consigne de température et la valeur mesurée par le thermostat agit en conséquence sur la marche du compresseur et sur celle du ventilateur.

Autre exemple : régulateur de vitesse d'un moteur électrique. Entrée : tension-image de la vitesse souhaitée, sortie : variateur de vitesse, contrôle par dynamo tachymétrique.

Relais électromagnétique : composant dont l'entrée est une bobine et la sortie des contacts électriques. L'excitation de la bobine crée un champ magnétique attirant l'armature des contacts (mise à 1). La mise à 0 des contacts est obtenue par des ressorts lorsque la bobine est désexcitée (relais version monostable).

Le relais électromagnétique bistable comporte deux bobines excitées momentanément. La mémorisation est obtenue mécaniquement (relais à accrochage) ou magnétiquement. Un relais électromagnétique de puissance s'appelle un contacteur.

Reprise de séquence : sélection de séquence particulière permettant au GRAFCET d'évoluer en effectuant un retour en arrière vers une séquence interne déjà rencontrée si une certaine condition est remplie. La branche de la reprise de séquence ne comporte pas d'étape.

Réseau de contacts : partie de schéma en langage graphique à contacts.

Exemple : en langage PL7-2 Télémécanique, un réseau de contacts comporte 4 lignes maximum.

ROM : acronyme de *Read Only Memory*. Mémoire morte à lecture seule.

S

Saut d'étape : sélection de séquence particulière permettant au GRAFCET d'évoluer en sautant volontairement des étapes si une certaine condition est remplie. La branche du saut d'étape ne comporte pas d'étape.

Sauvegarde de la situation

courante d'un GRAFCET : possibilité offerte par certains langages d'API et par certains logiciels de PGO. Notation de l'ordre de SauVegarde (SaVe en anglais) de la situation courante, suivi d'une mise en situation vide associé à l'étape i d'un GRAFCET supérieur G_s : $SV(G_n) > \{ * \}$
Notation du GRAFCET inférieur G_n dont la situation est sauvegardée puis vidée :
 $X_i > SV(G_n) > \{ * \}$

Sensibilité : le système est dit sensible au changement d'état d'une variable (condition ou événement) lorsque seul l'état d'une ou de plusieurs actions est modifié (actions conditionnelles), sans que la situation du GRAFCET soit changée. Si au contraire la situation évolue par franchissement d'une transition, on dit que le système est réceptif à cette variable.

Séquence : dans un GRAFCET, suite linéaire d'étapes et d'actions associées. La séquence peut être active (au moins une étape active) ou inactive (situation vide).

Séquence unique : structure d'un GRAFCET ne comportant ni divergence, ni convergence.

Séquences exclusives : séquences parallèles dont l'évolution simultanée a été rendue impossible par un choix approprié des expressions logiques des réceptivités. Le saut d'étape et la reprise de séquence sont des cas particuliers de sélection de séquence.

Séquences inclusives : séquences parallèles qui peuvent être, suivant les conditions d'évolution liées aux réceptivités, soit simultanées soit exclusives (parallélisme interprété).

Séquences simultanées : séquences évoluant en parallèle à partir d'une même transition (parallélisme structural).

Séquenceur : ensemble de composants accolés permettant de réaliser un automatisme décrit par GRAFCET. L'évolution du séquenceur étant de type asynchrone ne permet pas la réalisation d'une boucle à deux étapes.

Le module d'étape, composant essentiel du séquenceur, est constitué d'un opérateur logique séquentiel (mémoire bistable matérialisant l'étape) et d'opérateurs logiques combinatoires. Un ET permet le calcul de la fonction transition d'activation du module suivant. Un OU permet la désactivation du module précédent ou la RAZ générale du séquenceur. Les séquenceurs pneumatiques restent pratiquement les seuls en lice, les séquenceurs électriques et les séquenceurs électroniques ayant été supplantés par les API.

Seuil : valeur qu'une grandeur physique doit atteindre pour être prise en compte
Exemple du manostat : seuil maxi 5 bars \rightarrow état 1,
seuil mini 3 bars \rightarrow état 0.

SFC : (diagramme fonctionnel en séquence) désignation de la norme européenne EN 61131-3 qui est la transposition du diagramme fonctionnel de la norme française NF C03-190.

Noter que, dans cette norme européenne, la partie 3 relative aux langages de programmation pour API, les termes « fonction transition » et « réceptivité » sont ignorés et remplacés respectivement par les termes « transition » et « condition de transition ». Les 5 règles d'évolution du modèle GRAFCET n'y sont pas clairement définies. Au niveau des actions, cette norme élimine la commande S (Stored) et la remplace par S (Set) et R (Reset).

Signalisation : fonction assurée par les composants de la PR Peut être lumineuse ou sonore.

Situation : à un instant donné, ensemble des étapes actives d'un GRAFCET.

Notations : $S = \{i\}$ ou $S(G_n) = \{i\}$ où i est la seule étape active à l'instant considéré,
 $S = \{i,j,k\}$ ou $S(G_m, G_n, G_p) = \{i,j,k\}$ où i, j, k sont les étapes actives à l'instant considéré.

Sortie analogique : sortie à laquelle est appliquée une tension ou injecté un courant proportionnel à la valeur binaire traitée (sortie ANA).

Exemple : tension 0 à 10 V appliquée à l'entrée d'un régulateur de vitesse.

Sortie numérique : sorties binaires traitées globalement sous forme d'un mot (sortie NUM).

Exemple : mot appliqué à l'entrée du décodeur d'un afficheur numérique.

Sortie tout ou rien : sortie binaire pouvant prendre l'état 0 ou 1 (sortie TOR).

Exemple : signal appliqué à l'entrée d'une bobine d'un pilotage de distributeur électropneumatique.

Souris : périphérique se substituant au clavier dans les logiciels à mode graphique. La souris permet de piloter une flèche (pointeur) sur l'écran du moniteur. La souris dispose de 2 ou 3 boutons pour déclencher la commande du menu pointée par la flèche : clic sur le bouton gauche ou droit.

Sous-programme : structure correspondant à une séquence d'opérations rencontrée ou utilisée plusieurs fois dans le programme. Le concept d'étape de lancement d'un GRAFCET sous-programme (ou tâche) facilite l'analyse descendante du problème. Le tracé d'un GRAFCET de coordination des tâches devient possible sans qu'il soit nécessaire de connaître les détails du sous-programme (ou de la tâche). *En anglais : subroutine.*

Spécifications

fonctionnelles : elles caractérisent les comportements de la PC face aux informations provenant de la PO et de la PR. Elles permettent de faire l'inventaire des fonctions et des tâches à réaliser.

Spécifications

opérationnelles : elles caractérisent les comportements que doit avoir la PC et le système automatisé dans le contexte de production (performances globales, sécurité, protection, etc.).

Spécifications

technologiques : elles prennent en compte les choix technologiques des composants et des constituants de l'ensemble du système automatisé (PO, PR et PC).

Structuration

d'un programme : organisation modulaire du programme.

Structure d'un système automatisé décrit par

GRAFCET : description faisant notamment appel aux concepts de macro-représentation : GRAFCET sous-programmes, GRAFCET de tâches, macro-étapes et leurs expansions.

Synchronisme : caractérise des évolutions qui ont lieu en même temps ou dans un intervalle de temps fini et relativement faible (évolution entre deux tops d'horloge).

Syntaxe : règles régissant l'écriture et l'ordonnement des instructions dans un langage.

Système binaire : système de numération utilisant 2 symboles (base 2) : 0, 1.

Système décimal : système de numération utilisant 10 symboles (base 10) : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Système hexadécimal : système de numération utilisant 16 symboles (base 16) : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Système octal : système de numération utilisant 8 symboles (base 8) : 0, 1, 2, 3, 4, 5, 6, 7.

T

Tableau d'analyse : tableau définissant l'état des sorties en fonction de l'état des entrées.

Tableau de Karnaugh : procédé graphique de simplification des équations logiques.

Technoguides : fascicules servant de guide pour le choix technologique des constituants à retenir pour réaliser la PO et la PC notamment. Les technoguides sont diffusés par l'ADEPA.

Temporisateur : opérateur gérant le temps écoulé. Le temporisateur i est noté T_i . Il est défini :
– par son origine de lancement : étape i soit X_i ,
– par sa durée q ,
– par sa variable de temporisation notée $t_i/X_i/q$.

Temporisateur analogique : temporisateur où la prédétermination du retard ou du maintien est obtenue par réglage du courant de charge d'un condensateur (action sur une résistance réglable d'un temporisateur électronique) ou par réglage du débit de remplissage d'une capacité d'air (relais électromagnétique à temporisation pneumatique).

Temporisateur numérique : temporisateur basé sur l'exploitation d'un compteur ou d'un décompteur d'impulsions fournies par une base de temps ou horloge. Le signal est obtenu en sortie en comparant le contenu du compteur ou du décompteur avec le contenu du registre de présélection.

Terminal : composant comportant un clavier et un écran cathodique ou à cristaux liquides ou à plasma permettant le dialogue homme-machine.

Transcodage : traduction d'un langage codé dans un autre langage également codé.
Exemple : transcodage du binaire naturel en binaire réfléchi.

Transition : la transition indique la possibilité d'évolution entre une ou plusieurs étapes. La transition est soit validée soit non validée. La transition est franchissable et obligatoirement franchie lorsqu'elle est validée et que la réceptivité associée est vraie.

Transition puits : transition sans étape suivante. Elle conduit à une situation vide.

Transition source : transition sans étape précédente. Elle est toujours validée.

U

Unicité de mode : signifie que l'automatisme ne doit se trouver que dans un seul mode de marche ou d'arrêt à la fois. C'est au concepteur qu'incombe la vérification de l'unicité de mode et la recherche du moyen de l'assurer.

V

Validation

d'une transition : une transition est validée lorsque toutes ses étapes immédiatement précédentes sont actives.

Variable de situation : expression logique permettant de vérifier qu'un forçage a bien été réalisé.

Notation condensée : $S(\text{GRAFCET forcé}) = \{\text{situation forcée}\}$.

Exemple : la variable $S(\text{Gn}) = \{20\}$ vérifie que le GRAFCET Gn a bien été forcé sur la situation $\{20\}$. Cette variable doit être explicitée, elle a pour expression logique : $S(\text{Gn}) = \{20\} = X20./X21./X22./X23\dots$

Visualisation

alphanumérique : affichage de textes et de valeurs numériques.

Visualisation numérique : affichage de valeurs numériques (temps écoulé, température atteinte, comptage en cours, etc.).

X

Xi : notation du signal émis par l'étape i.

Si l'étape i est active, alors $X_i = 1$.

Si l'étape i est inactive, alors $X_i = 0$.

Y

Yi : notation de la fonction transition (logiciel CADEPA).

Z

Zoom : effet utilisé en informatique pour accéder à un détail. Exemple : en langage PL7-2, en effectuant un zoom à partir du pointage de la transition, on accède au réseau de contacts de la réceptivité associée.

A

action	7, 9, 12, 13, 14, 29, 31, 32, 33, 57, 60, 61, 62, 67, 76, 87, 89, 90, 111, 138, 153, 154, 157, 158, 160, 184, 189, 191, 192, 221, 234, 240, 245, 246
action à l'activation	87
action à la désactivation	87
action à niveau	62, 221, 234
action conditionnelle	62, 63, 221, 234
action conditionnelle à durée limitée	63, 234
action conditionnelle retardée	63, 222, 234
action continue	62, 87, 221
action continue temporisée	87, 234
action impulsionnelle	63, 87, 222, 234
action maintenue	64, 234
action mémorisée	64, 221, 234
actionneur	3, 4, 5, 14, 22, 24, 71, 94, 202, 205, 234, 235, 236, 237, 238, 241, 242
action ponctuelle	63, 222, 234
activation d'une étape	11, 12, 26, 62, 95, 116, 117, 118, 119, 131, 151, 160, 163, 231, 234, 237, 238, 239, 240, 243, 244, 245
ADEPA	2, 15, 25, 205, 246
AFCET	1
aiguillage	57, 58, 201, 216, 218, 234
aléa de séquence	123, 159, 230
algorithme	1, 8, 10, 61, 117, 134, 170, 171, 172, 174, 224, 234, 242
analyse combinatoire	226
analyse descendante	6, 23, 83, 102, 196, 234, 246
arborescence	4, 234
assembleur	76, 126, 127, 136, 147, 153, 162, 172, 234, 240
asservissement	61, 234
automate programmable industriel	2, 5, 15, 20, 26, 33, 83, 96, 111, 121, 123, 182, 187, 205, 217, 218, 234, 241
automatiser	3
AUTOMGEN société IRAI	95, 97, 182, 190 à 194
autorupture	231
AZERTY	234, 236, 243

B

base de temps	52, 53, 54, 142, 235
BFC	180, 181
bit	26, 29, 95, 96, 97, 98, 121, 125, 126, 191, 193, 235, 237, 241, 242, 243
bit bistable	121, 124, 126, 131, 133, 159, 160, 165, 166, 219, 235
bit d'étape	108, 244
bit drapeau	58, 202, 211, 212, 218, 237
bit glissant du C200	144
bit interne	33, 34, 53, 108, 125, 160, 171, 185, 186, 190, 219, 234, 235, 239, 240, 244
bit monostable	121, 124, 126, 127, 129, 130, 159, 165, 219, 235
bit synchrone	155, 156
bit système	87, 156, 175, 176, 235
bloc-fonction compteur	54
bloc-fonction décompteur	53, 54

blocs fonctionnels constructeurs (BFC)	180
bloc-fonction optionnel (OFB)	177
bloc-fonction temporisateur	53, 54
bloc-opératif	24, 231, 235
bloc opératif étendu	228, 235
boucle	26, 75, 100, 106, 118, 207, 208, 209, 235
boucle à deux étapes	26, 106, 118, 123, 125, 160, 231, 245
boucle à une seule étape	106
boucle ouverte	26, 100
boucle par boucle	207, 235
byte	235

C

CADEPA société Eurilor	125, 182 à 187
capacité mémoire	20, 25, 160, 235, 240
capteur fin de course	34, 51, 107, 200, 203, 205, 211, 228, 229, 231, 235
capteur-opérateur	4, 5, 9, 22, 25, 243
capteur-machine	5, 9, 22, 24, 202, 205, 236, 243
cause et effet	17
CEI	2
chien de garde	75, 98, 129, 235
chronogramme	1, 21, 35, 227, 228, 236, 237
composant de signalisation	4, 5, 236, 241
composant de visualisation	4, 5, 236, 241, 242
compteur	23, 26, 34, 39, 46, 53, 63
concepts du GRAFCET	29, 176, 183
condition toujours vraie	37
consigne	9, 14, 234, 236, 244
console de programmation	30, 33, 35, 83, 120, 123, 132, 143, 144, 153, 154, 157, 160, 164, 194, 234, 236, 239, 240, 241, 243, 245
contact électrique	2, 49, 51, 53, 74, 75, 87, 94, 98, 120, 121, 236, 237, 240, 244
contacteur	19, 22, 23, 31, 70, 94, 111, 231, 236, 243, 244
contradiction d'effet	229, 230, 236
contraintes	6, 7, 15, 23, 24, 27, 180, 196, 197, 236
contrôle de position	9, 236
contrôle de présence	9, 236
contrôle de résultat	9, 236
coordination asynchrone entre GRAFCET	68, 78, 237
coordination des tâches	6, 7, 8, 24, 83, 196, 198, 208, 209, 210, 211, 212, 214, 239, 240, 242, 246
coordination entre GRAFCET	27, 68, 77, 158, 196, 198, 208, 209, 211, 212, 237, 239, 240, 242, 245
coordination synchrone entre GRAFCET	79
corrélation d'événements	48, 237
curseur	144, 189, 237, 243
cycle automatique	4, 5, 6, 10, 12, 19, 198, 237, 241
cycle de lecture	48, 123, 125, 129, 130, 185, 237
cycle de production	70, 76, 236
cycle de scrutation du programme	49, 125, 129, 130, 185, 235, 236
cycle par cycle	19, 154, 237, 241

D

décompteur	34, 53, 54, 63, 237, 240, 243, 247
désactivation d'une étape	11, 12, 26, 39, 40, 47, 87, 123, 124, 131, 151, 160, 163, 237, 238, 239, 240, 244
détecteur de présence	34
diagramme-bloc	17, 18
diagramme des phases	1, 228, 237
diagramme fonctionnel en séquence (SFC)	245
dialogue homme-machine	5, 22, 27
dimension finesse	24
dimension point de vue	23
dimension spécifications techniques	23
distributeur	5, 14, 15, 19, 22, 31, 48, 64, 66, 70, 94, 120, 205, 228, 229, 230, 235, 236, 237, 238, 243, 245
drapeau	58, 136, 201, 202, 211, 212, 218, 237, 238

E

échelle à relais	237
entrée analogique	235, 236, 238
entrée numérique	238
entrée tout ou rien	191, 235, 238
erreur d'analyse	102
erreur d'écriture	56, 102, 106
erreur de forme	102, 103, 104, 106
erreur de syntaxe	102, 103, 120
étape	11, 12, 13, 15, 29, 30, 31, 33, 34, 39, 40, 60, 116, 121, 162, 238
étape d'attente	105, 107, 238
étape initiale	12, 30, 39, 40, 77, 78, 82, 90, 91, 117, 162, 238
étape initialisable	30, 91, 162, 238
étapes fusionnables	110, 111
étapes redondantes	108, 110, 111
événement	35, 36, 47, 48, 63, 234, 237, 238
évolution asynchrone	48, 117, 118, 119, 121, 238, 243
évolutions simultanées	47, 55, 59, 118, 243, 244
évolution synchrone	48, 117, 118, 237, 238, 243, 245
expansion de la macro-étape	68, 69, 85, 86, 87, 159, 238, 241, 246

F

figeage de la PO	70, 71, 94, 95, 223, 238
figeage de la PO après fin des mouvements en cours	94, 95, 238
figeage de la situation courante	62, 69, 70, 88, 92, 93, 94, 95, 98, 101, 152, 175, 176, 177, 193, 208, 210, 214, 223, 235, 238
fonction globale	6, 196, 234
fonction mémoire	19, 50, 64, 66, 67, 110, 116, 117, 119, 120, 121, 123, 124, 159, 164, 165, 185, 186, 219, 238
fonction opérative d'acquisition (FOA)	238
fonction opérative de commande (FOC)	19, 64, 238
fonction opérative de traitement (FOT)	25, 52, 54, 238
fonction principale	6, 24, 196, 197
fonction secondaire	6, 7, 24
fonction transition	40, 100, 116, 117, 120, 124, 125, 126, 127, 131, 132, 133, 134, 147, 159, 164, 165, 171, 172, 185, 186, 231, 239, 240, 245, 247

forçage à 0 des sorties	98, 223, 235, 239
forçage à 0 ou à 1 des E/S par la console	98, 239
forçage de situation	30, 70, 71, 88, 91, 92, 97, 101, 117, 151, 154, 158, 162, 163, 164, 170, 174, 184, 193, 223, 235, 238, 239, 247
franchissement de transition	33, 34, 40, 46, 48, 68, 69, 82, 88, 100, 123, 134, 147, 151, 193, 237, 239, 244
front descendant	35, 47, 48, 49, 50, 51, 63, 87, 189, 191, 234, 238, 239
front montant	35, 47, 48, 49, 50, 51, 63, 64, 67, 87, 180, 189, 191, 234, 238, 239
fusionnement des étapes	59, 108, 110, 111, 213, 219

G

GEMMA	21, 24, 39, 72, 76, 102, 187, 202, 205, 206, 207, 210, 235, 239, 241, 242
GRAF CET	1, 2, 3, 29, 30, 39, 102, 108, 116, 239
GRAF CET à deux étapes	108, 111
GRAF CET à étape active unique	55, 109
GRAF CET de conduite	24, 71, 94, 166, 167, 209, 210, 239, 240
GRAF CET de coordination des tâches	2, 24, 76, 80, 85, 86, 113, 196, 198, 208, 209, 211, 212, 214, 239, 240, 242, 246
GRAF CET de gestion de l'énergie	25, 71, 239, 240
GRAF CET de production	24, 70, 93, 94, 95, 166, 167, 168, 208, 210, 239, 240
GRAF CET de sécurité	24, 71, 166, 167, 209, 210, 239, 240
GRAF CET de tâche	24, 61, 62, 76, 80, 83, 86, 93, 94, 95, 192, 246
GRAF CET esclave	157, 158, 159, 181
GRAF CET hiérarchisés	24, 30, 71, 91, 99, 164, 182, 187, 239, 242
GRAF CET maître	157, 158, 159
GRAF CET particulier	24, 49, 64, 65, 218, 239
GRAF CET sous-programme	24, 62, 67, 68, 76, 77, 78, 79, 80, 83, 157, 192, 235, 246
GREPA	2

H

hiérarchie entre GRAF CET	24, 25, 30, 69, 71, 88, 89, 91, 99, 102, 158, 162, 166, 178, 179, 182, 187, 202, 209, 239, 240, 242
horloge	52, 54, 99, 182, 235, 239, 240, 243, 246
Huffman	1, 226, 242
hypothèses sur les durées relatives aux évolutions	29, 35, 46, 48, 237

I

ILS	203, 205, 216, 240
imprimante	182, 187, 237, 240, 243
incrément	46, 63, 236, 237, 240
incrémenter	46, 63, 191, 240
initialisation de la PC	39, 117, 154, 238, 240
interface	25, 60, 236, 240, 244
interrupteur à lame souple (ILS)	203, 205, 216, 240

J

journal de bord	61, 237, 240
jump (instruction, bobine)	75, 76, 139, 162, 240

L

langage AWL Festo	139, 140
langage de PGO Automgen Irai	95, 96, 97, 190 à 194
langage de PGO Cadepa	182 à 185

langage de PGO Omega	187 à 190
langage GRAFCET graphique	
ORPHEE April	97, 157 à 159, 180
langage GRAFCET graphique	
P4 V5 Alspa	154, 157
langage GRAFCET graphique PL7-2	
Télé mécanique	53, 96, 153, 154, 176
langage GRAFCET graphique PL7-3	
Télé mécanique	33, 49, 53, 67, 153, 154, 176
langage GRAFCET graphique PL7-3 V5	
Télé mécanique	95, 96, 177
langage GRAFCET littéral C100 Alspa	144, 145, 146, 160
langage GRAFCET littéral (assembleur)	
PB April	49, 96, 136, 150, 151, 172, 174
langage GRAFCET littéral PL7-1	
Télé mécanique	96, 152, 175, 176
langage GRAFCET littéral PL7-3	
Télé mécanique	96, 130, 137, 138, 148, 149, 173, 176
langage graphique à contacts PL7-2	
Télé mécanique	53, 74, 75, 96, 129, 133, 141, 148, 149, 169, 170, 176, 185, 244, 247
langage graphique à contacts PL7-3	
Télé mécanique	53, 74, 96, 129, 148, 149, 169, 170, 176, 185
langage graphique à contacts Siemens	186
langage graphique à contacts SMC April	128, 132
langage informatique C200 Alspa	138, 139
langage littéral LEA Alspa	157
langage littéral SMC April	73, 128, 132
liaison orientées	13, 29, 38, 59, 103, 182, 189, 220, 241
logigramme	1, 126, 127, 153, 160, 190, 191, 225, 241

M

macro-étape	62, 68, 76, 77, 81, 82, 83, 85, 86, 87, 191, 192, 221, 238, 241
macro-étape mono-active PL7-3	
Télé mécanique	87
macro-étape multi-active PL7-3	
Télé mécanique	87
macro-représentation	76, 77, 80, 84, 85, 86, 221, 241, 246
master-relay	161, 162, 241
marche automatique	92, 207, 241
marche étape par étape	100, 241
marche manuelle	15, 72, 207, 208, 215, 216, 241
marquage d'une étape active	30
matérialisation des concepts	
du GRAFCET	116
matérialisation des règles d'évolution	117
mécaniser	3
mécanisme	4, 9, 14, 22, 24, 32, 202, 234, 241
mémoire à effacement prioritaire	111, 120, 241
mémoire à inscription prioritaire	119, 125, 126, 219, 235, 241
méthode boucle par boucle	207
méthode d'Huffman	1, 226, 242
méthode en cascade	1, 228, 229, 231
méthode matricielle	1, 226, 242
microprocesseur	5, 15, 72, 76, 234, 240, 242
minuterie	23, 51, 52, 54
mise en référence	5, 7, 9, 11, 14, 16, 19, 22, 238, 242
modes de marche et d'arrêt	17, 21, 24, 27, 62, 71, 187, 196, 202, 205, 207, 216, 234, 235, 239
mot	34, 53, 54, 61, 75, 121, 123, 125, 142, 143, 147, 148, 149, 155, 160, 174, 175, 178, 180, 186, 190, 191, 235, 236, 238, 241, 242, 245

N

niveau de description	2, 23, 24, 26, 31, 32, 33, 60, 62, 242
-----------------------	--

O

occurrence d'événements	47, 242
octet	235, 240, 242
OFB MSIT	177, 178, 179
OMEGA V2-B6 société 3ip	187 à 190
opérateur	4, 5, 7, 9, 14, 22, 27, 34, 39, 64, 93, 144, 235, 236, 237, 242, 243
opérateur logique combinatoire	1, 5, 19, 35, 120, 231, 241, 242
opérateur logique séquentiel	1, 5, 116, 120, 231, 242, 245
ordre	4, 9, 22, 31, 32, 60, 62, 172, 173, 191, 221, 222, 234, 235, 236
ordre de demande de calcul	62, 72, 235
ordre de figeage de situation	62, 69, 93, 94, 95, 223, 235
ordre de forçage de situation	62, 69, 88, 89, 90, 91, 95, 97, 98, 101, 162, 163, 170, 171, 172, 173, 176, 178, 180, 181, 223, 235, 239
ordre de forçage de situation conditionnel	92, 223, 235
ordre de lancement de macro-étape	62, 68, 235
ordre de lancement de sous-programme	62, 67, 76, 78, 84, 235
ordre de lancement de tâche	62, 76, 80, 83, 84, 235
ordre de sélection des modes de marche et d'arrêt	62, 71, 235
organigramme	1, 21, 61, 72, 74, 134, 136, 137, 139, 190, 224, 234, 241, 242
organiphase	1, 233
ORPHEE société April	97, 157 à 159, 160, 180

P

parallélisme interprété	59, 60, 87, 243, 245
parallélisme structural	55, 56, 59, 64, 65, 82, 85, 87, 149, 238, 243, 245
partie commande	4, 5, 12, 15, 22, 23, 24, 29, 34, 39, 240, 243, 244
partie opérative	4, 5, 14, 16, 22, 24, 206, 235, 243
partie relation	4, 5, 22, 25, 235, 243
PGAO	125, 160, 176, 182 à 194, 244
pilotage de distributeur	1, 14, 15, 19, 20, 31, 48, 64, 66, 120, 205, 216, 228, 229, 230, 236, 237, 238, 243, 244
pointeur de souris	237, 243, 245
point de vue réalisateur	23, 24, 25, 27, 51, 60, 216, 242
point de vue PC	23, 24, 25, 27, 202, 242
point de vue processus	23, 24, 25, 27, 196, 242
position de référence	5, 19, 39, 102, 208, 214, 218, 242, 243
préactionneur	4, 5, 14, 15, 19, 22, 24, 25, 26, 31, 32, 60, 62, 71, 93, 94, 202, 205, 235, 237, 242, 243
présélection d'un compteur	23, 34, 52, 53, 63, 236, 240, 243
présélection d'un temporisateur	23, 52, 53, 63, 240, 243
procédé	15, 21, 24, 27, 196, 210, 242, 243
procédure d'arrêt d'urgence	24, 71, 90, 98, 205, 206, 239, 243
processus	4, 14, 15, 21, 24, 102, 108, 123, 209, 243
programmateur cyclique Télé mécanique	141, 142, 143, 173
pupitre de commande	3, 4, 9, 15, 16, 17, 19, 22, 23, 25, 27, 34, 117, 202, 208, 215, 216, 237, 243

	Q	
QWERTY		235, 236, 243
	R	
réceptivité		2, 11, 13, 29, 33, 34, 35, 37, 38, 40, 42, 55, 59, 70, 79, 82, 87, 93, 94, 95, 100, 102, 108, 110, 111, 116, 120, 123, 125, 222, 244, 246
réceptivité toujours vraie		37, 81, 118, 213
rectangle-état du GEMMA		206, 207
registre à décalage		43, 44, 84, 85, 86, 244
registre à empilage		44, 244
registre à désempilage		45, 244
registre pas à pas		123, 141, 143, 160, 173, 243, 244
registre pas à pas du C100 Alspa		144, 173
règle d'évolution 1		39, 117, 125, 244
règle d'évolution 2		40, 41, 48, 117, 118, 119, 124, 147, 231, 238, 244
règle d'évolution 3		40, 41, 42, 47, 48, 117, 118, 124, 147, 231, 238, 244
règle d'évolution 4		40, 42, 44, 47, 48, 79, 118, 119, 124, 147, 231, 238, 244
règle d'évolution 5		40, 42, 44, 51, 64, 106, 119, 125, 126, 228, 241, 244
règles d'évolution		1, 2, 11, 12, 14, 26, 29, 30, 33, 39, 42, 44, 81, 91, 117, 121, 123, 150, 151, 244, 245
régulation		61, 72, 158, 177, 244
relais-maître		161, 162, 241
remise à un (RAU)		244
remise à zéro (RAZ)		39, 66, 96, 176, 244
reprise de séquence		58, 118, 244
réseaux de contacts		75, 244, 247
réseaux de Pétri		1, 102
reset		58, 64, 66, 67, 96, 131, 133, 138, 139, 140, 173, 176, 178, 189, 191, 234, 235, 245
	S	
saut d'étapes		57, 75, 84, 145, 236, 245
scrutation		49, 129, 130, 145, 162, 185, 235
sélecteur rotatif pas à pas		1
sélection alternative par drapeau		58, 202, 211, 212, 237
sélection de séquence		47, 57, 245
sensibilité		63, 111, 245
séquence		1, 3, 4, 10, 12, 55, 58, 75, 76, 83, 229, 230, 231, 237, 238, 245, 246
séquence unique		55, 59, 87, 245
séquences exclusives		57, 87, 245
séquences inclusives		59, 245
séquences simultanées		55, 56, 87, 245
séquenceur pneumatique		1, 2, 5, 25, 26, 48, 111, 117, 118, 120, 121, 231, 232, 233, 238, 245
set		58, 64, 66, 67, 96, 131, 133, 138, 140, 156, 173, 176, 178, 189, 191, 234, 235, 245
SFC (diagramme fonctionnel en séquence)		245
situation		2, 4, 12, 14, 23, 29, 30, 33, 34, 38, 40, 47, 56, 62, 63, 69, 70, 71, 108, 109, 110, 160, 177, 178, 237, 239, 241, 244, 245
situation courante		69, 70, 88, 89, 96, 158, 176, 193, 214, 238, 244, 245
situation forcée		69, 70, 89, 97, 154, 162, 163, 164, 165, 166, 170, 174, 176, 184, 192, 223, 235, 239, 247
situation initiale		7, 9, 12, 30, 39, 100, 102, 117, 130, 138, 158, 159, 175, 176, 244

situation vide		39, 43, 55, 63, 70, 83, 158, 176, 193, 245
sortie analogique		5, 22, 61, 235, 245
sortie numérique		5, 22, 61, 245
sortie tout ou rien		61, 235, 245
sous-programme		24, 62, 67, 68, 76, 77, 78, 79, 80, 83, 84, 85, 86, 155, 156, 189, 192, 199, 235, 241, 243, 246
spécifications fonctionnelles		23, 24, 27, 30, 246
spécifications opérationnelles		23, 25, 205, 246
spécifications technologiques		23, 24, 27, 246
structure algorithmique		123, 134 à 141, 160, 161, 170 à 173, 224
structure basée sur la fonction mémoire		123, 124 à 133, 160, 161, 165 à 170
structure exploitant le langage calcul sur mots		123, 147 à 149, 160, 161, 174
structure exploitant le langage GRAFCET graphique		123, 153 à 158, 160, 161, 176 à 181
structure exploitant le langage GRAFCET littéral		123, 150 à 152, 160, 161, 174 à 176
structure exploitant les registres pas à pas		123, 140 à 146, 160, 161, 173
structure informatique		123, 134 à 141, 160, 161, 170 à 173, 224
structures d'implantation du GRAFCET dans les API		123 à 181
	T	
tableau d'analyse		226, 246
tableau de Karnaugh		226, 227, 246
technoguides de l'ADEPA		15, 25, 246
temporisateur		26, 34, 39, 50, 51, 52, 53, 54, 63, 100, 154, 158, 189, 190, 235, 240, 242, 246
temporisateur analogique		51, 246
temporisateur numérique		52, 53, 246
traitement postérieur PL7-2 et PL7-3		
Télé mécanique		97, 152, 154, 176
traitement préliminaire PL7-1, PL7-2 et PL7-3		97, 152, 154, 175, 176
traitement séquentiel PL7-2 et PL7-3		
Télé mécanique		86, 97, 152, 153, 176
transition		1, 2, 3, 11, 12, 13, 14, 26, 29, 30, 33, 34, 35, 38, 39, 40, 42, 43, 44, 46, 47, 48, 55, 57, 69, 70, 82, 83, 88, 100, 102, 103, 108, 110, 116, 120, 151, 153, 154, 162, 163, 164, 171, 172, 189, 213, 222, 237, 239, 240, 241, 244, 245, 246, 247
transition puits		43, 44, 246
transition redondante		108, 111
transition source		43, 44, 117, 246
	U	
unicité de mode		71, 207, 210, 239, 247
	V	
validation d'une transition		11, 12, 34, 40, 42, 43, 55, 57, 70, 116, 150, 151, 247
variable de situation		97, 163, 164, 247
	X	
Xi		19, 34, 40, 50, 51, 52, 53, 63, 96, 152, 158, 176, 185, 186, 238, 244, 246, 247
	Y	
Yi		185, 186, 247
	Z	
zoom		83, 143, 154, 247

Documents constructeurs

- APRIL (API PB et SMC, logiciel ORPHEE)
- CEGELEC (API C200, C300, C350, C370 ALSPA, langage GRAFCET graphique P4, version V5)
- FAMIC-EURILOR (logiciel CADEPA)
- FESTO (API FPC400, FPC405, FPC440, langages AWL et FST)
- IRAI (logiciel AUTOMGEN, versions 6 et 7)
- SCHNEIDER TELEMÉCANIQUE (API TSX 17, 47, 67, langages PL7-1, PL7-2, PL7-3, PL7-3 V4 et V5)
- SIEMENS (API S5-90U, S5-95U, S5-100U, S5-155U, langage STEP 5, GRAPH 5, GRAPH Mini)
- UXP (Atelier d'automatique Alograf)

Livres et articles de revues

- 7 facettes du GRAFCET (CEPADUES)
- Automatisme appliqué (Bossy, Mérat, CASTEILLA)
- Automatique et informatique industrielle (Blin, Danic, Le Garrec, Séité, Trolez, CASTEILLA)
- Comprendre et maîtriser le GRAFCET (Blanchard, CEPADUES)
- Fiches produits-automates programmables industriels (CETIM)
- GRAFCET, de nouveaux concepts (GREPA, CEPADUES)
- GRAFCET et logique programmée (Thelliez, Toulotte)
- Le GRAFCET (Bossy, Brard, Faugère, Merlaud, CASTEILLA)
- Le GRAFCET (ADEPA/AFCET, CEPADUES)
- Les automatismes industriels (Sourisse, HERMES)
- Les automates programmables (Sourisse, HERMES)
- Les automates programmables industriels (CETIM)
- Méthodologie d'analyse descendante (Merlaud, revue ITET, n°s 248 à 251)
- Mémotech Électrotechnique (Bourgeois, Cogniel, CASTEILLA)
- Le GEMMA – Modes de marches et d'arrêt – GRAFCET de coordination des tâches – Conception des systèmes automatisés de production sûrs (S. Moreno, E. Peulot, CASTEILLA)

Organismes

- ADEPA (Agence nationale pour le Développement de la Production Automatisée) : fascicules GRAFCET, GEMMA et TECHNOGUIDES
- AFCET (Association Française des sciences et technologies de l'information et des systèmes)
- CETIM (Centre Technique des Industries Mécaniques) : dossiers API

- GREPA (Groupe Equipement de Production Automatisée) : nouveaux concepts du GRAFCET
- IEC ou CEI (International Electrotechnical Commission) : norme 848 (1988)

Normes françaises

- NF C 03-103 : schémas-symboles électriques (1980)
- NF C 03-106 : symboles graphiques pour schémas électriques : électronique (1967)
- NF C 03-108 : symboles graphiques pour schémas électriques : opérateurs logiques binaires (1970)
- NF C 03-190 + R1 : Diagramme fonctionnel « GRAFCET » pour la description des systèmes logiques de commande (1995)
- NF C 03-212 : Symboles graphiques pour schémas. Opérateurs logiques binaires. Partie 12
- NF C 63-850 : Automates programmables (1982)
- NF E 04-056 : Transmissions hydrauliques et pneumatiques : représentation symbolique fonctionnelle des appareils et accessoires (1970)
- NF E 04-057 : Transmissions hydrauliques et pneumatiques : guide pour l'exécution des schémas de circuits
- NF E 49-600 : Transmissions hydrauliques et pneumatiques : Symboles pour opérateurs logiques binaires et connexes
- NF E 49-602 : Transmissions hydrauliques et pneumatiques : Symboles pour opérateurs séquentiels et connexes
- NF EN 61131-3 : Automates programmables (langages de programmation) (1993)
- NF Z 67-010 : Symboles et conventions pour les organigrammes des données et les organigrammes de programmation (1975)
- PR EN 60848 : Langage de spécification pour diagramme fonctionnel en séquence (2000)
- UTE C 03-190 : Établissement des diagrammes fonctionnels pour systèmes de commande. Diagramme fonctionnel GRAFCET (1990)
- Document UTE C 03-191 : Établissement des diagrammes fonctionnels pour systèmes de commande – Diagramme fonctionnel GRAFCET – Extension des concepts de base (1993)

Site des auteurs

- <http://perso.wanadoo.fr/edmond.peulot>