

GFK-0854-F
New In Stock!
GE Fanuc Manuals

[http://www.pdfsupply.com/automation/ge-fanuc-manuals/programming-](http://www.pdfsupply.com/automation/ge-fanuc-manuals/programming-software/GFK-0854-F)
[software/GFK-0854-F](http://www.pdfsupply.com/automation/ge-fanuc-manuals/programming-software/GFK-0854-F)

programming-software

1-919-535-3180

GRA31ET-S31 pour API Serie 90 Manuel utilisateur

www.pdfsupply.com

Email: sales@pdfsupply.com

GFK-0854-F

New In Stock!

~~GE Fanuc Manuals~~

<http://www.pdfsupply.com/automation/ge-fanuc-manuals/programming-software/GFK-0854-F>

programming-software

1-919-535-3180

GRA31ET-S31 pour API Serie 90 Manuel utilisateur

www.pdfsupply.com

Email: sales@pdfsupply.com



GE Fanuc Automation

Automates Programmables Industriels

***GRAFSET – SFC
pour API Série 90***

Manuel utilisateur

GFK-0854-FR

Février 1994

Utilisation de l'expression "Attention danger" et des termes "Attention" et "Remarque" dans ce document

Attention danger

L'expression "Attention danger" est utilisée pour mettre en évidence des risques de blessures dues aux tensions, aux courants, aux températures ou à d'autres grandeurs physiques.

Toutes les situations où un manque d'attention peut être source de blessures physiques ou de dommages pour l'équipement sont repérées par cette expression.

Attention

Le terme "Attention" est associé aux situations où un manque d'attention risque de conduire à des dégâts matériels.

Remarque

Les "Remarques" ont pour but d'attirer votre attention sur des informations particulièrement utiles à la compréhension et à la mise en oeuvre de l'équipement.

Ce document est basé sur des informations disponibles au moment de sa publication. Malgré nos efforts de précision, nous ne pouvons prétendre couvrir tous les détails et toutes les variations matérielles ou logicielles possibles, ni aborder tous les cas de figure de l'installation, du fonctionnement ou de la maintenance. Les caractéristiques décrites dans ce document peuvent être absentes de certains systèmes matériels ou logiciels. GE Fanuc Automation ne s'engage pas à avertir les possesseurs de ce document d'éventuelles modifications ultérieures.

GE Fanuc Automation ne fournit aucune garantie explicite, implicite ou statutaire, et décline toute responsabilité quant à la précision, à l'utilité, et au caractère complet ou suffisant des informations contenues dans ce document. GE Fanuc Automation ne donne aucune garantie de qualité marchande et d'aptitude à une utilisation donnée.

Les marques suivantes sont des marques déposées de GE Fanuc Automation North America, Inc. :

Alarm Master	CIMSTAR	Helpmate	PROMACRO	Series Six
CIMPLICITY	GENet	Logicmaster	Series One	Series 90
CIMPLICITY 90-ADS	Genius	Modelmaster	Series Three	VuMaster
CIMPLICITY PowerTRAC	Genius PowerTRAC	ProLoop	Series Five	Workmaster

Copyright 1989–1994 GE Fanuc Automation North America, Inc. Tous droits réservés

Le logiciel de programmation Logimaster™ 90 est utilisé pour configurer et programmer l'automate Série 90™. Ce manuel décrit comment utiliser la version GRAFCET – SFC (Graphe Commande Etape-Transition) du logiciel Logimaster pour programmer et créer des programmes utilisateur pour votre automate.

1. CONTENU DE CE MANUEL

Ce manuel est complémentaire du document *GFK-0263 Logimaster 90-70 Programming Software User's Manual* et du document *GFK-0265 Series 90-70 Programmable Controller Reference Manual*. Vous vous référerez à ces manuels de Logimaster 90-70 pour vous familiariser avec les caractéristiques du logiciel de programmation Logimaster 90-70. Le document *GFK-0854 Manuel utilisateur GRAFCET – SFC pour API Série 90*, ne contient que les informations propres à la version GRAFCET (SFC) du logiciel Logimaster 90-70 et ne reprend pas les informations déjà présentées dans les manuels *GFK-0263 Logimaster 90-70 Programming Software User's Manual*, qui sont communes aux versions GRAFCET et langage à relais du logiciel Logimaster 90-70.

Ce document est la première version du document *GFK-0854 Manuel utilisateur GRAFCET – SFC pour API Série 90*. Il contient les chapitres et annexes suivants :

Chapitre 1. Introduction : présente le langage Graphe commande étape-transition et décrit les éléments qui vous seront nécessaires pour créer un programme utilisateur.

Chapitre 2. Comprendre le langage GRAFCET : fournit un aperçu du langage GRAFCET (Graphe commande étape-transition).

Chapitre 3. L'éditeur GRAFCET : décrit les blocs de programme GRAFCET, comment insérer/éditer un réseau GRAFCET, et les fonctions disponibles dans l'éditeur GRAFCET.

Chapitre 4. Impression des blocs GRAFCET : décrit une sortie imprimée type d'un réseau GRAFCET en fournissant des exemples pour chaque section de la sortie imprimée.

Annexe A. Erreurs utilisateur courantes : décrit les erreurs utilisateur courantes.

2. AUTRES MANUELS À CONSULTER

GFK-0262 Series 90™-70 Programmable Controller Installation Manual

GFK-0263 Logimaster 90-70 Programming Software User's Manual

GFK-0265 Series 90-70 Programmable Controller Reference Manual

GFK-0356 Série 90™-30 Automate Programmable – Manuel d'installation

GFK-0466 Logimaster™ 90-30/20/Micro Programming Software User's Manual

3. VOS REMARQUES ET SUGGESTIONS SONT LES BIENVENUES

GE Fanuc Automation s'efforce d'éditer des documentations techniques de qualité. Après avoir utilisé ce manuel, merci de consacrer quelques instants à la page suivante, "Page de remarques", pour la compléter et nous la renvoyer.

GFK-0854-F

**Manuel utilisateur GRAFCET – SFC
pour API Série 90**

Cochez votre fonction principale SVP

- | | |
|----------------------------------------------|--------------------------------------------------------|
| <input type="checkbox"/> Concepteur système | <input type="checkbox"/> Programmeur |
| <input type="checkbox"/> Distributeur | <input type="checkbox"/> Responsable de maintenance |
| <input type="checkbox"/> Intégrateur système | <input type="checkbox"/> Opérateur |
| <input type="checkbox"/> Installateur | <input type="checkbox"/> Autre (à préciser ci-dessous) |

Si vous désirez une réponse personnelle, indiquez votre adresse postale complète :

SOCIETE : NOM :

ADRESSE :

..... PAYS :

Remettez cet imprimé directement à votre correspondant GE Fanuc ou envoyez-le à :

**GE Fanuc Automation France
45, rue du Bois Chaland
CE 2904 – Lisses
91029 EVRY Cedex**

Toutes vos remarques seront étudiées par du personnel qualifié.

REMARQUES

Si besoin, utilisez le verso de cette page.

CHAPITRE 1 – INTRODUCTION

1.	CE QU'IL VOUS FAUT	1-1
2.	VERSION MS-DOS	1-1
3.	INSTALLATION DU LOGICIEL	1-1
4.	ECRANS D'AIDE	1-2
5.	FONCTIONS DES TOUCHES	1-2

CHAPITRE 2 – COMPRENDRE LE LANGAGE GRAFCET

1.	ETAPES	2-4
2.	INSTRUCTIONS D' ACTIONS D'UNE ÉTAPE	2-5
3.	TRANSITIONS	2-6
4.	CYCLE DE SCRUTATION DE L'API SÉRIE 90-70	2-7
4.1.	Preprocessing	2-7
4.2.	Traitement séquentiel	2-8
4.3.	Postprocessing	2-8
5.	EVOLUTION DU GRAFCET	2-9
6.	EXEMPLES DE GRAPHES COMMANDE ÉTAPE-TRANSITION	2-11
7.	STRUCTURES DE CONTRÔLE DE BASE	2-12
7.1.	Séquence simple	2-12
7.2.	Divergence en OU d'une séquence	2-12
7.3.	Convergence en OU d'une séquence	2-13
7.4.	Divergence en ET d'une séquence	2-13
7.5.	Convergence en ET d'une séquence	2-14
8.	SAUTS D'ÉTAPE SOURCE ET DESTINATION	2-15
9.	EXEMPLES DE STRUCTURES DE CONTRÔLE DE BASE	2-16
10.	RÈGLES APPLICABLES AUX STRUCTURES DE CONTRÔLE DE BASE	2-18

Sommaire

CHAPITRE 3 – L'ÉDITEUR GRAFCET (GRAPHE COMMANDE ETAPE–TRANSITION)

1.	BLOCS DE PROGRAMME GRAFCET	3–2
1.1.	Temporisateurs d'étape	3–3
1.2.	Bits de défaut d'étape	3–3
1.3.	Bits d'étape	3–3
1.4.	Format d'un bloc GRAFCET	3–4
1.5.	Instructions de preprocessing et de postprocessing	3–5
1.6.	Programmation des instructions d'actions d'étape GRAFCET	3–6
1.7.	Sélection du langage du bloc	3–7
1.8.	Changer le mode d'affichage	3–9
2.	INSERTION/ÉDITION D'UNE TOPOLOGIE DE RÉSEAU GRAFCET	3–10
2.1.	Organisation des grilles	3–14
2.2.	Déplacement du curseur	3–17
2.3.	Attribution de noms aux étapes, transitions et sauts d'étape	3–17
2.4.	Insertion/édition d'un réseau GRAFCET (exemple)	3–18
2.5.	Suppression d'un élément	3–33
2.6.	Fonctions d'ouverture et de suppression d'espace	3–34
2.6.1.	Ouvrir la rangée	3–34
2.6.2.	Ouvrir la colonne	3–36
2.6.3.	Effacer la rangée	3–38
2.6.4.	Effacer la colonne	3–40
2.7.	Sortie du mode Insérer un programme ou Editer	3–42
2.8.	Affichage des erreurs dans le réseau	3–42
2.9.	Enregistrement des modifications apportées à un bloc GRAFCET	3–43
3.	FONCTION DE RECHERCHE	3–44
3.1.	Validation de la fonction de recherche	3–45
3.2.	La fonction Vers	3–49
3.3.	Fin de la recherche	3–49
4.	COMMENTAIRES	3–50
5.	OPTIONS ASSOCIÉES AUX BLOCS GRAFCET	3–51
5.1.	Journalisation des défauts pour des durées d'étape hors limites	3–51
5.2.	Configuration de plusieurs langages	3–53
6.	FONCTIONS DE MISE AU POINT	3–55
6.1.	SFC_RESET	3–56
6.2.	Changement d'état et forçages	3–57
6.3.	Historique d'évolution	3–58
6.4.	Vers..	3–59
6.5.	Réglage de la base de temps et des limites de durée d'étape pour un bloc GRAFCET	3–60
6.5.1.	Réglage de la base de temps	3–61
6.5.2.	Réglages des limites de durée d'étape	3–61
6.6.	Mise au point en mode mono-cycle	3–62
7.	FONCTIONS UTILITAIRES	3–63

7.1.	Transfert de l'API vers la console de programmation	3-63
7.2.	Enregistrement dans l'API de la console de programmation	3-63
7.2.1.	Transfert en mode Stop	3-63
7.2.2.	Transfert en mode Run	3-64
7.3.	Vérification d'un programme avec l'API	3-65

CHAPITRE 4 – IMPRESSION DES BLOCS GRAFCET

1.	EXEMPLE DE PAGE POUR UN RÉSEAU GRAFCET	4-3
2.	EXEMPLE DE PAGE POUR DES INSTRUCTIONS DE PREPROCESSING/POSTPROCESSING	4-4
3.	EXEMPLE DE PAGE D'INSTRUCTIONS D'ÉTAPE	4-5
4.	EXEMPLE DE PAGE D'INSTRUCTIONS DE TRANSITION	4-6
5.	EXEMPLE DE PAGE DE RÉFÉRENCES CROISÉES DE BLOC	4-7
6.	EXEMPLE DE PAGE DE RÉFÉRENCES CROISÉES GLOBALES	4-8
7.	EXEMPLE DE PAGE DE TABLE DES MATIÈRES	4-10

ANNEXE A – ERREURS UTILISATEUR COURANTES

1.	ERREURS GÉNÉRALES	A-1
2.	ERREURS D'INSTRUCTIONS DE TRANSITION	A-2
3.	ERREURS D'INSTRUCTIONS DE TRANSITION, D'ACTION D'ÉTAPE, ET DE PREPROCESSING/POSTPROCESSING	A-2
4.	ERREURS D'ÉLÉMENT GRAFCET GÉNÉRALES	A-2
5.	ERREURS GRAFCET DE PREMIER NIVEAU	A-3

Sommaire

Page laissée blanche intentionnellement

Tableaux

	Page
Tableau 2-1. Séquence d'événements pour les quatre premières scrutations d'un segment	2-10
Tableau 3-1. Modes d'affichage	3-9
Tableau 3-2. Mnémoniques GRAFCET	3-20

Figures

	Page
Figure 2-1. Cycle de scrutation d'API Série 90-70	2-7

Chapitre 1

Introduction

Logicmaster 90–70 est utilisé pour configurer et programmer l’automate Série 90–70. Dans la plupart des cas, vous écrivez les programmes utilisateur de l’API en langage à relais. Désormais, vous pouvez également choisir d’écrire votre programme en langage GRAFCET – SFC (Graphe Commande Etape–Transition). Ce manuel explique comment utiliser le langage GRAFCET avec le logiciel Logicmaster 90–70 pour programmer votre automate.

La description des caractéristiques et des fonctions du logiciel Logicmaster 90–70 sort du cadre de ce manuel. Pour toute information sur les caractéristiques et les fonctions du logiciel, veuillez vous reporter au document *GFK–0263 Logicmaster 90–70 Programming Software User’s Manual*. Pour toutes informations sur les instructions de programmation et les informations de temporisation, référez–vous au document *GFK–0265 Series 90–70 Programmable Controller Reference Manual*.

1. CE QU’IL VOUS FAUT

Le document *GFK–0263 Logicmaster 90–70 Programming Software User’s Manual*, présente le matériel et le logiciel nécessaires pour exécuter les progiciels Logicmaster 90–70. Revoyez cette liste dans le chapitre 1, *Introduction*, pour vous assurer de la compatibilité de votre système avant d’essayer de charger le logiciel ou de démarrer votre ordinateur.

2. VERSION MS–DOS

Pour exécuter le logiciel Logicmaster, MS–DOS Version 5.0 ou ultérieure doit être installé sur votre ordinateur.

Logicmaster fournit le support des claviers internationaux, selon la configuration de MS–DOS résidant sur l’ordinateur hôte. Consultez votre manuel utilisateur MS–DOS pour toutes informations de configuration.

3. INSTALLATION DU LOGICIEL

Le chapitre 2, *Utilisation*, du document *GFK–0263 Logicmaster 90–70 Programming Software User’s Manual*, décrit comment installer et lancer le logiciel de programmation Logicmaster. Avant de pouvoir commencer à utiliser les fonctionnalités et fonctions décrites dans ce manuel, vous devez d’abord installer le logiciel. Si vous n’avez pas encore installé le logiciel Logicmaster, faites–le en utilisant la procédure d’installation décrite dans le chapitre 2 du document *GFK–0263 Logicmaster 90–70 Programming Software User’s Manual*.

4. ECRANS D'AIDE

Logicmaster inclut des écrans d'aide détaillés. Ces écrans d'aide sont chargés sur le disque dur de votre console de programmation pendant la procédure d'installation du logiciel et sont constamment accessibles. Appuyez sur ALT-H pour accéder aux écrans d'aide.

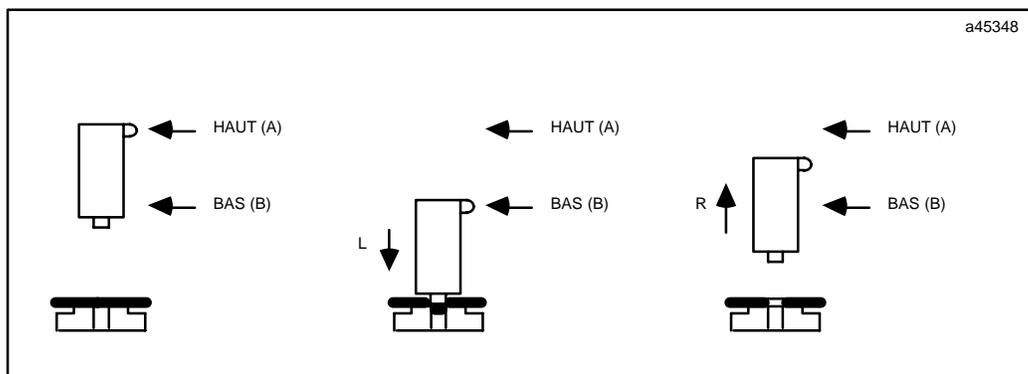
5. FONCTIONS DES TOUCHES

L'annexe E, *Fonctions des touches*, du document *GFK-0263 Logicmaster 90-70 Programming Software User's Manual*, liste les fonctions de clavier actives dans l'environnement logiciel Logicmaster.

Chapitre 2

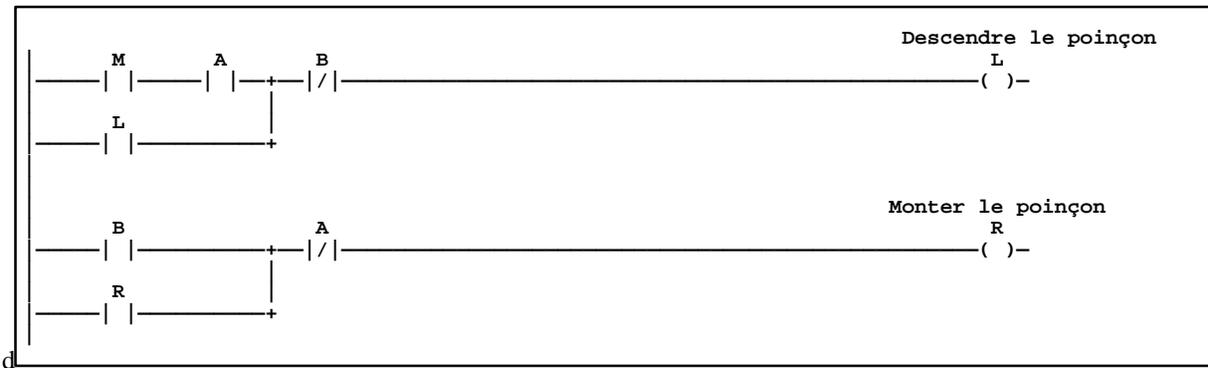
Comprendre le langage GRAFCET

La programmation en langage à relais traditionnel (RLD) a été largement utilisée par les ingénieurs électriciens pour concevoir et représenter des systèmes de contrôle logiques depuis l'invention du relais. Pour des systèmes de contrôle à logique combinatoire, où des sorties ou actions dépendent directement des états des entrées ou conditions, le langage RLD est excellent. Cependant, pour certains problèmes de contrôle, lorsque des actions de contrôle sont séquentielles ou dépendent du temps, des instructions de programme exclusivement en langage à relais peuvent devenir un peu lourdes. Prenons l'exemple simplifié d'un poinçon semi-automatique.



Dans cet exemple, un poinçon semi-automatique est initialement en position haute (A). Lorsque l'opérateur actionne le bouton poussoir Start (M), le poinçon descend, perce la plaque de métal à la position (B), et termine le cycle en retournant en position haute.

Dans un programme en langage à relais, vous ne pouvez vous contenter de prendre en compte le fait que l'action sur le bouton poussoir Start (M) déclenche le mouvement du poinçon. Vous devez aussi considérer que l'opérateur relâchera le bouton poussoir et qu'il peut réappuyer sur le bouton poussoir avant ou après que le poinçon soit en position de repos. Ces informations supplémentaires n'ont aucun effet sur le déplacement du poinçon et compliquent la description de la fonction.



Nous avons besoin d'une méthode qui nous permette de représenter des problèmes de contrôle séquentiel de manière séquentielle, montrant les diverses actions à mettre en oeuvre au cours d'une étape et indiquant les conditions qui doivent être satisfaites avant de passer à l'étape suivante.

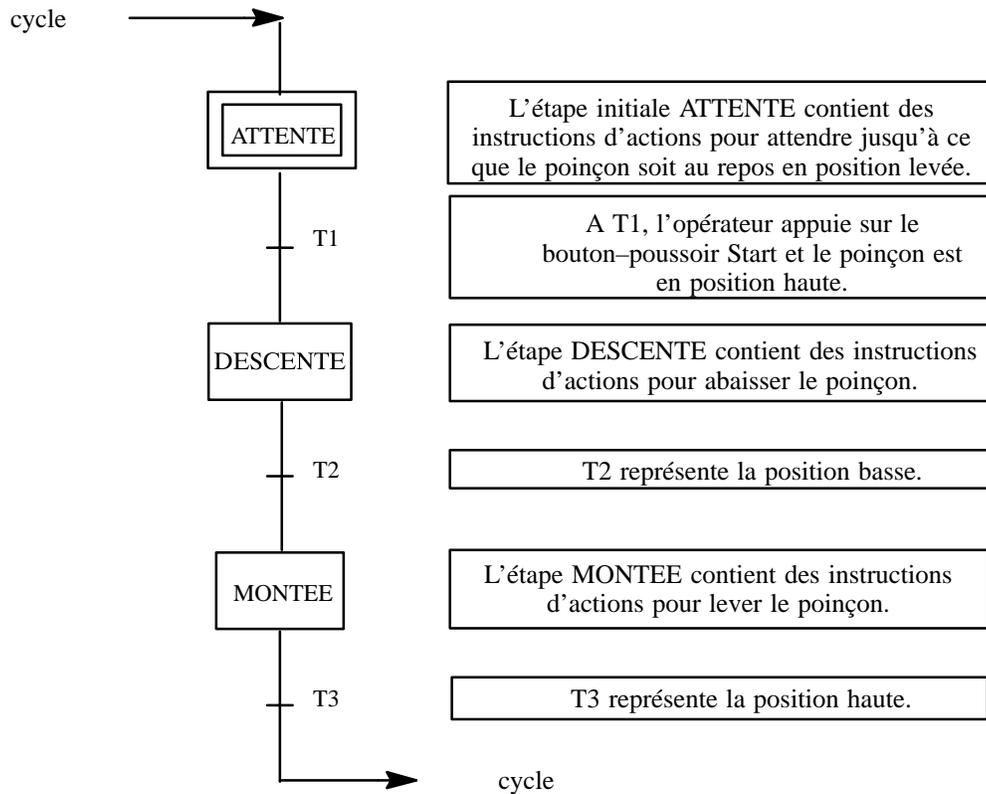
Le langage GRAFCET (Graphe Commande Etape-Transition) est une méthode spécialement développée pour décrire des systèmes de contrôle séquentiel industriels. GRAFCET est une méthode graphique qui représente les fonctions d'un système automatisé séquentiel sous forme d'une séquence d'étapes et de transitions. Chaque étape représente des commandes ou des actions qui peuvent être actives ou inactives. Le flux logique passe d'une étape à la suivante via une transition conditionnelle qui peut être vraie (1) ou fausse (0). Si la condition de transition est vraie (1), ce qui est indiqué en positionnant la variable de transition, le contrôle passe de la première étape (étape initiale), qui devient inactive, à l'étape suivante, qui devient alors active.

Chaque fonction de contrôle peut par conséquent être représentée par un groupe d'étapes et de transitions, appelé réseau GRAFCET (Graphe Commande Etape-Transition). Ces diagrammes peuvent alors être interconnectés par des liens dirigés, montrant le flux logique pour former un GRAFCET complet.

Remarque

Le terme SFC est utilisé en lieu et place du terme GRAFCET pour tous les affichages écran.

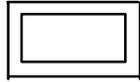
Reprenons l'exemple du poinçon semi-automatique de la page précédente. Il peut être représenté comme un GRAFCET, ainsi qu'illustré ci-dessous.



Dans l'exemple ci-dessus, l'étape initiale (ATTENTE) est représentée par un carré tracé avec des lignes doubles. Il contient les instructions d'actions exécutées initialement. Une étape est représentée par un carré qui peut également contenir un nom (par exemple, DESCENTE ou MONTEE) représentant la fonction de principe à cette étape. Les instructions d'actions associées à chaque étape ainsi qu'à chaque transition (T1, T2 et T3) peuvent être programmées en langage à relais en "zoomant" sur l'étape ou la transition correspondante.

1. ETAPES

Vous pouvez utiliser deux types d'étapes dans un graphe commande étape-transition : des étapes initiales et des étapes normales.



Etape initiale



Etape normale

Une **étape normale** est une étape qui doit être programmée en utilisant des instructions d'actions écrites en langage à relais. L'**étape initiale** est un type spécial d'étape normale. Il ne peut exister qu'une et une seule étape initiale par réseau GRAFCET.

L'étape initiale est la première étape exécutée après la réinitialisation du GRAFCET. (Pour plus d'informations sur l'utilisation de la fonction SFC_RESET pour réinitialiser le GRAFCET à son étape initiale, reportez-vous à la page 3-56.) L'étape initiale ne peut apparaître à l'intérieur d'une divergence en ET, mais peut apparaître n'importe où ailleurs.

Les étapes initiales sont construites avec des lignes horizontales doubles et des lignes latérales verticales doubles. Les étapes normales utilisent des lignes horizontales simples et des lignes latérales verticales simples.

2. INSTRUCTIONS D' ACTIONS D'UNE ÉTAPE

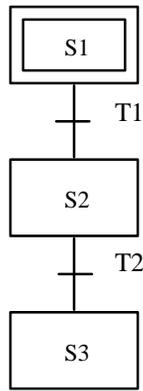
Chaque étape peut comporter des instructions d'actions composées de zéro segment ou plus programmés en langage à relais (RLD).

Remarque

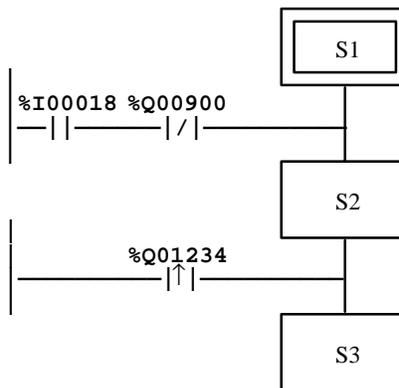
Lorsqu'une étape est désactivée, ses instructions d'actions sont exécutées une dernière fois, sans flux validant. Cela configure les bobines à leur état par défaut (une bobine normalement ouverte est hors-service par défaut, tandis qu'une bobine normalement fermée est en-service par défaut).

3. TRANSITIONS

Les instructions de transition sont programmées en langage à relais (RLD). Au minimum, chaque transition doit contenir un segment terminé par une bobine qui positionne sa variable de transition (c'est-à-dire, le nom de la transition, tel que T1).



L'exemple suivant montre comment les conditions de transition remplacent conceptuellement leurs transitions associées (T1 et T2) dans le graphe commande étape-transition affiché ci-dessus.



4. CYCLE DE SCRUTATION DE L'API SÉRIE 90-70

Le cycle de scrutation de l'API Série 90-70 (cycle) s'exécute continuellement dans l'ordre de preprocessing, traitement séquentiel et de postprocessing pour tous les blocs GRAFCET appelés et le bloc GRAFCET _MAIN lorsque l'API est en mode **RUN**.

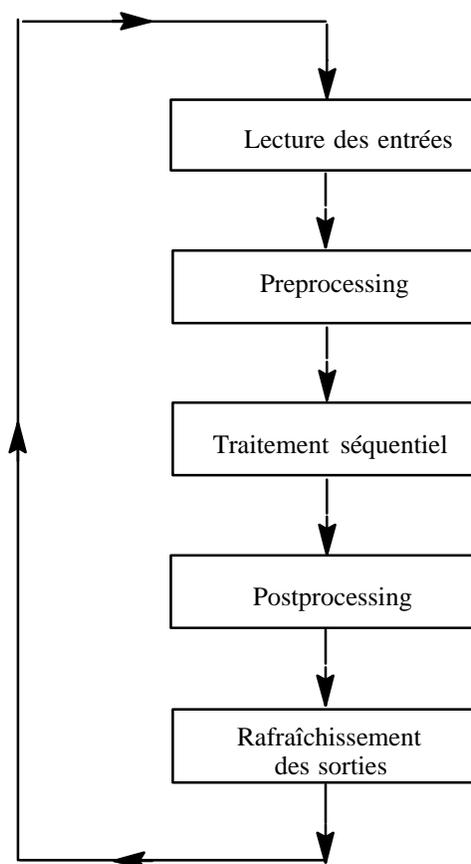


Figure 2-1. Cycle de scrutation d'API Série 90-70

4.1. PREPROCESSING

Cette section est traitée au début de la scrutation. Normalement, des instructions de preprocessing en langage à relais sont utilisées pour traiter, au début du cycle de scrutation, des événements qui pourraient affecter la section de traitement séquentiel du programme. Ces événements peuvent inclure :

- L'initialisation.
- Des commandes opérateur.
- La préconfiguration du GRAFCET à un état spécifique.

4.2. TRAITEMENT SÉQUENTIEL

Cette partie de la scrutation d'API consiste à faire évoluer le GRAFCET à son état suivant et à traiter les instructions d'actions de toutes les étapes devenues actives. L'API ne scrute que les instructions associées à des étapes actives et des transitions, ce qui réduit sensiblement le temps de scrutation. Pour plus d'informations, veuillez-vous reporter au paragraphe "Evolution du GRAFCET" ci-après.

4.3. POSTPROCESSING

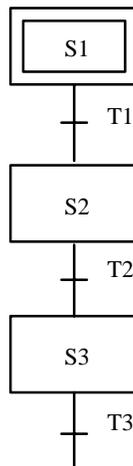
Cette section est traitée lorsqu'une évolution du GRAFCET s'est terminée. Elle peut contenir des instructions en langage à relais pour traiter des systèmes de verrouillage de sécurité, etc.

5. EVOLUTION DU GRAFCET

Une étape peut être active ou inactive. Une transition est active (évaluée uniquement) lorsque l'étape qui la précède immédiatement est active. Si une transition est évaluée à l'état vrai (ON), alors les étapes immédiatement précédentes sont rendues inactives et les étapes immédiatement postérieures sont rendues actives. Ce processus est "l'évolution du graphe commande étape-transition." La séquence d'opérations pour toute partie de traitement séquentiel donné de la scrutation de l'API est :

1. Evaluer les transitions qui suivent toutes les étapes actives.
2. Faire évoluer le graphe commande étape-transition. L'évolution de celui-ci s'effectue de manière descendante.
3. Exécuter les instructions d'actions associées aux étapes actives.

L'exemple simplifié de segment de graphe commande étape-transition ci-dessous montre un API exécutant les quatre premières scrutations d'un segment.



Lors de l'exécution initiale d'un bloc GRAFCET, le GRAFCET est considéré à l'état de réinitialisation. Puis l'étape initiale (S1) devient active, puis la transition T1. Les instructions d'actions associées à l'étape S1 sont alors exécutées.

A la deuxième scrutation, le système teste les instructions d'actions associées à la transition active T1 et les trouve à l'état vrai. L'étape S1 devient inactive ainsi que la transition T1. L'étape S2 devient active ainsi que la transition T2. Les instructions d'actions associées à l'étape S2 sont alors exécutées.

A la troisième scrutation, le système teste les instructions d'actions associées à la transition active T2 et les trouve à l'état faux. Comme la condition de transition n'est pas satisfaite, l'étape S2 reste active, de même que la transition T2. Les instructions d'actions associées à l'étape S2 sont de nouveau exécutées.

A la quatrième scrutation, le système teste de nouveau les instructions de transition associées à la transition active T2 et les trouve cette fois-ci à l'état vrai. La transition T2 devient donc inactive, ainsi que l'étape S2. L'étape S3 devient active, ainsi que la transition T3. Les instructions d'actions associées à l'étape S3 sont alors exécutées.

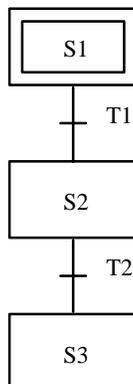
Le tableau ci-dessous résume cette séquence d'événements :

Tableau 2-1. Séquence d'événements pour les quatre premières scrutations d'un segment

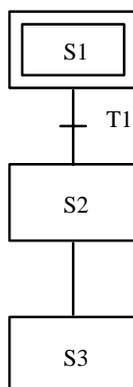
Scrutation	Etape S1	Transition T1	Etape S2	Transition T2	Etape S3	Transition T3
1	Active	Active	Inactive	Inactive	Inactive	Inactive
2	Inactive	Inactive	Active	Active	Inactive	Inactive
3	Inactive	Inactive	Active	Active	Inactive	Inactive
4	Inactive	Inactive	Inactive	Inactive	Active	Active

6. EXEMPLES DE GRAPHES COMMANDE ÉTAPE–TRANSITION

L'exemple suivant montre une partie d'un graphe commande étape–transition correctement structuré. Il comporte une étape initiale (S1), deux étapes normales (S2 et S3), et deux transitions (T1 et T2).



L'exemple suivant montre une partie d'un graphe commande étape–transition incorrectement structuré. Il comprend deux étapes (S2 et S3), sans transition entre ces deux étapes.

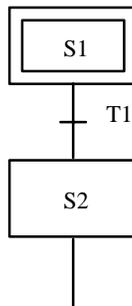


7. STRUCTURES DE CONTRÔLE DE BASE

Les exemples suivants illustrent les cinq structures de contrôle de base qui peuvent être utilisées dans un graphe commande étape-transition.

7.1. SÉQUENCE SIMPLE

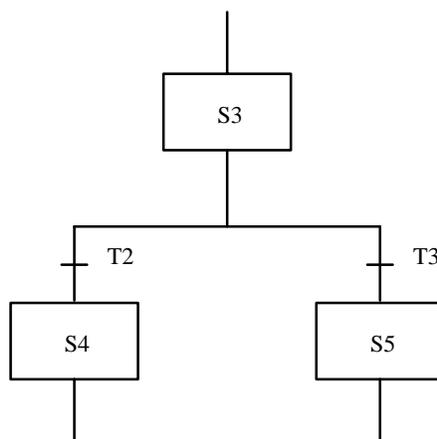
Dans une séquence simple, le contrôle passe de l'étape S1 à l'étape S2 **seulement** si l'étape S1 est active et si la transition T1 est évaluée à vrai.



7.2. DIVERGENCE EN OU D'UNE SÉQUENCE

Une divergence en OU doit être précédée par une étape. Le premier élément après une divergence en OU doit être une transition.

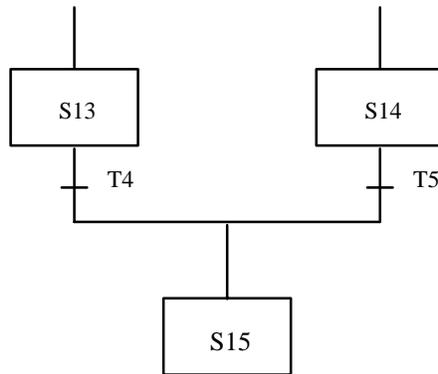
Dans une séquence de divergence en OU, le contrôle passe de l'étape S3 à S4 **seulement si** l'étape S3 est active et si la transition T2 est évaluée à vrai. Le contrôle passe de l'étape S3 à l'étape S5 **seulement si** la transition T2 n'est pas vraie, l'étape S3 est active et la transition T3 est vraie (priorité de transition de gauche à droite). Le système sélectionne exactement une branche de la divergence en OU.



7.3. CONVERGENCE EN OU D'UNE SÉQUENCE

Une branche convergente en OU ne peut être précédée que par des transitions. Elle doit être suivie d'une étape.

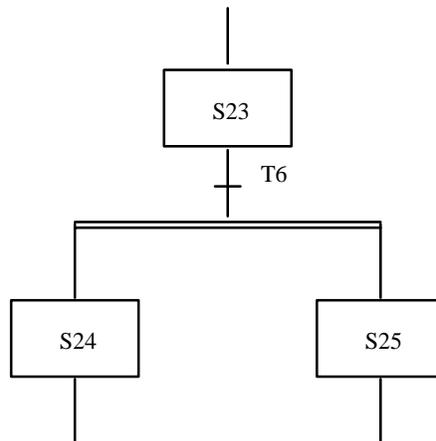
Dans une séquence de convergence en OU, le contrôle passe de l'étape S13 à l'étape S15 **seulement si** l'étape S13 est active et si la transition T4 est évaluée à vrai. Le contrôle passe de l'étape S14 à l'étape S15 **seulement si** l'étape S14 est active et la transition T5 est vraie.



7.4. DIVERGENCE EN ET D'UNE SÉQUENCE

Une branche divergente en ET doit être précédée d'une transition. Une étape doit être le premier élément d'une branche divergente en ET.

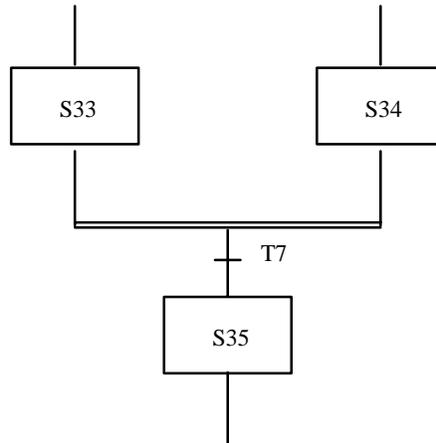
Dans une séquence divergente en ET, le contrôle passe de l'étape S23 à l'étape S24 et à l'étape S25 **seulement si** l'étape S23 est active et la transition T6 est évaluée à vrai. Les deux étapes passeront à l'état actif, bien que les instructions d'actions contenues dans une étape soient exécutées avant les instructions d'actions contenues dans l'autre étape. L'ordre d'exécution des étapes est indéfini.



7.5. CONVERGENCE EN ET D'UNE SÉQUENCE

Une convergence en ET ne peut être précédée que par des éléments d'étape. Elle doit être suivie par une transition.

Dans une convergence en ET, le contrôle passe de l'étape S33 et de l'étape S34 à l'étape S35 **seulement si** les étapes S33 et S34 sont toutes deux actives et que la transition T7 est évaluée à vrai.



Les instructions de transition pour T7 ne sont exécutées que lorsque toutes les étapes à la fin de la convergence en ET sont actives.

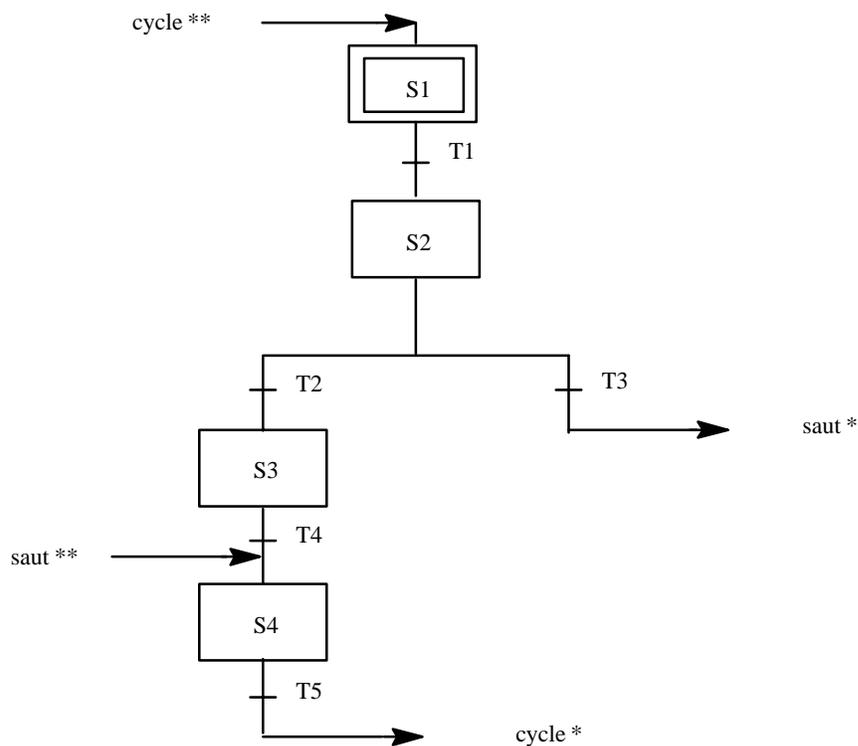
8. SAUTS D'ÉTAPE SOURCE ET DESTINATION

Les sauts d'étape source et destination sont utilisés pour créer des sauts vers l'avant et vers l'arrière dans un graphe commande étape-transition. Les mots *saut* et *cycle* dénotent des sauts d'étape dans le graphe commande étape-transition illustré ci-dessous. Les sauts vers l'arrière sont appelés des *cycles*.

Dans la séquence de saut vers l'avant illustrée ci-dessous, le contrôle passe de l'étape S2 à l'étape S4 seulement si l'étape S2 est active, la transition T2 est évaluée à faux, et la transition T3 est évaluée à vrai.

Dans la séquence de saut vers l'arrière (ou cycle) le contrôle passe de l'étape S4 à l'étape S1 seulement si l'étape S4 est active et la transition T5 est évaluée à vrai.

Des sauts d'étape source et destination ne peuvent pas être utilisés avant une transition. Des sauts d'étape source doivent être utilisés immédiatement après la transition. Des sauts d'étape destination doivent être utilisés immédiatement avant une étape. Les sauts d'étape ne peuvent être utilisés de façon adjacente à un branchement. Veuillez-vous référer à l'annexe A, *Erreurs utilisateur courantes*, pour d'autres restrictions.

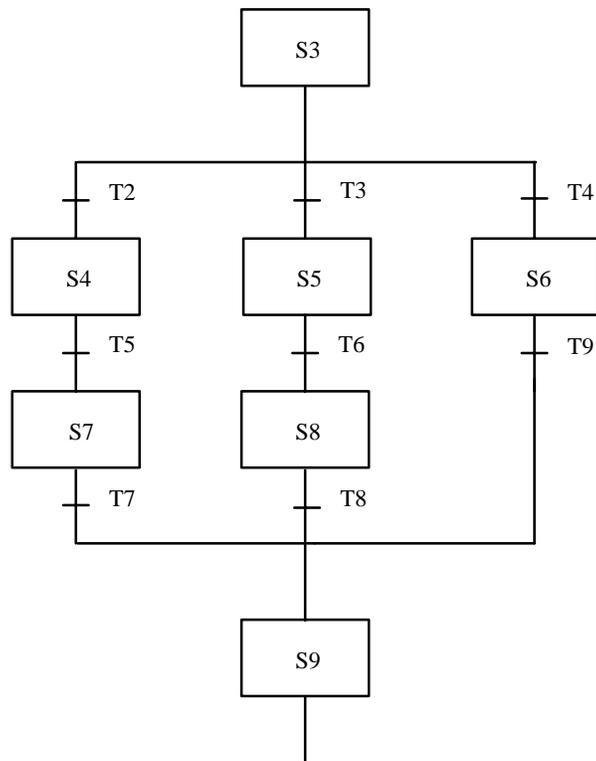


* Dénote un saut d'étape source.

** Dénote un saut d'étape destination.

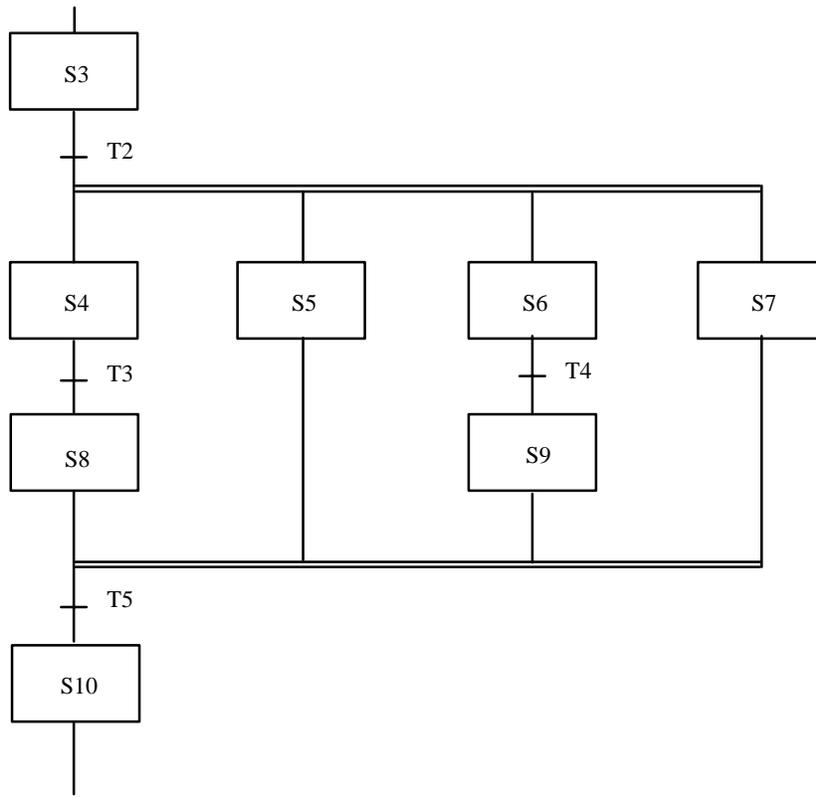
9. EXEMPLES DE STRUCTURES DE CONTRÔLE DE BASE

L'exemple suivant montre un segment de graphe commande étape-transition avec une divergence en OU.



Lorsque l'étape S3 a été active pendant une évolution, le système évalue les transitions nouvellement validées T2, T3 et T4. La première transition validée évaluée à vrai détermine la branche de la divergence en OU sélectionnée. Si plus d'une transition est évaluée à vrai au même moment, le système sélectionne la branche la plus à gauche dont la condition de transition est vraie.

L'exemple suivant montre un segment de graphe commande étape-transition avec une séquence de divergence en ET.



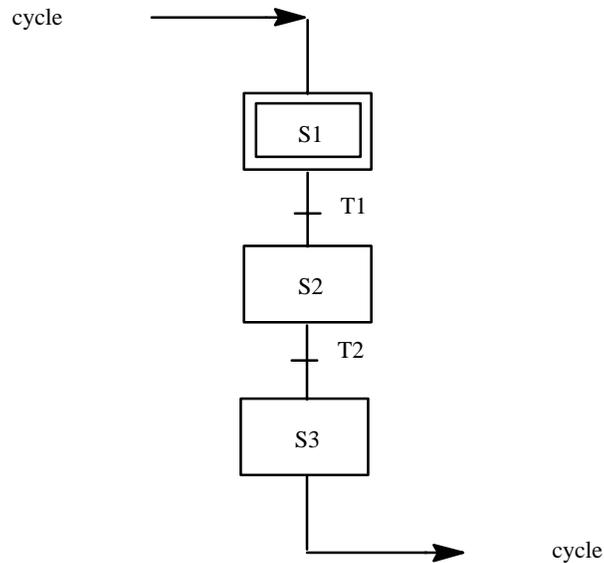
Lorsque l'étape S3 a été active pendant une évolution, le système évalue la transition T2 nouvellement validée. Lorsque la transition T2 est évaluée vraie, l'étape S3 devient inactive, et les étapes S4, S5, S6 et S7 deviennent toutes actives. (Toutes les branches de la divergence en ET sont exécutées en parallèle). La transition T2 est désactivée, et les transitions T3 et T4 sont validées. Lorsque les étapes S8, S5, S9 et S7 sont toutes actives et que la transition T5 est évaluée à vrai, toutes les branches de la divergence en ET sont terminées, et l'étape S10 devient active.

10. RÈGLES APPLICABLES AUX STRUCTURES DE CONTRÔLE DE BASE

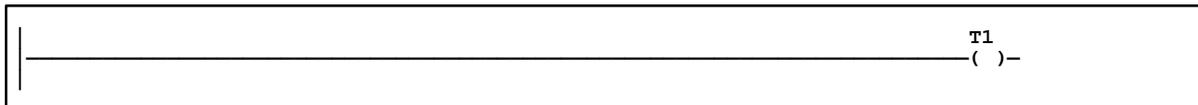
Lorsque vous construisez un graphe commande étape-transition, vous devez respecter les règles suivantes.

1. Une seule étape initiale peut être utilisée dans le graphe GRAFCET et elle ne peut pas être utilisée à l'intérieur d'une structure simultanée.
2. Deux étapes dans une séquence doivent toujours être séparées par une transition.

Dans le segment de graphe commande étape-transition **non valide** ci-dessous, un saut vers l'arrière (ou cycle) connecte directement deux étapes sans transition intermédiaire.

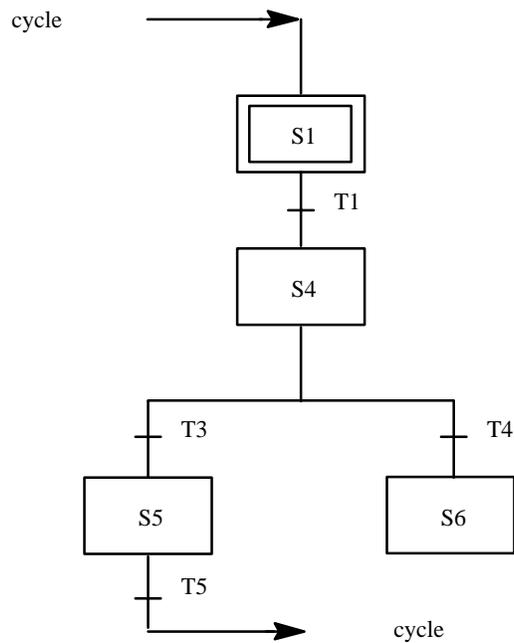


3. Deux transitions dans une séquence doivent toujours être séparées par une étape.
4. Chaque transition doit contenir des instructions qui au minimum définissent sa variable de transition. Par exemple, la transition T1 doit contenir un segment similaire à celui illustré ci-dessous de manière à positionner la variable de transition T1.



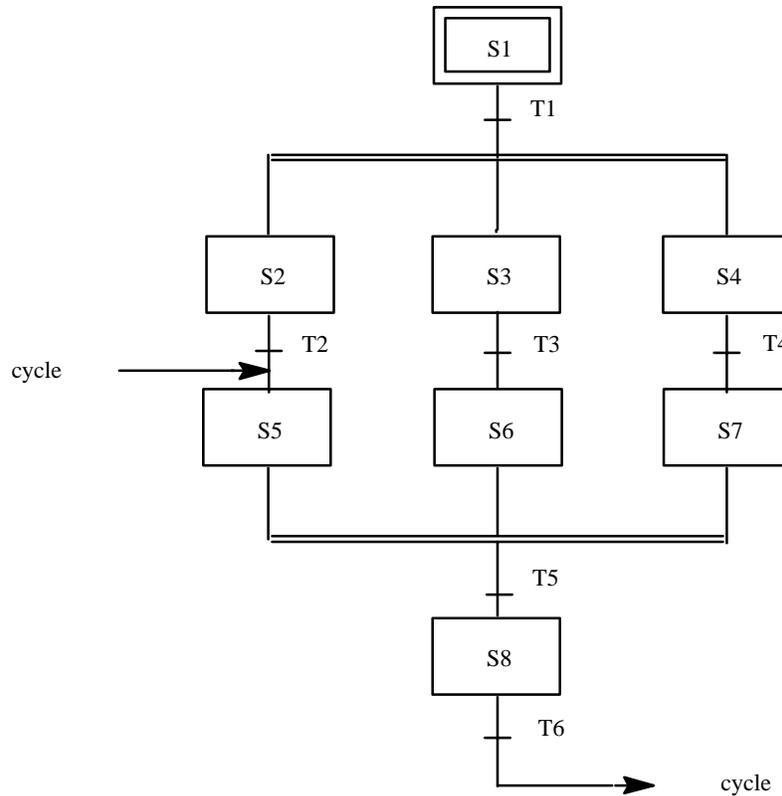
5. Des étapes terminales sont autorisées, sauf si vous les utilisez à l'intérieur d'une séquence de divergence en ET. Une étape terminale est une étape qui n'est pas suivie d'une transition. Une fois activée, elle ne peut être désactivée, sauf par une instruction SFC_RESET. (Pour plus d'informations sur la fonction SFC_RESET, voir à la page 3-56.)

Dans le segment de graphe commande étape-transition valide illustré ci-dessous, il n'y a pas de transition après l'étape S6. Ceci terminera le graphe commande étape-transition, et l'étape S6 continuera à s'exécuter à chaque évolution tant que le bloc est actif et que le graphe n'est pas réinitialisé.



6. Un saut ne peut pas pointer dans une branche d'une séquence de divergence en ET.

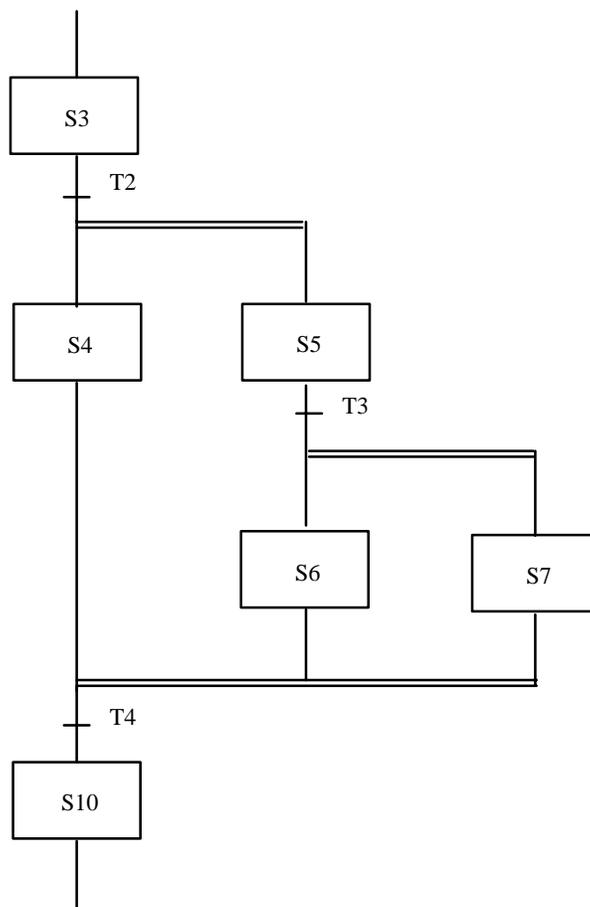
Dans le segment de graphe commande étape-transition **non valide** ci-dessous, l'opérateur a utilisé un saut pointant dans une séquence de divergence en ET. Le logiciel Logicmaster 90 indiquera le programme comme non-exécutable et empêchera son enregistrement dans l'API Série 90-70.



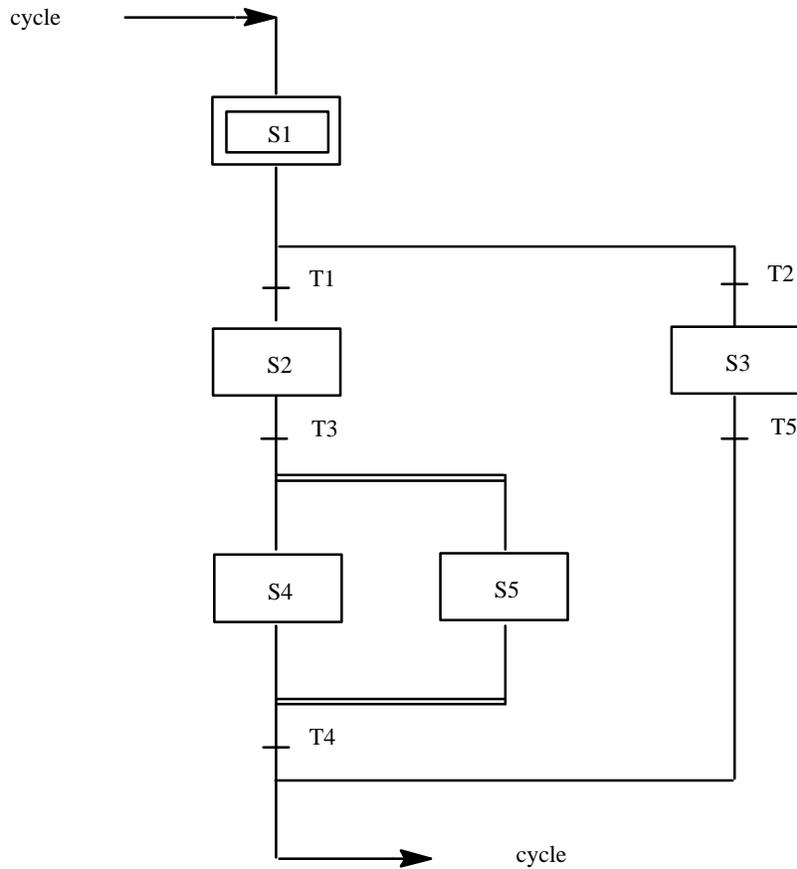
7. Un saut ne peut sauter hors d'une branche à l'intérieur d'une séquence de divergence en ET.

8. Deux points divergents du même type de séquence (divergences en ET ou en OU) peuvent avoir un point convergent commun. De même, un point divergent peut avoir plusieurs points convergents.

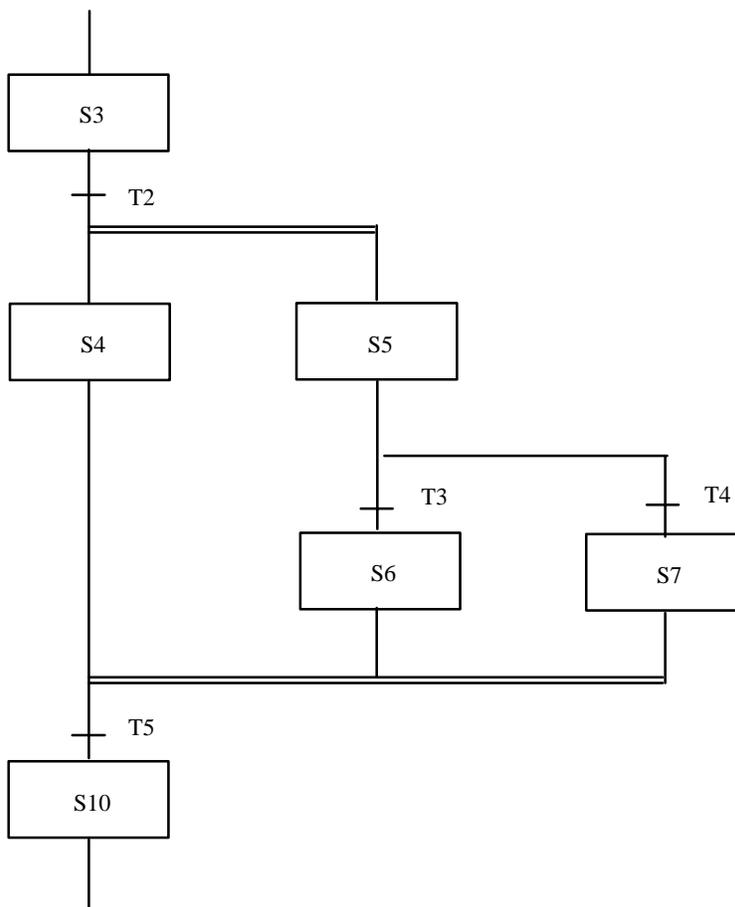
La partie ci-dessous d'un segment de graphe commande étape-transition valide illustre un point commun de convergence pour deux séquences de divergences en ET.



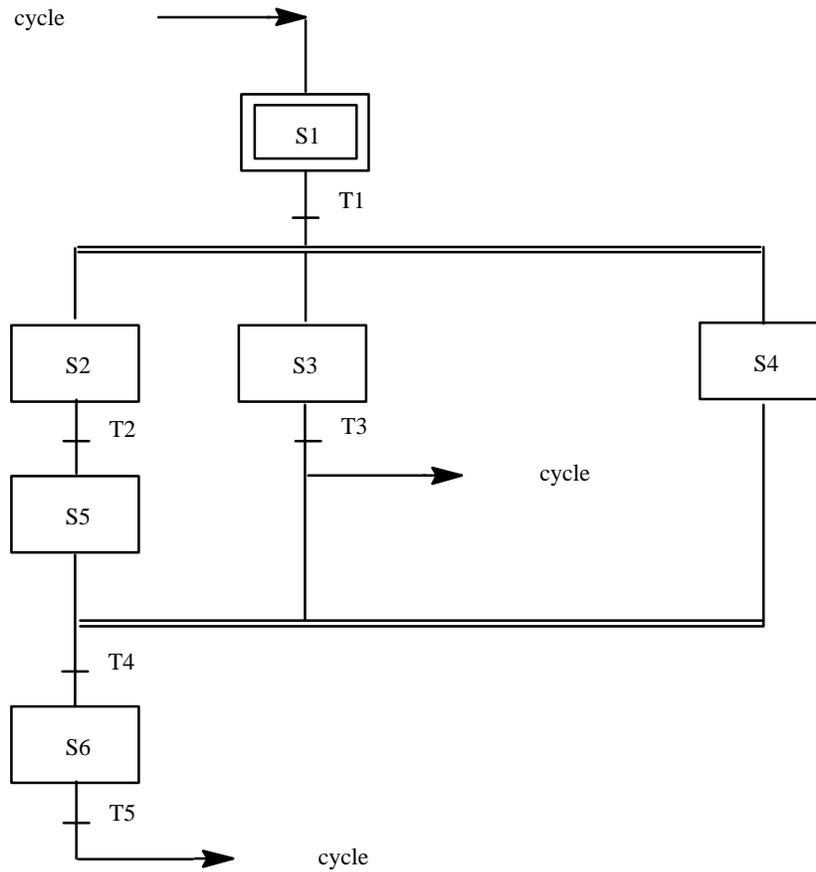
9. Les divergences et convergences de séquences doivent être correctement imbriquées. Par exemple, si une séquence de divergence en ET est effectuée après une divergence en OU, elle doit converger avant la convergence en OU. Le segment de graphe commande étape–transition valide ci–dessous illustre une imbrication correcte de séquences de différents types.



Ce segment de graphe commande étape–transition **invalide** illustre une imbrication incorrecte de deux séquences de types différents.



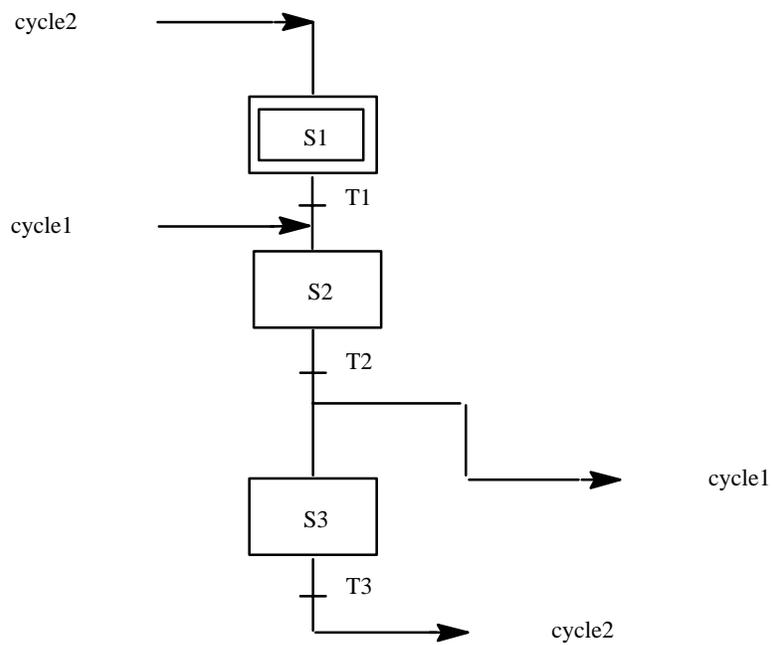
Dans le segment de graphe commande étape–transition **invalide** ci-dessous, l’opérateur a utilisé un saut hors d’une séquence de divergence en ET, de l’étape S3 à l’étape S1.



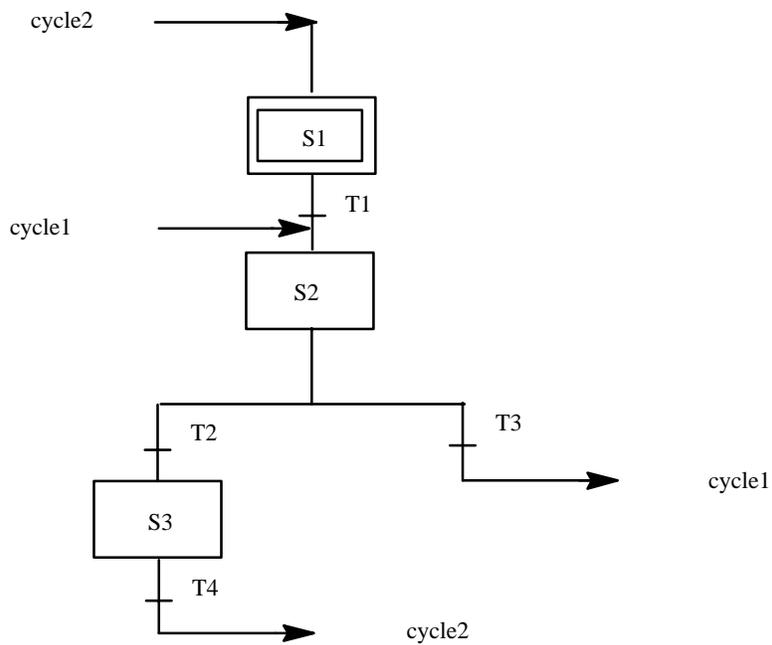
Remarque

Lorsque des séquences convergent, elles doivent le faire en respectant les règles d'imbrication. Des structures sélectives peuvent rester ouvertes, mais des structures simultanées doivent être fermées.

Le segment de graphe commande étape–transition ci–dessous est **invalide** car, lorsque l'étape S2 est active et que la transition T2 est évaluée à vrai, les deux étapes S2 et S3 deviennent simultanément actives à l'intérieur d'une séquence séquentielle.



Ce même segment de graphe commande étape–transition est correctement défini ci–dessous.



Page laissée blanche intentionnellement

Chapitre 3

L'éditeur GRAFCET (Grphe Commande Etape–Transition)

L'éditeur de graphe commande étape–transition est utilisé pour créer, modifier ou contrôler un bloc de programme écrit en langage GRAFCET.

Ce chapitre se répartit comme suit :

	Titre	Description	Page
1	Blocs de programme GRAFCET	Décrit la structure d'un bloc GRAFCET, comment sélectionner le langage du bloc, le déplacement de curseur dans un bloc, et les fonctions disponibles dans l'éditeur GRAFCET.	3–2
2	Insertion /Edition dans un réseau GRAFCET	Décrit comment créer un nouveau réseau GRAFCET ou modifier un réseau existant. Détaille également comment s'effectue l'affichage des erreurs dans le réseau et l'enregistrement des modifications dans un bloc GRAFCET.	3–10
3	Fonction de recherche	Décrit comment chercher un élément dans un bloc programmé en langage GRAFCET.	3–44
4	Commentaires	Décrit comment documenter des actions associées avec une étape particulière.	3–50
5	Options associées aux blocs GRAFCET	Décrit la consignation des défauts pour les dépassements de durée d'étape et la configuration de plusieurs langages.	3–51
6	Fonctions de mise au point	Décrit comment forcer un bloc GRAFCET à démarrer à partir de l'étape initiale, forcer des étapes à l'état actif ou inactif, forcer des transitions, suivre l'évolution d'un bloc et définir des temps d'exécutions minimal et maximal pour une étape.	3–55
7	Fonctions utilitaires	Décrit comment charger un programme de l'API vers la console de programmation, enregistrer un programme dans l'API à partir de la console de programmation, et vérifier un programme avec l'API.	3–63

1. BLOCS DE PROGRAMME GRAFCET

Tout bloc de programme, qu'il s'agisse d'un bloc **_MAIN** ou d'un bloc de sous-programme, peut être programmé en langage GRAFCET. Les blocs de sous-programmes paramétrés sont des blocs de sous-programmes en langage à relais et ne peuvent donc pas être programmés en GRAFCET. Il est cependant possible d'utiliser ces blocs dans les instructions en langage à relais d'un bloc GRAFCET.

Pour programmer un bloc en langage GRAFCET, il faut sélectionner le langage GRAFCET au moment de la création du bloc. La sélection du langage de programmation est décrite à la page 3–7.

Un bloc programmé en langage GRAFCET inclut le graphe commande étape–transition lui-même et toutes les instructions contenues dans chaque étape et chaque transition, ainsi que les instructions de preprocessing et de postprocessing. Un bloc GRAFCET peut contenir jusqu'à 255 étapes, 383 transitions et 255 sauts d'étape uniques. La taille d'un bloc logique est limitée à 16 Koctets. Pour un bloc GRAFCET, cela inclut la topologie GRAFCET et toutes les actions et instructions en langage à relais associées.

Remarque

Compte tenu de cette limite de 16 Koctets et des restrictions imposées à la complexité du réseau, il ne sera pas toujours possible d'intégrer les instructions correspondant à toutes les étapes dans un seul bloc. Vous pouvez éviter ce problème en plaçant les instructions correspondant à chaque étape dans un bloc distinct. Seules les instructions associées à l'étape seront alors utilisées avec un appel au bloc.

La liste ci-dessous résume les limites auxquelles sont soumis les blocs programmés en langage GRAFCET.

- 16K octets d'espace logique.
- 1 réseau GRAFCET.
- 255 étapes.
- 383 transitions.
- 255 sauts d'étape uniques.
- 32 branches parallèles simultanées.
- 32 étapes simultanément actives.
- 64 appels de blocs de programme.
- Taille de grille GRAFCET de 128 rangées par 8 colonnes.

Chaque étape, transition et saut d'étape utilisé dans un réseau GRAFCET doit avoir un nom. Ces noms sont placés dans la table de déclaration des variables du bloc et sont connus dans l'ensemble du bloc, y compris des instructions en langage à relais. Les noms ne peuvent pas être utilisés en dehors du bloc en cours, même s'il s'agit du bloc **_MAIN**. Les noms de transition ne peuvent être référencés que dans la transition qui les définit.

Bien qu'ils puissent être référencés sur l'ensemble du bloc GRAFCET, les étapes et les noms de transition ne peuvent être définis qu'une fois par bloc. Autrement dit, une étape ou une transition donnée ne peut se produire qu'une seule fois dans un réseau GRAFCET, et ne peut survenir que dans un réseau GRAFCET dans le bloc.

Les sauts d'étape ne peuvent apparaître que dans un réseau GRAFCET dans un bloc. A l'intérieur d'un réseau GRAFCET, il peut y avoir une occurrence d'un saut d'étape destination donné, mais il peut y avoir plusieurs occurrences du saut d'étape source correspondant.

1.1. TEMPORISATEURS D'ÉTAPE

Une valeur temporelle de l'étape (par exemple, S1.t) est associée à chaque étape. Il s'agit d'une valeur entière qui donne le nombre d'unités de temps, par rapport à la base de temps sélectionnée par l'utilisateur, depuis que l'étape a été activée. Elle peut être utilisée comme un paramètre en lecture-seule n'importe où dans le bloc sur un opérande de fonction en langage à relais requérant un seul mot d'entrée.

Lorsqu'une étape est activée, son temporisateur d'étape est réinitialisé à zéro avant de commencer à compter les unités de temps. Pour toutes informations sur la définition des durées d'étapes minimale et maximale, référez-vous au chapitre 3, § 6, *Fonctions de mise au point*.

1.2. BITS DE DÉFAUT D'ÉTAPE

Lorsqu'une étape n'atteint pas son temps d'activation minimal, ou lorsqu'elle dépasse son temps d'activation maximal, un bit de défaut d'étape (par exemple, S1.f), est mis à un. Par exemple, le bit de défaut de l'étape S1 est S1.f. Ce bit de défaut peut être référencé dans des instructions en langage à relais n'importe où dans le bloc, mais ne peut être modifié puisqu'il est en lecture seule. Il peut être utilisé avec toutes les instructions de contact, sauf les contacts transitionnels. Le bit de défaut conserve son état de défaut jusqu'à ce que l'étape redevienne active. Il est alors remis à zéro.

1.3. BITS D'ÉTAPE

Un bit d'étape booléen (par exemple S1.x) est également associé à chaque étape. Le bit d'étape indique l'état d'exécution de toutes les instructions d'actions à l'intérieur de l'étape. Il peut être référencé dans les instructions en langage à relais n'importe où dans le bloc pour déterminer si une étape donnée est active ou inactive. Le bit d'étape peut être utilisé sur tout contact non transitionnel. Cependant, il ne peut pas être modifié dans des instructions du programme utilisateur car il est en lecture seule.

Remarque

Les valeurs associées aux étapes, par exemple S1.t, S1.f et S1.x, sont effacées lorsque le dossier en cours est enregistré et à l'exécution d'une fonction **SFC_RESET**.

1.4. FORMAT D'UN BLOC GRAFCET

Le format général d'un bloc **_MAIN** en langage GRAFCET :

```
[ DEBUT DU PROGRAMME SFC GRAFCET ]
[   DECLARATIONS DES VARIABLES   ]
[   DECLARATIONS DES BLOCS       ]
[ DECLARATION DES INTERRUPTIONS   ]
[   PREPROCESSING                 ]
[   POSTPROCESSING                ] ← Réseau GRAFCET ici.
[   FIN DU PROGRAMME             ]
```

Le format général d'un bloc de sous-programme programmé en langage GRAFCET est :

```
[   DEBUT DU BLOC SFC BLOC 1     ]
[   DECLARATIONS DES VARIABLES   ]
[   PREPROCESSING                 ]
[   POSTPROCESSING                ] ← Réseau GRAFCET ici.
[   FIN DU BLOC                  ]
```

Les marqueurs [**DEBUT PROGRAMME SFC GRAFCET**], [**DECLARATIONS DES VARIABLES**], [**DECLARATIONS DES BLOCS**], [**DECLARATION DES INTERRUPTIONS**] et [**FIN DE PROGRAMME**] sont fonctionnellement identiques à ceux utilisés en langage à relais. Pour plus d'informations sur ces marqueurs, veuillez vous reporter au document *GFK-0263 Logimaster 90-70 Programming Software User's Manual*.

Seuls les marqueurs [**PREPROCESSING**] et [**POSTPROCESSING**], décrits dans le tableau suivant, sont spécifiques au langage GRAFCET.

Marqueur	Description
Instructions de preprocessing	Les instructions de preprocessing sont exécutées à chaque appel du bloc de programme, avant l'évolution des graphes commande étape–transition. Ces instructions sont optionnelles, mais si vous les utilisez, elles doivent être programmées dans un langage terminal, tel que le langage à relais. Il est possible d'accéder en lecture seule aux bits d'étape du graphe commande étape–transition, et d'accéder à tous les symboles de référence déclarés dans la table de déclarations des variables. Le nombre d'instructions qui peut être programmé n'est limité que par la taille du bloc de programme.
Instructions de postprocessing	Les instructions de postprocessing fonctionnent de manière similaire aux instructions de preprocessing mais sont exécutées après l'évolution des graphes commande étape–transition, autrement dit après l'exécution de toutes les instructions d'actions mais avant la sortie du bloc de programme.

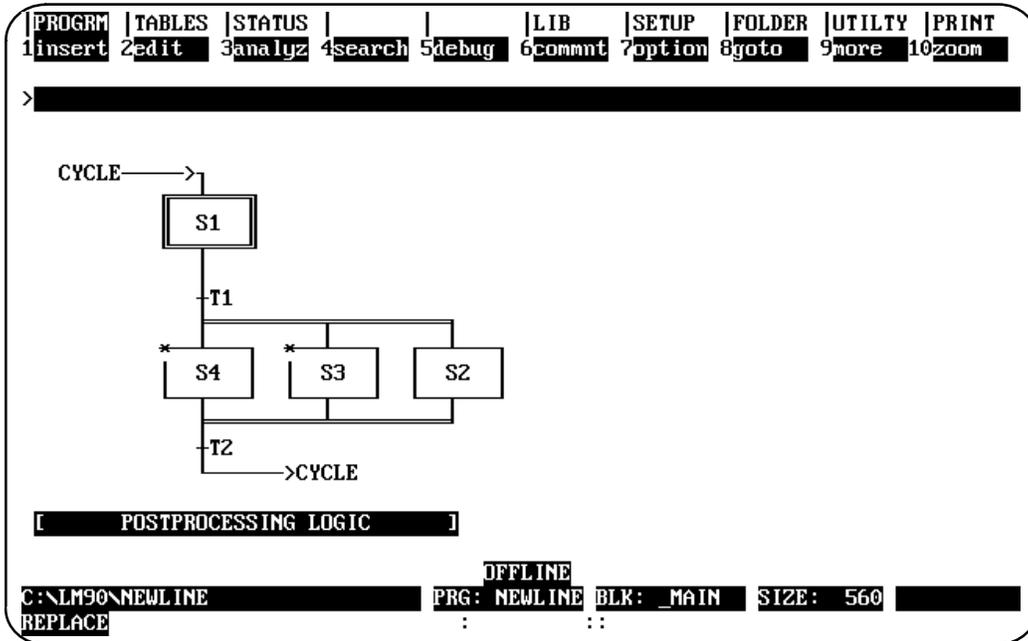
Remarque

Les instructions GRAFCET effectives dans un bloc de programme GRAFCET seront exécutées après les actions de preprocessing et avant les actions de postprocessing.

1.5. INSTRUCTIONS DE PREPROCESSING ET DE POSTPROCESSING

Le bloc GRAFCET contient deux lignes de marquage nommées [**PREPROCESSING**] et [**POSTPROCESSING**]. Ces lignes de marquage vous permettent d'accéder aux instructions qui précèdent et suivent chaque évolution GRAFCET. Pour insérer ou modifier ces instructions, vous pouvez déplacer le curseur sur la ligne de marquage sélectionnée puis appuyer sur la touche **Zoom (F10)**.

Dans l'écran donné en exemple ci-dessous, le curseur est positionné sur [**POSTPROCESSING**].



Lorsque vous appuyez sur la touche **Zoom (F10)**, vous obtenez l'écran ci-dessous.

```
PROGRAM | TABLES | STATUS | | | LIB | SETUP | FOLDER | UTILITY | PRINT
1insert 2edit 3modify 4search 5 | 6 | 7option 8goto 9more 10zoom
>
[ START OF POSTPROCESSING LOGIC ]
[ END OF POSTPROCESSING LOGIC ]
OFFLINE
C:\LM90\NEWLINE PRG: NEWLINE BLK: _MAIN SIZE: 560 RUNG 0001
REPLACE :
```

A partir de cet écran, vous pouvez entrer les instructions en langage à relais à exécuter après chaque évolution du bloc GRAFCET. La seule limite au nombre d'instructions est l'espace d'instructions disponible dans le bloc. Comme pour les instructions de preprocessing et les instructions d'étapes, les instructions JUMP et MCR doivent être résolues à l'intérieur de ce segment d'instructions.

Pour retourner au bloc GRAFCET, appuyez sur la touche **Escape**.

1.6. PROGRAMMATION DES INSTRUCTIONS D' ACTIONS D'ÉTAPE GRAFCET

Chaque étape peut comporter des instructions d'actions, composées de lignes d'instructions en langage à relais associées. La taille maximale des instructions associées à une action est limitée par la quantité de mémoire disponible dans le bloc courant. Pour afficher ces instructions en langage à relais, positionnez le curseur sur l'étape et appuyez sur **Zoom (F10)**.

Les instructions d'actions d'une étape peuvent être modifiées en utilisant les fonctions d'édition du logiciel Logicmaster. Pour plus d'informations, référez-vous au chapitre 3, *Edition de programmes*, dans le document *GFK-0263 Logicmaster 90-70 Programming Software User's Manual*.

Les instructions d'une action d'étape peuvent aussi accéder à d'autres blocs en langage à relais ou à d'autres blocs GRAFCET en utilisant l'instruction CALL du langage à relais. Pour plus d'informations sur l'utilisation de l'instruction CALL, référez-vous au document *GFK-0265 Series 90-70 Programmable Controller Reference Manual*.

Pour sortir des instructions d'actions et retourner à l'étape S1, appuyez sur la touche **Escape**.

1.7. SÉLECTION DU LANGAGE DU BLOC

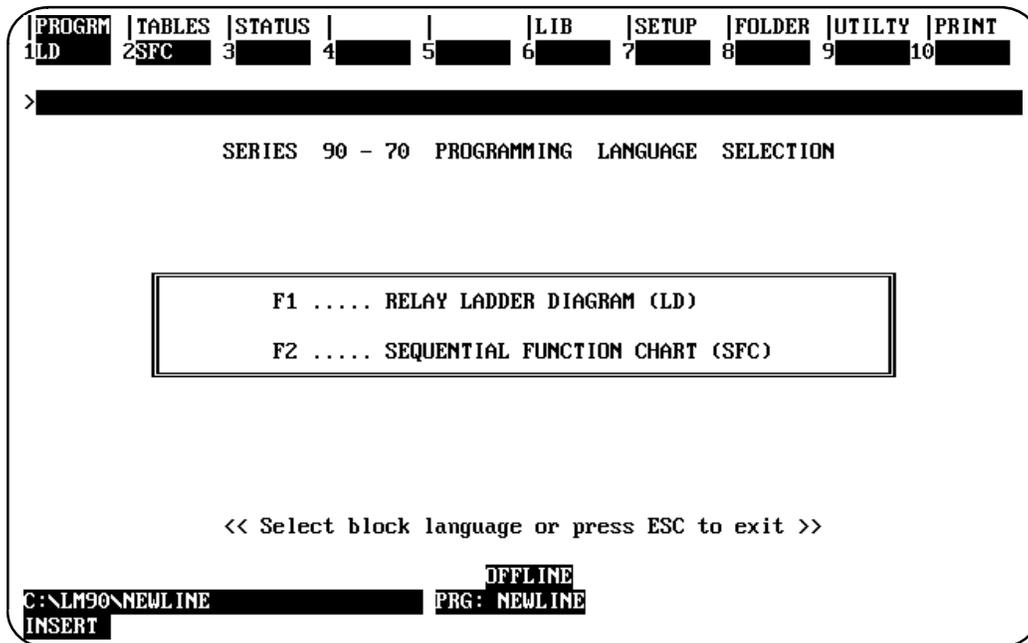
Tout bloc de programme, y compris le bloc **_MAIN**, peut être programmé en langage GRAFCET. Cependant les instructions correspondant aux actions et aux conditions de transition d'un bloc GRAFCET doivent être programmées en langage à relais.

Lorsque vous entrez la première fois dans un nouveau dossier (**_MAIN**), le logiciel Logimaster 90 vous demande de choisir un langage de programmation. Vous pouvez également sélectionner un langage lorsque vous créez des blocs de programme dans l'éditeur de déclaration de bloc de programme.

Remarque

Une fois que vous avez choisi un langage pour un bloc, vous ne pouvez plus le modifier. Pour changer le langage, vous devez supprimer la déclaration de bloc, et créer une nouvelle déclaration de bloc avec le langage désiré. Pour changer le langage du bloc **_MAIN**, vous devez créer un nouveau dossier avec le langage désiré pour **_MAIN**.

Après la création d'un nouveau dossier, appuyez sur la touche de fonction **Program (F1)** pour faire apparaître l'écran de Sélection du langage de programmation Série 90-70, montré ci-dessous.

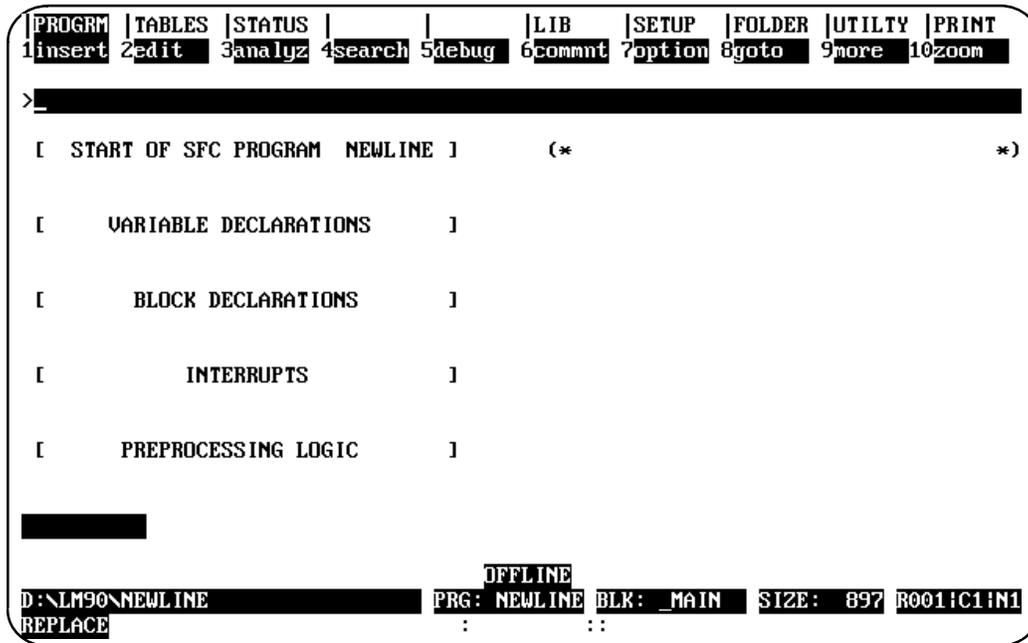


Pour sélectionner le langage à relais, appuyez sur Schéma à relais (**LD**) (**F1**). Pour sélectionner le langage GRAFCET, appuyez sur GRAFCET (**SFC**) (**F2**).

Si vous avez appuyé sur **SFC (F2)** pour sélectionner le langage GRAFCET, le système affiche le bloc GRAFCET. Les touches de fonctions affichées dans le bandeau supérieur de l'écran sont similaires aux sélections de touche affichées au premier niveau de l'éditeur en langage à relais.

```
PROGRAM | TABLES | STATUS | | | LIB | SETUP | FOLDER | UTILITY | PRINT
1insert 2edit 3analyz 4search 5debug 6commnt 7option 8goto 9more 10zoom
>
[ START OF SFC PROGRAM NEWLINE ] (* *)
[ VARIABLE DECLARATIONS ]
[ BLOCK DECLARATIONS ]
[ INTERRUPTS ]
[ PREPROCESSING LOGIC ]
[ POSTPROCESSING LOGIC ]
OFFLINE
C:\LM90\NEWLINE PRG: NEWLINE BLK: _MAIN SIZE: 373
INSERT :
```

Si vous sélectionnez un bloc existant, l'écran de sélection de langage n'apparaît pas. Le logiciel Logicmaster affichera directement l'éditeur GRAFCET ou langage à relais de premier niveau. Si le bloc était un bloc GRAFCET, l'affichage sera similaire à celui représenté ci-dessous. Le curseur apparaîtra dans la première rangée et la première colonne du premier réseau GRAFCET. Remarquez que la dernière entrée dans le coin droit de la deuxième ligne des informations d'état en bas de l'écran identifie l'emplacement actuel du curseur par le numéro de ligne, le numéro de colonne et le numéro de réseau GRAFCET. (Pour la version 5, le numéro de réseau GRAFCET sera toujours 1.)



1.8. CHANGER LE MODE D’AFFICHAGE

Vous disposez de deux modes d’affichage pour visualiser un graphe commande étape-transition. Ces modes d’affichage sont le mode **NORMAL** (le mode par défaut) et le mode **NUMERO**. Utilisez la séquence de touches ALT-N pour commuter entre ces deux modes d’affichage.

Tableau 3-1. Modes d’affichage

Mode	Description
Normal	En mode NORMAL , les étapes et les transitions prennent trois lignes d’écran. Le système utilise des noms d’étape et de transition définis par l’utilisateur, plutôt que des numéros. Vous pouvez visualiser, éditer et surveiller des programmes GRAFCET en mode NORMAL .
Numéro	Le mode NUMERO est très similaire au mode NORMAL . La seule différence est que le système fournit lui-même des noms d’étape et de transition, tels que S1, S2, T1 et T2, plutôt que d’utiliser des noms définis par l’utilisateur.

2. INSERTION/ÉDITION D'UNE TOPOLOGIE DE RÉSEAU GRAFCET

L'éditeur GRAFCET fournit une fonction d'insertion et une fonction d'édition. La fonction d'insertion permet de créer un nouveau réseau GRAFCET, tandis que la fonction d'édition est utilisée pour modifier un réseau GRAFCET existant.

```

PROGRAM | TABLES | STATUS |  |  | LIB | SETUP | FOLDER | UTILITY | PRINT
1insert 2edit 3analyz 4search 5debug 6commnt 7option 8goto 9more 10zoom
>
[ VARIABLE DECLARATIONS ]
[ BLOCK DECLARATIONS ]
[ INTERRUPTS ]
[ PREPROCESSING LOGIC ]
[ POSTPROCESSING LOGIC ]
[ END OF PROGRAM ]
OFFLINE
C:\LM90\NEWLINE PRG: NEWLINE BLK: _MAIN SIZE: 373
REPLACE : ::
    
```

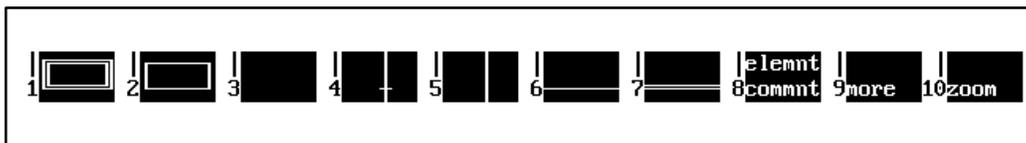
Touche	Fonction	Description	Voir Page
F1	Insérer un programme	Crée un nouveau réseau GRAFCET. Le nouveau réseau sera placé immédiatement avant le curseur. Dans un bloc GRAFCET vide, placez le curseur sur le marqueur [PREPROCESSING], et appuyez sur la touche Insérer un programme (F1). Le nouveau réseau sera placé avant le marqueur [POSTPROCESSING].	3-10
F2	Editer	Modifie un réseau GRAFCET existant. Placez le curseur sur le réseau que vous souhaitez modifier, et appuyez sur la touche Editer (F2) .	3-10
F3	Analyser	Détermine les erreurs existantes dans le réseau, et vérifie si le bloc est exécutable. Vous trouverez plus d'informations sur la fonction d'analyse à la fin de ce paragraphe.	3-42
F4	Rechercher	Recherche un ou plusieurs éléments dans un bloc GRAFCET.	3-44
F5	Debug	Donne accès aux fonctions de mise au point qui vous aideront à mettre au point un bloc GRAFCET.	3-55
F6	Commentaires	Associe des commentaires détaillés aux étapes dans un réseau GRAFCET.	3-50
F7	Options	Donne accès au paramétrage des options liées aux programmes contenant des blocs GRAFCET.	3-51
F8	Vers...	Déplace le curseur à l'emplacement de l'étape, transition ou nom de saut d'étape GRAFCET spécifié dans le bloc GRAFCET.	3-59

Touche	Fonction	Description	Voir Page
F9	Autres	Donne accès à des fonctions supplémentaires de l'éditeur.	
F10	Zoom	Effectue un zoom dans une déclaration de variable, une déclaration de bloc, ou une déclaration d'interruption. Pour plus d'informations sur l'utilisation de la fonction zoom sur ces lignes de marquage, référez-vous au document <i>GFK-0263 Logicmaster 90-70 Programming Software User's Manual</i> . Sur le marqueur [PREPROCESSING] ou [POSTPROCESSING], le zoom vous permet de zoomer dans l'éditeur de diagramme en langage à relais de premier niveau, et d'afficher les instructions de preprocessing ou de postprocessing en langage à relais. La fonction zoom peut également être utilisée sur des étapes ou des transitions dans un réseau GRAFCET. Lorsque vous effectuez un zoom dans une étape, vous pouvez visualiser et éditer les instructions d'actions associées à cette étape. Lorsque vous effectuez un zoom dans une transition, vous pouvez visualiser et éditer les instructions associées à la transition.	
La touche Autres (F9) affiche les fonctions d'édition supplémentaires suivantes :			
F6	Supprimer	Placez le curseur sur le réseau GRAFCET à supprimer et appuyez sur la touche Supprimer (F6) ou ALT-D . Puis confirmez l'effacement. <i><u>Une fois supprimé, le réseau ne peut pas être restauré.</u></i>	
F7	Renommer	Renomme chaque étape et transition dans un bloc GRAFCET. Les numéros d'étape et de transition sont normalement attribués dans l'ordre de création des étapes et des transitions. Les noms automatiques générés pour les étapes et les transitions sont basés sur ces numéros. Renommez les étapes et les transitions pour affecter les numéros dans un ordre qui ne dépendra que de la topologie GRAFCET.	3-17
F9	Autres	Donne accès à d'autres fonctions de l'éditeur.	
F10	Zoom	Voir la description de la fonction Zoom (F10) ci-dessus.	

Vous pouvez utiliser **ALT-A** pour abandonner une fonction dans l'éditeur GRAFCET. Une fois la demande d'abandon confirmée, le système restaure la version sur disque du bloc en cours, et toutes les modifications effectuées depuis le dernier enregistrement du bloc de programme sont perdues.

La séquence **ALT-D** peut être utilisée pour supprimer un réseau GRAFCET. Elle fonctionne de manière similaire à la touche **F6** décrite ci-dessus.

Lorsque vous appuyez sur les touches **Insérer un programme (F1)** ou **Editer (F2)**, le système modifie les affectations des touches de fonctions comme suit :



Touche	Fonction	Mnémo- nique	Description	Voir Page
F1	Etape initiale	&istep	Appuyez sur F1 pour entrer une étape initiale dans un réseau GRAFCET. Il peut y avoir une et une seule étape initiale dans le réseau GRAFCET.	2-4
F2	Etape normale	&step	Appuyez sur F2 pour entrer une étape normale dans un réseau GRAFCET. Une étape normale est une étape qui doit être programmée avec des instructions d'actions écrites en langage à relais.	2-4
F4	Transition	&trans	Appuyez sur F4 pour entrer une transition dans un réseau GRAFCET.	2-6
F5	Liaison verticale	&verlnk	Appuyez sur F5 pour entrer une liaison verticale.	
F6	Divergence en OU	&selbrch	Appuyez sur F6 pour entrer une divergence en OU.	2-12
F7	Divergence en ET	&simbrch	Appuyez sur F7 pour entrer une divergence en ET.	2-13
F8	Commentaire	&commnt	Appuyez sur F8 pour entrer un commentaire. Les commentaires ne peuvent être placés que sur des instructions d'étape.	3-50
F9	Autres		Appuyez sur F9 pour afficher des éléments GRAFCET ou des options supplémentaires.	
F10	Zoom		Appuyez sur F10 pour effectuer un zoom dans une étape ou une transition. Lorsque vous zoomez dans une étape, vous pouvez visualiser les instructions d'actions associées à cette étape. Lorsque vous zoomez dans une transition, vous pouvez visualiser les instructions associées à cette transition.	

Lorsque vous appuyez sur **Autres (F9)**, vous accédez aux touches de fonctions supplémentaires suivantes :



Touche	Fonction	Mnémo- nique	Description	Voir Page
<i>La touche Autres (F9) affiche les éléments supplémentaires suivants :</i>				
F1	Saut d'étape destination	&dstcnc	Appuyez sur F1 pour entrer un saut d'étape destination. Vous devez nommer le saut d'étape destination avant de pouvoir déplacer le curseur.	2-15
F2	Saut d'étape source	&srcnc	Appuyez sur F2 pour entrer un saut d'étape source. Vous devez nommer le saut d'étape source avant de pouvoir déplacer le curseur.	2-15
F3	Ouvrir la rangée		Appuyez sur F3 pour ajouter deux rangées immédiatement au-dessus de l'emplacement du curseur.	3-34
F4	Ouvrir la colonne		Appuyez sur F4 pour ajouter une colonne à la position du curseur.	3-36
F5	Effacer l'élément		Appuyez sur F5 (ou ALT-D) pour supprimer des éléments de branchement ou effacer un élément d'étape, de transition ou de saut d'étape.	3-33
F6	Effacer la rangée		Appuyez sur F6 pour supprimer la ligne désignée par la position du curseur ainsi que la ligne suivante.	3-38
F7	Effacer la colonne		Appuyez sur F7 pour supprimer la colonne à la position du curseur.	3-40
F8	Analyser le réseau		Appuyez sur F8 pour vérifier la topologie, afficher un message sur la première erreur rencontrée, et positionnez le curseur sur cette erreur.	3-42
F9	Autres		Appuyez sur F9 pour revenir au premier niveau des éléments GRAFCET.	
F10	Zoom		Appuyez sur F10 pour effectuer un zoom dans une étape ou une transition. Lorsque vous zoomez dans une étape, vous pouvez visualiser les instructions d'actions associées à cette étape. Lorsque vous zoomez dans une transition, vous pouvez visualiser les instructions associées à cette transition.	

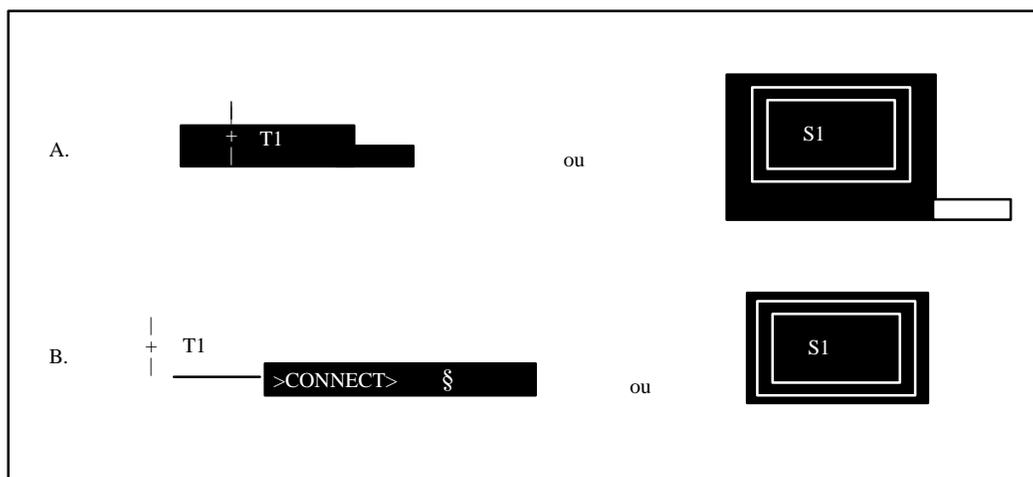
2.1. ORGANISATION DES GRILLES

Chaque réseau GRAFCET peut contenir jusqu'à 128 rangées et 8 colonnes d'éléments. Lorsque le bandeau bas est affiché, vous ne pouvez afficher que 6 rangées par 8 colonnes du réseau ; sans le bandeau bas, vous pouvez afficher une rangée supplémentaire. Le bandeau bas peut être désactivé en appuyant sur ALT-N.

Les rangées de numéros impairs sont désignées comme des rangées de transition, et les rangées de numéros pairs sont des rangées d'étape. Le logiciel Logicmaster applique automatiquement l'élément que vous sélectionnez à la rangée appropriée.

Les rangées de transition peuvent contenir des éléments de transitions, de branchement, de saut d'étape et de liaison verticale. Les rangées d'étape peuvent contenir les deux types d'éléments d'étape et d'éléments de liaison verticale. Une position particulière dans une rangée de transition peut contenir une transition, avec un élément de branchement sur la ligne supérieure de la rangée, et un autre élément de branchement sur la ligne inférieure de la rangée.

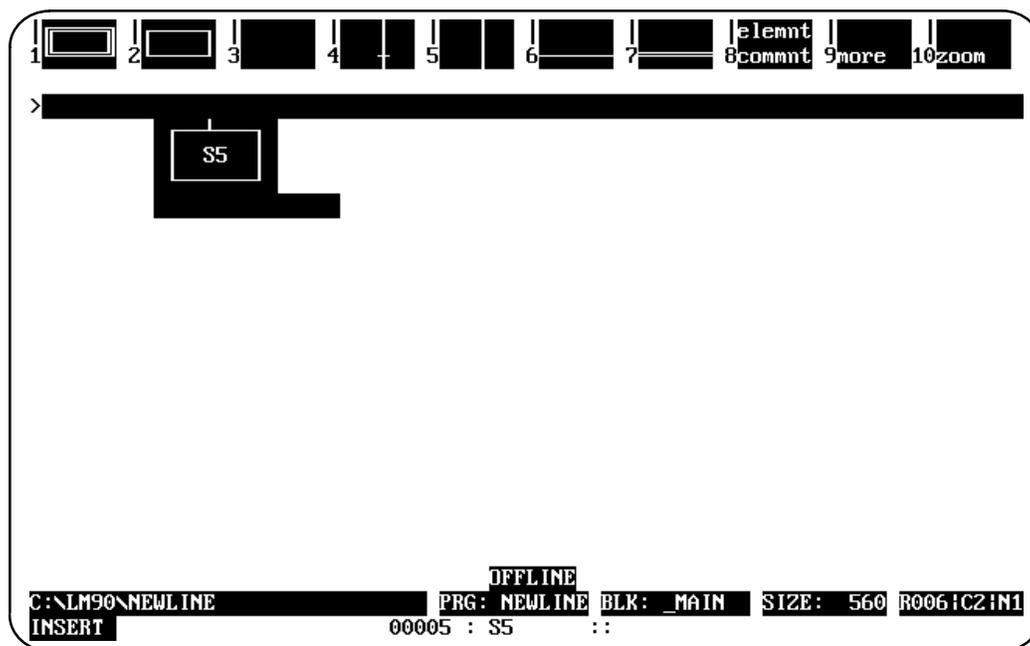
La forme du curseur en mode **INSERER UN PROGRAMME** ou **EDITER** change, selon que le curseur est sur une rangée d'étape ou de transition. Deux formes de curseur sont fournies sur les rangées d'étape et de transition, ainsi qu'illustré ci-dessous.



Le curseur A est utilisé pour indiquer où sera inséré un branchement. Le curseur B est utilisé sur des éléments de saut d'étape, des branchements, ainsi que sur des transitions si un branchement existe déjà sous la transition ou l'étape.

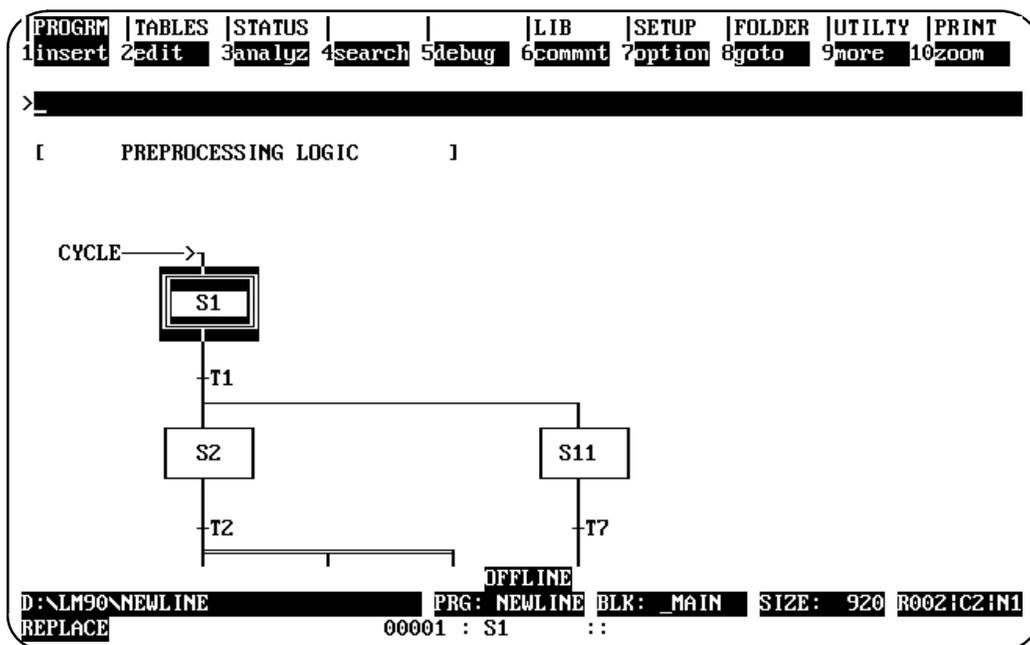
A l'emplacement d'une rangée de transition, vous pouvez ajouter une transition ou un branchement. A partir du curseur A, le branchement est ajouté à la position indiquée par le segment inférieur le plus long du curseur.

Le curseur de rangée d'étape est illustré ci-dessous :



La ligne de curseur inférieure la plus longue du curseur A indique où le branchement va être inséré. Les branchements de divergence en OU et de convergence en ET sont entrés à partir de la rangée d'étape. Les branchements de convergence en OU et de divergence en ET sont entrés à partir de la rangée de transition.

S'il existe un branchement sous une étape, le système affiche le curseur d'étape B, ainsi qu'illustré dans l'écran suivant. Le branchement aura son propre curseur, comme décrit précédemment.



2.2. DÉPLACEMENT DU CURSEUR

Vous disposez dans l'éditeur GRAFCET de premier niveau des touches de curseur suivantes :

Touches	Nom	Description
[↑] et [↓]	Touches de curseur haut et bas	Déplace le curseur respectivement d'une rangée vers le haut ou vers le bas, à l'intérieur du graphe commande étape-transition. Le système fait défiler le graphe aux limites de l'affichage. S'il est sur la première rangée du réseau GRAFCET, le curseur sera placé sur la ligne de marqueur des instructions de preprocessing. S'il est sur la dernière rangée du réseau GRAFCET, le curseur sera placé sur la ligne de marqueur des instructions de postprocessing.
[←] et [→]	Touches de curseur gauche et droite	Déplace le curseur d'une colonne respectivement vers la gauche ou vers la droite, dans le graphe commande étape-transition. Un déplacement au-delà du côté gauche ou du côté droit de la zone de 8 colonnes entraînera un bouclage sur la rangée immédiatement supérieure ou inférieure, respectivement, et le curseur sera positionné sur le côté opposé.
[CTRL][↑] [CTRL][↓] [CTRL][←] [CTRL][→]	Touche CTRL-Curseur	Déplace le curseur sur la première ou la dernière rangée du réseau GRAFCET, en préservant la position de colonne. Ru déplace le curseur sur la première ou la dernière colonne du réseau GRAFCET, en conservant la position de rangée. (Ces touches sont applicables aux modes INSERER UN PROGRAMME et EDITER uniquement.)
[PgUp] et [PgDn]	Touches Page Up et Page Down	Déplace le curseur de six rangées à la fois vers le haut ou vers le bas respectivement. Le curseur est toujours placé dans la colonne 1 de la rangée.
[Home]	Touche Home	Place le curseur au début du marqueur de programme/bloc.
[End]	Touche End	Place le curseur à la fin du marqueur de programme/bloc.

2.3. ATTRIBUTION DE NOMS AUX ÉTAPES, TRANSITIONS ET SAUTS D'ÉTAPE

Les noms système S1 à S9999 sont réservés dans un bloc GRAFCET pour être utilisés comme des noms d'étapes. De même, les noms de transitions T1 à T9999 sont réservés aux transitions, et les noms de sauts d'étape C1 à C9999 sont réservés pour des sauts d'étape. Vous ne pouvez pas utiliser ces noms comme symboles de référence, noms d'étiquettes, ou noms de bloc de programme.

Les numéros d'étapes et de transitions sont attribués dans l'ordre de création de ces étapes et transitions. Les noms automatiques générés pour ces étapes et transitions sont basés sur ces numéros. Cependant, vous pouvez Renommer chaque étape et transition dans un bloc GRAFCET en appuyant sur **Autres (F9)** puis sur **Renommer (F7)** dans l'éditeur GRAFCET. Quand vous renumérotez ainsi les étapes et les transitions, les numéros sont réattribués selon un ordre qui ne dépend que de la topologie GRAFCET.

Pour changer le nom automatique, entrez le nom souhaité sur la ligne de commande avec le curseur positionné sur l'étape ou la transition, et appuyez sur la touche **Enter**. Le nom doit respecter les règles des identificateurs Logimaster 90. (Pour plus d'informations sur les identificateurs, veuillez vous référer au chapitre 3, § 5, *Table de déclaration des variables*, dans le document *GFK-0263 Logimaster 90-70 Programming Software User's Manual*.)

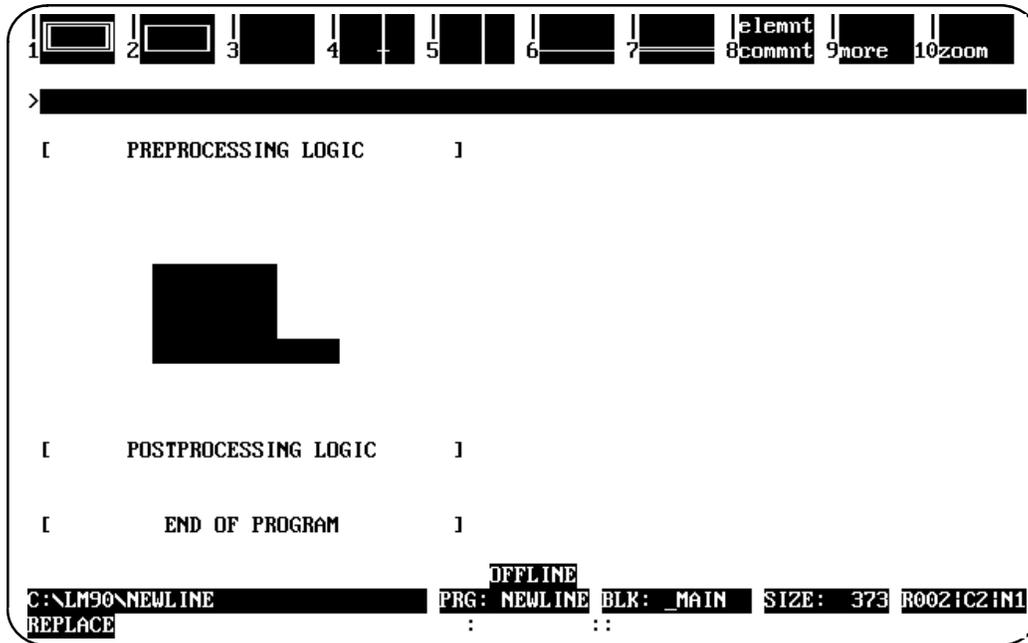
2.4. INSERTION/ÉDITION D'UN RÉSEAU GRAFCET (EXEMPLE)

Cet exemple vous présente pas à pas l'édition d'un réseau GRAFCET.

1. Entrez un nouveau dossier, et sélectionnez le langage GRAFCET. Le système affiche l'écran suivant :

```
|PROGRAM |TABLES |STATUS | | |LIB |SETUP |FOLDER |UTILITY |PRINT
1insert 2edit 3analyz 4search 5debug 6commnt 7option 8goto 9more 10zoom
>
[ START OF SFC PROGRAM NEWLINE ] (* *)
[ VARIABLE DECLARATIONS ]
[ BLOCK DECLARATIONS ]
[ INTERRUPTS ]
[ PREPROCESSING LOGIC ]
[ POSTPROCESSING LOGIC ]
OFFLINE
D:\LM90\NEWLINE PRG: NEWLINE BLK: _MAIN SIZE: 479
REPLACE :
```

- Appuyez sur **Insérer un programme (F1)** pour créer un nouveau réseau GRAFCET, ou appuyez sur **Editer (F2)** pour éditer un réseau GRAFCET existant. Lorsque vous appuyez sur **Insérer (F1)**, le système affiche l'écran ci-dessous.



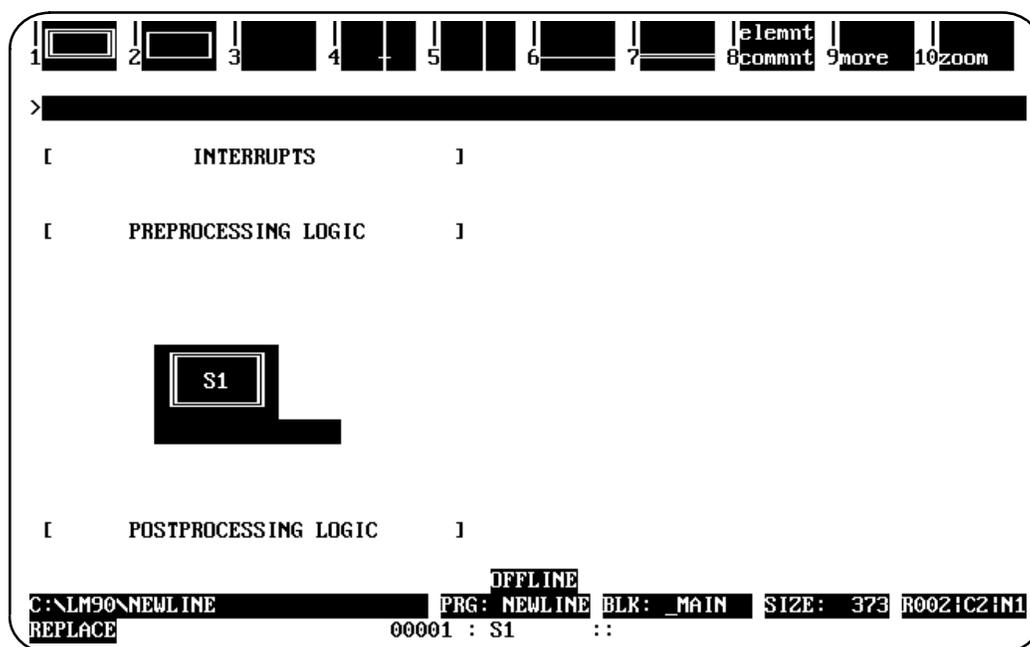
Les touches de fonctions affichent les éléments GRAFCET de base. Vous pouvez également entrer un de ces éléments en saisissant son mnémonique sur la ligne de commande.

Tableau 3-2. Mnémoniques GRAFCET

Mnémonique	Nom
&istep	Etape initiale.
&step	Etape normale.
&trans	Transition.
&verlnk	Liaison verticale.
&selbrch	Ouvrir une divergence en OU.
&simbrch	Ouvrir une divergence en ET.
&srcnet	Saut d'étape source.
&dstcnct	Saut d'étape destination.

Les rangées de numéros impairs (un, trois, cinq, etc.) sur un réseau GRAFCET sont des rangées de transitions. Les rangées de numéros pairs (deux, quatre, six, etc.) sont des rangées d'étapes. La position actuelle du curseur est indiquée dans le bandeau bas sur l'écran précédent par rangée 2, colonne 2. Vous pouvez donc entrer un élément d'étape à cet emplacement. Si vous entrez une position, le logiciel Logicmaster 90 tentera d'entrer l'élément sur la 3ème rangée sous la position actuelle du curseur, car les transitions ne peuvent être entrées que sur des rangées de numéros impairs. Le curseur se positionnera sur la rangée suivante même si votre tentative d'entrée d'une transition échoue.

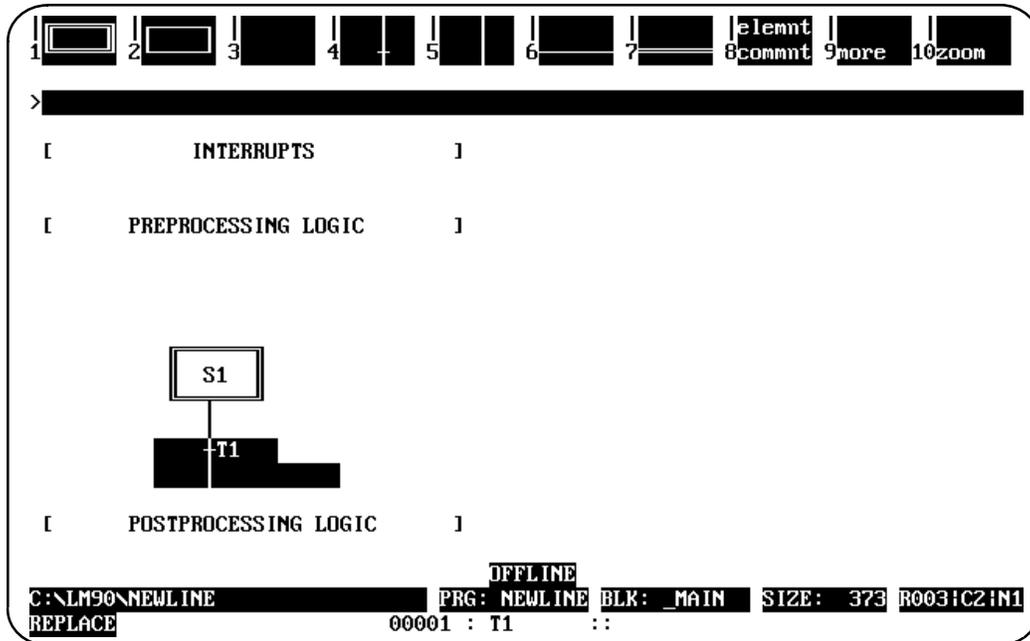
- Appuyez sur la touche **F1** pour entrer une étape initiale.



Le logiciel Logicmaster entre automatiquement le nom d'étape par défaut S1.

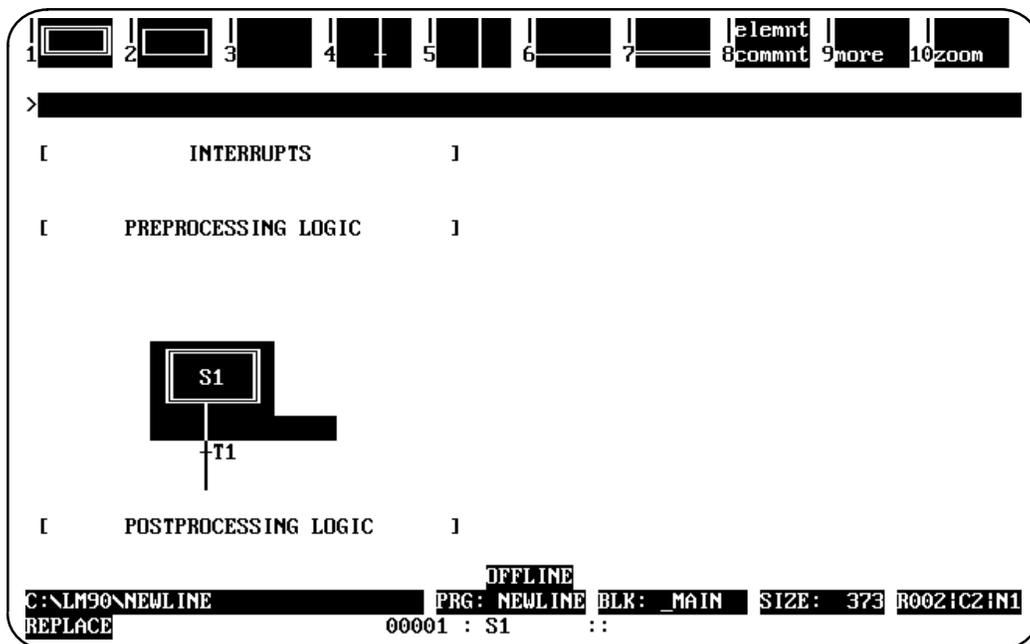
Si vous saisissez un autre nom sur la ligne de commande, le logiciel vérifiera si le nom est disponible et dans ce cas l'appliquera à la place du nom par défaut S1. (Pour plus d'informations sur l'attribution de noms aux étapes, voir page 3-17.)

4. Après avoir entré l'étape initiale, continuez à créer le réseau GRAFCET, ou appuyez sur la touche **Zoom (F10)** pour programmer l'étape. (Un réseau GRAFCET valide doit comporter une étape initiale).
5. Une fois l'étape entrée dans le réseau GRAFCET, le curseur reste sur l'élément. Vous pouvez entrer un type d'étape différent, effectuer un zoom dans l'étape, la supprimer, entrer une transition, ou entrer un branchement. Si vous entrez une transition en appuyant sur la touche **F4**, la transition est entrée et le curseur automatiquement avancé, ainsi qu'illustré ci-dessous.

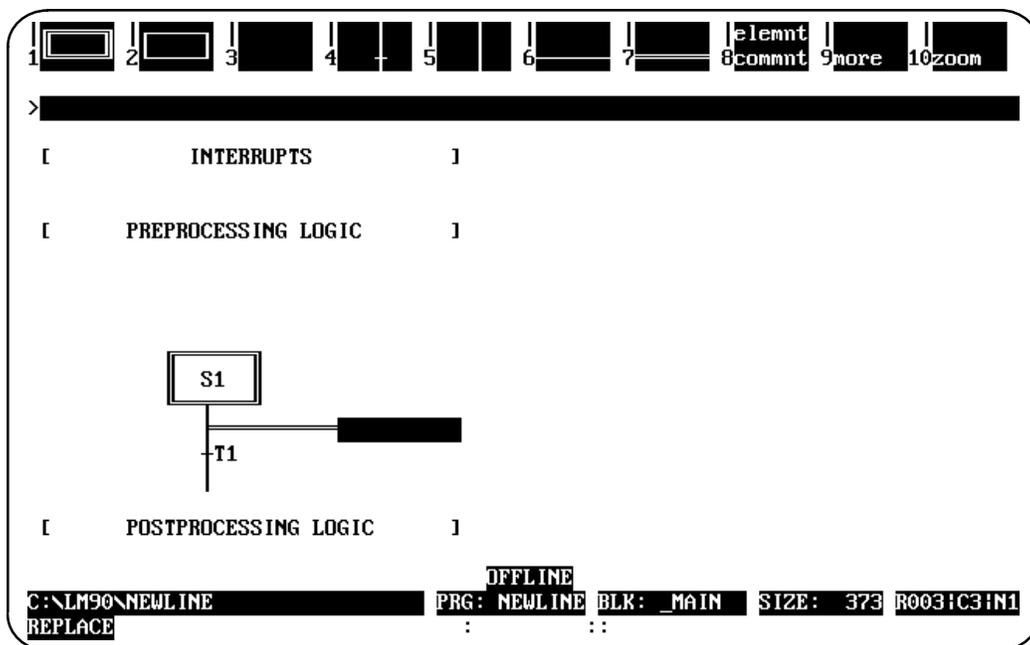


Le logiciel Logicmaster attribue également un nom à la transition. (Pour plus d'informations sur l'attribution de noms aux transitions, se reporter à la page 3-17).

6. L'emplacement de la patte inférieure du "L" du curseur en forme de L indique où sera placée l'instruction de branchement, si vous décidez d'en entrer une à cet emplacement. Si vous déplacez le curseur sur cet emplacement, le système affiche l'écran suivant.

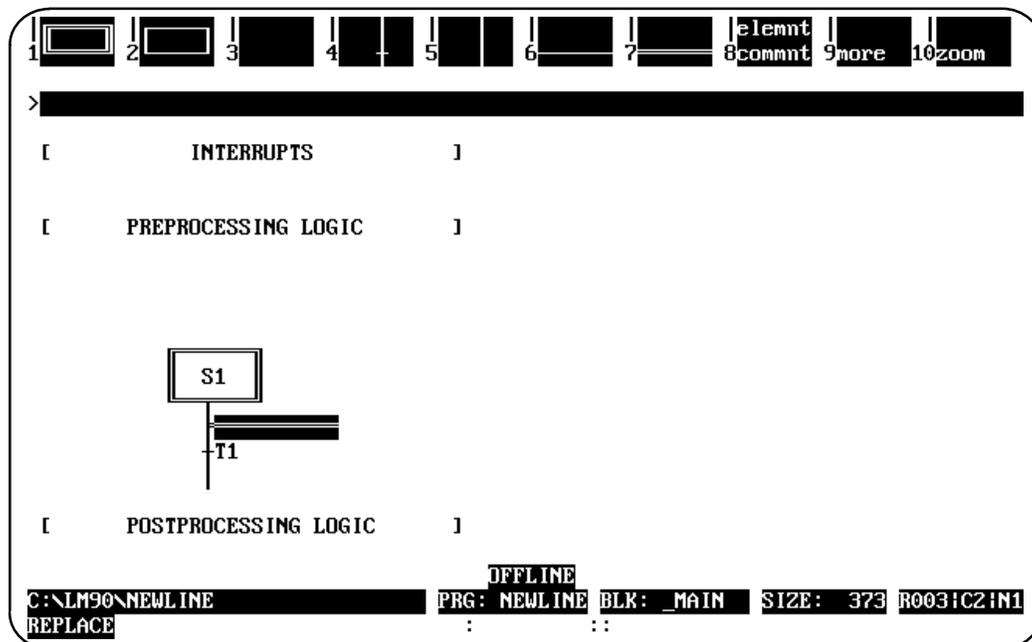


Si vous ouvrez une divergence en ET en appuyant sur la touche **F7**, le système affiche l'écran suivant.



Remarquez que le curseur a automatiquement avancé pour vous permettre d'étendre le branchement avec la prochaine touche de définition de branchement.

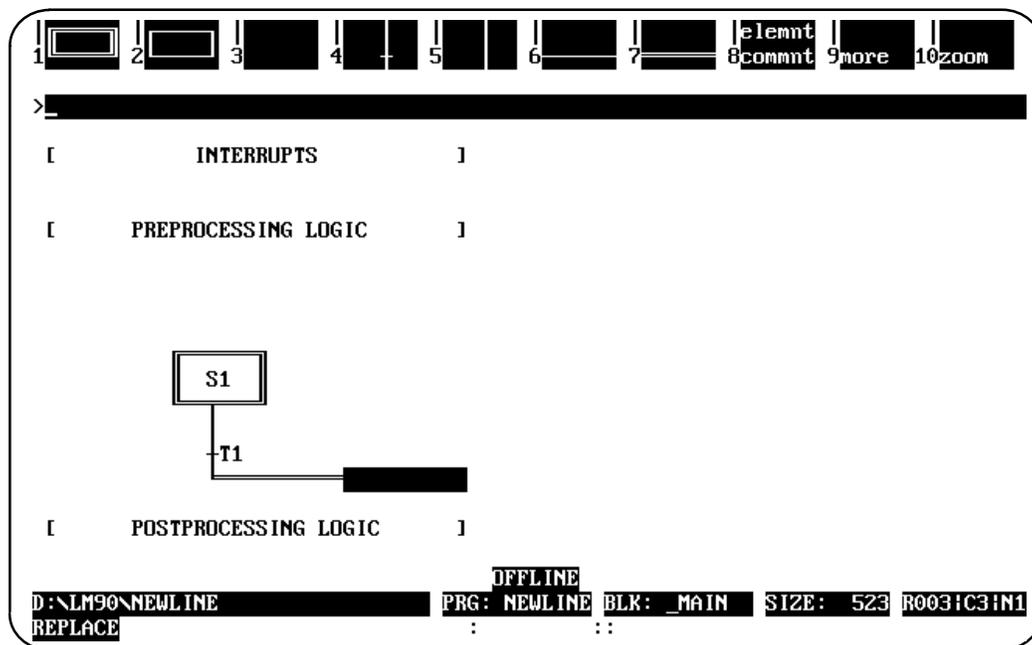
7. Pour déplacer le branchement sous la transition T1 et retirer le branchement que vous venez d'entrer, positionnez le curseur sur la partie la plus à gauche du branchement. Puis appuyez sur la touche de divergence en ET ou en OU qui correspond au branchement à supprimer. (Ou appuyez sur **ALT-D**.) Les touches de fonctions de branchement agissent comme des touches à bascule, permettant tour à tour d'insérer ou de supprimer un branchement.



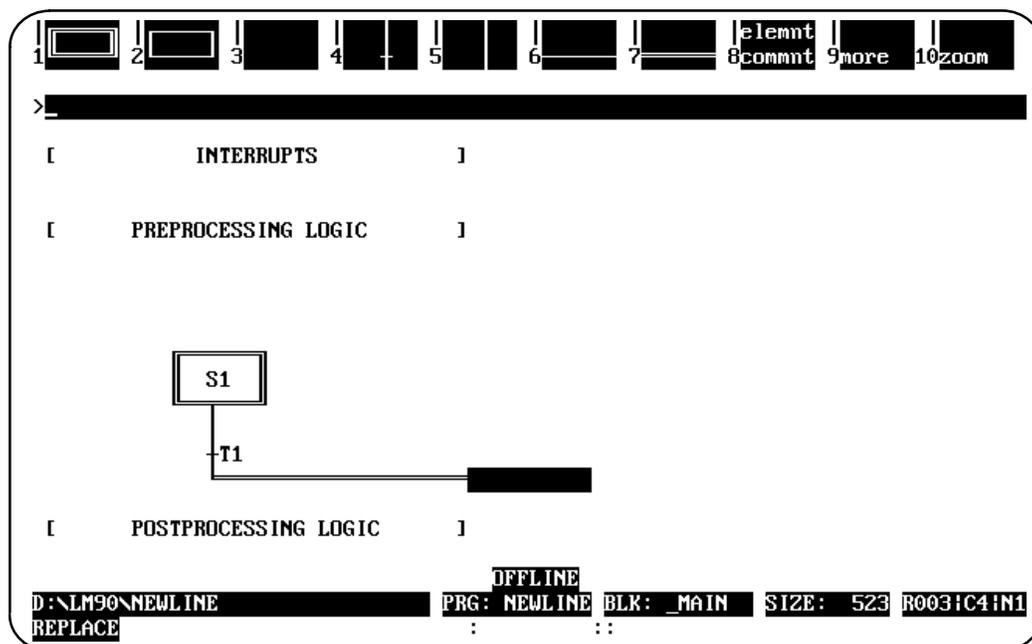
Remarque

La fonction Effacer élément (Appuyez sur **Autres, F9**, pour afficher cette touche) ou **ALT-D** peut également être utilisée pour supprimer n'importe quel élément GRAFCET. Pour plus d'informations sur la suppression d'un élément, se reporter à la page 3-33.

- Pour terminer le déplacement du branchement sous la transition T1, déplacez le curseur vers le bas sur la rangée de transition et appuyez sur **Ouvrir une divergence en ET (F7)**. Le curseur se déplace automatiquement vers la droite.



- Pour continuer le branchement, appuyez sur la touche de fonction **Ouvrir une divergence en ET (F7)**.

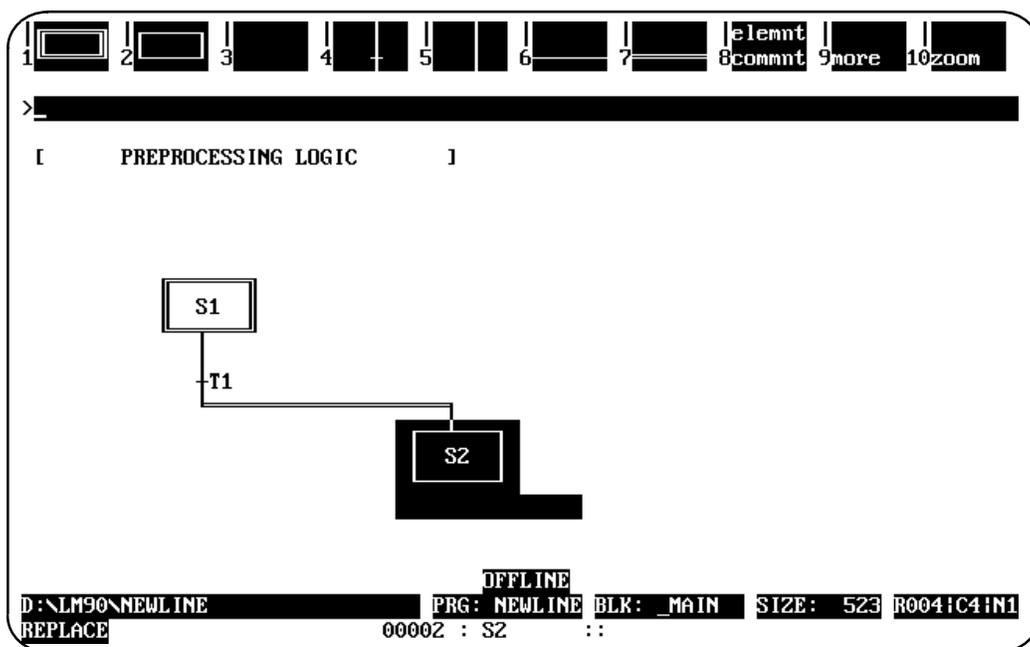


Vous êtes maintenant prêt à programmer les trois branches de la divergence en ET qui ont été créées.

Remarque

Vous n'êtes pas obligé d'entrer toutes les instructions dans un ordre descendant. Vous pouvez déplacer le curseur vers le bas et développer certaines des instructions qui vont suivre la convergence de ce branchement, par exemple, puis connecter les instructions après avoir terminé la série d'instructions simultanées.

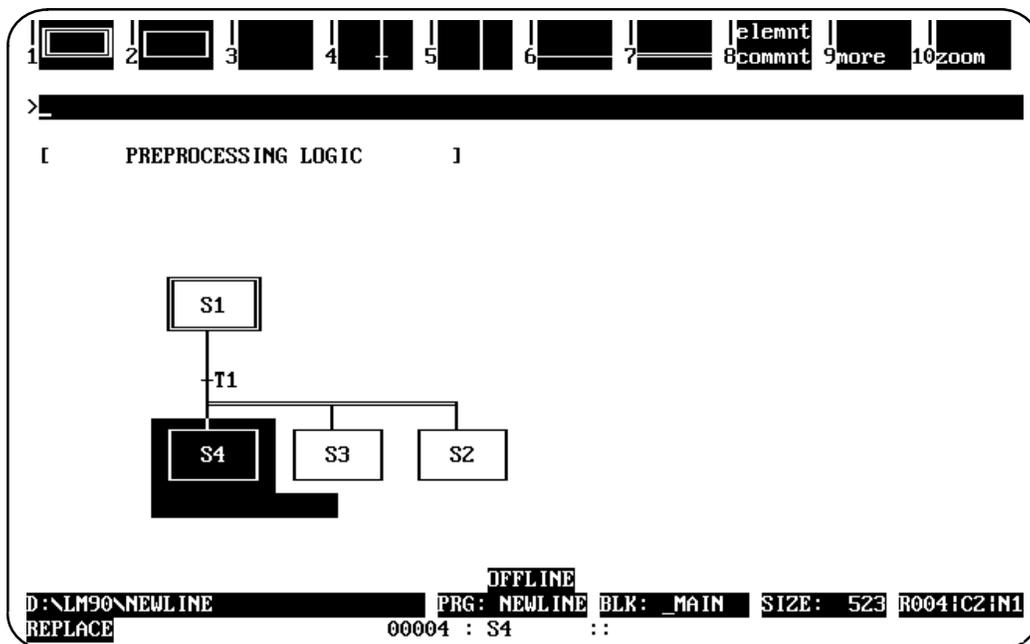
- Appuyez sur la touche de fonction **Etape (F2)** pour ajouter une étape sous le connecteur de branchement. Cette nouvelle étape est positionnée dans la branche la plus à droite contenue à l'intérieur du curseur (voir l'écran ci-dessous).



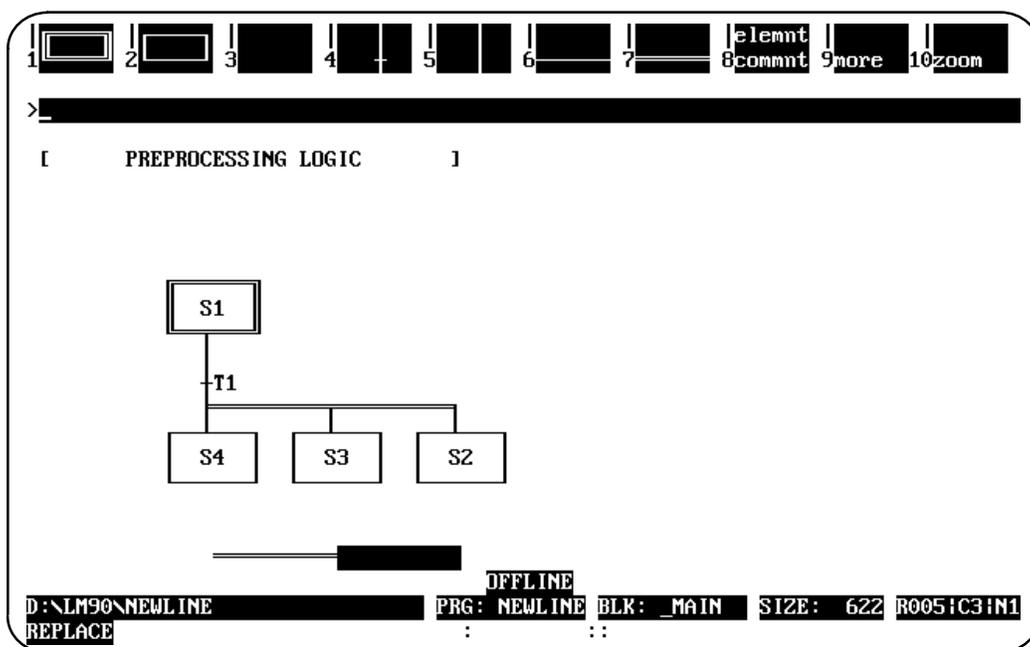
Remarque

Si vous appuyez sur la touche **Curseur bas** avant d'appuyer sur la touche de fonction **Etape (F2)**, l'étape sera entrée au même emplacement que si l'élément d'étape était ajouté à partir de la rangée précédente.

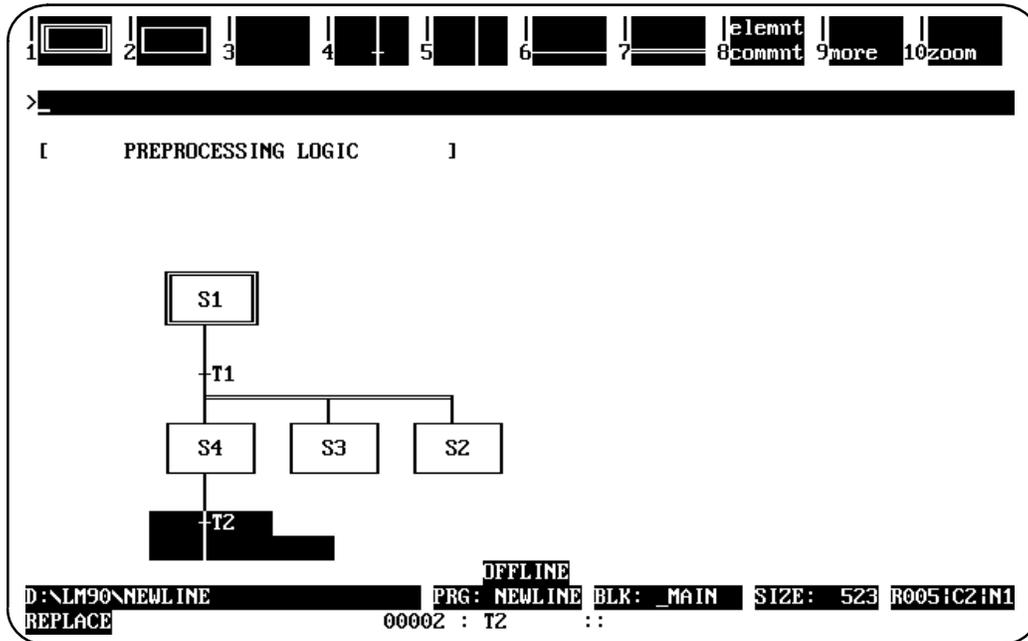
11. Le logiciel Logicmaster attribue le nom par défaut S2 à la seconde étape ajoutée à ce programme. Si vous déplacez le curseur vers la gauche et que vous réappuyez deux fois sur la touche **Etape (F2)**, des étapes seront affectées aux trois branches.



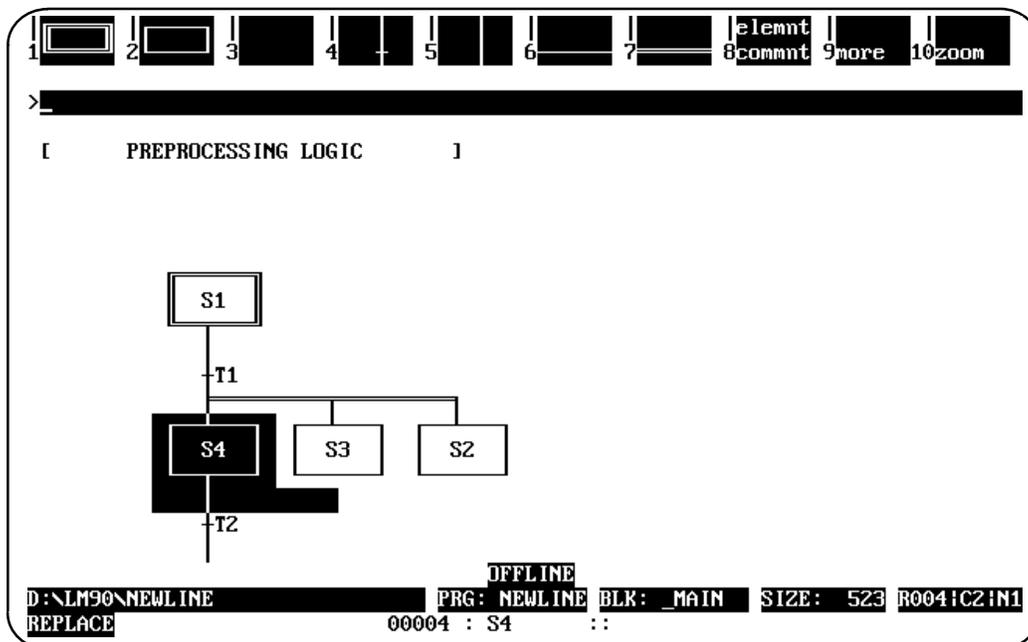
12. Si vous essayez de déplacer le curseur vers le bas et d'entrer une branche de divergence en ET inférieure pour fermer la séquence, vous verrez que le branchement est positionné sous l'emplacement de la transition.



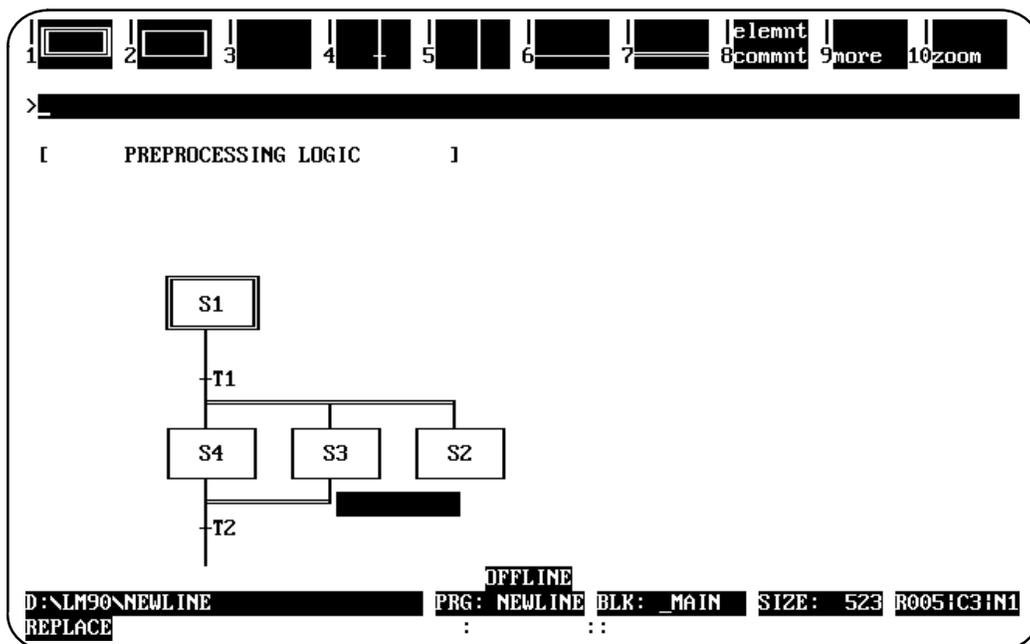
13. N'appuyez **pas** ici sur la touche **Branchement pour divergence en ET (F7)**. Vous pouvez entrer une transition à cette position en appuyant sur la touche **F4**.



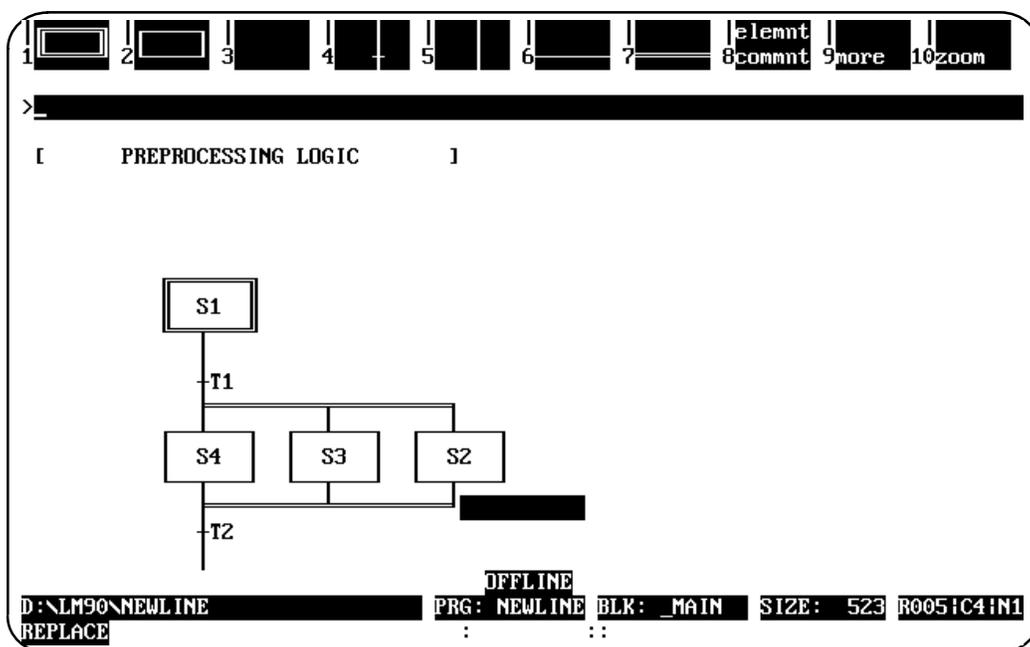
14. Appuyez sur la touche **flèche vers le haut** pour positionner le curseur à l'emplacement correct pour l'élément de branchement pour divergence en ET.



15. Appuyez sur la touche **Ouvrir une divergence en ET (F7)** pour entrer la première branche.



16. Réappuyez sur **Ouvrir une divergence en ET (F7)** pour ajouter une seconde branche afin de terminer la séquence.

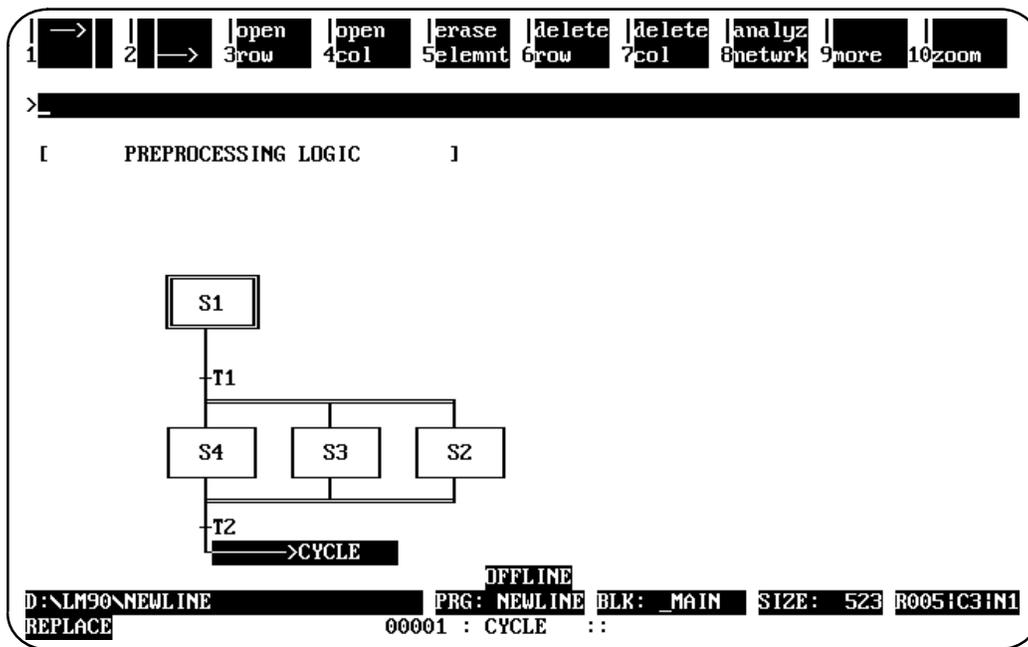


17. Pour terminer ce réseau GRAFCET, vous pouvez ajouter un saut d'étape source et un saut d'étape destination. Vous pouvez afficher les touches de fonctions correspondant à ces éléments GRAFCET en appuyant sur la touche **Autres (F9)**.

Remarque

Les instructions de saut d'étape ne sont autorisées **que** dans des rangées de transition de numéro impair. Les sauts d'étape source doivent être ajoutés **après les transitions** ; les sauts d'étape destination doivent être ajoutés **avant les étapes**. Ils sont entrés de la même manière que des branchements. L'emplacement du saut d'étape qui va être entré est indiqué par l'emplacement du curseur.

18. Pour entrer un saut d'étape source, positionnez d'abord le curseur sur la transition T2 en appuyant deux fois sur la touche **Curseur gauche** et une fois sur la touche **Curseur bas**. Puis entrez un nom (par exemple **CYCLE**) sur la ligne de commande, et appuyez sur la touche **Saut d'étape source (F2)**.

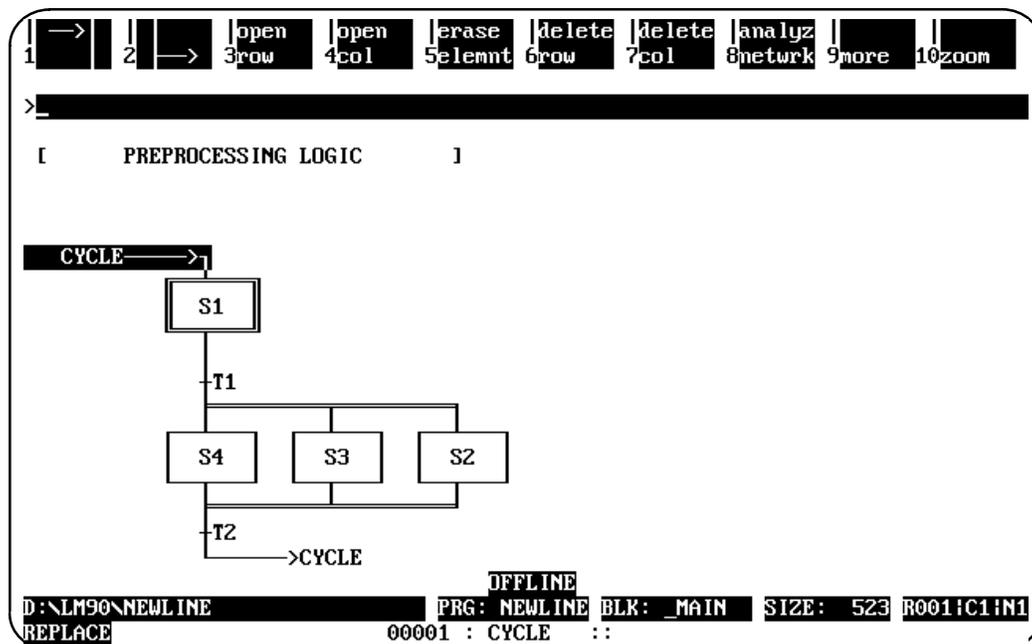


Si vous appuyez sur **F2** avant d'entrer un nom sur la ligne de commande, le logiciel Logicmaster insérera le saut d'étape et remplira le champ de nom par des points d'interrogation. Ceux-ci indiquent qu'un nom de saut d'étape est attendu. Vous ne pourrez pas déplacer le curseur hors de ce saut d'étape avant d'avoir entré un nom.

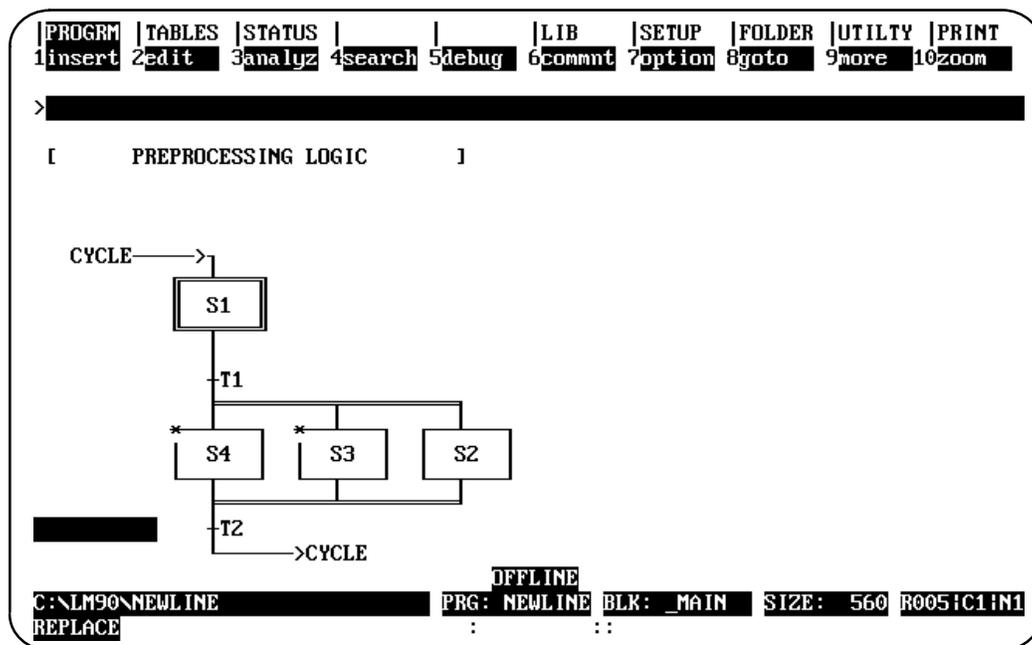
Remarque

Vous pouvez utiliser **ALT-D** pour supprimer à tout moment l'entrée de saut d'étape. Pour plus d'informations sur la suppression d'un élément, reportez-vous à la page 3-33.

19. Pour cet exemple, pour ajouter un saut d'étape destination au-dessus de l'étape initiale S1, positionnez le curseur dans le coin supérieur gauche du réseau en appuyant sur la touche **Home**. Puis entrez le nom **CYCLE** sur la ligne de commande et appuyez sur **Saut d'étape destination (F1)**.



20. La touche **Commentaire (F8)** donne accès au bloc de commentaire disponible pour chaque élément d'étape. Pour ajouter un commentaire, appuyez sur **Autres (F9)** quand les touches de fonctions de l'éditeur GRAFCET sont affichées dans le bandeau haut de l'écran. Puis appuyez sur **F8**. Pour plus d'informations sur les commentaires, référez-vous au § 4. *Commentaires*, à la page 3-50. Pour plus d'informations sur l'instruction COMMENT du logiciel Logicmaster, référez-vous au chapitre 3, *Edition de programmes*, dans le document GFK-0263 *Logicmaster 90-70 Programming Software User's Manual*.



Lorsqu'un commentaire est associé à un élément d'étape, un astérisque apparaît dans le coin supérieur gauche du graphique d'étape, ainsi qu'illustré pour les étapes S3 et S4 dans l'écran montré ci-dessus.

21. Pour entrer des instructions d'actions pour une transition, positionnez le curseur sur cette transition et appuyez sur la touche **Zoom (F10)**. Les instructions d'actions pour une transition doivent, au minimum, positionner sa variable de transition. Par exemple, les instructions d'actions correspondant à la transition T1 doivent contenir un segment similaire à celui illustré ci-dessous pour positionner la transition T1.

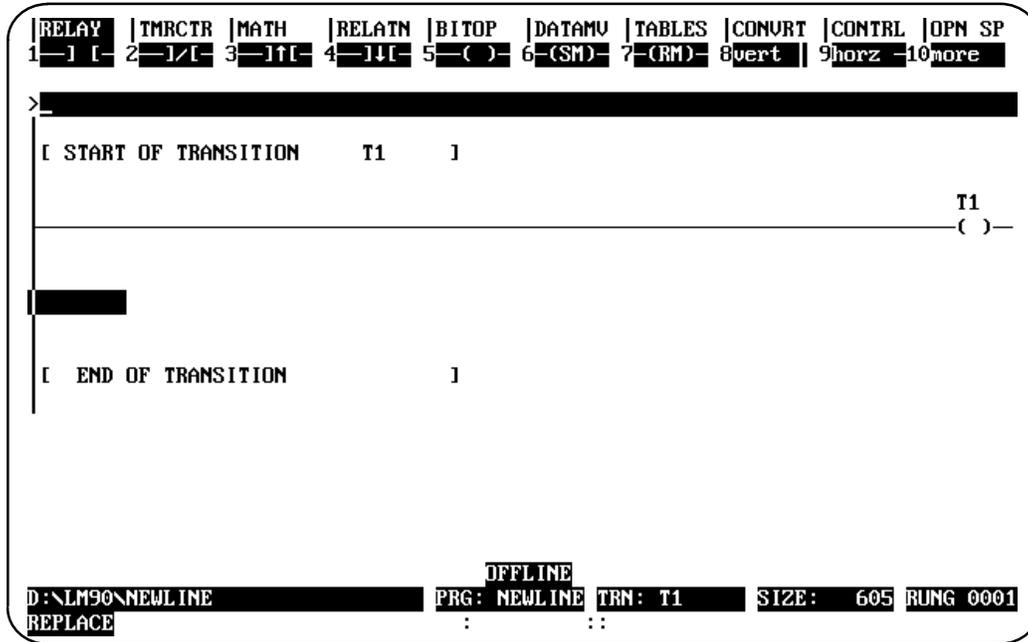


Pour ajouter cette ligne d'instructions de transition, positionnez le curseur sur la transition T1 et appuyez sur la touche **Zoom (F10)**. Le système affiche alors l'écran suivant.

```
PROGRAM | TABLES | STATUS | | | LIB | SETUP | FOLDER | UTILITY | PRINT
1insert 2edit 3modify 4search 5 | 6 | 7option 8goto 9more 10zoom
>
[ START OF TRANSITION T1 ]
[ END OF TRANSITION ]

OFFLINE
D:\LM90\NEWLINE PRG: NEWLINE TRN: T1 SIZE: 605 RUNG 0001
REPLACE :
```

Puis appuyez sur **Insérer un programme (F1)** pour entrer de nouvelles instructions (ou **Editer (F2)** pour éditer les instructions existantes pour cette transition). Remarquez que les touches propres au langage à relais apparaîtront lorsque vous aurez appuyé sur **Insérer un programme (F1)** ou **Editer (F2)**. Appuyez sur **F5** pour afficher le segment suivant.



Pour entrer des instructions d'actions pour une étape, procédez de la même manière que pour une transition. Positionnez le curseur sur l'étape et appuyez sur la touche **Zoom (F10)**. Puis appuyez sur **Insérer un programme (F1)** ou **Editer (F2)** pour entrer de nouvelles instructions en langage à relais ou éditer des instructions existantes.

2.5. SUPPRESSION D'UN ÉLÉMENT

La fonction Effacer élément (appuyez sur **Autres, F9**, pour afficher cette touche) ou **ALT-D** peut être utilisée pour supprimer n'importe quel élément GRAFCET, tel qu'une étape ou une transition.

Lorsque vous effacez une étape ou une transition, vous supprimez également les instructions correspondantes. Lorsque vous utilisez la fonction **Effacer élément (F9)** sur une étape avec des instructions d'actions ou une transition avec des instructions en langage à relais, vous devez confirmer l'action. La fonction Effacer élément supprime également toutes les entrées connexes dans la table des symboles. Le nom d'étape, de transition ou de saut d'étape peut ensuite être utilisé sur un autre élément.

Le remplacement d'un élément par un autre élément ayant un nom différent a le même effet que l'exécution d'une opération d'effacement d'un élément. Par exemple si vous appuyez sur la touche **Etape (F2)** avec un nom sur la ligne de commande, vous remplacerez (supprimerez) l'étape actuelle par l'étape nouvellement définie.

2.6. FONCTIONS D'OUVERTURE ET DE SUPPRESSION D'ESPACE

Les touches de fonctions **Ouvrir la rangée (F3)** et **Ouvrir la colonne (F4)** peuvent être utilisées pour étendre des séquences de branchements individuels ou déplacer un réseau entier.

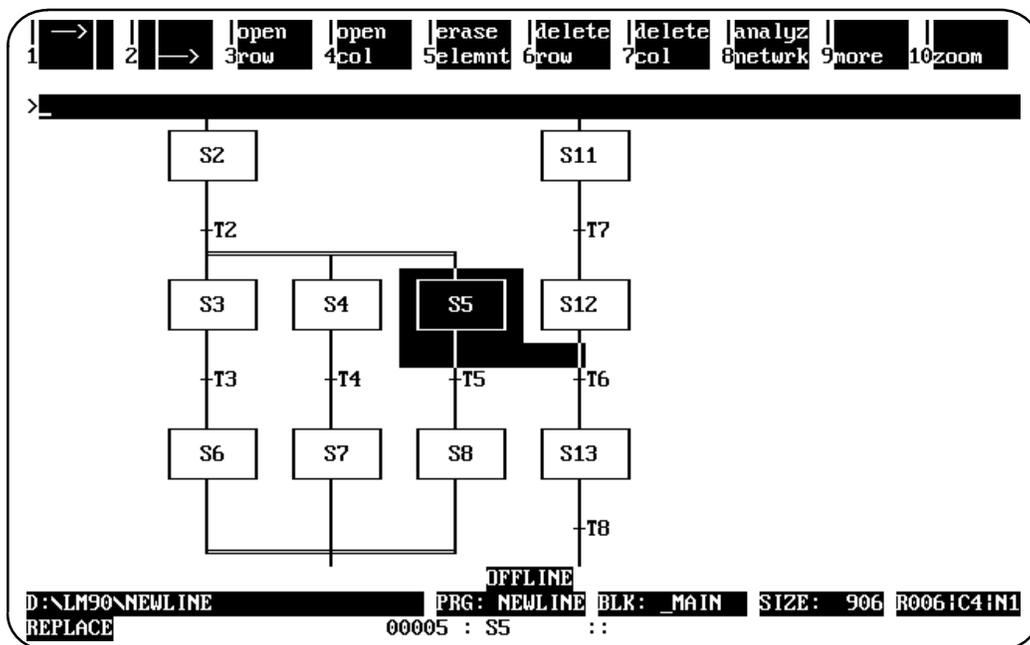
Les touches **Effacer la rangée (F6)** et **Effacer la colonne (F7)** peuvent être utilisées pour supprimer des espaces de séquences de branchements individuels ou déplacer un réseau entier.

2.6.1. Ouvrir la rangée

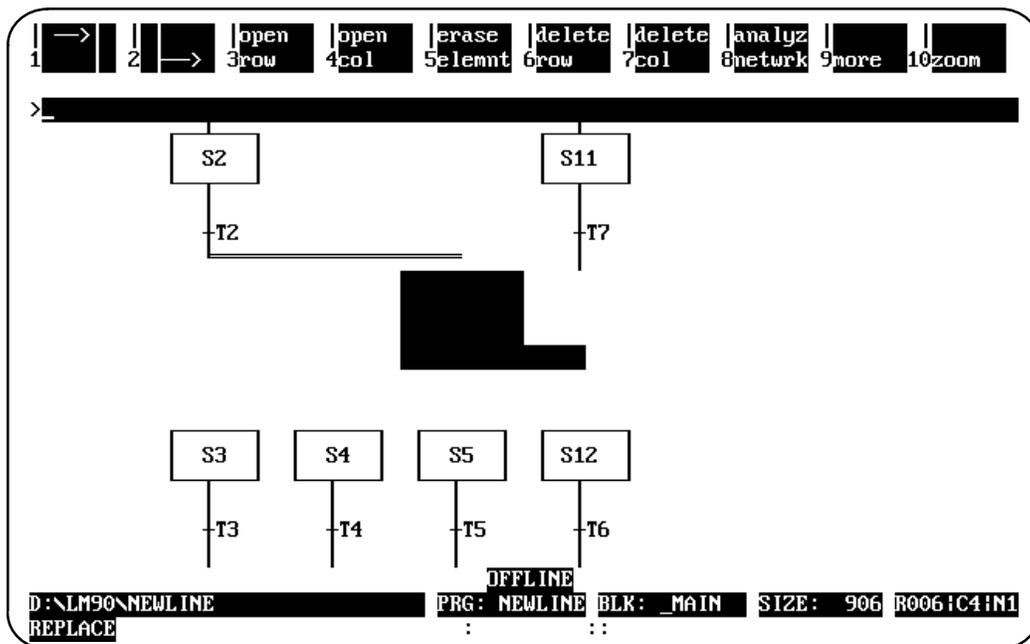
La fonction **Ouvrir la rangée (F3)** ajoute deux rangées immédiatement au-dessus de l'emplacement du curseur en déplaçant la rangée désignée par le curseur de deux rangées vers le bas. Le système ajoute deux rangées de manière à conserver les étapes sur des rangées paires et les transitions sur des rangées impaires. Le curseur reste à son emplacement de rangée d'origine. Les deux dernières rangées (les rangées 127 et 128) doivent être vides pour pouvoir utiliser la fonction Ouvrir la rangée.

Dans l'exemple ci-dessous, la rangée contenant l'étape S5 est déplacée de deux rangées vers le bas en positionnant le curseur sur l'étape S5 et en appuyant sur la touche **Ouvrir la rangée (F3)**.

Avant :



Après :

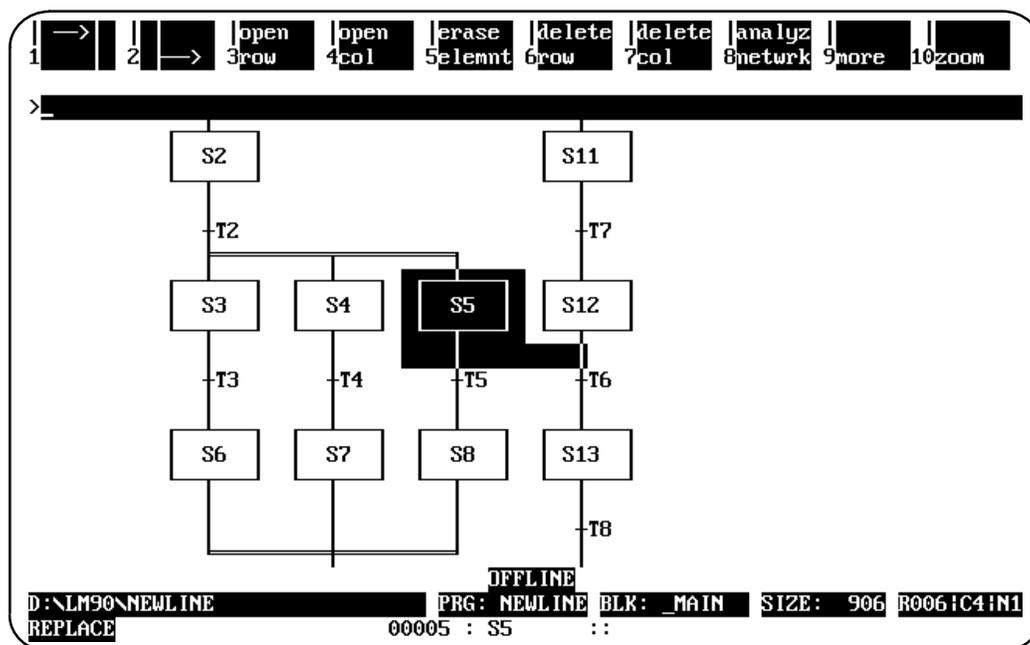


2.6.2. Ouvrir la colonne

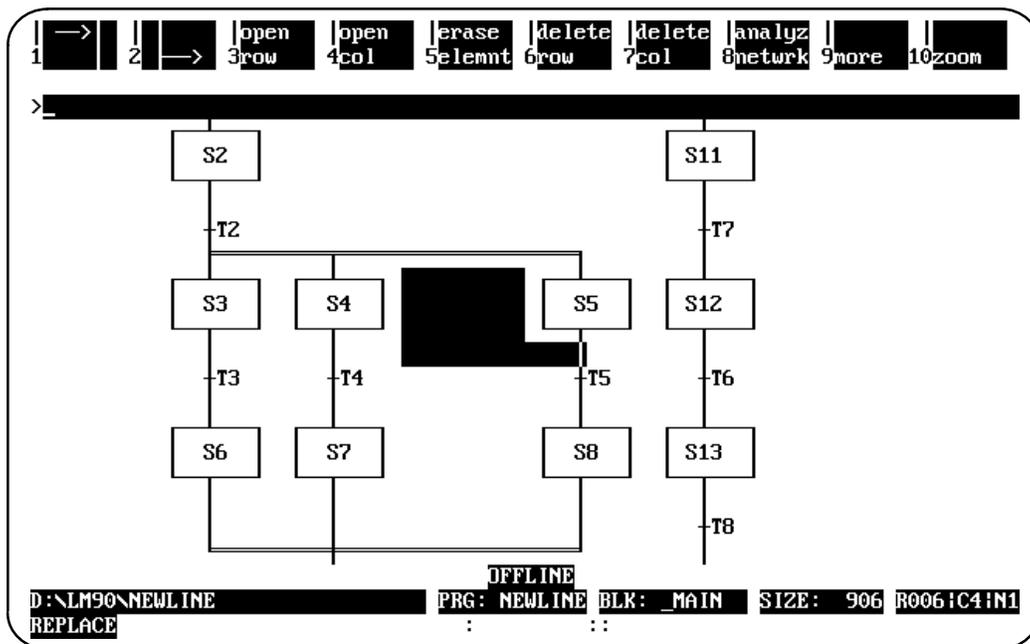
La fonction **Ouvrir la colonne (F4)** ajoute une colonne à la position du curseur en déplaçant la colonne désignée par le curseur d'une position vers la droite. Le curseur reste à son emplacement de colonne d'origine, et tous les branchements sont étendus de manière à préserver les structures de contrôle. Tous les éléments à la droite du curseur (rangées 1 à 128) seront déplacés. La dernière colonne doit être vide pour pouvoir utiliser la fonction Ouvrir la colonne.

Dans l'exemple suivant, la colonne contenant l'étape S5 est déplacée d'une colonne vers la droite en positionnant le curseur sur l'étape S5 et en appuyant sur la touche **Ouvrir la colonne (F4)**.

Avant :



Après :

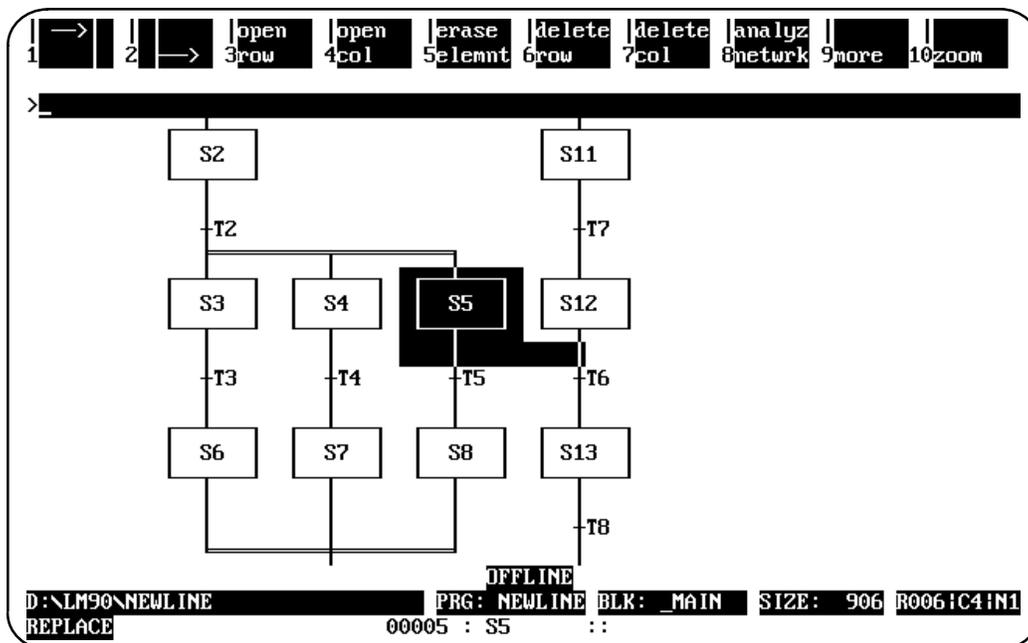


2.6.3. Effacer la rangée

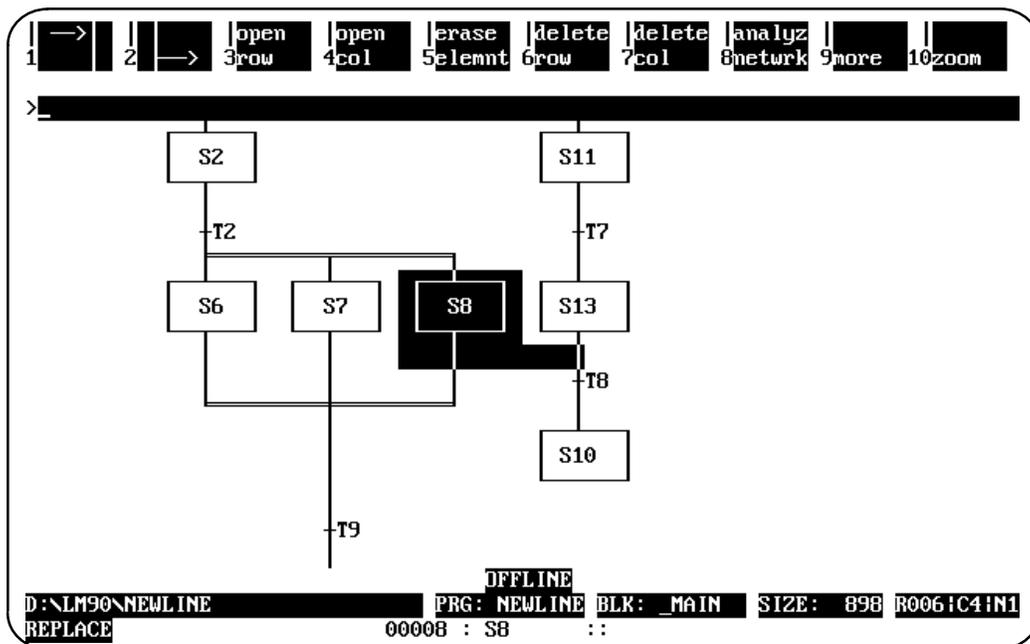
La fonction **Effacer la rangée (F6)** supprime la rangée à l'emplacement du curseur ainsi que la rangée suivante. Le curseur reste à son emplacement de rangée d'origine, et tous les branchements sont raccourcis de manière à préserver les structures de contrôle.

Dans l'exemple suivant, la rangée contenant l'étape S5 et la rangée qui suit sont supprimées en positionnant le curseur sur l'étape S5 et en appuyant sur la touche **Effacer la rangée (F6)**.

Avant :



Après :



2.6.4. Effacer la colonne

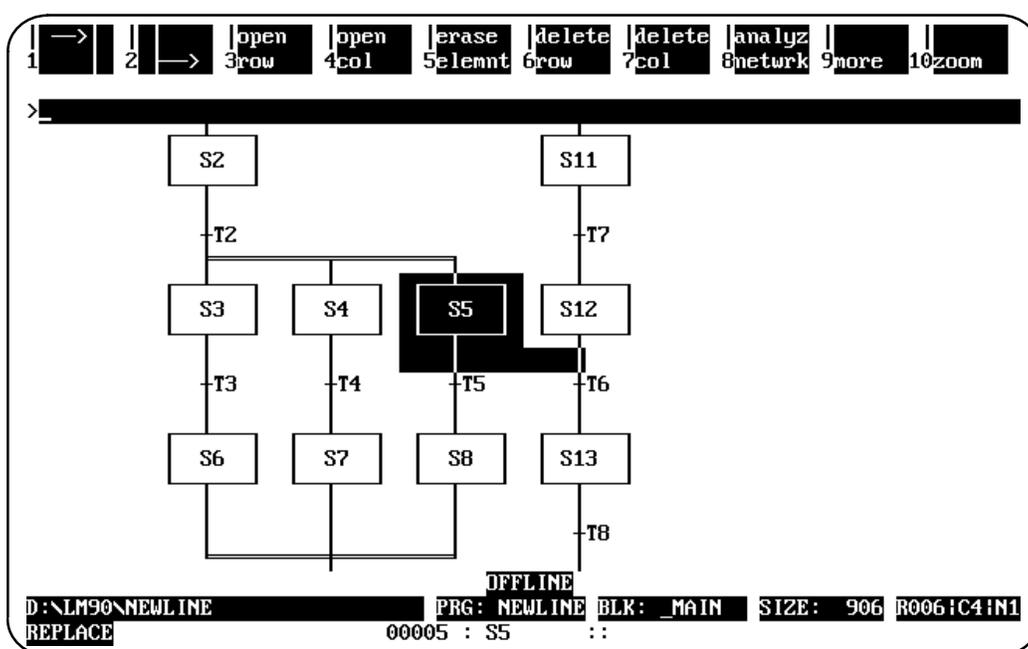
La fonction **Effacer la colonne (F7)** supprime la colonne à la position du curseur en déplaçant la colonne sur laquelle se trouve le curseur d'une position vers la gauche. Le curseur reste à son emplacement de colonne d'origine, et tous les branchements sont raccourcis de manière à préserver les structures de contrôle. Remarque

Remarque

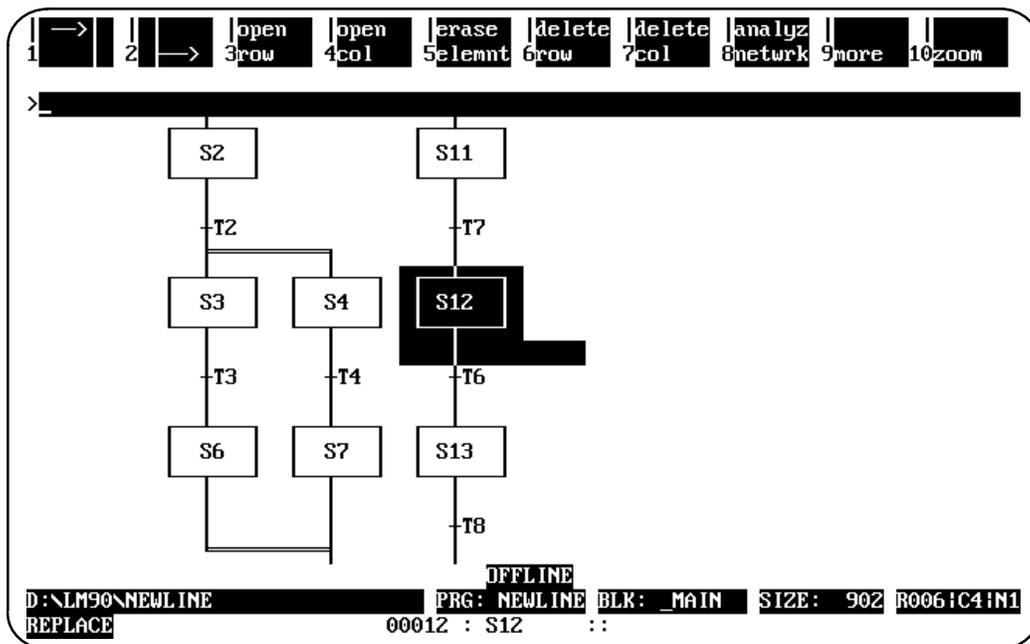
Tous les éléments de toute la colonne occupée par le curseur seront supprimés.

Dans l'exemple suivant, la colonne contenant l'étape S5 est supprimée en positionnant le curseur sur l'étape S5 et en appuyant sur la touche **Effacer la colonne (F7)**.

Avant :



Après :



2.7. SORTIE DU MODE INSÉRER UN PROGRAMME OU EDITER

Pour sortir du mode **INSERER UN PROGRAMME** ou **EDITER** et sauvegarder le réseau GRAFCET, appuyez sur la touche **Escape** ou sur la touche **Enter** (Acceptation). Le système affichera des messages d'erreur ou avertissements relatifs au réseau et positionnera le curseur à la première erreur rencontrée dans le réseau GRAFCET.

2.8. AFFICHAGE DES ERREURS DANS LE RÉSEAU

Vous pouvez identifier et afficher les erreurs dans le réseau GRAFCET par l'une des méthodes suivantes :

- Au niveau commande, vous pouvez appuyer sur la touche **Analyser (F3)** pour contrôler la validité du réseau GRAFCET et l'exécutabilité du bloc. Les trois premières erreurs détectées sont affichées, et le curseur est positionné sur la première erreur.
- En mode **INSERER UN PROGRAMME** ou **EDITER**, appuyez sur **Autres (F9)** puis sur **Analyser le réseau (F8)** pour contrôler la validité du réseau GRAFCET. La première erreur détectée est affichée, et le curseur positionné sur cette erreur. En mode **INSERER UN PROGRAMME** ou **EDITER**, le système vérifie également l'absence d'erreurs dans le réseau quand vous appuyez sur la touche **Escape** pour accepter le réseau.
- La fonction Analyser peut également être utilisée pour contrôler la validité d'une étape ou d'une transition lorsque vous utilisez des instructions en langage à relais dans une étape ou une transition. Ce contrôle est effectué automatiquement quand vous appuyez sur la touche **Escape** pour accepter les instructions d'actions. La première erreur rencontrée est affichée. Ce type d'erreur se produira par exemple, si vous essayez d'accepter une transition sans instructions.

Lorsque vous effectuez un zoom hors d'un bloc GRAFCET, le système effectue automatiquement une analyse complète du réseau GRAFCET dans le bloc. S'il détecte des erreurs, il affiche un message identifiant le réseau GRAFCET qui contient l'erreur. Vous n'êtes pas tenus de corriger des erreurs avant de sortir d'une session d'édition ou d'insertion. Cependant, un bloc GRAFCET contenant des erreurs n'est pas exécutable et ne peut pas être enregistré dans l'API.

2.9. ENREGISTREMENT DES MODIFICATIONS APPORTÉES À UN BLOC GRAFCET

Les modifications apportées à un bloc GRAFCET sont automatiquement enregistrées sur disque lorsque :

1. Vous effectuez un **zoom** dans des niveaux inférieurs du bloc.
2. Vous faites un **Escape** vers des niveaux supérieurs du bloc.
3. Vous appuyez sur la touche **Escape** pour sortir du mode **INSERER UN PROGRAMME** ou **EDITER**
4. Vous appuyez sur **ALT-U** pour forcer un enregistrement sur disque. (**ALT-U** peut être utilisé à tout moment.)
5. Vous appuyez sur **ALT-S** pour enregistrer dans un API (disponible uniquement en mode **EDITE BLOC**).

Selon le type de modifications apportées au bloc et si le bloc est toujours exécutable, vous pourrez enregistrer un bloc GRAFCET sur un API pendant que l'API est en mode **STOP** ou **RUN**. Les modifications autorisées sont les modifications des instructions en langage à relais associées aux étapes ou transitions, ou aux instructions de preprocessing ou postprocessing. En outre, le bloc ne doit pas contenir d'erreurs.

Quand des modifications sont effectuées en-ligne, le bandeau bas change de **PROGRAMME EGAL** à **EDITE BLOC**. Une fois l'enregistrement terminé, le bandeau bas reprend le libellé **PROGRAMME EGAL**.

Cependant, si vous modifiez la topologie d'un réseau GRAFCET, ou s'il y a des erreurs, le bloc ne peut pas être enregistré sur un API en fonctionnement (mode **RUN**) à partir de l'éditeur. Si vous modifiez le réseau, le bandeau bas affichera le libellé **PROGRAMME NON EGAL**. Si vous essayez alors d'enregistrer sur l'API avec la fonction **TRANSFERT EN MODE RUN** (RMS), l'API générera une erreur.

3. FONCTION DE RECHERCHE

La fonction Recherche permet de rechercher un élément dans un bloc programmé en langage graphe commande étape-transition. La touche Rechercher fonctionne de la même manière que dans un bloc en langage à relais. Pour plus d'informations sur la fonction de recherche, référez-vous au § 11, *Fonction de recherche*, dans le chapitre 3, Edition de programme, du document *GFK-0263 Logicmaster 90-70 Programming Software User's Manual*.

De plus, vous pouvez rechercher les éléments GRAFCET suivants, bien que les remplacements ne soient pas autorisés :

- Une étape ou un nom d'étape, <nom d'étape>.x, <nom d'étape>.t, <nom d'étape>.f.
- Une transition ou un nom de transition.
- Un nom de saut d'étape.
- Un élément en langage GRAFCET :
 - &istep.
 - &step
 - &trans
 - &selbrch
 - &simbrch
 - &verlnk
 - &srcnct
 - &dstnct

Une recherche locale opère sur toutes les instructions GRAFCET dans le bloc, même si elle est appelée à partir d'une étape ou transition particulière. Une recherche globale demandée à partir d'une topologie GRAFCET bouclera pour rechercher à partir du début du bloc et jusqu'au point de départ de la recherche. Lorsque la recherche est effectuée, elle commence par l'élément GRAFCET suivant dans l'ordre de parcours. La recherche continuera et bouclera depuis le début pour inclure le point d'origine et ses instructions sous-jacentes. Une recherche locale, cependant, commence à la position spécifiée et ne boucle pas.

Remarque

Une recherche arrière ne peut être exécutée que sur des blocs en langage à relais. Sur des instructions en langage GRAFCET, la recherche se fera toujours par ordre de parcours, même si vous sélectionnez une recherche arrière.

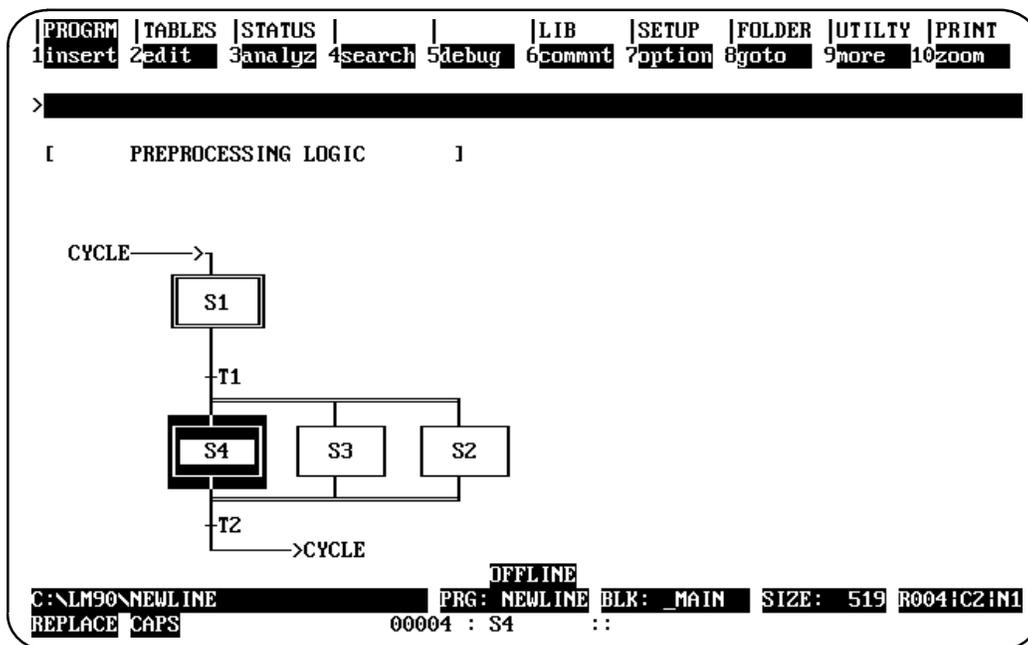
Dans des topologies exécutables, l'ordre de parcours commence par l'étape la plus en haut à gauche dans la topologie et se poursuit en fonction des branchements de la topologie, à partir de cette étape. Lors d'une divergence en OU ou en ET, les branches sont examinées de gauche à droite. Les sauts d'étape destination sont examinés immédiatement avant les étapes qu'ils précèdent. Les sauts d'étape source sont examinés immédiatement après les transitions qu'ils suivent. Dans des topologies non-exécutables, l'ordre de parcours suit l'ordre gauche-droite et de haut en bas des éléments GRAFCET, tels qu'ils apparaissent sur l'écran.

3.1. VALIDATION DE LA FONCTION DE RECHERCHE

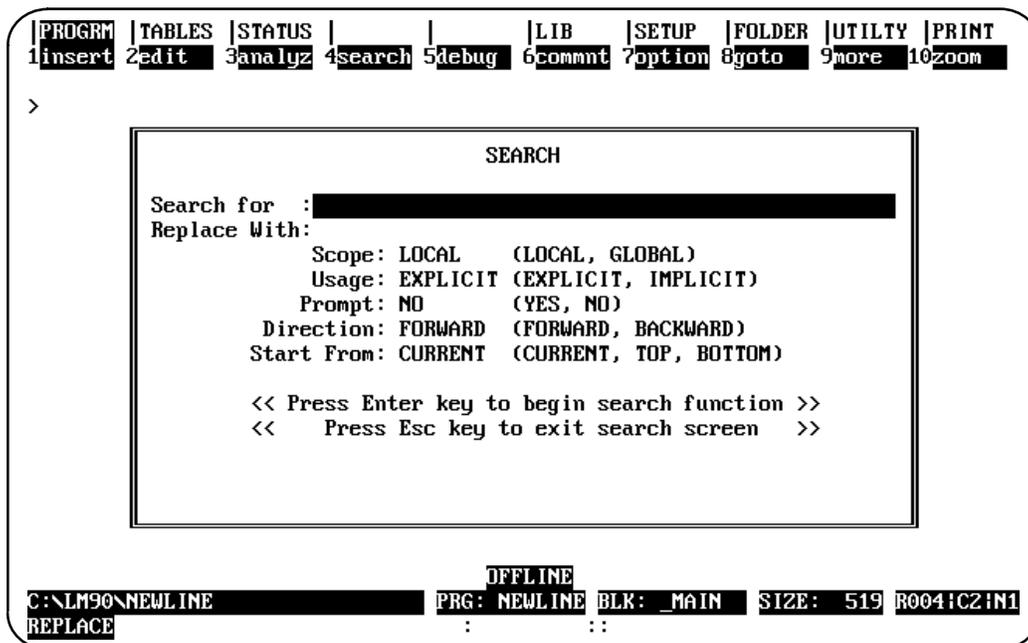
Vous accédez à la fonction de recherche en appuyant sur la touche **Rechercher (F4)** à partir des touches de fonctions de l'éditeur GRAFCET.

L'exemple suivant illustre l'utilisation de la fonction de recherche.

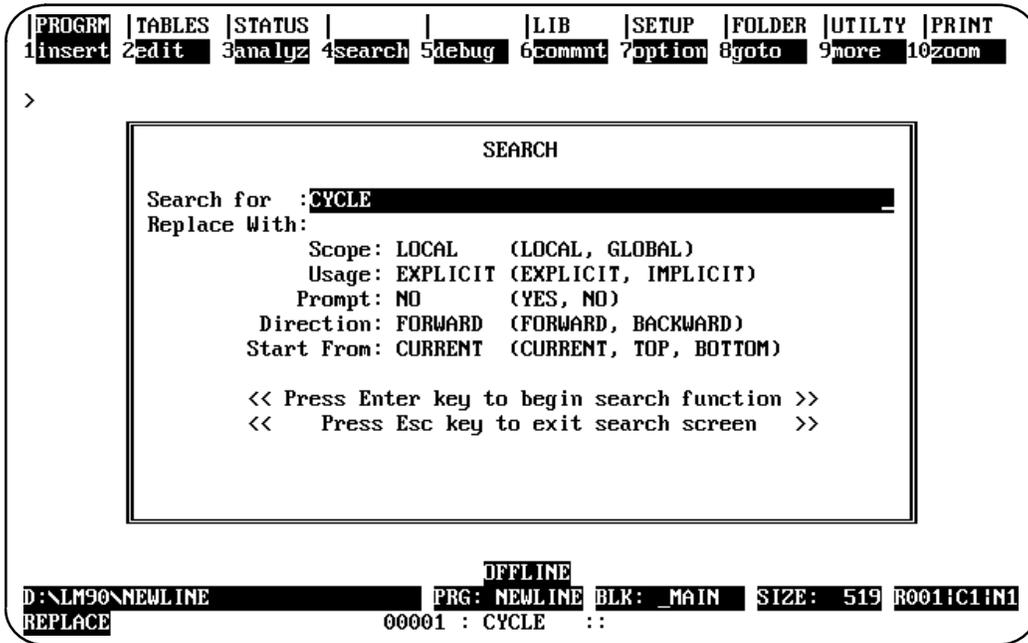
1. Positionnez le curseur sur l'étape S4.



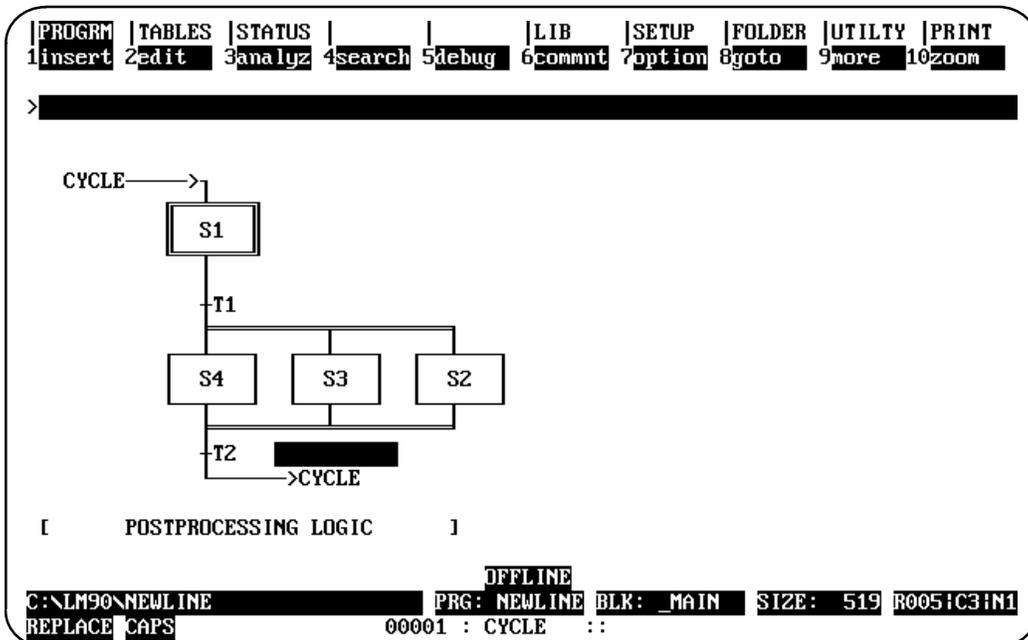
- Appuyez sur la touche **Rechercher** (**F4**) pour accéder à la fonction de recherche et afficher la fenêtre de recherche initiale.



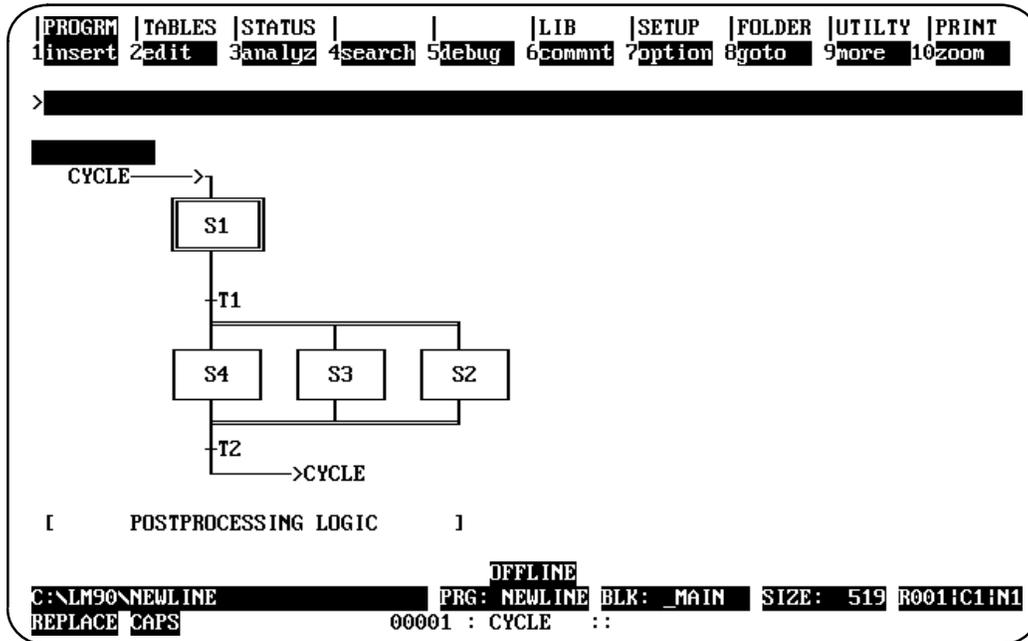
3. Pour cet exemple, notre cible de recherche sera le saut d'étape **CYCLE**. Entrez le mot **CYCLE** dans le champ *Recherche de*.



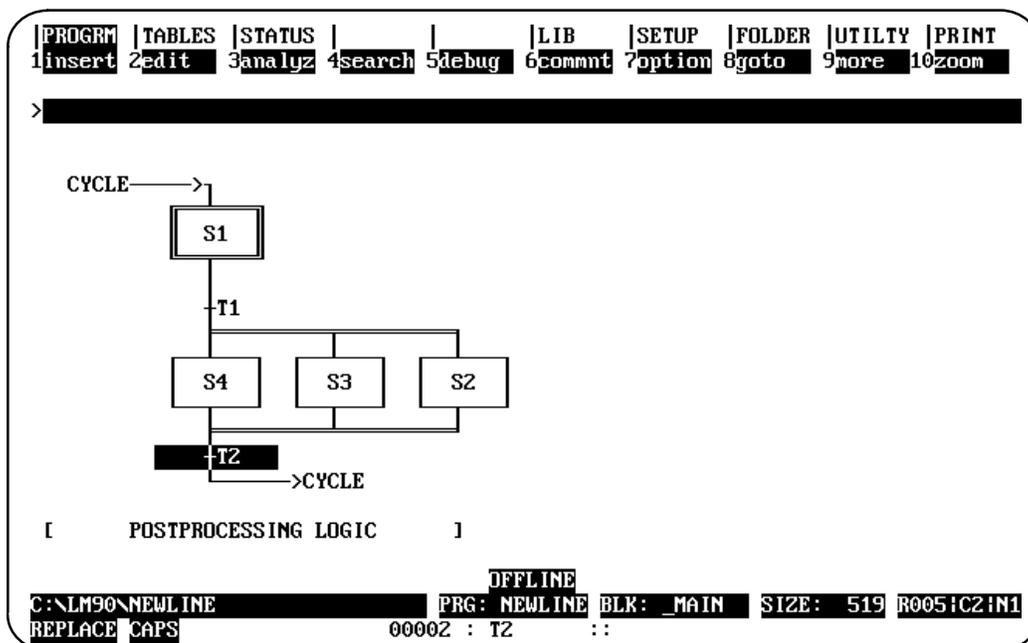
4. Appuyez sur la touche **Enter** pour commencer la recherche de la cible **CYCLE**. La cible est localisée dans le graphe commande étape-transition.



5. Réappuyez sur la touche **Rechercher** (F4) puis sur la touche **Enter** (ou appuyez sur **ALT-Rechercher** (F4)) pour chercher l'occurrence suivante du **CYCLE** cible.



6. Cette fois, n'appuyez que sur la touche **Rechercher** (F4). Puis préparez une recherche pour la transition **T2** en entrant **T2** dans le champ **Recherche de**. Lorsque vous appuyez sur la touche **Enter**, le système affiche l'écran ci-dessous.

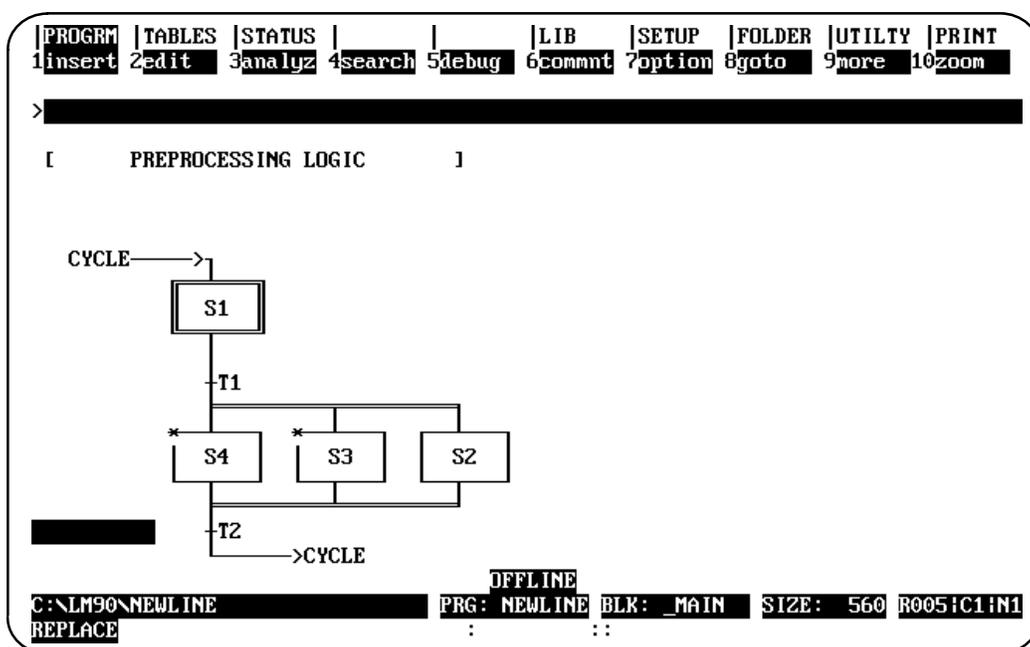


4. COMMENTAIRES

Les commentaires d'étape vous permettent de documenter des actions associées à une étape particulière. Ces commentaires d'étape peuvent être visualisés quand vous éditez le programme ou imprimés sur le listing du programme.

Pour appeler l'éditeur de texte de commentaire, positionnez le curseur sur l'étape et appuyez sur la touche **Commentaire (F6)**. L'éditeur de texte de commentaire fonctionne de la même manière que pour l'instruction COMMENT du langage à relais dans le progiciel de programmation Logicmaster 90. Pour plus d'informations sur l'instruction COMMENT du logiciel Logicmaster 90, référez-vous au chapitre 3, *Edition de programmes*, dans le document *GFK-0263 Logicmaster 90-70 Programming Software User's Manual*.

Lorsqu'un commentaire est associé à un élément d'étape, un astérisque apparaît dans le coin supérieur gauche du graphique d'étape, ainsi qu'illustré pour les étapes S3 et S4 dans l'écran montré ci-dessous.



5. OPTIONS ASSOCIÉES AUX BLOCS GRAFCET

Les options suivantes sont disponibles pour tous les blocs en langage GRAFCET dans un programme :

- La journalisation des défauts pour des temporisations d'étape hors limite.
- Les options de sélection du langage du bloc de programme.

5.1. JOURNALISATION DES DÉFAUTS POUR DES DURÉES D'ÉTAPE HORS LIMITES

Lorsqu'une limite temporelle minimale ou maximale est définie pour une étape, le système positionne à un bit d'étape <nom d'étape>.f à chaque fois que la durée de l'étape est hors de ces limites. De plus, si vous validez l'option de journalisation des défauts, l'API consignera un défaut dans la table des défauts de l'API lorsque cela se produit.

Pour valider l'option de journalisation des défauts, appuyez sur **RACK e/s (F1)** pour entrer la configuration d'E/S à partir du logiciel de configuration Logimaster 90.

1. Dans le logiciel de configuration, utilisez les touches de curseur pour positionner le curseur sur l'emplacement 1 du bac 0. L'emplacement 1 contient le module d'UC.

The screenshot shows the Logimaster 90 configuration software interface. At the top, there is a menu bar with options: RACK, COPY, REF UU, DELETE, UNDEL, CFGSEL, and others. Below the menu bar, there is a table with 9 columns representing different rack positions. The table is titled "PROGRAMMED CONFIGURATION". The first column is labeled "PS" and contains "PWR711" and "100W". The second column is labeled "1" and contains "CPU 771" and "256 KB". The third column is labeled "2" and contains "MDL 240" and "I AC 16". The fourth column is labeled "3" and contains "PCM 711" and "64 KB". The fifth column is labeled "4" and contains "BEM 731" and "Devices BUS1: 0". The sixth column is labeled "5" and contains "ADC 701" and "ADC". The seventh, eighth, and ninth columns are empty. At the bottom of the screen, there is a status bar with the text "OFFLINE", "D:\LM90\NEWLINE", "PRG: NEWLINE", and "CONFIG VALID".

PS	1	2	3	4	5	6	7	8	9
PWR711	CPU 771	MDL 240	PCM 711	BEM 731	ADC 701				
100W	256 KB	I AC 16	PCM	GBC1	ADC				
	Ref Adr %I00001		64 KB	Devices BUS1: 0					

OFFLINE
D:\LM90\NEWLINE PRG: NEWLINE CONFIG VALID
REPLACE

- Appuyez sur **Zoom (F10)** pour effectuer un zoom dans l'écran d'informations du processeur.

```

RACK 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10
cpu    |   |   |   |   |   |   |   | expbd |
>
SERIES 90-70 MODULE IN RACK 0 SLOT 1
SOFTWARE CONFIGURATION
SLOT 1 Catalog #: IC697CPU771 771 CPU EXPANDABLE MEM
CPU 771
256 KB
IOScan-Stop: NO      Baud Rate : 19200
Passwords : ENABLED Parity : ODD
Tmr Faults : DISABLED Stop Bits : 1
Data Bits : 8
Modem TT : 0 1/100 Second / Count
Idle Time : 10 Seconds
Mode : SNP
Watchdog Tmr: 200 msec
The parameters on next page are supported by PLC Rev 4.0 or higher.
If PLC Firmware version is less than 4.0, do not modify them.

<< More Config Data Exists; PgDn for Next Page, PgUp for Previous Page >>
OFFLINE
C:\LM90\NEWLINE PRG: NEWLINE CONFIG VALID
REPLACE
    
```

- Positionnez le curseur sur le champ **Tmr Faults**, et utilisez la touche Tab pour commuter la sélection du paramètre sur **VALIDE**. (La sélection par défaut pour ce paramètre est **DEVALIDE**.)

```

RACK 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10
cpu    |   |   |   |   |   |   |   | expbd |
>
SERIES 90-70 MODULE IN RACK 0 SLOT 1
SOFTWARE CONFIGURATION
SLOT 1 Catalog #: IC697CPU771 771 CPU EXPANDABLE MEM
CPU 771
256 KB
IOScan-Stop: NO      Baud Rate : 19200
Passwords : ENABLED Parity : ODD
Tmr Faults : ENABLED Stop Bits : 1
Data Bits : 8
Modem TT : 0 1/100 Second / Count
Idle Time : 10 Seconds
Mode : SNP
Watchdog Tmr: 200 msec
The parameters on next page are supported by PLC Rev 4.0 or higher.
If PLC Firmware version is less than 4.0, do not modify them.

<< More Config Data Exists; PgDn for Next Page, PgUp for Previous Page >>
OFFLINE
C:\LM90\NEWLINE PRG: NEWLINE CONFIG VALID
REPLACE
    
```

La modification est automatiquement enregistrée sur disque lorsque vous sortez de la configuration d'E/S (en appuyant sur la touche **Escape**) et sera enregistrée sur l'API à la prochaine opération d'enregistrement de la configuration de programme.

Pour plus d'informations sur la configuration des modules d'UC, référez-vous au § 2, *Configuration des modules d'UC*, dans le chapitre 11, *Configuration des E/S*, du document *GFK-0263 Logimaster 90-70 Programming Software User's Manual*.

5.2. CONFIGURATION DE PLUSIEURS LANGAGES

Si vous souhaitez que tous les blocs de programme soient programmés dans le même langage, à savoir le langage à relais ou le langage GRAFCET, vous pouvez définir l'option "langages multiples" à "langage unique". Dans ce cas, vous n'aurez pas besoin de sélectionner un langage lorsque vous entrerez la première fois un nouveau bloc.

Remarque

Même si vous décidez de programmer tous vos blocs en langage GRAFCET, vous devrez quand même utiliser le langage à relais pour les parties du bloc GRAFCET ne laissant pas le choix du langage (c'est-à-dire les actions de preprocessing, les actions de postprocessing, les instructions de transition, et les instructions d'actions d'étape).

Pour changer la configuration de langage de bloc :

1. Appuyez sur la touche **Options (F7)** à partir du premier niveau de l'éditeur GRAFCET ou Langage relais.

```

PROGRAM | TABLES | STATUS |   |   | LIB | SETUP | FOLDER | UTILITY | PRINT
1coilck 2ed opt 3   4   5   6   7   8   9   10
>

PROGRAM EDITOR SETUP OPTIONS

F1 ... Multiple Coil Use
F2 ... Editor Options

ID:   RUN/OUT EN   2ms SCAN MONITOR L4 ACC: WRITE LOGIC LOGIC EQUAL
C:\LM90\NEWLINE PRG: NEWLINE BLK: MAIN SIZE: 573 R001:C1:IN1
REPLACE 00001 : CYCLE  ::
    
```

2. Puis appuyez sur la touche **Options Editeur (F2)**.

```
PROGRAM | TABLES | STATUS |   |   | LIB | SETUP | FOLDER | UTILITY | PRINT
1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | 10      |
>
      E D I T O R   O P T I O N S

Automatically Insert References into Variable Declaration Table?   N (Y/N)
Automatically Create %U, %UR References, If No Reference Specified? N (Y/N)
Block Language Choices for New Blocks are :      SFC AND RLD

      << Press ENTER key to Change Setting >>
      <<   Press ESC Key to Exit   >>

ID:      RUN/OUT EN      2ms SCAN  MONITOR  L4 ACC: WRITE LOGIC  LOGIC EQUAL
C:\LM90\NEWLINE      PRG: NEWLINE  BLK: _MAIN  SIZE: 573  R001C1I1N1
REPLACE      00001 : CYCLE  ::
```

3. Placez le curseur sur la ligne :

Les choix de langages pour les nouveaux blocs sont :

et appuyez sur la touche **Tab** pour sélectionner un langage. Les choix possibles sont **SFC** et **RLD**, **RLD**, ou **SFC** (respectivement GRAFCET et langage Relais, langage Relais, ou GRAFCET).

Remarque

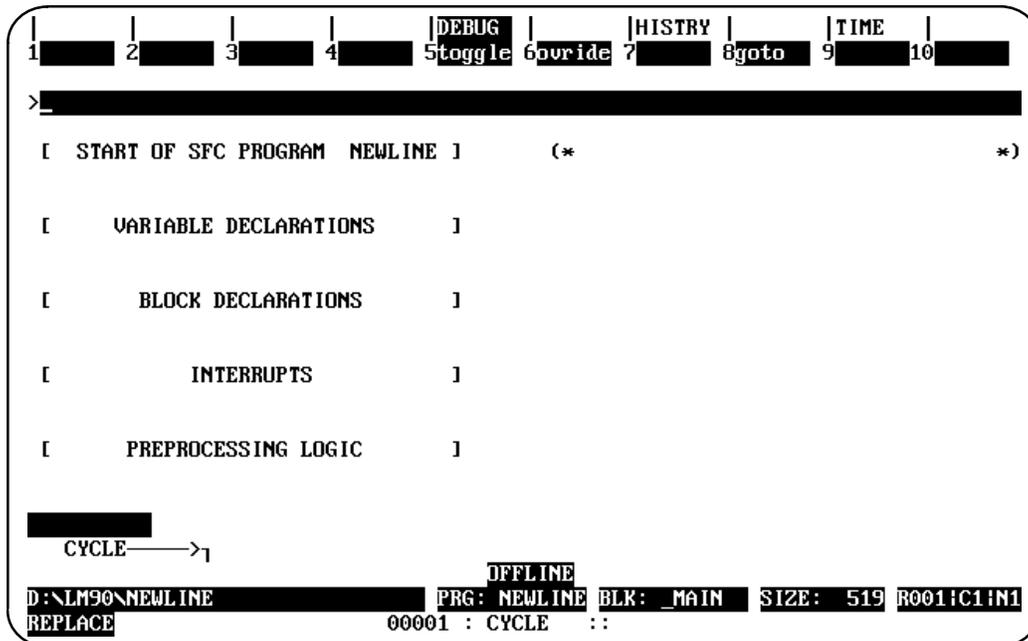
L'option langages multiples peut être changée aussi souvent que désiré. Votre sélection prend effet immédiatement ; cependant, elle n'affecte que le choix du langage pour les nouveaux blocs. Si vous sélectionnez par exemple l'option **RLD**, tous les blocs déjà programmés en langage GRAFCET resteront en GRAFCET lorsque vous y accédez par la fonction Zoom.

6. FONCTIONS DE MISE AU POINT

Les fonctions de mise au point suivantes peuvent vous aider à mettre au point un graphe commande étape-transition :

- Définition (avec forçage optionnel) des étapes à l'état actif ou inactif, indépendamment du flux logique.
- Définition (avec forçage optionnel) des transitions, indépendamment des conditions logiques.
- Suivi de l'historique d'évolution d'un bloc GRAFCET.
- Réglage de la base de temps du réseau.
- Réglage des temps d'exécution minimal et maximal pour une étape.
- Surveillance des défauts de temporisation d'étape.
- L'instruction SFC_RESET permet de réinitialiser le réseau en utilisant des instructions en langage relais.
- Mise au point sur un seul cycle (scrutation).
- Vous pouvez effectuer un zoom dans des étapes ou des transitions à partir de l'écran de mise au point.

Vous accédez aux touches de fonctions correspondant aux fonctions de mise au point en appuyant sur la touche **Debug (F5)** à partir du premier niveau de l'éditeur GRAFCET.

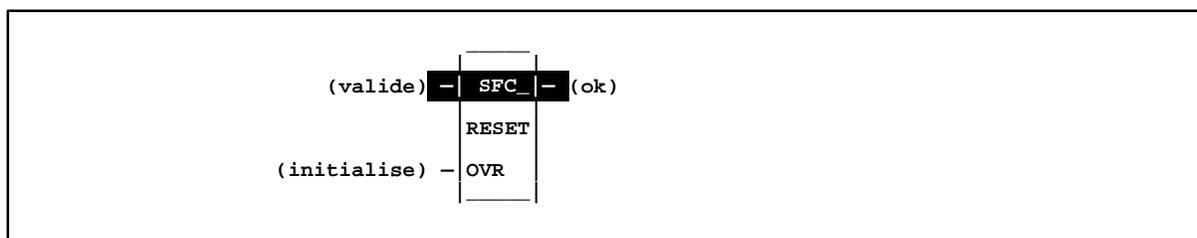


6.1. SFC_RESET

Vous pouvez également contrôler le mode d'exécution par des instructions en langage à relais en utilisant la fonction de réinitialisation GRAFCET **SFC_RESET**.

La fonction **SFC_RESET** peut être utilisée pour réinitialiser le réseau GRAFCET dans le bloc à l'étape initiale. Les blocs GRAFCET étant rémanents, vous devez utiliser la fonction **SFC_RESET** pour forcer un bloc GRAFCET à commencer à partir de l'étape initiale à la mise sous tension ou à partir d'une transition **STOP-TO-RUN**.

La fonction **SFC_RESET** a la forme suivante :



Les entrées “valide” et “initialise” sont utilisées pour contrôler l'exécution du bloc GRAFCET. “valide” est la seule entrée obligatoire. L'entrée OVR (forçage) est optionnelle. Si l'entrée “valide” est active, la fonction **SFC_RESET** commande au bloc GRAFCET de commencer à l'étape initiale. Les temporisateurs d'étape sont initialisés, et les bits d'étape <nom d'étape>.f mis à zéro. Si OVR est active, les forçages sont effacés.

Remarque

Les valeurs de <nom d'étape>.t, <nom d'étape>.f et <nom d'étape>.x sont effacées lorsque la fonction **SFC_RESET** est exécutée.

Lorsque le système reçoit une demande de réinitialisation, il l'envoie à l'interpréteur GRAFCET. Lorsque l'interpréteur GRAFCET est ensuite appelé pour faire évoluer le réseau GRAFCET dans le bloc, toutes les étapes actuellement actives sont désactivées. Toutes les instructions d'actions associées aux étapes actives sont alors exécutées sans flux validant. Puis le réseau GRAFCET passe à un état où son étape initiale est activée, les instructions activées sont exécutées.

Paramètres :

Paramètre	Description
valide	Lorsque la fonction est validée, l'opération est effectuée.
OVR	OVR efface les bits de forçage GRAFCET.

6.2. CHANGEMENT D'ÉTAT ET FORÇAGES

Si le logiciel Logicmaster 90 est **ONLINE** et **EGAL** à l'API, les transitions peuvent être basculées et forcées exactement comme des contacts et des bobines en langage à relais. Pour changer l'état d'une transition, positionnez le curseur sur la transition à modifier et appuyez sur la touche **Changement d'état (F5)** dans les touches de fonctions de mise au point, sur la touche de fonction **F12**, ou sur la touche “-” du clavier. Si la transition était précédemment active (ON), elle sera désactivée (OFF).

Remarque

L'effet de changement d'état d'une transition ne peut durer que pendant un cycle.
Le cycle suivant, la transition sera réévaluée et la valeur basculée sera remplacée.

Pour garder la transition à son état basculé, il faut d'abord la forcer en utilisant la touche de fonction **Forçage (F6)** dans les touches de fonctions de mise au point, la touche de fonction **F11**, ou la touche “*” du clavier, puis la basculer à l'état désiré.

Attention

Soyez prudent lorsque vous utilisez cette fonctionnalité pour conserver une transition dans son état basculé. Le programme GRAFCET peut être placé dans un état non valide, que le logiciel Logicmaster 90 aurait normalement empêché en imposant des restrictions sur le programme GRAFCET.

Lorsqu'une transition est forcée, son nom clignote pour indiquer cet état de forçage.

Les fonctions de changement d'état et de forçage peuvent être utilisées pour contrôler la branche d'exécution d'un réseau GRAFCET. Par exemple, pour désactiver une branche d'une divergence en OU tout en laissant les autres branches s'exécuter normalement :

1. Forcer la transition en-tête de la branche à désactiver.
2. Basculer la transition forcée à l'état OFF.

A l'entrée de la divergence en OU, toutes les branches, à l'exception de la branche désactivée, seront exécutées normalement. Dans une série similaire d'étapes et de transitions, vous pouvez désactiver toutes les branches de la divergence en OU sauf une.

6.3. HISTORIQUE D'ÉVOLUTION

La touche de fonction **Historique (Shift-F7)** vous permet de visualiser et d'enregistrer un historique d'évolution du bloc GRAFCET en cours. Il n'est pas nécessaire que l'API soit en fonctionnement pour activer la fonction d'historique d'évolution, mais il devra être en fonctionnement pour les mises à jour. Les informations indiquant si l'historique d'évolution du bloc GRAFCET doit être enregistré sont enregistrées **uniquement** dans l'API. Elles ne sont **pas** enregistrées dans le dossier en cours avec les autres informations relatives à ce bloc.

Pour afficher l'écran suivant, appuyez sur la touche **Historique (Shift-F7)** à partir des touches de fonctions de mise au point pour le bloc désiré.

```

1 On      2 Off    3 Clear  4          5 DEBUG  6          7 HISTRY  8          9 TIME   10
>
SFC EVOLUTION HISTORY FOR BLOCK _MAIN IS ON

  TIME  NTKW ACTIVE STEPS
-----
1 00:02:03 1 S1
2 00:02:03 1 -S50 +S49 S48
3 00:01:01 1 S50 +S49 S48
4 00:01:00 1 S50 S49 S48

ID:      RUN/OUT EN  2ms SCAN  ONLINE  L4 ACC: WRITE LOGIC  LOGIC EQUAL
C:\LM90\NEWLINE   PRG: NEWLINE  BLK: _MAIN  SIZE: 573 R005 C1 IN1
REPLACE      :      ::

```

Pour permettre la collecte de l'historique d'évolution du bloc GRAFCET actuel, appuyez sur **On (F1)**. Si la fonction est validée, le système enregistre et affiche un historique des huit dernières évolutions du bloc GRAFCET. L'évolution la plus récente est d'abord affichée (autrement dit, dans l'exemple d'écran ci-dessus, l'entrée 4 est antérieure à l'entrée 1). Chaque évolution est horodatée. Chaque étape active pour cette évolution est enregistrée.

Pour désactiver la collecte de l'historique d'évolution, appuyez sur **Off (F2)**. Pour effacer l'historique d'évolution, appuyez sur **Efface (F3)**.

Si une étape dépasse sa durée maximale ou n'atteint pas sa durée minimale, elle est marquée dans l'historique d'évolution par un "+" ou un "-", respectivement. Une étape dépassant son temporisateur maximal est une évolution distincte du bloc GRAFCET et est listée comme une entrée distincte dans le journal d'historique. Une étape qui n'atteint pas sa durée minimale sera consignée avec la (ou les) prochaine(s) étape(s) qui devient active. En effet, une étape qui dépasse son temporisateur maximal est consignée au moment du dépassement du temporisateur maximal. La condition de durée minimale n'est pas évaluée avant que l'étape ne devienne inactive et avant que l'étape(s) suivante ne devienne active.

Par exemple, dans l'écran précédent, la troisième entrée enregistre le fait que l'étape S49 a dépassé sa durée maximale à 1h01. La deuxième entrée a été générée à 2h03 lorsque le réseau GRAFCET a évolué des étapes S50, S49 et S48 à l'étape S1. A ce point, S50 n'avait pas atteint sa durée minimale, et le système a généré la deuxième entrée pour enregistrer ce fait.

L'API doit disposer d'une quantité fixe de mémoire utilisateur pour chaque bloc GRAFCET pour supporter cette fonction. La mémoire est utilisée pour stocker les informations d'évolution.

Attention

Une fois la fonction activée, l'historique est enregistré continuellement dans l'API jusqu'à ce que la fonction soit désactivée en appuyant sur Off (F2), ou qu'un programme soit enregistré dans l'API en mode STOP. Si vous sortez d'un bloc GRAFCET dont l'historique d'évolution est enregistré, la cadence de scrutation de l'API sera affectée à moins que vous ne désactiviez l'historique d'évolution avant de sortir du bloc GRAFCET.

6.4. VERS..

La fonction Vers.. vous permet de vous placer rapidement sur une étape, transition ou saut d'étape spécifique dans un réseau GRAFCET. Pour utiliser cette fonction, vous devez d'abord être positionné dans le réseau GRAFCET. Entrez le nom de l'étape, transition ou saut d'étape désiré sur la ligne de commande, et appuyez sur la touche **Vers.. (F8)** disponible dans les touches de fonctions de mise au point.

Si vous appuyez sur **Vers.. (F8)** sans entrer de nom sur la ligne de commande et que la console de programmation est **ONLINE** et **EGAL** à l'API, le curseur avancera à l'étape active suivante dans le réseau GRAFCET.

6.5. RÉGLAGE DE LA BASE DE TEMPS ET DES LIMITES DE DURÉE D'ÉTAPE POUR UN BLOC GRAFCET

Pour utiliser cette fonction pour un bloc, appuyez sur la touche **Durée (Shift-F9)** à partir des touches de fonctions de mise au point. L'écran ci-dessous apparaît :

```

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|
>

TIME BASE FOR SFC NETWORK 1 IN BLOCK _MAIN : 0.1 SECONDS

STEP      MINIMUM TIME    MAXIMUM TIME    ELAPSED TIME    FAULT
-----
S1        NONE            NONE            0.0             -
S4        3.00            4.00            0.0             -
S3        NONE            NONE            0.0             -
S2        NONE            NONE            0.0             -

<< Use the TAB key to cycle through time base values >>
<< Press ALT-A or ESC to exit >>

ID:      RUN/OUT EN  2ms SCAN  ONLINE  L4 ACC: WRITE LOGIC  LOGIC EQUAL
C:\LM90\NEWLINE  PRG: NEWLINE  BLK: _MAIN  SIZE: 573 R0051C1IN1
REPLACE  :      ::
    
```

Le système affiche le champ **Base de temps** du premier réseau GRAFCET dans le bloc, suivi d'une liste des étapes initiale(s) et normale(s) dans le réseau. Vous pouvez utiliser les touches de curseur pour parcourir cet écran. Puis positionnez le curseur sur les différents champs **Étape Temps minimum etc.** pour définir leurs valeurs.

Lorsque le logiciel Logicmaster 90 est connecté et **EGAL** à l'API, les temps d'étape écoulés et les étapes actives sont affichés sur cet écran. Les temps d'étape écoulés sont indiqués dans l'unité de la base de temps du réseau GRAFCET. La dernière colonne de l'écran indique si la durée d'étape maximale a été dépassée (indiqué par le signe "+") ou si la durée d'étape minimale n'a pas été atteinte (indiquée par un signe "-"). Le nom de l'étape active est affiché en vidéo inversée.

6.5.1. Réglage de la base de temps

Vous pouvez utiliser l'écran ci-avant pour régler la base de temps pour le réseau GRAFCET dans le bloc graphe commande étape-transition. La base de temps est utilisée pour tous les temporisateurs d'étape dans le réseau GRAFCET. La valeur de la base de temps peut être 1 seconde, 0,1 seconde (par défaut), ou 0,01 seconde. Pour sélectionner une valeur pour la base de temps, positionnez le curseur sur le champ **Base de temps**, et appuyez à plusieurs reprises sur la touche **Tab** jusqu'à l'affichage de la base de temps désirée. Vous pouvez également régler la base de temps en entrant la valeur et en appuyant sur la touche **Enter**.

La modification de la base de temps du réseau doit se faire **OFFLINE**, que le logiciel Logicmaster 90 soit ou non connecté à un API. La nouvelle base de temps est enregistrée sur disque lorsque vous quittez cet écran et enregistrée sur l'API dès qu'un **TRANSFERT EN MODE STOP** de tout le programme est exécuté.

Remarque

Changer la base de temps est considéré comme une modification du réseau. Un changement de base de temps entraîne donc l'état **PROGRAMME NON EGAL** et nécessite que l'API soit en mode **STOP** pour pouvoir enregistrer le dossier.

6.5.2. Réglages des limites de durée d'étape

Vous pouvez également définir des durées d'étape minimale et maximale pour chaque étape initiale ou normale en positionnant le curseur sur le champ de durée de l'étape considérée, en entrant une limite de durée d'étape puis en appuyant sur la touche **Enter**.

Les limites de durée doivent être exprimées comme des nombres décimaux dans la plage 0,1 à 32767, ou par la valeur par défaut spéciale **NONE**, qui indique qu'il n'y a pas de limite de durée minimale ou maximale sur l'étape.

Toutes les valeurs de durée sont automatiquement adaptées pour être dans la plage autorisée par la base de temps. Par exemple, si la base de temps est 0,01 seconde et que vous entrez une durée d'étape maximale de 400 secondes, la valeur sera automatiquement définie à la valeur maximale possible de 327,67 secondes. Si la base de temps est 1 seconde et que vous entre une durée minimale de 1,5 secondes, celle-ci sera changée à 2 secondes.

Remarque

Si vous changez la base de temps du réseau après avoir spécifié des durées d'étape minimale et maximale, ces dernières seront automatiquement ajustées dans la plage de la nouvelle base de temps.

Si l'API est **EGAL** à la console de programmation, les durées d'étape minimale et maximale peuvent être modifiées **ONLINE**. Les nouvelles durées d'étape sont envoyées immédiatement à l'API ; cependant les durées d'étape modifiées ne sont pas appliquées à une étape active et ne sont prises en compte que lorsque cette étape devient inactive. Les modifications sont enregistrées sur disque lorsque vous quittez cet écran et enregistrées sur l'API dès qu'un **TRANSFERT EN MODE STOP** de tout le programme est exécuté.

Remarque

Si vous voulez que les défauts de base de temps apparaissent dans la table des défauts, vous devez valider le paramètre **Tmr Faults** dans la configuration de l'UC. Pour plus d'informations, référez-vous au § 5, *Options associées aux blocs GRAFCET*.

6.6. MISE AU POINT EN MODE MONO-CYCLE

La mise au point en mode mono-cycle (scrutation) vous permet d'arrêter l'API pour visualiser ou surveiller le programme après un cycle en appuyant sur **ALT-G**. ALT-G peut être utilisé pendant que vous affichez le programme, les tables de référence, la table des défauts de l'API, ou la table des défauts d'E/S. Pour utiliser cette fonction, la console de programmation doit être en mode **ONLINE**, et l'API doit être en mode **STOP/SCRUTATION**. Si l'API est en mode **RUN**, appuyez sur **ALT-R** pour basculer le mode de l'API sur **STOP**, ou sélectionnez le mode **STOP** dans l'écran Run /Stop API en utilisant les fonctions d'état.

Pour effectuer une première scrutation, tapez **0** sur la ligne de commande avant d'appuyer sur **ALT-G**. Si la ligne de commande est vide ou que vous entrez **1** sur la ligne de commande, le système effectue une mise au point en mode mono-cycle sur une autre scrutation que la première.

Lorsque l'API exécute des cycles en mode mono-cycle, les temporisateurs ne sont pas incrémentés. Vous devrez forcer une valeur dans le registre CG du temporisateur pour simuler un dépassement de temporisation.

7. FONCTIONS UTILITAIRES

7.1. TRANSFERT DE L'API VERS LA CONSOLE DE PROGRAMMATION

Vous pouvez charger des programmes contenant des blocs GRAFCET de l'API vers le logiciel Logimaster 90 exactement comme des programmes sans blocs GRAFCET. Pour plus d'informations sur la fonction de transfert, référez-vous au chapitre 9, *Utilitaires*, dans le document *GFK-0263 Logimaster 90-70 Programming Software User's Manual*.

7.2. ENREGISTREMENT DANS L'API DE LA CONSOLE DE PROGRAMMATION

La fonction d'enregistrement est utilisée pour copier des instructions de programme, des données de configuration et/ou des tables de références de la console de programmation vers l'API. (Le logiciel de configuration ne permet d'enregistrer que des données de configuration).

7.2.1. Transfert en mode Stop

Le **TRANSFERT EN MODE STOP** d'un programme qui contient des blocs GRAFCET est similaire à l'enregistrement d'un programme sans blocs GRAFCET. Pour plus d'informations sur la fonction d'enregistrement, référez-vous au chapitre 9, *Utilitaires*, dans le document *GFK-0263 Logimaster 90-70 Programming Software User's Manual*.

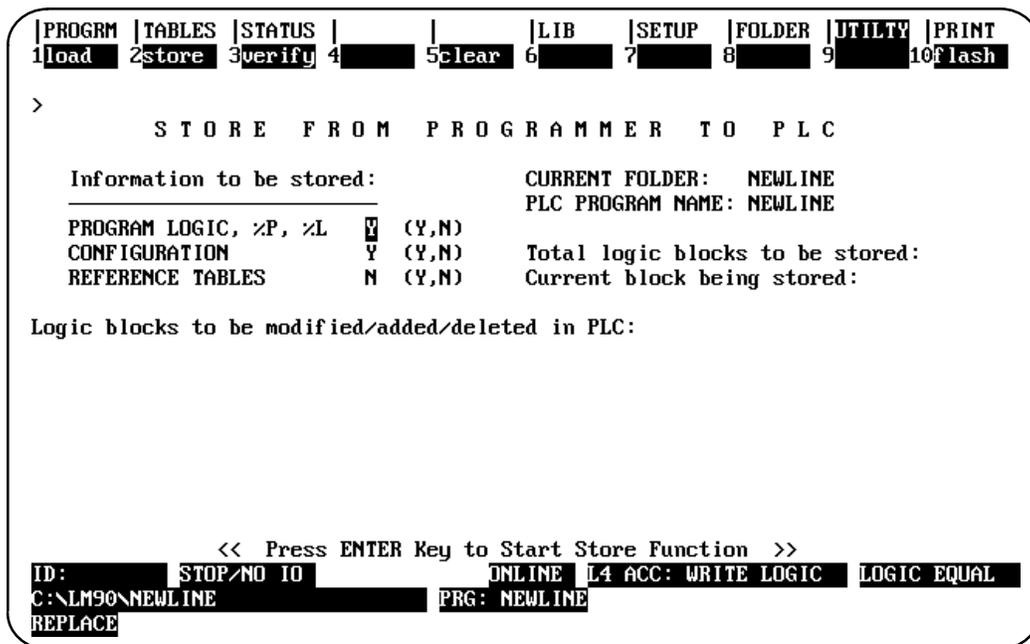
Le logiciel Logimaster doit d'abord déterminer si le programme actuel contient des blocs GRAFCET et, si c'est le cas, si les blocs GRAFCET sont supportés par la version actuelle de l'interpréteur GRAFCET sur l'API. Si l'un des blocs GRAFCET n'est pas supporté par la version actuelle de l'interpréteur GRAFCET, l'enregistrement ne sera pas autorisé. Si tous les blocs sont supportés, le **TRANSFERT EN MODE STOP** sera exécuté.

Un **TRANSFERT EN MODE STOP** complet enregistrera chaque bloc dans le programme sur l'API. Dans ce type d'enregistrement, tous les blocs GRAFCET sont réinitialisés à leur état initial, ce qui inclut l'initialisation des temporisateurs d'étape et la réinitialisation des bits d'étape <nom d'étape>.f à OFF.

Remarque

L'enregistrement du bloc GRAFCET entraînera l'effacement de tous les bits de contrôle de topologie GRAFCET. En conséquence, le programme commencera à l'étape initiale lorsque vous passerez en mode **RUN**. La mémoire de l'API (tables de références) ne sera pas effacée à moins que vous ne sélectionnez cette option sur l'écran de transfert.

Pour initialiser un **TRANSFERT EN MODE STOP**, appuyez sur **api (F2)** dans le menu des fonctions utilitaires.



Vous pouvez enregistrer trois types de données dans l'API à partir de la console de programmation : des programmes, des données de configuration, et/ou des tables de références. A la première apparition de cet écran, seuls les programmes sont définis à **Oui** ; c'est la sélection par défaut. Tous les autres champs ont par défaut la valeur **Non**.

Pour enregistrer toutes les données, vous devez changer les sélections pour les tables de références et la configuration sur **Oui**. Pour n'enregistrer qu'une partie des données, sélectionnez **Non** pour les données que vous ne souhaitez pas enregistrer. Vous pouvez aussi utiliser la touche Tab pour permuter la sélection des options.

Une fois les sélections effectuées, appuyez sur la touche **Enter** pour commencer le processus d'enregistrement.

7.2.2. Transfert en mode Run

Dans un **TRANSFERT EN MODE RUN**, seuls les blocs modifiés sont enregistrés sur l'API. Un **TRANSFERT EN MODE RUN** ne peut être effectué que si les blocs GRAFCET à enregistrer présentent la même topologie de réseau que celle actuellement enregistrée dans l'API. Vous ne pouvez modifier que les instructions en langage à relais pour les étapes et les transitions existantes.

Remarque

Des blocs GRAFCET ne peuvent pas remplacer des blocs en langage relais, et vice-versa, pendant un **TRANSFERT EN MODE RUN**.

7.3. VÉRIFICATION D'UN PROGRAMME AVEC L'API

Si le programme dans le dossier actuel est identique au programme dans l'API, le système affiche un message pour indiquer la fin de la vérification et qu'aucune différence n'a été détectée. L'état d'égalité de programme sera défini à EGAL. Si le programme dans le dossier actuel n'est pas équivalent au programme dans l'API, la fonction de vérification listera les blocs qui sont différents.

Pour plus d'informations sur la vérification d'un programme avec l'API, référez-vous au chapitre 9, *Utilitaires*, dans le document *GFK-0263 Logimaster 90-70 Programming Software User's Manual*.

Page laissée blanche intentionnellement

Vous pouvez imprimer un réseau GRAFCET en demandant l'impression de la **TOPOLOGIE GRAFCET**. Le système imprime alors les instructions associées aux étapes et aux transitions en utilisant les options choisies pour l'impression de programmes en langage à relais.

La sortie imprimée d'un bloc GRAFCET commence avec le réseau GRAFCET, en listant les étapes que contient le réseau. Après l'impression du réseau, le système imprime les instructions en langage relais de preprocessing et de postprocessing, puis les instructions en langage relais associées à chaque étape et enfin les instructions en langage relais associées à chaque transition.

La table des variables est imprimée après toutes les instructions. Pour un bloc GRAFCET **_MAIN**, les tables de déclaration de blocs de programme et de blocs d'interruption suivent la table des variables. Les tables de références croisées et d'utilisation des références sont les derniers listings imprimés pour le bloc.

Chaque section de la sortie imprimée est identifiée dans la table des matières, qui est imprimée à la fin du listing.

Remarque

Le Chapitre 10, *Fonctions d'impression*, du document *GFK-0263 Logicmaster 90-70 Programming Software User's Manual*, décrit les paramètres à définir et les options à sélectionner pour votre impression. Référez-vous le cas échéant au chapitre 10 pour remplir les écrans d'impression nécessaires.

Le tableau ci-dessous décrit les éléments imprimés sur l'exemple de sortie imprimée présenté à partir de la page 4-3.

Page imprimée	Description
Réseau GRAFCET	Le réseau GRAFCET est identifié par une étiquette de la forme <<SFC network 1>>, similaire à l'étiquette <<RUNG 1>> utilisée sur les segments en langage à relais. Cette étiquette est reprise en haut de chaque page d'instructions de topologie GRAFCET imprimée pour le réseau.
Instructions de preprocessing et de postprocessing	Les instructions de preprocessing et de postprocessing en langage à relais sont imprimées après les topologies de réseau et avant les instructions en langage relais de la première action.
Instructions d'étape	Le listing des instructions d'étape, un développé de chaque étape en langage à relais, suit les instructions en langage à relais de preprocessing et de postprocessing. Tout texte de commentaire d'étape associé à une étape sera imprimé avant les instructions correspondant à cette étape.
Instructions de transition	La section instructions de transition de la sortie imprimée contient les instructions en langage à relais pour chaque transition associée avec le graphe commande étape-transition. Les instructions de transition sont imprimées après les instructions de toutes les étapes.
Références croisées du bloc	<p>Les références croisées de bloc sont imprimées immédiatement après les instructions correspondant au bloc. Les références croisées incluent le nom de l'étape ou de la transition, le numéro de segment et l'instruction où est utilisée chaque référence. Le nom d'étape ou de transition est imprimé entre crochets, comme <STEP1>. La liste des numéros de segments qui suit un nom d'étape ou de transition inclut les segments où la référence est utilisée sur l'instruction indiquée à l'intérieur de l'étape ou de la transition. Comme une étape ne peut apparaître que dans un seul réseau GRAFCET, vous pouvez rapidement localiser une étape particulière en parcourant le listing de chaque réseau.</p> <p>La table de références croisées d'identificateur de bloc comprend des informations de références croisées pour les variables <nom d'étape>.x, <nom d'étape>.f et <nom d'étape>.t et les noms de transition. Comme les noms d'étape et les transitions sont locaux au bloc GRAFCET où ils sont utilisés, les variables d'étape et les noms de transition ne sont pas inclus dans la table de références croisées d'identificateur global.</p> <p>Remarque : Les références trouvées dans les instructions de preprocessing ou de postprocessing associées à un bloc GRAFCET sont notées comme étant dans une "étape" <_PRE> ou <_POST>, respectivement (voir exemple ci-dessous).</p>
Références croisées globales	Les références croisées globales comprennent le nom de bloc, le nom d'étape ou de transition, le numéro de segment et l'instruction pour chaque adresse ou identificateur de référence utilisé dans le programme.
Table des matières	Chaque section de la sortie imprimée est identifiée dans la table des matières, qui est imprimée à la fin du listing.

2. EXEMPLE DE PAGE POUR DES INSTRUCTIONS DE PREPROCESSING/POSTPROCESSING

```

02-16-94 10:40 GE Fanuc Série 90-70 DOCUMENTATION (v5.00) Page 50
Logicmaster 90 SFC Logic Listing

***** DEVELOPPE DES INSTRUCTIONS DE PREPROCESSING *****

| [ START OF PREPROCESSING ]
|
| << RUNG 1 >>
|
| +-----+
| +-----+ ADD_+-
| | INT |
| | |
| TOTAL --+I1 Q+- TOTAL
| | |
| CONST --+I2 |
| +00001 +-----+
|
| [ END OF PREPROCESSING ]

***** DEVELOPPE DES INSTRUCTIONS DE POSTPROCESSING *****

| [ START OF POSTPROCESSING ]
|
| << RUNG 1 >>
|
| +-----+
| +-----+ ADD_+-
| | INT |
| | |
| TOTAL --+I1 Q+-TOTAL2
| | |
| CONST --+I2 |
| +00001 +-----+
|
| [ END OF POSTPROCESSING ]

***** DEVELOPPE DE L'ETAPE S1 *****

| [ START OF ACTION FOR STEP S1 ]
|
| << RUNG 1 >>
|
| +-----+ +-----+ +-----+ NO-ERR
| +DATA_+ +-----+ +CALL FAULT +---( )--
| | INIT_ | | INT | | (EXTERNAL) |
| | ASCII | | | |
| | Q+-%G00001 CONST --+IN Q+-%G00185 CONST --+X1 Y1+-
| | LEN | +00000 | LEN | 0001 | |
| | 00022 | | 00001 | | |
| +-----+ +-----+ CONST --+X2 Y2+-
| | | | 0005 +-----+

Program: EXETST C:\70FOLDER\EXETST Block: RESET
    
```


5. EXEMPLE DE PAGE DE RÉFÉRENCES CROISÉES DE BLOC

```

02-16-94 10:38 GE Fanuc Série 90-70 DOCUMENTATION (v5.00) Page 15
Logicmaster 90 SFC Logic Listing

=====
===== INTERNAL (%M) CROSS REFERENCES
=====
REF. NICKNAME REF. DESCRIPTION / CROSS REFERENCES
-----
%M00001 NOHLT If set to 1, no halting/test
-] [- <HOLD> 2
%M00002 DONE Indicate the end of testing
-(S) <END> 1
-(R) <REDME> 3

=====
===== TEMPORARY (%T) CROSS REFERENCES
=====
REF. NICKNAME REF. DESCRIPTION / CROSS REFERENCES
-----
%T00001 CONT Continuation flag
-] [- <T10> 1 <T11> 1 <T12> 1 <T13> 1 <T14> 1 <T15> 1
<T16> 1 <T17> 1 <T20> 1 <HOLD> 1
-(S) <REDME> 2 <OPEN> 1 <HOLD> 2
-(R) <RSET> 1 <HOLD> 1

=====
===== REGISTER (%R) CROSS REFERENCES
=====
REF. NICKNAME REF. DESCRIPTION / CROSS REFERENCES
-----
%R00001 BASE Base address of DEVLIB table
FBIO <REDME> 1
%R00004 PATH Execution path selection
FBIO <T2> 1 <T3> 1 <T4> 1 <T5> 1 <T6> 1 <T7> 1 <T8> 1
<T9> 1 <T18> 1 <T19> 1 <REDME> 1 <NEXT> 1

=====
===== IDENTIFIER NAME CROSS REFERENCES
=====
IDENTIFIER REF. DESCRIPTION / CROSS REFERENCES
-----
T1 -( ) <T1> 1
T2 -( ) <T2> 1
T3 -( ) <T3> 1
T4 -( ) <T4> 1
T5 -( ) <T5> 1
T6 -( ) <T6> 1
T7 -( ) <T7> 1
T8 -( ) <T8> 1
T9 -( ) <T9> 1
T10 -( ) <T10> 1
T11 -( ) <T11> 1
T12 -( ) <T12> 1
T13 -( ) <T13> 1
T14 -( ) <T14> 1
T15 -( ) <T15> 1
T16 -( ) <T16> 1

Program: EXETST C:\70FOLDER\EXETST Block: _MAIN

```

6. EXEMPLE DE PAGE DE RÉFÉRENCES CROISÉES GLOBALES

REF.	BLOCK	NICKNAME	REF. DESCRIPTION / CROSS REFERENCES
02-16-94 10:43 GE Fanuc Série 90-70 DOCUMENTATION (v5.00) Page 89 Logicmaster 90 SFC Logic Listing			
===== INTERNAL (%M) GLOBAL CROSS REFERENCES =====			
%M00001	_MAIN	NOHLT	If set to 1, no halting/test -] [- <HOLD> 2
%M00002	_MAIN	DONE	Indicate the end of testing -(S) <END> 1 -(R) <REDME> 3
===== TEMPORARY (%T) GLOBAL CROSS REFERENCES =====			
REF.	BLOCK	NICKNAME	REF. DESCRIPTION / CROSS REFERENCES
%T00001	_MAIN	CONT	Continuation flag -] [- <T10> 1 <T11> 1 <T12> 1 <T13> 1 <T14> 1 <T15> 1 <T16> 1 <T17> 1 <T20> 1 <HOLD> 1 -(S) <REDME> 2 <OPEN> 1 <HOLD> 2 -(R) <RSET> 1 <HOLD> 1
	RST_MSG		-(S) <S13> 3
	RACKGEN		-(S) <S10> 3
	RESET		-(S) <S3> 4
	CBASE		-(S) <S7> 4
	CNVCTRL		-(S) <S5> 4
	GOTSG		-(S) <S5> 4
	ALLSTOP		-(S) <S3> 4
%T00002	RST_MSG	NO-ERR	Error status flag -]/[- <S1> 2 <S2> 3,6,9 <S3> 2 <S8> 2 <S4> 2 <S6> 2 <S11> 2 <S7> 2 <S12> 2 <S5> 2<S9> 2 <S13> 2 -() <S1> 1 <S2> 2,5,8 <S3> 1 <S8> 1 <S4> 1 <S6> 1 <S11> 1 <S7> 1 <S12> 1 <S5> 1<S9> 1 <S13> 1
	RACKGEN		-]/[- <S1> 2 <S2> 3,6 <S3> 2 <S4> 2 <S5> 2<S6> 2 <S7> 2 <S8> 2 <S9> 2 <S10> 2 -() <S1> 1 <S2> 2,5 <S3> 1 <S4> 1 <S5> 1<S6> 1 <S7> 1 <S8> 1 <S9> 1 <S10> 1
	RESET		-]/[- <S1> 2 <S2> 2 <S3> 3 -() <S1> 1 <S2> 1 <S3> 2
	CBASE		-]/[- <S1> 3 <S2> 2,4,6 <S3> 2 <S4> 2 <S5> 2 <S6> 2 <S7> 3 -() <S1> 2 <S2> 1,3,5 <S3> 1 <S4> 1 <S5> 1 <S6> 1 <S7> 2
	CNVCTRL		-]/[- <S1> 3 <S2> 2 <S3> 2 <S4> 2 <S5> 3 -() <S1> 2 <S2> 1 <S3> 1 <S4> 1 <S5> 2
	GOTSG		-]/[- <S1> 3 <S2> 2 <S3> 2 <S4> 2 <S5> 3 -() <S1> 2 <S2> 1 <S3> 1 <S4> 1 <S5> 2
	ALLSTOP		-]/[- <S1> 3 <S2> 2 <S3> 3 -() <S1> 2 <S2> 1 <S3> 2
Program: EXETST C:\70FOLDER\EXETST			Global Cross Reference Tables

REF.	BLOCK	NICKNAME	REF. DESCRIPTION / CROSS REFERENCES
%G00177	RST_MSG		FBIO <S1> 1 <S13> 1
	RACKGEN		FBIO <S1> 1 <S10> 1
	CNVCTRL		FBIO <S1> 1
%G00185	RESET		FBIO <S1> 1 <S3> 2
	CNVCTRL		FBIO <S5> 1

===== IDENTIFIER NAME GLOBAL CROSS REFERENCES =====

IDENTIFIER	BLOCK	REF. DESCRIPTION / CROSS REFERENCES
FAULT	RST_MSG	CALL <S1> 1 <S13> 1
	RACKGEN	CALL <S1> 1 <S10> 1
	RESET	CALL <S1> 1 <S3> 2
	CBASE	CALL <S1> 1 <S7> 2
	CNVCTRL	CALL <S1> 1 <S5> 2
	GOTSG	CALL <S1> 1 <S5> 2
	ALLSTOP	CALL <S1> 1 <S3> 2
ESET	RST_MSG	CALL <S3> 1 <S8> 1 <S4> 1 <S6> 1 <S11> 1<S7> 1 <S12> 1 <S5> 1 <S9> 1
	RACKGEN	CALL <S3> 1 <S4> 1 <S5> 1 <S6> 1 <S7> 1 <S8> 1
	CBASE	CALL <S3> 1 <S4> 1 <S5> 1
	CNVCTRL	CALL <S2> 1 <S3> 1 <S4> 1
	GOTSG	CALL <S2> 1 <S3> 1 <S4> 1
	ALLSTOP	CALL <S2> 1
ECHK	RACKGEN	CALL <S9> 1
	CBASE	CALL <S6> 1
	CNVCTRL	CALL <S5> 1
	GOTSG	CALL <S5> 1
	ALLSTOP	CALL <S3> 1
ASSERT	RST_MSG	CALL <S2> 1,4,7
	RACKGEN	CALL <S2> 1,4
	CBASE	CALL <S2> 1,3,5
	CNVCTRL	CALL <S1> 2
	GOTSG	CALL <S1> 2
ADDSERT	RST_MSG	CALL <S2> 2,5,8
	RACKGEN	CALL <S2> 2,5
RESET	_MAIN	Pre & Post Processing
		CALL <PPP> 1
CLEAR	_MAIN	Clear fault tables
		CALL <RSET> 1
STOP	RST_MSG	Stop the PLC
		CALL <S1> 2 <S2> 3,6,9 <S3> 2 <S8> 2 <S4> 2 <S6> 2 <S11> 2 <S7> 2 <S12> 2 <S5> 2<S9> 2 <S13> 2
	RACKGEN	CALL <S1> 2 <S2> 3,6 <S3> 2 <S4> 2 <S5> 2<S6> 2 <S7> 2 <S8> 2 <S9> 2 <S10> 2
	RESET	CALL <S1> 2 <S2> 2 <S3> 3
	CBASE	CALL <S1> 3 <S2> 2,4,6 <S3> 2 <S4> 2 <S5> 2 <S6> 2 <S7> 3
	CNVCTRL	CALL <S1> 3 <S2> 2 <S3> 2 <S4> 2 <S5> 3

7. EXEMPLE DE PAGE DE TABLE DES MATIÈRES

02-16-94 10:43	GE Fanuc Série 90-70 DOCUMENTATION (v5.00)	Contents	1
***** TABLE DES MATIERES DU PROGRAMME *****			
_MAIN			1
SFC Network 1			
Topology			1
Step Logic			
REDME			2
RSET			2
SIMPL			2
SIMUL			3
JPCY			3
SELEC			3
PPP			4
CMPD1			4
CMPD2			5
OPEN			5
NEXT			5
END			6
HOLD			6
Transition Logic			6
Program Block Declaration Table			13
Cross Reference Tables			15
RST_MSG			17
SFC Network 1			
Topology			17
Step Logic			
S1			18
S2			18
S3			23
S8			23
S4			24
S6			24
S11			25
S7			25
S12			26
S5			26
S9			27
S10			27
S13			28
Transition Logic			28
Cross Reference Tables			33
Non-printable Blocks Lists			38
Global Cross Reference Tables			39
%L Xref and Use Tables			43
Program: EXETST	C:\70FOLDER\EXETST	TABLE OF CONTENTS	

De manière générale, tout ce qui peut générer une erreur dans un bloc programmé en langage à relais sera aussi une erreur dans la partie en langage à relais d'un bloc de programme en langage GRAFCET. De plus, plusieurs vérifications globales sont effectuées sur un bloc de programme en langage à relais complet, qui contrôlent notamment s'il n'y a pas d'étiquettes manquantes, de sauts MCR vers l'arrière, de END_FORs manquants, ou trop d'appels de bloc de programme. Ces types de contrôles globaux sont aussi appliqués individuellement à chaque section en langage à relais d'un bloc GRAFCET. Par exemple, chaque section d'instructions d'action doit comporter une étiquette correspondante pour chaque saut unique à l'intérieur de cette section d'instructions en langage à relais. La seule exception à cette règle est le contrôle du nombre d'appels de bloc de programme. Pour un bloc en langage GRAFCET, le contrôle est effectué sur le bloc GRAFCET tout entier, et non sur les différentes sections d'instructions en langage à relais.

Cette annexe décrit des conditions d'erreur propres aux blocs GRAFCET et aux instructions en langage à relais qui sont utilisées dans ces blocs. Elle ne décrit pas les erreurs de langage à relais courantes que l'on peut retrouver à la fois dans les blocs programmés en langage à relais et les blocs GRAFCET.

1. ERREURS GÉNÉRALES

Le système détectera les erreurs suivantes à l'entrée dans un segment en langage à relais :

- Une référence à un nom de variable de transition hors des instructions de transition en langage à relais pour la variable.
- Une tentative de modification (accès en écriture) d'un bit d'étape <nom d'étape>.x ou <nom d'étape>.f.
- Une tentative de modification (accès en écriture) d'une valeur <nom d'étape>.t.

2. ERREURS D'INSTRUCTIONS DE TRANSITION

Le système détectera les conditions d'erreur suivantes lorsque vous appuyez sur la touche **Escape** pour sortir d'une section d'instructions de transition. Le bloc restera non-exécutable tant que les erreurs n'auront pas été corrigées.

- La variable de transition n'a pas été définie.
- Plus d'un segment normal.
- Moins d'un segment normal.

3. ERREURS D'INSTRUCTIONS DE TRANSITION, D'ACTION D'ÉTAPE, ET DE PREPROCESSING/POSTPROCESSING

Le système détectera les conditions d'erreur suivantes lorsque vous appuyez sur la touche **Escape** pour sortir d'une section d'instructions de transition, une section d'instructions d'action, ou une section d'instructions de preprocessing ou de postprocessing. Le bloc GRAFCET restera non-exécutable tant que les erreurs n'auront pas été corrigées.

- Un nom de saut/d'étiquette est utilisé dans une autre section d'instructions en langage à relais dans ce bloc.
- Un nom de MCR/END_MCR est utilisé dans une autre section d'instructions en langage à relais dans ce bloc.

Les conditions d'erreur suivantes **doivent** être corrigées immédiatement :

- Un saut MCR vers l'arrière.
- Trop de FOR imbriqués.
- Trop de EXIT_FOR.
- Le transfert de plus de 16 octets de données entre deux blocs fonctionnels (instructions). Vous pouvez utiliser des registres sur la sortie d'un bloc fonctionnel et l'entrée du bloc fonctionnel (instruction) suivant.

4. ERREURS D'ÉLÉMENT GRAFCET GÉNÉRALES

Les erreurs suivantes sont détectées à l'entrée et doivent être corrigées immédiatement :

- L'entrée d'une étape, une transition ou un saut d'étape dans la mauvaise rangée.
- Le déplacement du curseur en dehors d'un saut d'étape avant d'avoir nommé celui-ci.
- Une tentative d'ajouter plus de 255 étapes, 383 transitions, ou 255 sauts d'étape uniques.
- Une tentative d'extension du réseau GRAFCET au-delà de 128 rangées ou 8 colonnes.

5. ERREURS GRAFCET DE PREMIER NIVEAU

Les erreurs suivantes peuvent se produire au niveau principal du réseau GRAFCET. Elles sont détectées quand vous appuyez sur la touche **Escape** dans ce niveau. Le bloc restera non-exécutable tant que les erreurs n'auront pas été corrigées.

- Deux transitions directement connectées.
- Deux étapes directement connectées.
- Une transition qui n'est pas précédée par une étape.
- Une transition qui n'est pas suivie par un étape ou un saut d'étape source.
- Une étape et un saut d'étape source directement connectés.
- Des segments GRAFCET déconnectés. Deux sections ou plus de réseaux GRAFCET qui ne sont pas connectées par des liaisons verticales, des branches horizontales ou des sauts d'étape.
- Des branches voisines non adaptées. Une branche de divergence en OU voisine d'un segment de branchement parallèle.
- Début incomplet d'une branche de divergence en ET. Vous devez avoir une transition ainsi qu'une branche de divergence en ET.
- Fin incomplète d'une branche de divergence en ET. Vous devez avoir une transition ainsi qu'une branche de divergence en ET.
- Début incomplet d'une branche de divergence en OU. Vous devez avoir une transition ainsi qu'une branche de divergence en OU.
- Fin incomplète d'une branche de divergence en OU. Vous devez avoir une transition ainsi qu'une branche de divergence en OU.
- Plus d'une étape initiale.
- L'étape initiale apparaît dans une structure de contrôle simultanée.
- L'étape ou transition apparaît plus d'une fois dans le réseau GRAFCET.
- Saut d'étape destination manquant.
- Plusieurs occurrences du même saut d'étape destination.
- Saut d'étape source manquant. (Il s'agit d'un message d'avertissement, la correction n'est pas obligatoire.)
- Des sauts d'étape source et destination ne sont pas autorisés dans une structure de contrôle simultanée.
- Plus de 32 branchements parallèles dans une même structure de contrôle.
- Imbrication incorrecte des structures de contrôle. Une structure simultanée doit être fermée ; autrement dit, elle doit converger. De plus, elle doit être fermée dans l'ordre inverse de son ouverture.
- Une liaison dirigée en GRAFCET diverge en deux branches sans l'utilisation d'une structure de contrôle divergente.

Outre la détection des erreurs au niveau principal du graphe commande étape-transition quand vous appuyez sur la touche **Escape** de l'éditeur, le système détectera aussi toutes les erreurs à tous les autres niveaux du bloc GRAFCET et affichera un message concernant la première erreur rencontrée dans la zone de message. Vous pouvez choisir de corriger l'erreur à cet instant, ou continuer à quitter l'éditeur. La plupart des erreurs listées ici marqueront le bloc comme "non-exécutable." Autrement dit le bloc peut être enregistré, mais les erreurs doivent être corrigées avant que le programme puisse être enregistré dans l'API.

Page laissée blanche intentionnellement

A

Action, 2–5
 ALT–D (suppression), 3–33
 ALT–S (enregistrer sur l’API), 3–43
 ALT–U (forcer un enregistrement), 3–43
 ALT–H (écrans d’aide), 1–2

B

Bits d’étape, 3–3
 Bits de défaut, 3–3
 Blocs de programme, 3–2
 changement d’état et forçages, 3–57
 changer le mode d’affichage, 3–9
 commentaires d’étape, 3–50
 configuration de plusieurs langages, 3–53
 enregistrement dans l’API de la console de programmation, 3–63
 transfert en mode Run, 3–64
 transfert en mode Stop, 3–63
 enregistrement de l’historique d’évolution d’un bloc, 3–58
 enregistrement des modifications d’un bloc GRAFCET, 3–43
 fonction d’impression, 4–1
 impression d’un réseau GRAFCET, 4–3
 instructions d’action, 4–5
 instructions de preprocessing et de postprocessing, 4–4
 instructions de transition, 4–6
 références croisées de bloc, 4–7
 références croisées globales, 4–8
 table des matières, 4–10
 fonctions d’ouverture d’espace, 3–34
 fonctions de suppression d’espace, 3–34
 fonctions utilitaires, 3–63
 format d’un bloc GRAFCET, 3–4
 insertion/édition d’un réseau GRAFCET, 3–10
 exemple, 3–18
 instructions de preprocessing et de postprocessing, 3–5
 journalisation des défauts pour des durées d’étape hors limites, 3–51
 limites, 3–2
 marqueurs, 3–4
 options, 3–51
 organisation des grilles, 3–14
 programmation des actions GRAFCET, 3–6
 réglage de la base de temps, 3–61
 réglage des limites de durée d’étape, 3–61
 sélection du langage du bloc, 3–7

transfert de l’API vers la console de programmation, 3–63
 transfert en mode Run, 3–64
 transfert en mode Stop, 3–63
 utilisation de la fonction de recherche, 3–44
 utilisation de la fonction Vers, 3–49, 3–59
 utilisation des fonctions de mise au point, 3–55
 vérification d’un programme avec l’API, 3–65

Branchements, 2–12
See also Structures de contrôle de base

C

Commentaires, 3–50
 Convergence en ET d’une séquence, 2–14
 Convergence en OU d’une séquence, 2–13
 Cycle, 2–15

D

Début du programme, 3–4
 Déclarations des blocs de programme, 3–4
 Déclarations des variables, 3–4
 Dépannage, affichage des erreurs dans le réseau, 3–42
 Divergence en ET d’une séquence, 2–13
 Divergence en OU d’une séquence, 2–12

E

Ecrans d’aide, 1–2
 Éditeur GRAFCET. *See* Edition d’un bloc de programme
 Edition d’un bloc de programme, 3–1
 blocs de programme GRAFCET, 3–2
 changement d’état et forçages, 3–57
 changer le mode d’affichage, 3–9
 commentaires d’étape, 3–50
 configuration de plusieurs langages, 3–53
 enregistrement dans l’API de la console de programmation, 3–63
 transfert en mode Run, 3–64
 transfert en mode Stop, 3–63
 enregistrement de l’historique d’évolution d’un bloc, 3–58
 enregistrement des modifications d’un bloc GRAFCET, 3–43
 fonctions d’ouverture d’espace, 3–34
 fonctions de suppression d’espace, 3–34

- fonctions utilitaires, 3–63
- insertion/édition d'un réseau GRAFCET, 3–10
 - exemple, 3–18
- instructions de preprocessing et de postprocessing, 3–5
- journalisation des défauts pour des durées d'étape hors limite, 3–51
- limites des blocs, 3–2
- options, 3–51
- organisation des grilles, 3–14
- programmation des actions GRAFCET, 3–6
- réglage de la base de temps, 3–61
- réglage des limites de durée d'étape, 3–61
- sélection du langage du bloc, 3–7
- transfert de l'API vers la console de programmation, 3–63
- transfert en mode Run, 3–64
- transfert en mode Stop, 3–63
- utilisation de la fonction de recherche, 3–44
- utilisation de la fonction Vers, 3–49, 3–59
- utilisation des fonctions de mise au point, 3–55
- vérification d'un programme avec l'API, 3–65

Editor, SFC. *See* Editing a program block

Effacement d'un élément, 3–33

Élément qualificatif d'une action, 2–5

Enregistrement dans l'API de la console de programmation, 3–63

Erreurs d'instructions, A–2

Erreurs d'instructions de transition, A–2

Erreurs générales, A–1

Erreurs utilisateur courantes, A–1

- erreurs d'élément GRAFCET générales, A–2
- erreurs d'instructions, A–2
- erreurs d'instructions de transition, A–2
- erreurs générales, A–1
- erreurs GRAFCET de premier niveau, A–3

Étape initiale, 2–4

Étape normale, 2–4

Étapes, 2–4

- bit d'étape, 3–3
- bit de défaut, 3–3
- changement d'état et forçages, 3–57
- commentaires, 3–50
- durées minimale et maximale d'étape, 3–3, 3–60
- élément qualificatif d'une action, 2–5
- étape initiale, 2–4
- étape normale, 2–4
- journalisation des défauts pour des durées d'étape hors limite, 3–51
- noms, 3–17
- programmation des actions GRAFCET, 3–6

- rangées d'étape, 3–20

Evolution, historique d'évolution d'un bloc, 3–58

Evolution du graphe commande étape–transition, 2–9

Exemples de graphes commande étape–transition, 2–11

F

Fin du programme, 3–4

Fonction d'impression, 4–1

- impression d'un réseau GRAFCET, 4–3
- instructions d'action, 4–5
- instructions de preprocessing et de postprocessing, 4–4
- instructions de transition, 4–6
- références croisées de bloc, 4–7
- références croisées globales, 4–8
- table des matières, 4–10

Fonction de recherche, 3–44

- mettre fin à une recherche, 3–49
- utilisation, 3–45

Fonctions d'ouverture d'espace, 3–34

- ouvrir la colonne, 3–36
- ouvrir la rangée, 3–34

Fonctions de mise au point, 3–55

- changement d'état et forçages, 3–57
- enregistrement de l'historique d'évolution d'un bloc, 3–58
- réglage de la base de temps, 3–61
- réglage des limites de durée d'étape, 3–61
- Vers, 3–49, 3–59

Fonctions de suppression d'espace, 3–34

- effacer la colonne, 3–40
- effacer la rangée, 3–38

Fonctions des touches, 1–2

Fonctions utilitaires, 3–63

- enregistrement dans l'API de la console de programmation, 3–63
- transfert de l'API vers la console de programmation, 3–63
- vérification d'un programme avec l'API, 3–65

G

Grafcet

- cycle, 2–15
- élément qualificatif d'une action, 2–5
- erreurs d'élément GRAFCET générales, A–2
- erreurs GRAFCET de premier niveau, A–3
- erreurs utilisateur courantes, A–1

étape, 2–4
 évolution du graphe commande étape–transition, 2–9
 exemples de graphes commande étape–transition, 2–11
 fonction d'impression, 4–1
 impression d'un réseau GRAFCET, 4–3
 instructions d'action, 4–5
 instructions de preprocessing et de postprocessing, 4–4
 instructions de transition, 4–6
 références croisées de bloc, 4–7
 références croisées globales, 4–8
 table des matières, 4–10
 saut, 2–15
 saut vers l'arrière, 2–15
 sauts d'étape source et destination, 2–15
 structures de contrôle de base, 2–12
 convergence en ET d'une séquence, 2–14
 convergence en OU d'une séquence, 2–13
 divergence en ET d'une séquence, 2–13
 divergence en OU d'une séquence, 2–12
 exemples, 2–16
 règles, 2–18
 séquence simple, 2–12
 transition, 2–6

H

Historique d'un bloc, 3–58

I

Instructions, 3–5
 Instructions de postprocessing, 3–4, 3–5
 Instructions de preprocessing, 3–4, 3–5
 Interruptions, 3–4
 Introduction, 1–1

J

Journalisation des défauts pour des durées d'étape hors limite, 3–51

L

Langage, sélection pour le bloc, 3–7
 Language, configuration de plusieurs langages, 3–53
 Logiciel, installation du logiciel, 1–1

M

Marqueurs, 3–4
 début de programme, 3–4
 déclarations des blocs de programme, 3–4
 déclarations des variables, 3–4
 fin de programme, 3–4
 instructions de postprocessing, 3–4
 instructions de preprocessing, 3–4
 interruptions, 3–4
 Mise au point, A–1
 erreurs d'élément GRAFCET générales, A–2
 erreurs d'instructions, A–2
 erreurs d'instructions de transition, A–2
 erreurs générales, A–1
 erreurs GRAFCET de premier niveau, A–3
 Mnémoniques, 3–20
 Mode d'affichage
 normal, 3–9
 numéro, 3–9
 Modes d'exécution, contrôler le mode d'exécution avec un SFC_RESET, 3–56
 MS-DOS, 1–1

N

Noms d'étape, 3–17
 Noms de transition, 3–17
 Noms réservés, 3–17

O

Organisation des grilles, 3–14

P

Programmation des actions GRAFCET, 3–6
 Programme, blocs de programme. *See* Program blocks

R

Rangées d'étape, 3–20
 Rangées de transition, 3–20
 Réseau
 affichage des erreurs dans le réseau, 3–42
 insertion/édition d'un réseau GRAFCET, 3–10
 exemple, 3–18

organisation des grilles, 3–14

S

Saut, 2–15

Saut d'étape destination, 2–15

Saut d'étape source, 2–15

Saut vers l'arrière, 2–15

Sauts d'étape, 2–15, 3–29

 saut d'étape destination, 2–15, 3–30

 saut d'étape source, 2–15, 3–29

Séquence simple, 2–12

SFC_RESET, 3–56

Structures de contrôle de base, 2–12

 convergence en ET d'une séquence, 2–14

 convergence en OU d'une séquence, 2–13

 divergence en ET d'une séquence, 2–13

 divergence en OU d'une séquence, 2–12

 exemples, 2–16

 règles, 2–18

 séquence simple, 2–12

Suppression d'un élément, 3–33

T

Touches ALT, 1–2

Touches CTRL, 1–2

Touches de curseur, 3–17

Transfert de l'API vers la console de programmation,
3–63

Transfert en mode Run, 3–64

Transfert en mode Stop, 3–63

Transitions, 2–6

 changement d'état et forçages, 3–57

 noms, 3–17

 rangées de transition, 3–20

V

Valeurs temporelles, durées minimale et maximale
d'étape, 3–3, 3–60

Vérification d'un programme avec l'API, 3–65

Vers, 3–49, 3–59