

GRAFCET specification language for sequential function charts

The European Standard EN 60848:2002 has the status of a
British Standard

ICS 29.020

National foreword

This British Standard is the official English language version of EN 60848:2002. It is identical with IEC 60848:2002. It supersedes BS 7716:1993 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee GEL/3, Documentation and graphical symbols, which has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international/European committee any enquiries on the interpretation, or proposals for change, and keep the UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

A list of organizations represented on this committee can be obtained on request to its secretary.

From 1 January 1997, all IEC publications have the number 60000 added to the old number. For instance, IEC 27-1 has been renumbered as IEC 60027-1. For a period of time during the change over from one numbering system to the other, publications may contain identifiers from both systems.

Cross-references

The British Standards which implement international or European publications referred to in this document may be found in the BSI Standards Catalogue under the section entitled “International Standards Correspondence Index”, or by using the “Find” facility of the BSI Standards Electronic Catalogue.

A British Standard does not purport to include all the necessary provisions of a contract. Users of British Standards are responsible for their correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

This British Standard, having been prepared under the direction of the Electrotechnical Sector Policy and Strategy Committee, was published under the authority of the Standards Policy and Strategy Committee on 27 May 2002

Summary of pages

This document comprises a front cover, an inside front cover, the EN title page, pages 2 to 52, an inside back cover and a back cover.

The BSI copyright date displayed in this document indicates when the document was last issued.

Amendments issued since publication

Amd. No.	Date	Comments

© BSI 27 May 2002

ISBN 0 580 39757 2

EUROPEAN STANDARD

EN 60848

NORME EUROPÉENNE

EUROPÄISCHE NORM

April 2002

ICS 29.020

English version

GRAFCET specification language for sequential function charts (IEC 60848:2002)

Langage de spécification GRAFCET
pour diagrammes fonctionnels en séquence
(CEI 60848:2002)

Entwurfssprache GRAFCET
für Ablauf-Funktionspläne
(IEC 60848:2002)

This European Standard was approved by CENELEC on 2002-04-01. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the Central Secretariat has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Czech Republic, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Malta, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Central Secretariat: rue de Stassart 35, B - 1050 Brussels

© 2002 CENELEC - All rights of exploitation in any form and by any means reserved worldwide for CENELEC members.

Foreword

The text of document 3B/344/FDIS, future edition 2 of IEC 60848, prepared by SC 3B, Documentation, of IEC TC 3, Information structures, documentation and graphical symbols, was submitted to the IEC-CENELEC parallel vote and was approved by CENELEC as EN 60848 on 2002-04-01.

The following dates were fixed:

- latest date by which the EN has to be implemented
at national level by publication of an identical
national standard or by endorsement (dop) 2003-01-01
- latest date by which the national standards conflicting
with the EN have to be withdrawn (dow) 2005-04-01

Annexes designated "normative" are part of the body of the standard.

Annexes designated "informative" are given for information only.

In this standard, annex ZA is normative and annexes A, B and C are informative.

Annex ZA has been added by CENELEC.

Endorsement notice

The text of the International Standard IEC 60848:2002 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following note has to be added for the standard indicated:

IEC 61131-3 NOTE Harmonized as EN 61131-3:1993 (not modified).

CONTENTS

INTRODUCTION.....	4
1 Scope and object.....	5
2 Normative references.....	5
3 Terms and definitions	5
4 General principles.....	7
4.1 Context.....	7
4.2 GRAFCET, a behaviour specification language	8
4.3 GRAFCET, short presentation.....	8
4.4 Syntax rule	10
4.5 Evolution rules.....	10
4.6 Input events.....	11
4.7 Internal events.....	11
4.8 Output modes.....	12
4.9 Application of the evolution rules.....	12
4.10 Comparison between the two output modes	15
5 Graphical representation of the elements	16
6 Graphical representation of sequential structures	29
6.1 Basic structures.....	29
6.2 Particular structures	32
7 Structuring.....	35
7.1 Partition of a grafcet	35
7.2 Structuring using the forcing of a partial grafcet	37
7.3 Structuring using the enclosure.....	38
7.4 Structuring using the macro-steps.....	41
Annex A (informative) Example of control of a press.....	42
Annex B (informative) Example: Automatic weighing-mixing.....	43
Annex C (informative) Relations between the GRAFCET of IEC 60848 and the SFC of IEC 61131-3.....	49
Annex ZA (normative) Normative references to international publications with their corresponding European publications	52
Bibliography	51

INTRODUCTION

The main reason for the revision of this standard is the desire of the users to increase the standardised specification language with new concepts, allowing a structured and hierarchical description.

Otherwise, in addition to the descriptive and functional aspects of the first edition, it now seems necessary to add the formal and behavioural aspects, which are essential for the definition of a real specification language.

For all these reasons, an overall review of the document is required.

This standard is mainly for people (design engineers, realisation engineers, maintenance engineers, etc.) who need to specify the behaviour of a system (control-command of an automatic machine, safety component, etc.). This specification language should also serve as a communication means between designers and users of automated systems.

GRAFCET SPECIFICATION LANGUAGE FOR SEQUENTIAL FUNCTION CHARTS

1 Scope and object

This International Standard defines the GRAFCET¹⁾ specification language for the functional description of the behaviour of the sequential part of a control system.

This standard specifies the symbols and the rules for the graphical representation of this language, as well as for its interpretation.

This standard has been prepared for automated production systems of industrial applications. However no particular area of application is excluded.

Methods of development of a specification that makes use of GRAFCET are beyond the scope of this standard. One method is for example the "SFC language" specified in IEC 61131-3, which defines a set of programming languages for programmable controllers.

NOTE See annex C for further information on the relations between IEC 60848 and implementation languages such as the SFC of IEC 61131-3.

2 Normative references

The following normative documents contain provisions, which, through reference in this text, constitute provisions of this International Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of IEC and ISO maintain registers of currently valid International Standards.

IEC 60050-351:1998, *International Electrotechnical Vocabulary (VEI – Part 351: Automatic control*

IEC 60617-12:1997, *Graphical symbols for diagrams – Part 12: Binary logical elements*

3 Terms and definitions

For the purposes of this International Standard, the following definitions apply. The definitions of the terms preceded by an asterisk apply only in the context of the GRAFCET specification language. The chosen order is the alphabetic one.

3.1

* action

GRAFCET language element associated with a step, indicating an activity to be performed on output variables

¹⁾ GRAFCET, GRAPhe Fonctionnel de Commande Etape Transition.

3.2**chart, graph**

graphical presentation describing the behaviour of a system, for example the relations between two or more variable quantities, operations or states

3.3*** directed link**

GRAFCET language element indicating the evolution paths between steps by connecting steps to transitions and transitions to steps

3.4*** grafcet chart**

function chart using GRAFCET

NOTE The "grafcet chart" can, in short form, be called "grafcet".

3.5*** input event**

event characterized by the change of at least one value of all input variables of the sequential part of the system

3.6*** internal event**

event characterized by an input event associated with the situation of the sequential part of the system

3.7*** interpretation**

part of the GRAFCET enabling the linkage of:

- the input variables and the structure, by the means of the transition-condition;
- the output variables and the structure, by the means of the actions

3.8*** situation**

name of the state of the system described by grafcet and characterised by the active steps at a given instant

3.9*** step**

GRAFCET language element used for the definition of the state of the sequential part of the system

NOTE 1 A step can be active or inactive.

NOTE 2 The set of active steps represents the situation of the system.

3.10*** structure**

part of the GRAFCET enabling the description of the possible evolution between situations

3.11**system**

a set of interrelated elements considered in a defined context as a whole and separated from their environment

[IEV 351-11-01]

NOTE 1 Such elements may be material objects and concepts as well as their results (e.g. forms of organisation, mathematical methods, programming languages).

NOTE 2 The system is considered to be separated from the environment and from the other external systems by an imaginary surface, which cuts the links between them and the system.

NOTE 3 The language GRAFCET can be use to describe the logical behaviour of any kind of system.

3.12

* transient evolution

evolution characterized by the clearing of several successive transitions on the occurrence of a single input event

3.13

* transition

GRAFCET language element indicating a possible evolution of the activity between two or more steps

NOTE The possible evolution is realised by clearing the transition.

3.14

* transition-condition

GRAFCET language element associated with a transition indicating the result of a boolean expression

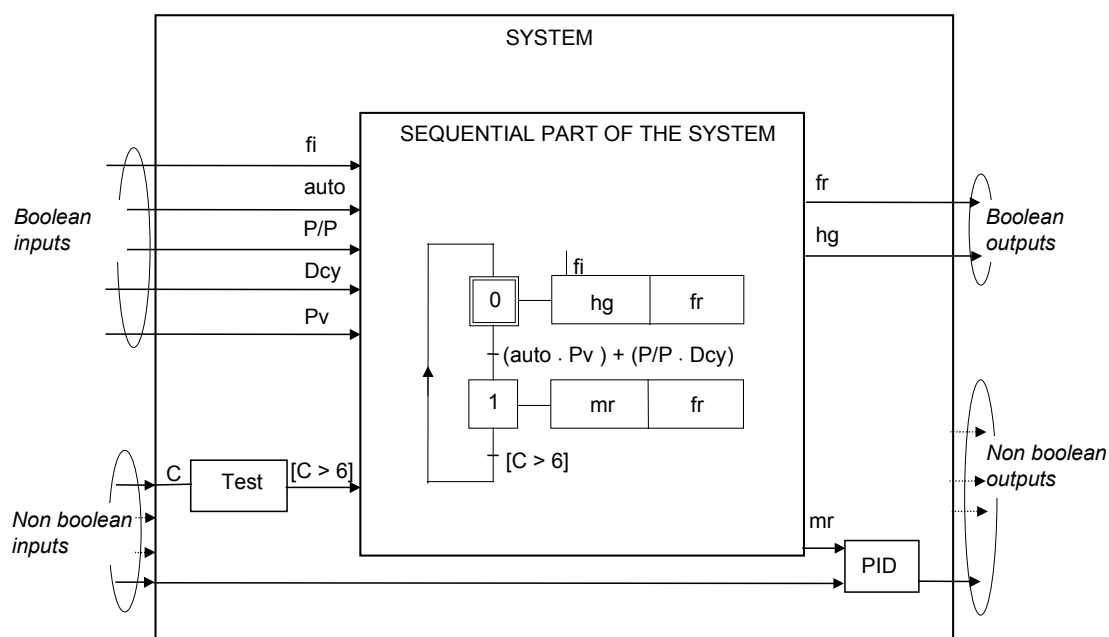
NOTE The transition-condition can be either true or false.

4 General principles

4.1 Context

The implementation of an automated system requires, in particular, a description relating cause and effect. To do this, the logical aspect of the desired behaviour of the system will be described.

The sequential part of the system, which is accessed via Boolean input and output variables, is the logical aspect of this physical system. The behaviour indicates the way in which the output variables depend on the input variables (see note to figure 1). The object of the GRAFCET is to specify the behaviour of the sequential part of the systems.



NOTE The sequential part of the system is characterised by its input variables, its output variables, and its behaviour. This sequential part comprises only input and output boolean variables. However the GRAFCET specification language by extension allows the description of the behaviour of the non Boolean variables (for example: evaluation of an assertion or allocation of a numeric value for a variable).

Figure 1 – Graphical representation of the sequential part of a system

4.2 GRAFCET, a behaviour specification language

The GRAFCET specification language enables a grafcet to be created showing the expected behaviour of a given sequential system. This language is characterized mainly by its graphic elements, which, associated with an alphanumerical expression of variables, provides a synthetic representation of the behaviour, based on an indirect description of the situation of the system.

The behaviour description on states is the following: the "monomarked" states correspond to the GRAFCET *situations*, which implies the uniqueness of the situation at a given instant. The states are connected to each other by means of an evolution condition, which allows the passage from one situation to another one to be described.

For reasons of convenience, the behaviour description based on states is better replaced by a description based on *steps* called GRAFCET. In the GRAFCET, several steps may be active simultaneously, the situation being then characterized by the set of active steps at the considered moment. The evolution of one set of steps to another are translated by one or several transitions, each characterized by:

- its preceding steps,
- its succeeding steps,
- its associated transition-condition.

NOTE These reasons lead to the syntax rule enforcing the alternation step-transition.

4.3 GRAFCET, short presentation

The GRAFCET is used for the design of grafcet charts to provide a graphical and synthetic representation of the sequential systems behaviour. The representation (figure 2) distinguishes:

- the *structure*, which allows possible evolutions between the situations to be described,
- the *interpretation*, which enables the relationship between input, output variables and the structure (evolution, assignation and allocation rules are necessary to achieve this interpretation).

4.3.1 The structure comprises the following basic items

- *Step* (definition: 3.9, symbol 1). A step is either active or inactive, the set of the active steps of a grafcet chart at any given instant represents the *situation* of this grafcet at this instant.
- *Transition* (definition: 3.13, symbol 7). A transition indicates that an evolution of the activity between two or more steps may evolve. This evolution is realized by the clearing of the transition.
- *Directed link* (definition: 3.3, symbol 10). A directed link connects one or several steps to a transition, or a transition to one or several steps.

4.3.2 The following elements are used for the interpretation

- *Transition-condition* (definition: 3.14, symbol 13). Associated with each transition, the transition-condition is a logical expression which is true or false and which is composed of *input variables* and/or *internal variables*.
- *Action* (definition: 3.1). The action indicates, in a rectangle, what shall be done to the *output variable*, either by assignation (continuous action, symbol 20), or allocation (stored action, symbol 26).

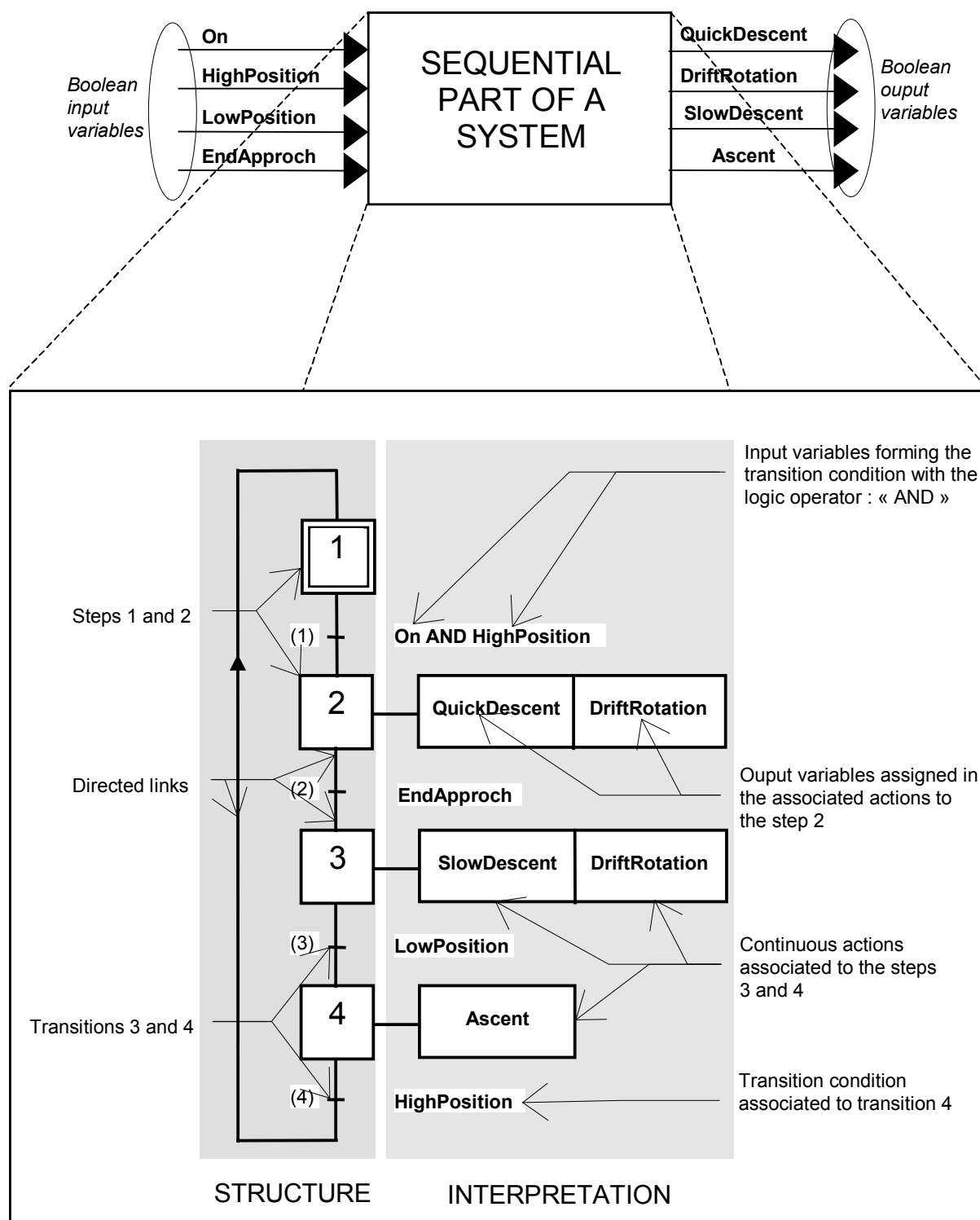


Figure 2 – Structure and interpretation elements used in a grafcet chart to describe the behaviour of a sequential part of the system defined by its input and output variables

4.4 Syntax rule

Step transition and transition step alternation shall always be respected whatever the sequence.

Consequences:

- Two steps shall never be connected directly by a directed link.
- The directed link shall only connect a step to a transition or a transition to a step.

4.5 Evolution rules

As each situation is characterized by the set of active steps at a given instant, the GRAFCET evolution rules only affect the application, on the steps, of the evolution principle between the situations of the sequential part of the system.

4.5.1 Initial situation

The initial situation is the situation at the initial time. Therefore it is described by the set of steps active at this time. The choice of the situation at the initial time depends on the methodology relating to the type of sequential part of the system considered.

Rule 1: The initial situation, chosen by the designer, is the situation at the initial time.

4.5.2 Clearing of a transition

Rule 2: A transition is said to be enabled when all immediately preceding steps linked to this transition are active. The clearing of a transition occurs:

- when the transition is ENABLED,
- AND WHEN its associated transition-condition is TRUE.

4.5.3 Evolution of active steps

Rule 3: The clearing of a transition simultaneously provokes the activation of all the immediate succeeding steps and the deactivation of all the immediate preceding steps.

4.5.4 Simultaneous evolutions

The evolution between two active situations implies that no other intermediate situation is possible, the change from one representation of the situation by a set of steps to another representation is instantaneous.

Rule 4: Several transitions, which can be cleared simultaneously, are simultaneously cleared.

4.5.5 Simultaneous activation and deactivation of a step

If a step is included in the description of the preceding situation and in that of the following one, it can therefore only remain active.

Rule 5: If during the operation, an active step is simultaneously activated and deactivated, it remains active.

4.6 Input events

The evolution rules show that only a change in the values of the input variables may cause the evolution of the grafcet. This change called "input event" shall be defined by the preceding value and the succeeding value of all the input variables to characterise this *single* event. In practice, a set of input events is specified only by the characterised state change (rising edge or falling edge) of one or several boolean input variables.

NOTE The rising edge of a logical variable, indicated by the sign " $\uparrow$ " in front of a boolean variable, indicates that this rising edge is only true for the change from value 0 to value 1 of the variable concerned. The falling edge of a logical variable noted by the sign " $\downarrow$ " in front of a boolean variable, indicates that this falling edge is only true for the change from value 1 to value 0 of the variable concerned.

It is said that "the event occurs" at the date of the change of state of the input variables which characterize it.

4.6.1 Input events specification

The input events specification is implemented by a logical expression of one or several characteristic variables, often in a transition-condition. More rarely, it may also directly specify an internal event (see 4.7).

EXAMPLE 1:
 $\uparrow a$ The expression " $\uparrow a$ " describes the set of all input events for which the preceding value of the input variable a is 0 and its succeeding value is 1, regardless of the value of the other input variables of the system.

EXAMPLE 2:
 $a \cdot \uparrow b$ The expression " $a \cdot \uparrow b$ " describes the set of all input events for which the succeeding value of the input variable a is 1, and the preceding value of the input variable b is 0 and its succeeding value is 1, regardless of the value of the other input variables of the system.

EXAMPLE 3:
 a The expression " a " describes the sets of all input events for which the succeeding value of the input variable a is 1, regardless of the value of the other input variables of the system.

NOTE Used in a transition-condition, this expression could lead to a transient evolution (see 3.12)

4.7 Internal events

Only certain input events could occur from a given situation. The connection between a situation and input event, which may occur from this situation, is called internal event (see 3.6). This notion is mainly used by the designer to condition an output allocation to a set of internal events (see 4.8.2). The description of a set of internal events is realized by one of the following ways.

4.7.1 Internal events described by the step activation

The step activation, noted graphically (symbol 27), describes the set of internal events each of which has this step activation as a consequence.

4.7.2 Internal events described by the deactivation of a step

The graphically noted deactivation of a step, (symbol 28), describes the set of the internal events each of which have this step deactivation as consequence.

4.7.3 Internal events described by the clearing of a transition

The graphically noted clearing of a transition (symbol 29), describes the set of internal events each of which have the clearing of this transition as consequence.

4.8 Output modes

The actions enable links to establish the connection between the evolution of the grafcet chart and the outputs. Two output modes, continuous mode or stored mode, describe how the outputs depend on the evolution and on the system inputs.

4.8.1 Continuous mode (assignation on state)

In the continuous mode, *the association of an action with a step* indicates that an output variable has a true value if the step is active and if the assignation condition is verified. The assignation condition is a logical expression of the input variables and/or the internal ones (see symbol 22). If one of the conditions is not met and provided that no other action relating to the same output meets the conditions, the output variable concerned takes the false value.

Assignation refers to imposing the value of the output variables (true or false).

The set of the local assignation (relating to the active steps at a given instant) defines the assignation of *all* the output variables for this situation.

Assignation rule: For a *given situation*, the value of the outputs relating to the *continuous* actions is assigned:

- to the true value, for each output relating to the actions associated with active steps and for which the assignation conditions are verified,
- to the false value, for the other outputs (which are not assigned to the true value).

4.8.2 Stored mode (allocation on event)

In the stored mode, *the association of an action to internal events* is used to indicate that an output variable takes and maintains the enforced value if one of these events occurs.

Explicit representations are necessary to describe the association of the actions with the events (activation step, deactivation step, clearing of a transition, etc.).

The value of an output relating to a stored action remains unchanged until a new specified event modifies its value.

Allocation refers to storing, at a considered moment, a determined value affected to an output variable.

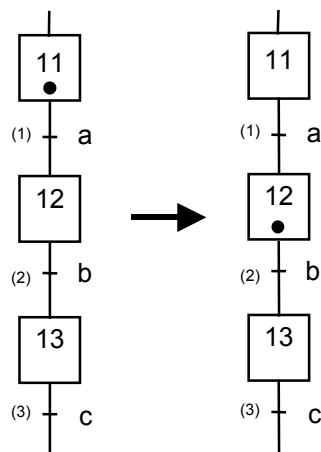
Allocation rule: The value of an output, relating to a stored action and associated to an event, is allocated to the indicated value, if the specified internal event occurs; the value of this output is null at the initialisation.

4.9 Application of the evolution rules

Intuitive interpretation of the evolution, called “step by step”, designates the progressive way which allows, on the occurrence of an input event and from the preceding situation, to determine the succeeding situation of this event, by the successive application of the evolution rules on each transition. The interpretation facility is a device to enable an indirect specification of the evolution, but the designer shall take care that the clearing of the transitions on this path does not involve the effective activation of the intermediate situations.

4.9.1 Non transient evolution

In general, the evolution is non-transient, which means that the input event only leads to one evolution stage (the simultaneous clearing of one or more transitions).



EXAMPLE: "Non transient evolution"

Preceding situation: step 11 active, $a=0$, $b=0$ and $c=0$.

Intuitive interpretation of the evolution:

The change in the value "a" involves the clearing of the transition (1) and the activation of the step 12, the transition (2) can not be cleared, because $b=0$, the subsequent situation is therefore: step 12 active.

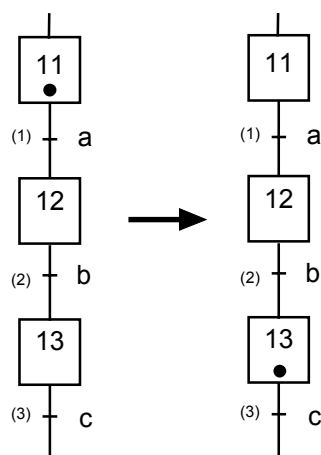
Real interpretation of the evolution:

The occurrence of one of the input events such as the value of a changes from 0 to 1 leads straight to the subsequent situation: step 12 active.

4.9.2 Transient evolution

In some cases, the application of the evolution rules can lead to successively clearing some transitions (in several evolution stages) if the transition-conditions associated with the subsequent transitions are already true, when the first transitions considered are cleared. The corresponding description, referred to as transient, uses the path taken to indicate how to move from a preceding situation to a succeeding situation (see 3.9).

The corresponding intermediate steps, referred to as *unstable* are not activated, but we consider that they have been "virtually" activated and deactivated along the intuitive evolution path, as well as for the corresponding transitions which have been "virtually" cleared.



EXAMPLE: "Transient evolution"

Preceding situation: step 11 active, $a=0$, $b=1$ and $c=0$.

Intuitive interpretation of the evolution:

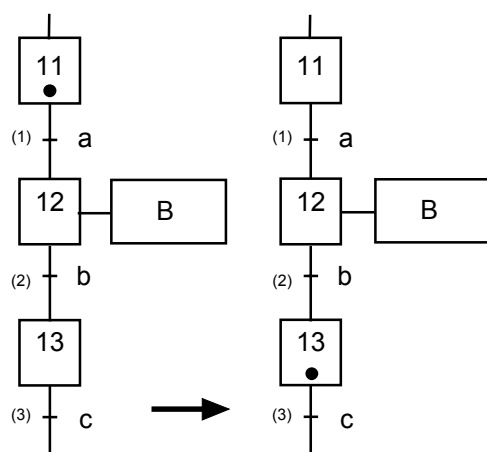
The change in the value "a" involves the clearing of the transition (1) and the virtual activation of the step 12, then the transition (2) is virtually cleared, because $b=1$, leading to the succeeding situation: step 13 active.

Real interpretation of the evolution:

The occurrence of one of the input events, such as the value of a change from 0 to 1, leads to the succeeding situation: step 13 active.

4.9.3 Consequence of a transient evolution on the assignments

The assignment of an output value by a continuous action associated with a step, which is an unstable step in the case of a transient evolution, is not effective, since the step is not really activated (see 4.8.1).



EXAMPLE: “Continuous action associated with an unstable step”

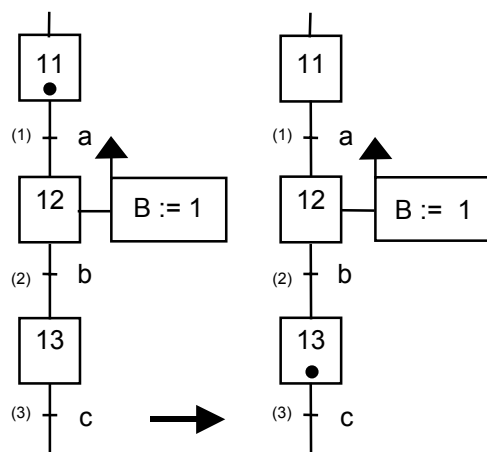
Preceding situation: step 11 active, $a=0$, $b=1$ and $c=0$.

The occurrence of one of the input events such as the value of “a” changes from 0 to 1, leads straight to the subsequent situation: step 13 active.

The preceding situation (step 11 active) and the succeeding situation (step 13 active) assign the value 0 to the output variable B. The unstable step 12 being not really activated, the assignation of B to the value 1 is not effective on the transient evolution.

4.9.4 Consequence of a transient evolution on the allocations

The allocation to a determinate value of an output by a stored action (symbol 26) associated to a step, which is an unstable step in the case of a transient evolution, is effective since this allocation is associated to the events releasing this evolution (see 4.8.2).

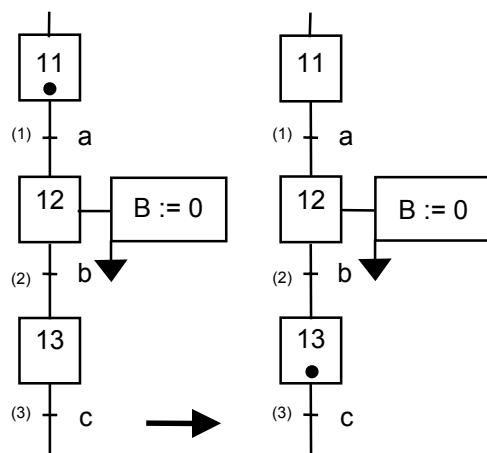


EXAMPLE 1: “Stored action associated with the activation of an unstable step”

Preceding situation: step 11 active, $a=0$, $b=1$ and $c=0$.

The occurrence of one of the input events such as the value of “a” changes from 0 to 1, leads straight to the subsequent situation: step 13 active.

The allocation of the value 1 to the output variable B is realized on the occurrence of one of the input events having the real or the virtual activation of the step 12 as consequence.



EXAMPLE 2: “Stored action associated with the deactivation of an unstable step”

Preceding situation: step 11 active, $a=0$, $b=1$ and $c=0$.

The occurrence of one of the input events such as the value of “a” changes from 0 to 1, leads straight to the subsequent situation: step 13 active.

The allocation of the value 0 to the output variable B is realized on the occurrence of one of the input events having the real or the virtual deactivation of the step 12 as consequence.

4.10 Comparison between the two output modes

The choice of the output mode depends on the practice and methodology used. However the designers attention is drawn to the important differences between the two modes.

4.10.1 Determination of the value of the outputs

- In continuous mode, all the outputs are assigned according to the *situation*, to the *true value* for the outputs explicitly indicated in the actions associated to the active steps, and to the *false value* for the other ones which are implicitly set by omission (see assignation rule, 4.8.1).
- In the stored mode, *only* the considered outputs are modified according to the *indicated value*, the other stored values of the outputs remain unchanged (see allocation rule, 4.8.2).

4.10.2 Analysis of the value of the outputs for a grafcet-chart at a defined instant

- In the continuous mode, the knowledge of the situation and the value of the inputs *is sufficient* to determine the value of the outputs (see 4.8.1).
- In the stored mode, the knowledge of the situation and the value of the inputs *is not sufficient*, the preceding evolutions shall also be known to determine the value of the outputs (see 4.8.2).

4.10.3 Actions relative to transient evolution

- In the continuous mode, the actions associated with an unstable step *are not taken* into consideration because this step is not activated (see 4.9.1).
- In the stored mode, the actions associated with events and in relation with a transient evolution *are taken* into consideration because the triggered events releasing this evolution occur (see 4.9.2).

4.10.4 Possible conflict on the value of the outputs

- In the continuous mode, the assignation principles ensure every assignation conflict on the particular output to be avoided.
- In the stored mode, the allocation rules do not *allow the possible assignation conflicts on a same output to be avoided*. The designer shall ensure that two contradictory allocations can not occur simultaneously.

NOTE 1 Both output modes can be used in one specification in GRAFCET, but the value of an output variable is determined either by assignation or by allocation. The specification of an allocation to an output variable (stored mode), excludes this output variable of any assignation (continuous mode).

NOTE 2 Clause 5 gives the graphic symbols which enable the stored actions (indicated by explicit representation according to the set of specified events) to be distinguished from the continuous ones (indicated by absence of any representation).

NOTE 3 In the frequent case of the specification of control system behaviour, the current industrial practice forces the employment of the continuous mode for all the outputs to the actuators, and the stored mode for describing internal control tasks. These tasks, such as the incrementation of a counter, or the modification of the value for a numerical register, refer to internal variables, which are not necessarily Boolean ones. The internal tasks associated with the stored actions, as well as the calculation of expressions associated with transition-conditions, are not described in the present standard, but are associated by the use of the logical description of the grafcet evolutions. The designers should take care to make their descriptions consistent and clear.