13.12.2004

SIEMENS

SIMATIC

SFC for S7
Sequential Function Chart

Manual

Edition 01/2005
A5E00345250-01

13.12.2004

Safety Guidelines

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring to property damage only have no safety alert symbol. The notices shown below are graded according to the degree of danger.

Danger

indicates that death or severe personal injury **will** result if proper precautions are not taken.

Warning

indicates that death or severe personal injury **may** result if proper precautions are not taken.

Caution

with a safety alert symbol indicates that minor personal injury can result if proper precautions are not taken.

Caution

without a safety alert symbol indicates that property damage can result if proper precautions are not taken.

Attention

indicates that an unintended result or situation can occur if the corresponding notice is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel.** Within the context of the safety notices in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Prescribed Usage

Note the following:

Warning

This device and its components may only be used for the applications described in the catalog or the technical description, and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens.
Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG.
The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Siemens AG
Automation and Drives
Postfach 4848, 90327 Nuremberg, Germany

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Preface

Purpose of the Manual

This manual supports you when you create sequential control systems and parameter controls. It provides you with an overview of the following:

- The basics of sequential control systems
- Working with the SFC Editor
- Sequential control systems on the programmable controller
- Putting sequential control systems into operation and monitoring and testing them
- Documenting SFC charts

You will find a detailed description of the software and procedures in the **SFC online help**.

This manual "SFC for S7" provides you with the information you require to use the SFC configuration tool in conjunction with CPUs in SIMATIC S7 programmable controllers (PLCs). If you use other target systems (for example, SIMADYN D), please read the **additional documentation** for this target system.

How Sections for Specific Systems are Indicated

If sections, paragraphs or even individual sentences in this S7 manual relate solely to S7 users, this is indicated by **[S7]**. This means that the information is relevant only to S7 or is different in other systems. In this case, if you use a different PLC, you will find the information you require in the manual for your specific system.
If the **[S7]** label is in a title, the entire section applies only to S7; if the label is at the start of a paragraph, the paragraph is solely relevant to S7. In lists, the **[S7]** label applies only to the particular list.

Audience

This manual is intended for personnel involved in configuring, commissioning, and service.
Basic experience of working with PCs and Windows is assumed.

Validity

This manual is valid for the SFC software version V6.0 or higher.
You will find the latest information that could no longer be included in this manual along with instructions on installation in the README.TXT file accompanying the product.

Standard

The SFC software is based on the international standard DIN EN 61131-3 (IEC 1131-3) for programming languages for programmable logic controllers.

Further Support

If you have any technical questions, please get in touch with your Siemens representative or agent responsible.

You will find your contact person at:

http://www.siemens.com/automation/partner

You will find a guide to the technical documentation offered for the individual SIMATIC Products and Systems here at:

http://www.siemens.com/simatic-tech-doku-portal

The online catalog and order system is found under:

http://mall.ad.siemens.com/

Training Centers

Siemens offers a number of training courses to familiarize you with the SIMATIC S7 automation system. Please contact your regional training center or our central training center in D 90327 Nuremberg, Germany for details:

Telephone: +49 (911) 895-3200.

Internet: http://www.sitrain.com

Technical Support

You can reach the Technical Support for all A&D products

- Via the Web formula for the Support Request
  http://www.siemens.com/automation/support-request
- Phone: + 49 180 5050 222
- Fax: + 49 180 5050 223

Additional information about our Technical Support can be found on the Internet pages http://www.siemens.com/automation/service

Service & Support on the Internet

In addition to our documentation, we offer our Know-how online on the internet at:

http://www.siemens.com/automation/service&support

where you will find the following:

- The newsletter, which constantly provides you with up-to-date information on your products.
- The right documents via our Search function in Service & Support.
- A forum, where users and experts from all over the world exchange their experiences.
- Your local representative for Automation & Drives.
- Information on field service, repairs, spare parts and more under "Services".

Contents

1 Basics of Sequential Control Systems

Introduction

This chapter explains the basics of sequential control systems.

Here, you will learn what a sequential control system is and what it is used for. You will get to know the terminology and elements of SFC and the rules governing the topology of sequencers.

1.1 Introduction

What is SFC?

SFC allows the graphic configuration and commissioning of sequential control systems (SFC = sequential function chart). These are transferred to an automation system and executed there. A sequential control system allows status- or event-driven execution of production processes based on sequencers.

Sequential control systems can be used, for example, to describe the manufacture of products as event-controlled processes (recipes).

With a sequential control system, functions from basic automation (typically created with CFC) are controlled by operating and state changes and executed selectively.

SFC provides two independent variants of sequential control systems for different application scenarios:

- SFC chart
- SFC type with SFC instances.

SFC Chart

An SFC chart contains a standardized interface to allow control of the SFC by the user program or by the user and up to 8 sequencers for formulating the sequential control system ("gray sequencer"). The sequential control system accesses the blocks and signals of basic control directly and is therefore not reusable.

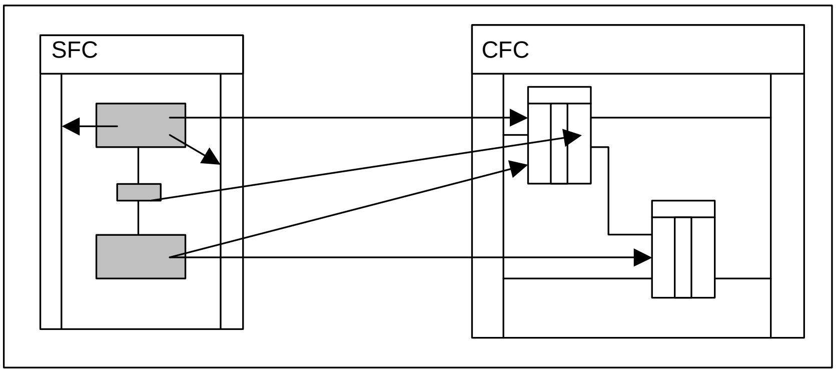


Fig. 1-1 : Interaction of an SFC Chart and CFC

SFC Type / SFC Instance

An SFC type contains a standardized interface to allow control of the SFC by the user program or by the user and up to 32 sequencers for formulating the sequential control system ("gray sequencer"). The interface can also be expanded by the user ("gray interface"). The SFC type accesses only its own interface and can therefore be used as often as required as an SFC instance.

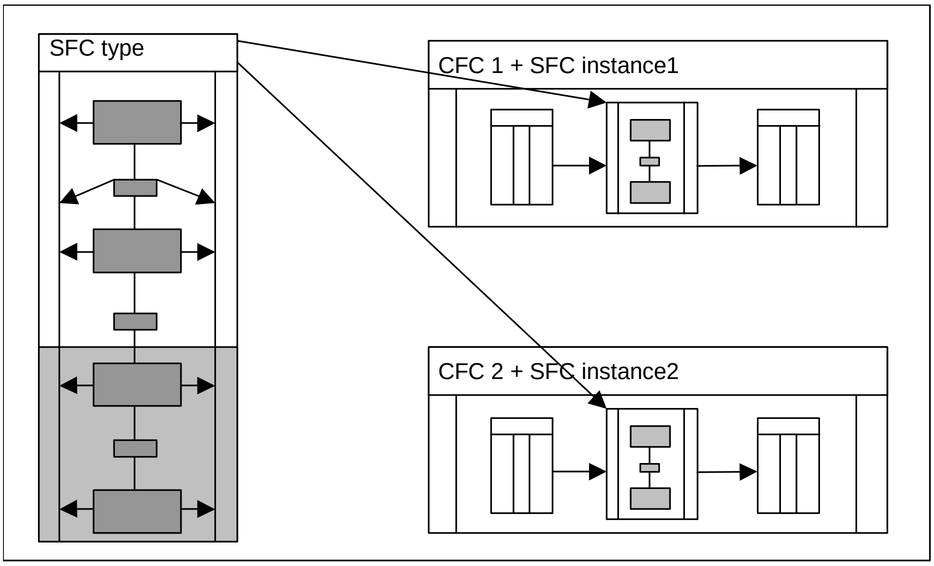


Fig. 1-2 : Interaction of SFC Type/Instance and CFC

An SFC instance is derived from an SFC type and initially has properties identical to those of the SFC type. By making changes to the SFC instance in CFC or SFC, a limited range of instance-specific adaptations can be made.

Selection Criteria for SFC Chart or SFC Type

An SFC chart is used when a sequential control system is required that will be used once and that accesses several subareas when controlling the production plant.

An SFC type is used when a sequential control system is required that will be used more than once and accesses its own subarea when controlling the production plant. Changes to an SFC type are made centrally in the SFC type and automatically passed on to all SFC instances.

The criteria for deciding which of the sequential control variants to use are therefore in the following order:

- Area to be influenced
- Reusability
- Modifiability

If, for example, the area to be influenced is extensive, but the sequential control system must be capable of being used more than once, it may nevertheless be necessary to implement the sequential control system as an SFC chart and to accept that reusability is possible only in the form of copies. In this case, there is then naturally no chance of central modification.

On the other hand, if the area of influence is not extensive, both an SFC chart and an SFC type can be used. Here, the decision depends on the other criteria.

What is a sequential control system or a sequencer?

A sequential control system allows the structuring of production processes by breaking them down to consecutive steps. The steps define the actions to be executed and therefore the activities required in the production plant, for example:

- Turn motor on,
- Open valve etc.

The passing of control from one step to the next is triggered by a transition with a defined step enabling condition, for example:

- Temperature reached,
- Reactor empty, etc.

As a higher-level structuring tool, sequencers can be used that define state- or event-dependent subsequences, for example:

- Sequence for production,
- Sequence for holding,
- Sequence for error handling, etc.

The relationship between the status or event and execution of the sequencer is defined in the start condition of the sequencer. The sequencer itself is formulated as a series of steps and transitions.

A sequential control system contains at least one sequencer. Its start condition is initialized so that the sequencer executes when the sequential control system is in the "active" status (condition: RUN = TRUE). This sequencer then represents the actual production procedure. The start condition can, of course, also be modified.

Where are sequential control systems used?

The typical applications of sequential control systems involve processes and plants with discontinuous characteristics. Sequential control systems can, nevertheless, also be used for continuous processes and plants, for example for approach and withdrawal movements, operating point changes, and state changes due to faults etc.

Such systems can be used at various levels of a process or plant:

- Device control level (open valve, start motor .....)
- Group control level (dosing, stirring, heating, filling .....)
- Unit level (tank, mixer, scales, reactor .....)
- Plant level (synchronization of units and common resources, for example routing)

SFC charts can be used at all levels, while SFC types are generally used at the group control and unit levels since requirements for reusability at the higher levels are more unlikely.

Terms

The SFC editor is a tool for creating sequential control systems.

In these descriptions, the term "SFC" is used both as a chart, type, or the editor depending on the context.

How the Editor Works

Using the SFC editor, you create your sequential control system using graphic tools. The elements of the SFC are positioned in the sequencer according to fixed rules. You do not need to be aware of details such as algorithms or the assignment of machine resources but can concentrate solely on the technological aspects of your configuration.

After creating the sequencers as sequences of steps and transitions, you change to the configuration of the object properties of SFC, sequencers, steps and transitions and formulate the required properties; in other words, you configure the operating parameters of the SFC, the start conditions of the sequencers and the actions of the steps and the step enabling conditions of the transitions.

After configuration, you compile the executable machine code with SFC, download it to the CPU, and test it with the SFC test functions.

1.2 Getting Started

Where is the best place to start?

If you have never worked with SFC, you should start with an SFC chart to become familiar with the basics step by step. Based on this you can try out

- the handling; in other words, creating, configuring, compiling, and downloading a sequential control system,
- the functionality of a sequential control system on the CPU and
- testing; in other words, starting, aborting, ... a sequential control system

Once you know how to use sequential control systems, you should take a closer look at SFC types and SFC instances so that you can later decide which of the two sequential control variants is most suitable for a concrete task.

The SFC type and SFC instance introduce in new aspects in working with sequential control systems. With the SFC chart, you can get going immediately; in other words, an SFC chart can be generated, compiled, downloaded and tested immediately.

With an **SFC type**

- you create an abstract sequential control system not directly related to a concrete application as a typical template with the aim of using it more than once. Only after generating and adapting an SFC instance and interconnecting it with blocks of basic control do you have a situation comparable with that when using an SFC chart. You can now
- compile, download, and test the SFC instance.
  Indirectly, you are, of course, also testing the SFC type.

To interconnect an **SFC instance** with basic control, an interface must be created for the SFC type. At the interface of the SFC type, you can

- add new I/Os directly or
- create the required I/Os technologically as characteristics of the SFC type.

You can, of course, create and test the sequential control system step by step both when using an SFC chart as well as with an SFC type (SFC instance).

Procedure

See also:

Procedure: Creating an SFC Chart, Section 1.2.1

Procedure: Creating an SFC Type, Section 1.2.2

Procedure: Creating an SFC Instance, Section 1.2.3

1.2.1 Procedure: Creating an SFC Chart

The order of the steps outlined below represents the procedure for configuring SFC charts:

1. **Create the project structure:**
   Create a project structure in the SIMATIC Manager in which you can configure CFC/SFC charts.
   See: "Creating the Project Structure", Section 1.3.2.
2. **Specify the chart properties:**
   When you specify the chart properties, you can change the chart name and add a comment.
   See: "Adapting the Chart Properties", Section 2.5.1.
3. **Adapt the operating parameters and runtime properties**
   By setting the operating parameters, you specify the behavior of the sequential control system, such as the mode (manual, auto), step control mode (T, C, T and C...) and other chart execution options (cyclic operation, time monitoring, autostart, ...).
   See: "Adapting the Chart Properties", Section 2.5.1.

   The runtime properties of an SFC chart determine how the SFC chart is included in the execution of the entire structure on the CPU (in the window of the CFC runtime editor).
   See: "Runtime Properties", Section 2.7.
4. **Create sequencers:**
   The newly created SFC already contains a sequencer (RUN). You create further sequencers with the context-sensitive menu (right mouse button) using the tab of the current sequence or by copying sequencers from the templates of the SFC Library.
   See: "Creating the Sequencer Topology", Section 2.8.1.
5. **Configure sequencer properties:**
   For each sequencer, you configure the start condition and, as an option, the action for preprocessing and postprocessing and, if required, the OS comment.
   See: "Configuring Multiple Sequencers", Section 2.8.2.
6. **Create the topology of the sequencers:**
   The SFCs are used to configure sequential control systems by inserting the steps and transitions for one or more sequencers and adding further elements as necessary.
   See: "Creating SFC Elements", Section 2.9.
7. **Configure steps** (in the Object Properties dialog)**:**
   Actions are formulated in the steps. The actions contain statements with which the values of block inputs and of shared addresses can be changed or runtime groups or other SFC charts can be activated and deactivated.
   See: "Editing Object Properties: Steps", Section 2.11.2.
8. **Configure transitions** (in the Object Properties dialog)**:**
   Conditions are formulated in the transitions. The conditions read the values of block I/Os, of shared addresses or, for example, the state (active/inactive) of runtime groups or other SFC charts. When the conditions of a logic operation are met, the following step becomes active during execution of the SFC and its actions are executed. See: "Editing Object Properties: Transitions", Section 2.11.3.

9. **Compile program:**
   During compilation, the CFC and SFC charts of the active chart folder are converted to an executable user program (compile entire program).
   See: "Compile", Section 2.12.1.

10. **Download program:**
    After compilation, you can download the program to the target system (CPU).See: "Download", Section 2.12.2.

11. **Test the program:**
    After compiling and downloading, you can test the program in process mode or in laboratory mode. Using the SFC test functions, you can run the sequential control system in various operating modes and step control modes and monitor and modify the values of addresses on the CPU. You can also influence the most important operating modes (STOP, clear/reset, RUN, ...) on the CPU.
    See: "Testing and Commissioning", Section 4.

Note:

Prior to compiling (point 9), you must interconnect the external view of the SFC chart or the SFC instance (of the SFC type) for AUTO mode in the CFC chart.

See also: "What is the external view of an SFC chart", Section 1.4.4.

Tips:

- In the "SFC Library", you will find the "ChartStates" SFC chart in SFC Library > Blocks+Templates > Templates. This already contains several sequencers for state-oriented processing of the sequential control system. You can copy this chart and use it as an example.
- For a first run through from creating to testing a sequential control system, you can compile, download and test the SFC chart "ChartStates" (or a newly created SFC chart) unchanged. The SFC chart is processed in MAN mode; in other words, you can monitor and influence the processing of the SFC chart in test mode with the available test functions.
- To stop the SFC chart from being processed "too quickly" in the automation system, you can, when necessary, move the SFC chart in the run sequence, for example to the "OB32" task. In the test settings (Debug menu >Test Settings) in SFC, you can select a suitable monitoring cycle.

1.2.2 Procedure: Creating an SFC Type

The order of the steps outlined below represents the procedure for configuring SFC types:

1. **Create the project structure:**
   Create a project structure in the SIMATIC Manager in which you can configure CFC charts and SFCs.
   See: "Creating the Project Structure", Section 1.3.2.

   Note: You can create an SFC type not only in the project structure but also in a library.

2. **Create an SFC Type**
   In the Component view of the SIMATIC Manager, insert an SFC type as a new object in the chart folder. The SFC type is entered in the block catalog of CFC.
   See: "Creating an SFC type", Section 2.4.2.

   Tip: In the "SFC Library", you will find the "TypeStates" SFC type in SFC Library > Blocks+Templates > Templates. This already contains several sequencers for state-oriented processing of the sequential control system. You will also find the "TypeCtrlStrategy" SFC type containing control strategy-oriented processing of the sequential control system. You can copy these templates and adapt them to your own purposes.

   Adapt SFC properties
   You can modify the general properties (name, author, family, FB number, comment), adapt the operating parameters and change the version.
   By setting the operating parameters, you specify the behavior of the sequential control system, such as the mode (manual, auto), step control mode (T, C, T and C...) and other chart execution options (cyclic operation, time monitoring, autostart, ...).
   See: "Adapting the Type Properties", Section 2.5.2.

   Create the interface
   The SFC type already has a standard interface that must normally be extended for the interconnections of SFC instances with basic control.
   See: "Using the Characteristics Editor and the Interface Editor", Section 2.2.3.

   For more detailed information, refer to the SFC online help topics:

   The standard interface of the SFC type
   Interface expansions "Inputs/Outputs"
   Interface expansions "Characteristics"

3. **Create sequencers:**
   First open the SFC type. The newly created SFC already contains a sequencer (RUN). You create further sequencers with the context-sensitive menu (right mouse button) using the tab of the current sequence or by copying sequencers from the templates of the SFC Library.
   See: "Creating the Sequencer Topology", Section 2.8.1.

4. **Configure sequencer properties:**
   For each sequencer, you configure the start condition and, as an option, the action for preprocessing and postprocessing and, if required, the OS comment.
   See: "Configuring Multiple Sequencers", Section 2.8.2.

5. **Create the topology of the sequencers:**
   The SFCs are used to configure sequential control systems by inserting the steps and transitions for one or more sequencers and adding further elements as necessary.
   See: "Creating SFC Elements", Section 2.9.

6. **Configure steps** (in the Object Properties dialog)**:**
   Actions are formulated in the steps. The actions contain statements with which the values of inputs of the chart's own interface can be changed.
   See: "Editing Object Properties: Steps", Section 2.11.2.

7. **Configure transitions** (in the Object Properties dialog)**:**
   Conditions are formulated in the transitions. The conditions read the values of I/Os of the charts own interface. When the conditions of a logic operation are met, the following step becomes active during execution of the SFC and its actions are executed.
   See: "Editing Object Properties: Transitions", Section 2.11.3.

8. **Create SFC instances**
   See also: "Procedure: Creating an SFC Instance, Section 1.2.3

1.2.3 Procedure: Creating an SFC Instance

The order of the steps outlined below represents the procedure for configuring SFC instances:

1. **Open the CFC chart**
   Open the CFC chart in which you want to interconnect an SFC instance with the blocks of basic control.

   From the block catalog of CFC or from a library, place the SFC type in the chart to create an SFC instance.
   See: "The SFC Instance", Section 2.3.

2. **Specify the SFC instance properties**
   In CFC, you can change the general properties (name, comment) in the object properties of the SFC instance.
   See: "Adapting the Instance Properties", Section 2.5.3.

3. **Adapt operating parameters and options of the instance**
   In CFC, open the SFC instance and in the Properties dialog, adapt the operating parameters that decide how the instance executes on the CPU.
   As an option, you can decide which of the control strategies provided by the SFC type should be used for the SFC instance.
   See: "Adapting the Instance Properties", Section 2.5.3.

4. **Assign parameters and interconnect the SFC instance**
   You make the parameter setting for the I/Os of the SFC instance in CFC using the Object Properties or in SFC using the Interface Editor "Inputs/Outputs".

   In CFC, you interconnect the I/Os of the SFC instance with the I/Os of the CFC blocks or with shared addresses or you create textual interconnections.

5. **Compile and download program:**
   When you compile the current chart folder, an executable user program is generated (entire program) that you can then download to the CPU.
   See: "Compiling and Downloading", Section 2.12.

6. **Test the program:**
   After compiling and downloading, you can test the program in process mode or in laboratory mode. Using the test functions, you can run the sequential control system in various operating modes and step control modes and monitor and modify the values of addresses on the CPU. You can also influence the most important operating modes (STOP, RUN, clear/reset, ...) on the CPU.
   See: "Testing and Commissioning", Section 4.

1.3 SFC in the STEP 7 Environment

The **SIMATIC manager** is the graphic user interface used for all CPUs and to coordinate the tools and objects. It manages tools and data and is used, among other things, to created and modify a project structure (CPU, CFC/SFC charts) and to start the SFC Editor.

Fig. 1-3 shows how SFC fits into the STEP 7 and PCS 7 environment:

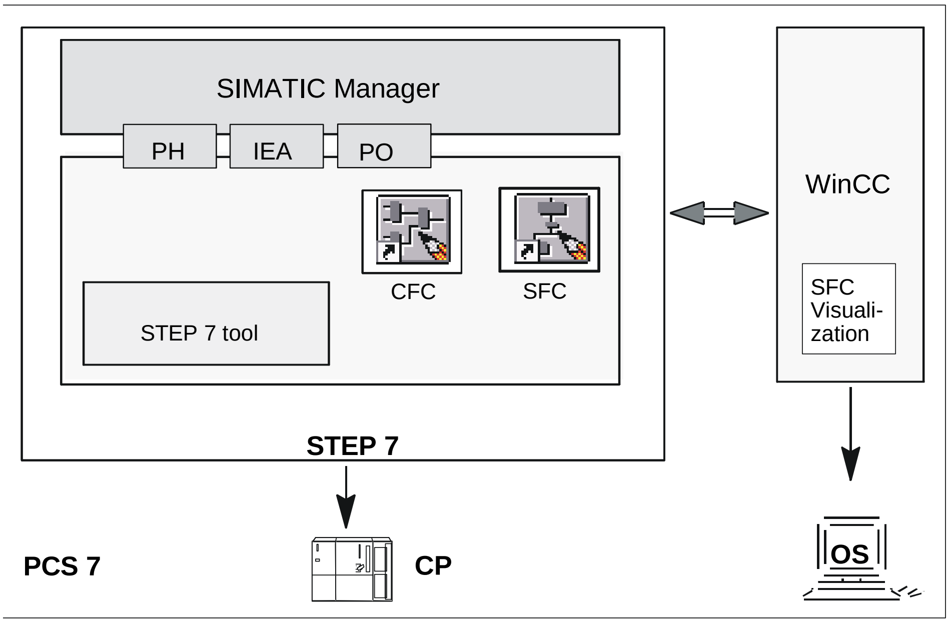


Fig. 1-3 : SFC in the STEP 7 / PCS 7 Environment

Key:

PH (plant hierarchy), **IEA** (Import/Export Assistant) and **PO** (process object view) are components of the Process Control System (PCS 7) and expand the SIMATIC Manager. **WinCC** is the operator control and monitoring system in PCS 7 (here with the optional package for SFC Visualization).

1.3.1 SFC and Plant Hierarchy

The plant hierarchy (PH) allows charts to be arranged and managed not only from the point of view of running them on a CPU, but also according to technological or plant criteria (for example an SFC chart for device control, group control, or unit control).

If the SFC chart was assigned to a plant hierarchy folder, the path of the plant hierarchy is added to the chart name. You can use the naming scheme of your plant as the criteria for arranging charts in the project.

For more detailed information on the plant hierarchy, refer to the online help of the PH.

1.3.2 Creating the Project Structure

You can create a project in the following ways:

1. Start the SIMATIC Manager and then the PCS 7 Wizard (if this is not already started automatically) by selecting "File > 'New Project' Wizard...". Work through the dialogs displayed by the WizardIn the ”Which other objects are you using?” dialog, among others, the "SFC chart" option is preset in the "CPU objects" box.

   The wizard creates a single project or multiproject in the plant view and in the component view. Apart from the actual project, a multiproject also includes a master data library.
   During the creation of the project, certain defaults are set that you can modify later if necessary.

2. The STEP 7 Wizard was started in the SIMATIC Manager. (The setting PCS 7 or STEP 7 is made with "Options > Customize..." in the "Wizards" tab of the dialog.)
   Using this wizard, you create a STEP 7 project; in other words the SIMATIC station, the CPU, the S7 program and the block folder.

   - Select the menu command "Insert > S7 Software > Chart Folder" and insert a chart in the chart folder.

   - Select the chart folder and then the menu command "Insert > S7 Software > SFC".

3. Create a new project in the SIMATIC Manager using menu commands.

   - Select "File > New...", enter the name of the project in the dialog box and confirm with "OK".

   - Insert an S7 program in the project (Insert > Program > S7 Program).

   - Insert a chart folder in the S7 program (Insert > S7 Software > Chart Folder).

   - Insert a chart in the chart folder (Insert > S7 Software > SFC).

The recommended procedure is to use the **PCS 7 Wizard** (point 1) because this is both faster and is less liable to errors.

1.4 SFC Chart, SFC Type, SFC Instance and External View

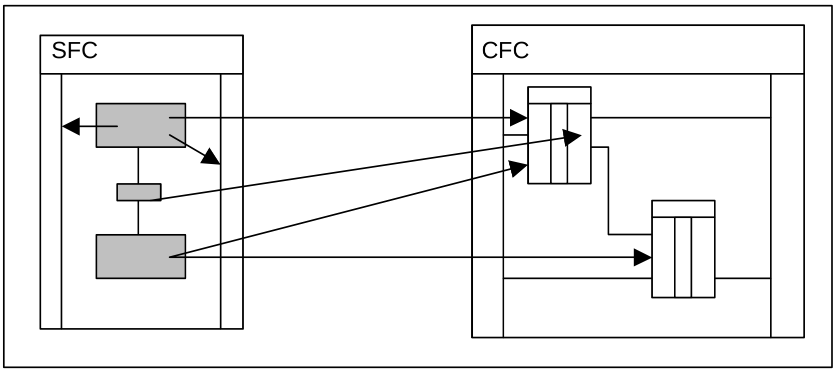
Overview

Below, you will see detailed information on the usage and functionality of an SFC chart, SFC type, SFC instance, and the external view of the SFC chart.

1.4.1 What is an SFC chart?

An SFC chart is a sequential control system that normally controls a larger subarea of a plant and is only used once since the same control task does not occur again in the plant.

The SFC chart has a series of properties, contains a standardized interface for user program or user control of the SFC and allows the configuration of up to 8 sequencers with which the control function is formulated ("gray sequencer"). The sequential control system accesses the blocks and signals of basic control directly and is therefore not reusable.



The properties of the SFC chart include name, author and comment as descriptive data and the operating parameters for the CPU that decide the execution (for example mode, step control mode, execution options).

See also: "How an SFC Executes", Section 3.2.

You can also decide whether or not the SFC chart can be controlled and monitored on the OS (requires the optional package "SFC Visualization" on the OS). As further properties, you can configure the message properties and the footer data of the SFC chart.

The standard interface of the SFC chart includes the necessary I/Os for the user program to control the SFC chart in AUTO mode or for the user to control and monitor the chart in MAN mode. The control of the SFC chart in AUTO mode is configured in the external view of the SFC chart.

See also: "What is the external view of the SFC chart?", Section 1.4.4.

You will find detailed information on the standard interface of the SFC chart in the online help of SFC in the topic: "The Standard Interface of the SFC Chart".

The actual sequential control system is formulated by the sequencers.

See also: "What is a sequencer?", Section 1.5.1.

To configure the sequential control system, you must know the mechanisms for processing an SFC and its sequencers on the CPU.

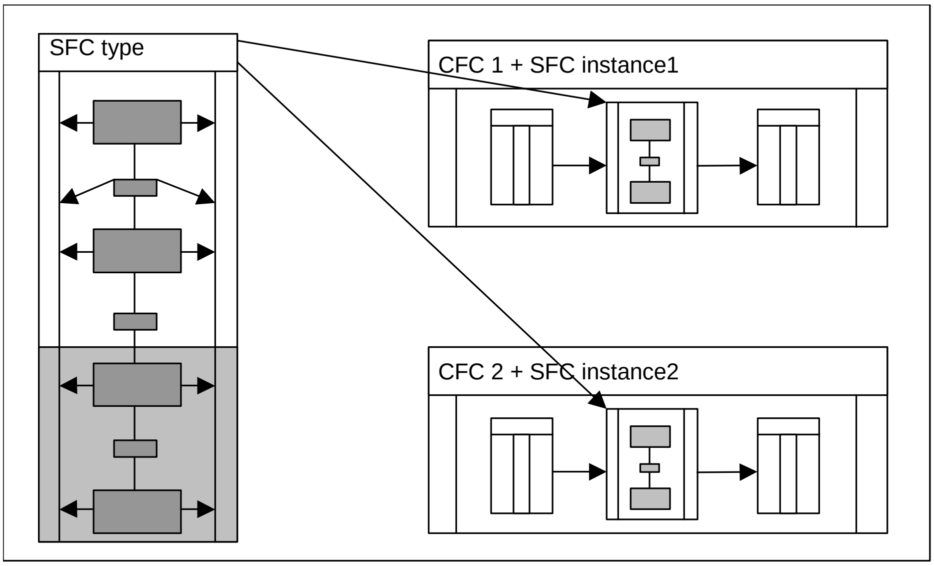
See also: "How an SFC Executes", Section 3.2 and "The Behavior of a Sequential Control System", Section 3.3.

1.4.2 What is an SFC type / an SFC instance?

An SFC type is a sequential control system that normally controls a smaller subarea of a plant (unit, part of a plant) when the control task occurs repeatedly in similar form in the plant. The SFC type therefore creates an abstract sequential control system not directly related to a concrete application as a type-defined template. After it has been created, the SFC type can be used repeatedly by generating SFC instances from it.

To allow this, the SFC type has a series of properties. It contains a standardized interface to allow control of the SFC by the user program or by the user and allows configuration of up to 32 sequencers for formulating the control function ("gray sequencer"). The interface can also be expanded by the user ("gray interface"). The SFC type accesses only its own interface and can therefore be used as often as required as an SFC instance.

An SFC instance is derived from an SFC type and initially has properties identical to those of the SFC type. By making changes to the SFC instance in CFC or SFC, a limited range of instance-specific adaptations can be made. Only after interconnecting the SFC instance with blocks and signals of basic control, do you have an executable sequential control system.



The properties of the SFC type and SFC instance include the name and comment as descriptive data and the SFC type also has the properties author, version and family and FB number as information for the compilation. Both for the SFC type and the SFC instance, you can specify the operating parameters for the CPU that determine how the instance executes on the CPU (for example mode, step control mode, execution options).

You can also set further options and specify whether the SFC type and SFC instance are relevant for SIMATIC BATCH and which of the defined control strategies will be released for use with the SFC type or SFC instance. As further properties, you can configure the message properties and the footer data of the SFC type and SFC instance.

SFC types and SFC instances can be made available on the OS (default: Yes), so that the SFC instances can be controlled and monitored there with the "SFC Visualization" optional package. If you do not want SFC instances to be available on the OS, you must remove the "OCM possible" property from the instance.

The standard interface of the SFC type includes the necessary I/Os for the user program to control an SFC instance in AUTO mode or for the user to control and monitor the instance in MAN mode.

You will find detailed information on the standard interface of the SFC type in the online help of SFC in the topic: "The Standard Interface of the SFC Type".

To interconnect an SFC instance with basic control, the interface must normally be expanded in the SFC type. You can do this by adding new I/Os to the interface of the SFC type directly or can create the required I/Os technologically as characteristics of the SFC type. The latter should be the preferred method. See also: "Using the Characteristics Editor and the Interface Editor", Section 2.2.3 and "Configuring with the Characteristics Editor", Section 2.2.4.

The actual sequential control system is formulated by the sequencers. See also "What is a sequencer?", Section 1.5.1.

To configure the sequential control system, you must know the mechanisms for processing an SFC and its sequencers on the CPU.

See also: "How an SFC Executes", Section 3.2 and "The Behavior of a Sequential Control System", Section 3.3.

1.4.3 SFC Chart and SFC Type Compared

What do an SFC chart and SFC type have in common?

- Standard interface for external control of the SFC (MAN/AUTO)
- Sequencers for formulating the control functions of the SFC

What is the difference between an SFC chart and SFC type?

SFC chart	SFC type
Direct access to basic control	Access to basic control over interface
Can be used once	Can be used more than once
Can be modified locally	Can be modified centrally
Interface cannot be expanded	Interface can be expanded
Maximum of 8 sequencers can be configured	Maximum of 32 sequencers can be configured

1.4.4 What is the external view of an SFC chart?

Display

The external view of the SFC chart represents the SFC chart in a **CFC chart** as a block; in other words, with its interface. To distinguish it from CFC blocks and hierarchical charts, the external view has the "SFC chart" icon in the header.

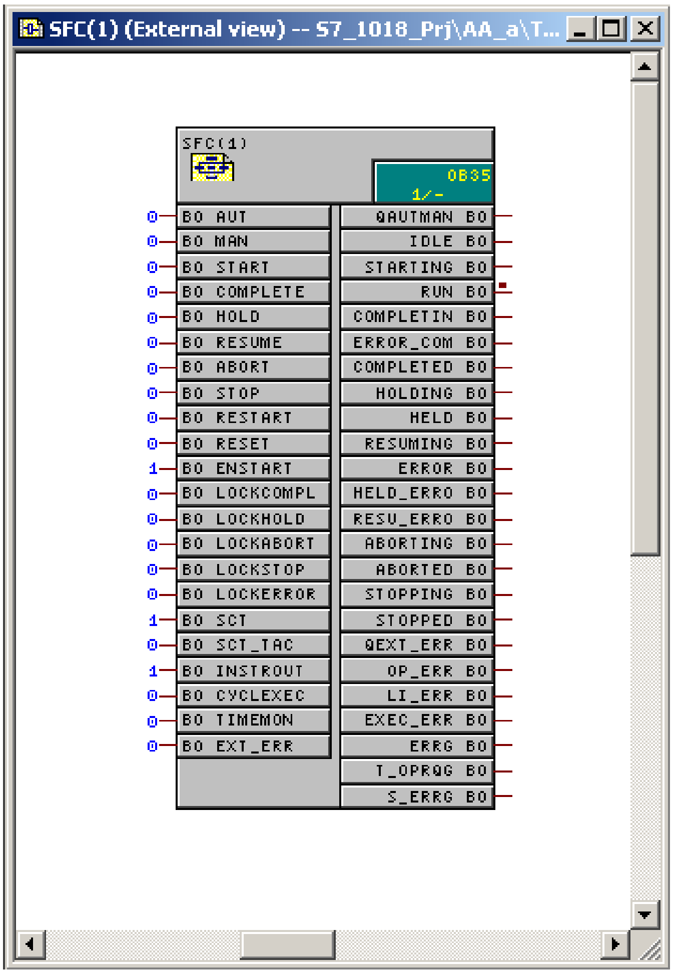


Fig. 1-4 : External View of the SFC Chart

Purpose

In the external view, the interface of the SFC chart can have parameters set and can be interconnected for the AUTO mode so that the status of the SFC chart and the control signals derived from it can be obtained for the SFC chart. This allows the processing of the SFC chart to be influenced by the user program. If the SFC chart is used in MAN mode, only the I/Os that are also processed in MAN mode are of significance in the external view.

For a description of the interface of the SFC chart, refer to the online help of SFC.

Interconnection

You can assign textual interconnections to the I/Os and/or interconnect them with compatible I/Os of other objects or with shared addresses. All interconnections are made using the sheet bar; in other words, you cannot place any objects (blocks etc.) in this window.

Properties

You can display the object properties for the entire interface (double-click in the header of the external view) or for the individual I/Os (double-click on an interconnection).

If you double-click in the box of the runtime properties, the window for the run sequence opens.

1.5 SFC Elements

1.5.1 What is a sequencer?

With sequencers, status-dependent and event-driven execution is possible in SFC.

When it is created, each sequencer is given a consecutive number. This is required for the "programmed target steps" and to interpret the outputs of the SFC in the CFC view.

An SFC chart can include up to 8 sequencers and an SFC type up to 32 sequencers that can be controlled by defining different **start conditions**.

One sequencer is displayed in the working window of the SFC. You can change to a different sequencer easily, using the tabs at the bottom edge of the window.

When you create a new chart/type, a sequencer with the name "RUN" and the start condition RUN=TRUE is created (Note: The corresponds to a chart of V5). The start conditions are formulated in the same way as the transition conditions (see Section 1.5.4, "What is a Transition?"). An empty start condition, in contrast to the transition, is evaluated as FALSE; in other words, the sequence path is never executed.

Each sequencer contains not only this start condition but also the "Priority" attribute with which a start order can be specified when there are simultaneously satisfied conditions for several sequencers ("Start Condition" tab in the "Sequencer Properties" dialog). If the priority is the same and the condition is satisfied, the position of the tab decides the order in which they are processed on the CPU (analogous to the alternative sequence, see Section 3.3.6, Processing an Alternative Sequence).

You can also configure an additional action for each sequencer. The action consists of a part known as the preprocessing that executes after the start of the sequencer in every cycle before the steps and transitions are processed, and a part that executes after processing of the steps and transitions known as the postprocessing. This, for example, allows you to make prior settings or pass on the results of sequencer execution.

1.5.2 What are Sequence Path Elements?

An SFC chart consists of 1 to 8 and an SFC type of 1 to 32 sequencers each with a sequence of sequence path elements. These elements include the following:

- Step
- Transition

And below a sequence (can be freely positioned):

- Text

The remaining elements are structures made up of different elements:

- Sequence
- Simultaneous sequence
- Alternative sequence
- Loop
- Jump

Identifying "Steps" and "Transitions"

The basic elements, step and transition, have a **name** that is unique within the sequencer. When the editor creates one of these elements, it assigns a consecutive number that you can modify and change to a name with up to 16 characters. This name must not consist exclusively of numbers. The original number is retained by the element and is displayed in the object properties in the **Number** field. This is required for the "programmed target steps" and to interpret the outputs of the SFC in the CFC view.

You can use the optional **comment** to add comments about the functionality of the element. A comment can take up several lines and consists of up to 80 characters although only 16 characters are displayed to the right of the chart element.

If you position the mouse pointer on the step or the comment, the name with a maximum of 16 characters is displayed and the comment with up to 50 characters as brief information.

Resources

When programming and setting parameters for steps, transitions, start condition and pre- and postprocessing in SFCs, you can access the interface of the SFC, CFC block I/Os, runtime groups, other SFCs, and all shared addresses of the user program. With the textual interconnections, you can also access objects that do not yet exist. When you configure steps and transitions in SFC types, it is only possible to use its interface I/Os alongside the textual interconnections.

Screen Display

- All the elements of an SFC, including the links are displayed in white with black print in the unselected and unedited state.
- Selected elements are displayed in blue.
- Edited steps or transitions (whose object properties have been changed) are displayed in gray with black print.

Note:

The colors named here are the default settings, some of which can be modified (refer to the SFC online help).

1.5.3 What is a step?

The standard is a control entity for processing the assigned actions on the CPU. Per step, you can configure up to three actions (Initialization, Processing, Termination).

What are actions?

An action is a series of statements and is formulated as:

- Assignments for setting parameters of CFC blocks or shared resources, for example:
  Destemp := 100
  XYZ.pump.on := TRUE
- Activating or deactivating an SFC or a runtime group, for example:
  SFC_1.INTONOFF := TRUE
  ABL_1.EN := FALSE

What types of steps exist?

A start step, a maximum of 253 normal steps and a final step can be used in a sequencer. When a sequencer is created, an initial step, a final step, and one transition are created automatically. These three elements form the initial state of a sequencer that you can edit by adding further chart elements.

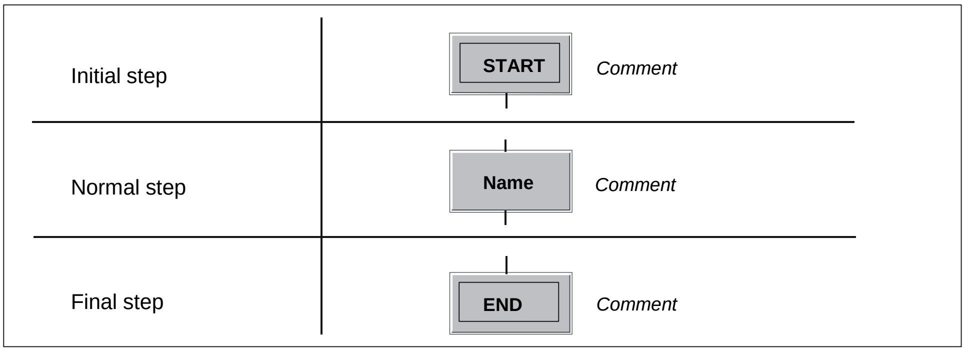


Fig. 1-5 : Icons of the Step Types

The **start step** is activated and the start of the sequencer and the corresponding actions are executed depending on the status of the successor transition. The **final step** does not have a successor transition; all actions are processed exactly once.

All other steps apart from the initial and final steps are **normal steps**.

The initial step and final step can neither be copied, cut, nor deleted. This makes sure that there is always one initial and one final step in a sequencer.

1.5.4 What is a transition?

A transition contains the conditions with which a sequential control system passes control from one step to the next. Several conditions can be logically combined using Boolean operators. The result of the logic operation decides whether control is passed to the next step.

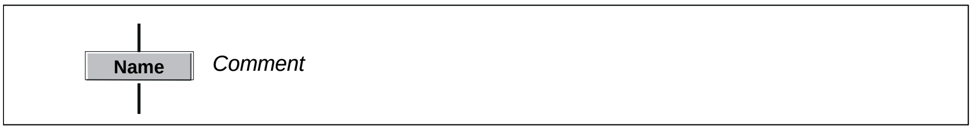


Fig. 1-6 : Icon of a Transition

1.5.4.1 How is a transition condition created?

The result of a transition condition is obtained from a Boolean expression that is formed by logically combining the values of shared addresses, textual interconnections, CFC block I/Os, runtime group state, and SFC chart state. In SFC types, it is only possible to use its interface I/Os alongside the textual interconnections.

During compilation, an empty transition is given the default value TRUE. This default is necessary because while the formulation of a condition is optional, a defined value is required on the AS to allow control to be passed on.

If several transitions become valid at the same time (in alternative sequences, in a loop or in jumps) the system automatically assigns priority from left to right in descending order.

1.5.5 What is a text?

With the "Text" chart element, you can insert any text you require in a chart.

The text object is a box with a character string with one or more lines. During the analysis phase, for example, this allows you to insert descriptive texts into the SFC that can later be replaced by automation functions.

These free texts are not embedded in the topological sequencer structure and are therefore not repositioned if you change the topology but remain where they are.

1.5.6 What is a sequence and what is a sequencer?

A sequence is a path made up of steps and transitions that can be created with a selectable length and inserted in the chart.

A closed sequence forms a sequence path in a sequential control system, for example between the divergence and convergence of a simultaneous or alternative sequence. In the SFC a full sequencer can also be called a sequence path, this runs from the initial step to the final step (see Fig. 1-7).

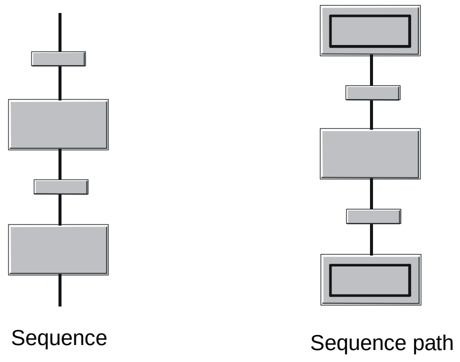


Fig. 1-7 : Sequence of the Sequencer

1.5.7 What is a simultaneous sequence?

If the control diverges along two or more sequence paths that should be executed at the same time, simultaneous sequences are used.

Simultaneous sequences consist of at least two sequence paths that are executed at the same time.

A simultaneous sequence is always preceded by a transition (or an alternative sequence). The simultaneous sequence paths end in a simultaneous convergence that is always followed by a transition (or alternative sequence).

The successor transition executes only when all actions of the steps at the end of every sequence path have been executed (apart from the "termination" action) and the condition for passing on control is satisfied (synchronization).

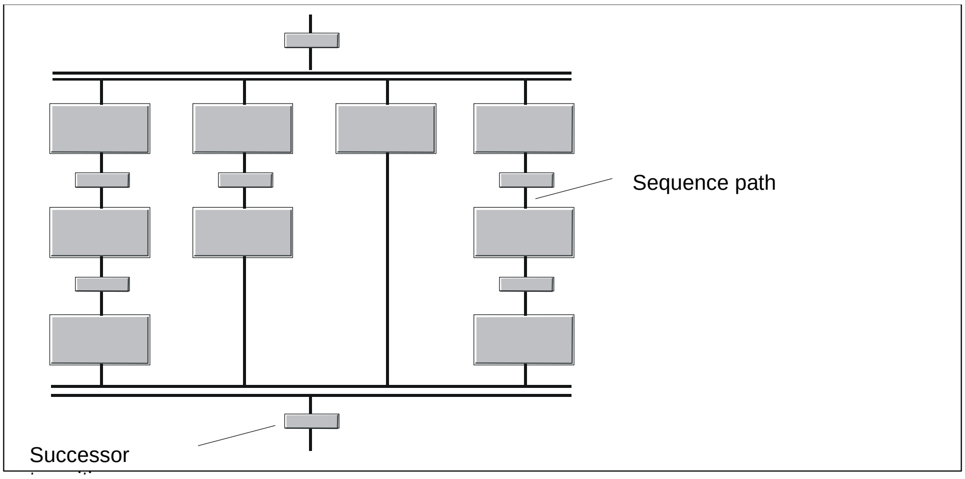


Fig. 1-8 : Example of a Simultaneous Sequence with Four Sequence Paths

1.5.8 What is an alternative sequence?

If control diverges into two or more sequence paths of which one and only one should be executed, alternative sequences are used.

An alternative sequence consists of at least two sequence paths, of which only one will be executed dependent on the state of the first transition in the individual sequence paths. This means that the path that will be selected is the path whose transition is true first. If more than one transition is true simultaneously, the sequence path furthest left with a true transition is executed.

Alternative sequences must be preceded and followed by a step (or simultaneous sequence).

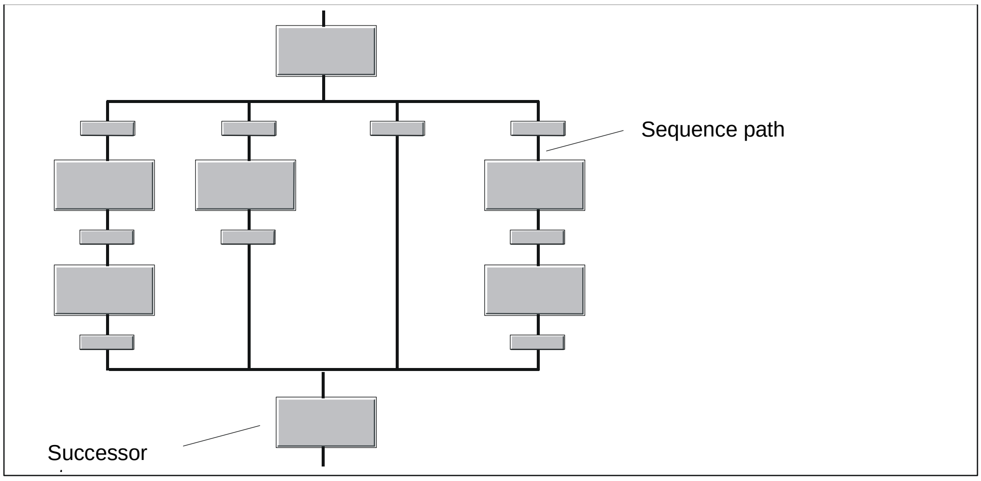


Fig. 1-9 : Example of an Alternative Sequence with Four Sequence Paths

1.5.9 What is a loop?

If you want a section of the sequencer to be repeated depending on a transition, you use a loop.

A loop consists of a sequence and a return path with a transition that encloses the sequence (see Fig. 1-10). The start of the loop must be immediately following a step and the return path must converge again immediately before a step.

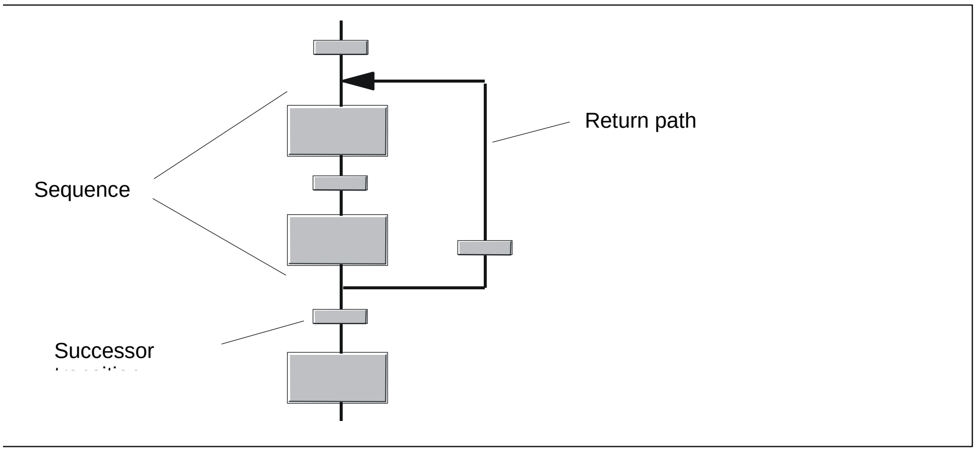


Fig. 1-10 : Example of a Loop

The transition of the return path is processed after the successor transition.

If the successor transition and the return path transition are true at the same time, the step (or simultaneous sequence) following the successor transition is executed.

Note:

Return paths out of or into simultaneous or alternative sequences are not possible.

1.5.10 What is a jump?

Depending on a transition condition, jumps can be used to continue the execution of the sequence path at any step within the same sequence path.

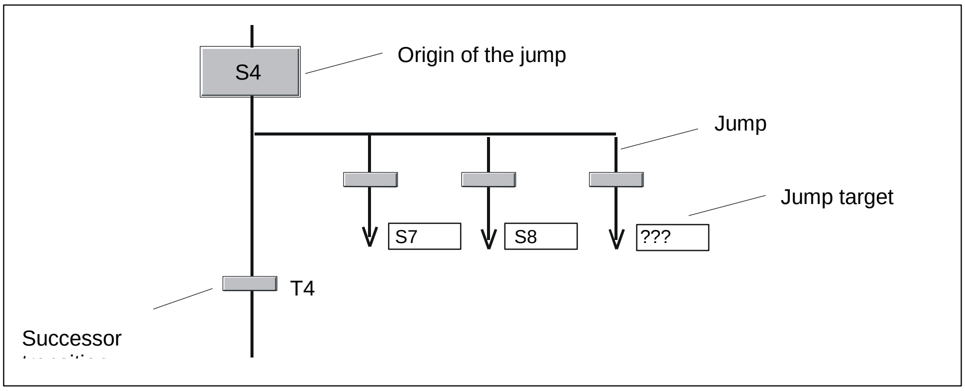


Fig. 1-11 : Example of the Origin of a Jump with Three Jumps

A jump always leads immediately out of the sequence following a step (origin of the jump). Several jumps are also possible.

A jump consists of an initiating transition and an arrow specifying the jump target. The name of the step to which control is passed if the transition is true is specified as the name of the step or **???**, if the jump destination is as yet unknown.

The transition of the jump is processed after the successor transition.

If the successor transition one or more jump transitions are true at the same time, the step (or simultaneous sequence) following the successor transition is executed.

The origin and target of a jump must always be a step.

Note:
With jumps into or out of a sequence path of a simultaneous sequence, consider the consequences for the program execution on the CPU.
For more detailed information, refer to Section 2.9.5, "Creating a Jump".

2 Working with the SFC Editor

Introduction

With the SFC editor, you can create sequential control systems graphically and specify the actions and step control conditions. From beginning (creating the chart or type) to end (compiling and downloading to the CPU), the editor provides all the functions required.

How to use the editor is described in this chapter.

2.1 Configuration: SFC Chart

In SFC, there is initially the object type "SFC chart" along with the object type "SFC type". The SFC chart allows the definition of sequential control systems using a predefined interface and process signals to the plant being automated. The SFC chart must first be compiled and it is then downloaded to the automation system where it executes.

2.1.1 General Information on Configuring the SFC Chart

The SFC chart has an interface that is created when the SFC chart is generated and includes the standard interface derived from the SFC chart template (block @SFC_RTS). This is required to provide SFC system functionality (operating modes, operating states, step control modes, etc.) at the interface of the SFC chart.

The elements of the standard interface cannot be moved or deleted and no new elements can be added. The initial value, comment, and the attributes can be modified.

During configuration, only the interface I/Os and any required process signals can be used to formulate step assignments or the transition and start conditions. Here, textual interconnections are also possible.

The SFC chart is therefore not self-contained and can only be used once. To use a chart more than once, the SFC chart must be copied and completely adapted to the "new environment" since the process signals must normally be replaced by others.

See also: **Procedure: Creating an SFC Chart**, Section 1.2.1

2.1.2 Changing the Configuration of the SFC Chart

Changes to the topology (step/transition sequence, jump destination change) and to the step or transition configurations are made in the SFC chart and do not become effective until after compiling and downloading. If changes are made to the topology, the rule is that inactive sequencers can be downloaded at any time, whereas if the sequencers are active, the SFC chart must first be turned off before downloading. Changes to the step and transition configuration can be downloaded at any time even if the SFC chart is currently being processed on the CPU.

After completing the changes, the OS must be compiled and downloaded to ensure that the current data is available on the OS station.

2.2 The SFC Type

In SFC, there is not only the object type "SFC chart" but also the object type "SFC type". The SFC type allows the definition of sequential control systems including an interface. The sequential logic of the SFC type is based solely on the interface I/Os of the SFC type; in other words in contrast to an SFC chart, an SFC type cannot access every process signal. The SFC type cannot run alone. Like a function block type, an SFC type must be placed in a CFC chart to obtain a executable object, in this case, an SFC instance. To execute an SFC instance, both the SFC type and the SFC instance are loaded on the automation system.

2.2.1 General Information on Configuring the SFC Type

The SFC type has an interface analogous to the SFC chart. It is created when an SFC type is generated and already contains the SFC type standard interface, derived from the SFC type template "@SFC_TYPETEMPLATE" (FB247). The standard interface is required to provide SFC system functionality (operating modes, operating states, step control modes, etc.) at the interface of the SFC type.

Interface Expansions

You can add further I/Os to the interface using the interface editor and add characteristics in the Characteristics editor. The elements of the characteristics cannot be moved or deleted and no new elements can be added.

Elements created explicitly can be manipulated as required.

Assignments and Conditions

During configuration, only the interface I/Os can be used to formulate step assignments or the transition and start conditions. In other words, addresses in assignments or conditions are always references to I/Os of the interface. Here, textual interconnections are also possible. This means that the SFC type is self-contained and there are no external accesses from the SFC type which bypass the interface.

Notes

- SFC types cannot be assigned to a hierarchy folder in the plant view since they themselves are not relevant to execution (from the perspective of the process to be automated).
- An SFC type does not have any runtime properties, since it is not relevant to execution of the program. An SFC type cannot be installed in the run sequence.

2.2.2 Creating and Modifying an SFC Type

There are two possible ways of creating and modifying an SFC type:

- **Creating and modifying in a library** has the advantage that the master for the SFC type is always in the library and that the test project can still run until a new version of the SFC type is adopted.
- **Creating and modifying in a project** has the advantage that every change to the SFC type can be checked immediately since you are working directly with the master.

2.2.2.1 Creating the SFC Type in the Project

You can only create an SFC type when a PCS 7 project already exists.

Basic Procedure:

- Creating (SIMATIC Manager):
  - Create a new SFC type ("Insert New Object > SFC Type")
  - Adapt the SFC type name
  - Open the SFC type
- Adapt (SFC Editor):
  - Adapt the properties (SFC > Properties);
    “General" Tab: Name, author, family, FB number, comment;
    "Operating Parameters CPU": Options for "Defaults" and "Start Options"
    "Options" tab: SIMATIC BATCH, control strategy selection
    "Version" tab: Version identifier (0.0 - 15.15)
- Characteristics (SFC Editor)
  - Definition of the characteristics ("View > Characteristics")
  - Add the control strategies
  - Add the setpoints
  - Add the process values, block contacts etc.
- Sequencers (SFC Editor)
  - Add/edit the sequencers
  - Edit the start condition (sequencer properties)
  - If applicable, edit preprocessing/postprocessing of the sequencer
  - Configure the sequencer

Note:

There are predefined sequencer templates available in the "SFC Library" (SIMATIC Manager). You can copy these templates and adapt them to your own purposes.

2.2.2.2 Changing the Configuration of the SFC Type

Modifications to the interface of the SFC type are transferred to the SFC instances immediately. This means that the SFC type and its instances can only be downloaded to the CPU in RUN mode if all the SFC instances of this SFC type are deactivated or are deactivated briefly during downloading. The instances are deactivated during the download following operator confirmation and restarted after the download, again following operator confirmation. The execution of the instance then depends on the process state and on the configuration of the instances (especially the start conditions).

While changes are being downloaded, the system prevents the SFC instances from being processed on the CPU and prevents access to the SFC instances via the interconnections in the CFC.

Changes to the topology (step/transition sequence, jump destination change) and to the step or transition configurations are made in the SFC type and do not become effective in the SFC instances until after compiling and downloading. If changes are made to the topology, the rule is that inactive sequencers can be downloaded at any time, whereas if the sequencers are active, the SFC instances must first be turned off before downloading. Changes to the step and transition configuration can be downloaded at any time even if SFC instances of the SFC type are currently being processed on the CPU.

After completing the changes, the OS must be compiled and downloaded to ensure that the current data is available on the OS.

2.2.3 Using the Characteristics Editor and the Interface Editor

The Characteristics editor is used for the technological configuration of an interface for an SFC type. You can only work with the editor if an SFC type is open in SFC.

The interface I/Os of the SFC type are defined directly in the Interface editor

(started with **View > Inputs/Outputs** or ▤) and/or using the Characteristics editor

(started with **View > Characteristics** or ⌶). The Interface editor and the Characteristics editor box are displayed alternatively. This means that it is not possible to define technological characteristics and interface I/Os at the same time. You can define characteristics and I/Os in any order.

Note:

Individual I/Os can be defined in the Interface editor. These are available in addition to the I/Os defined in the Characteristics editor. These I/Os are unknown in the Characteristics editor and cannot be manipulated in it.

Setpoints, for example, are created and assigned values during the definition of characteristics. From this definition, the Characteristics editor generates the required I/Os that are displayed in the Interface editor after it is opened and that can be used to configure the sequence logic. The I/Os generated by the Characteristics editor cannot be modified in the Interface editor (except for system attributes, initial value, comment), in other words, the Characteristics editor is the master in terms of these I/Os.

Grouping of I/Os is only possible using the Characteristics editor. These groups are taken into account during compilation in the applicable validations, status and message processing. Visualization/operator control on an OS station (for example of setpoints) using the SFC type faceplate is also only possible if the Characteristics editor is used.

2.2.4 Configuring with the Characteristics Editor

The Characteristics of the SFC Type

A characteristic is the technology-oriented functionality of an SFC type that is described by a set of attributes (for example, I/O name, data type, initial value, high limit, ...).

The following characteristics are available for creating the type:

Characteristic	Meaning
Control strategies	Control strategies are used for the structuring of an SFC type as required for process control. With the Characteristics editor, you can define control strategies so that they can be used when configuring the sequencers (for example Heat, Cool, ...). The control strategy can be set by the operator or by a higher-level controller (for example SIMATIC BATCH). Control strategies are optional.
Setpoints	Setpoints control the SFC type or are used by it to control underlying automation functions. Setpoints can be set by the operator or by a higher-level controller (for example SIMATIC BATCH). A setpoint is assigned to the control strategies that exist when it is created. You can change the assignment individually for each control strategy.
Process values	Process values control the SFC type based on process signals (for example, the value for the fill level).
Control values	Control values are used by the SFC type to control external logic.
Parameters	Parameters modify the behavior of the SFC type. For example, parameters can be used to select alternative sequences or to configure start conditions.
Bit memory	Bit memory is local data memory of the SFC instances belonging to the SFC type. Here, data can be stored temporarily and reused later. The data is retained until it is overwritten or deleted.
Timers	Timer objects are used to configure time-driven sequences in the SFC type. These can be local to a step or related to steps and sequencers. Timed routines are started in steps, stopped, and the elapsed time queried in transitions or start conditions. Timed routines can be varied with different modes (pulse, extended pulse, on delay, retentive on delay, off delay).
Note texts	Note texts are used to display information relating to execution that may require action on the part of the operator. A note text is displayed on the OS in the SFC faceplate and can be acknowledged there.
Block contacts	Block contacts represent blocks of the basic control. When a block contact is created, I/Os of the block to be represented are created in the interface of the SFC type. These can then be used when configuring the SFC type instead of the block that will later be interconnected with an SFC instance.
Position texts	Position texts are used to display the current progress in execution or the status of execution of the SFC. A position text is displayed on the OS in the SFC faceplate.

2.2.4.1 Note on Subsequent Modifications to Control Strategies

Control strategies that have already been used, for example in start conditions, in transitions, in the assignment to setpoints and in the control strategy release must not be deleted or moved in the Characteristics editor, otherwise the control strategies receive new numbers. The numbers of the control strategies in the existing utilization are not adapted. The access to these numbers and mechanisms that relate to these numbers then no longer function as originally intended.

Example:

You no longer want to use the control strategy with number "2" and delete it.

The control strategies are renumbered. The previous control strategy "3" is given the number "2", control strategy "4" number "3", etc.

If you no longer want to use a control strategy, do **not** delete the control strategy but rename it, for example calling it "not used". The numbering and functionality of the other control strategies is then retained.

2.2.4.2 Configuring and Using Note Texts and Position Texts

The note texts and position texts are configured in the Characteristics editor and an identification number is assigned to each text. When you create a text, the editor first assigns the next free number, this can be changed (1 through 32767) however the number must be unique for all note texts. The numbering can also include gaps.

For more detailed information on configuration procedures and using the note and position texts, refer to the SFC online help, topic "Using Note and Position Texts".

2.2.5 I/Os for Characteristics

To define the interface I/Os or I/O elements of a characteristic, there are predefined descriptions of the required I/Os.

The interface I/Os or I/O elements for the control strategies, note texts, and position texts characteristics are already present in the standard interface of an SFC type.

For the other characteristics, the names of the individual I/O elements are made up of the I/O names configured in the characteristic and a fixed part.

For more detailed information on the characteristics, refer to the SFC online help:

- **The Standard Interface of the SFC Type**
- **Interface Expansions "Characteristics"**
- **Interface Parameter Assignments "Characteristics"**

2.2.5.1 Block Contacts

To interface blocks of basic control using block contacts, you must specify the I/Os relevant for the connection to an SFC type in the block types.

You do this in the language in which the block type was written by assigning the system attribute "S7_contact=true" to the block I/O. The technological blocks from the PCS 7 Library are already suitably prepared. When necessary, you can make project-specific adaptations to the relevant I/Os for the supplied block types.

Note:

If the system attribute "S7_contact" is added or modified later at the I/Os of a block type that is used as a block contact in the SFC type, this has no effect on the block I/Os of the SFC type.

If you want these changes to be adopted for the SFC type, note the following:

- The modified block type must be imported into the CFC (menu command "Options > Block Types"). This also updates all block instances.
- In SFC, you must update the block contacts with the menu command "Options > Block Contacts". Changes to the interface of the SFC type and all SFC instances then take effect.

If different groups with relevant I/Os are required for a block type, this can be achieved as follows:

- The set of all required I/Os is defined for the block type or
- the block type is copied or an empty block is created with the block interface of the original block reduced to the relevant I/Os. The required I/Os are assigned "S7_contact=true" in the block type. The resulting block contact (the SFC instance) can later be interconnected with an instance of a different "suitable" block type.

By defining an instance of the "block contact" characteristic in the Characteristics editor, the selected block I/Os are then created in the SFC type. The I/Os of the type IN and OUT are created as their opposites in the SFC type. The IN_OUT I/Os of the block type are created as OUT in the SFC type. This allows a concrete CFC instance of the block type to be interconnected with the corresponding I/Os of an SFC instance. An instance of the "block contact" characteristic is therefore always assigned to exactly one block type. Which block type this is, is entered in the Block column in the right-hand pane.

This allows the independent configuration, commissioning, and visualization of blocks of basic control. At the same time, SFC types can be configured on the basis of block contacts and later interconnected with the blocks of basic control.

2.2.5.2 Attributes for Characteristics

Below, you can see which attributes are available for configuring the characteristics (assignment) and their meaning.

Assignment of the Attributes to the Characteristics

Characteristics → ***Attributes ↓***	**Control Strategy**	**Set-points**	**Process Values**	**Control Values**	**Parameters**	**Bit Memory**	**Timers**	**Position Texts**	**Note Texts**	**Block Contacts**
Name	x	x	x	x	x	x	x	x	x	x
Number	x							x	x	
Data type		x	x	x	x	x				
I/O name		x	x	x	x	x	x			x
Comment	x	x	x	x	x	x	x	x	x	x
<Setpoint>	x									
Initial value		x	x	x	x	x	x			
Text length		x	x	x	x	x				
Precision		x	x	x	x	x				
Unit		x	x	x	x					
Low limit		x								
High limit		x								
Text0		x								
Text1		x								
Enumeration		x								
Archive		x								
Test		x								
Setpoint ID		x								
Process value ID		x								
Material		x								
BATCH ID		x								
Block										x

Meaning of the Attributes

Name

The name is the name of the characteristic that must be unique for all characteristics of a type. It can be a maximum of 24 characters long and must not contain any special characters except „_" (in particular, it must not contain blanks) and is not relevant for the interface I/Os.

Number

The number is required to identify the corresponding characteristic uniquely.

With the control strategies, the number is managed by the system and assigned consecutively from 1 through 32 according to its position in the Characteristics editor. The control strategy <no> is assigned bit <no-1> in the SELCS I/O of the SFC type. This bit is set when the corresponding control strategy was configured. With an SFC type and SFC instance, the bit is deleted when the corresponding control strategy is not to be used (SFC > Properties > Options > Control Strategy Selection).

With note and position texts, the number is initially the first free number, however, this can be changed by the user to any unique number within the range 1.. 32767.

Data type

Permitted data types for characteristics are BOOL, INT, DINT, REAL, and STRING. For setpoints, the data types PI and PO are also available and effectively represent a REAL setpoint but they have the additional attributes "Material" and "Batch ID". Depending on the characteristic and its data type, other fields can either be edited or are locked.

I/O name

The I/O name is required to generate the interface I/Os belonging to the characteristic. The I/O name is derived from characters of the name. The I/O name can then be modified later. For setpoints and timers up to 16 characters are allowed and for block contacts up to 10 characters, while the other characteristics can have up to 24 characters.

When the interface I/Os are created, several I/Os are generated depending on the characteristic. A postfix is appended to the I/O name so that the I/Os are named uniquely.

The maximum lengths of the I/O names result from the different ways of generating interface I/Os. With setpoints and timers, the postfixes are restricted to a maximum of 8 characters by the system. With block contacts, the length of the postfixes depends on the block being used.
No postfix is appended to the other characteristics so that the I/O name can be up to 24 characters long.

Comment

The comment can be used to describe the characteristic in greater detail. The comment can be a maximum of 80 characters long and can contain any special characters.

<Setpoint>

Setpoint selection for the "Control strategy" characteristic. Name of each created setpoint is displayed as a column. This allows the required setpoint to be assigned to the control strategy by selecting the corresponding column.

The assignment of setpoint to control strategy is stored in the "<setpoint-I/O name>_CS" I/O. The coding is the same as with the SELCS I/O; in other words control strategy <no> is assigned bit <no-1>. This bit is set when the setpoint is assigned to the control strategy; in other words, it is required to run the control strategy.

Initial value

The initial value corresponds to the value of the characteristic when no current value is available.

Text length

With the STRING data type, the text length decides the maximum length of the string (number of characters: 1 through 254).

Precision

With the REAL, PI, and PO data types, the precision decides the number of decimal places to be displayed (0 through 7).

Unit

With the INT, DINT, REAL, PI, and PO data types, a unit can be defined. This is stored as the system attribute "S7_unit" on the interface I/Os.

The units are included as a basic set in the ES data management and can be added to or modified in the SIMATIC Manager as "Shared declarations".

Low limit

With the INT, DINT, REAL, PI, and PO data types, a low limit can be defined for the range of values. This is stored in the "<I/O name>_LL" I/O.

High limit

With the INT, DINT, REAL, PI, and PO data types, an upper limit can be defined for the range of values. This is stored in the "<I/O name>_HL" I/O.

Text0

With the BOOL data type, this specifies the value identifier for the value FALSE. This is stored as the system attribute "S7_string_0" on the interface I/Os.

Text1

With the BOOL data type, this specifies the value identifier for the value TRUE. This is stored as the system attribute "S7_string_1" on the interface I/Os.

Enumeration

With the BOOL, INT and DINT data types, an enumeration can be assigned here. This is stored as the system attribute "S7_enum" on the interface I/Os. Enumerations are defined as "Shared declarations" in the SIMATIC Manager. The name of this enumeration is then entered here (from a drop-down list box) as an attribute. The enumeration is then available for operator control and monitoring of SFC instances in SIMATIC BATCH and in the SFC faceplates.

Note: If "S7_enum" is used, the "S7_string_0" and "S7_string_1" system attributes are ignored.

Archive

From the drop-down list box, you can select whether the current value of the "<I/O name>_AO" output is archived, not archived, or entered in the long-term archive.

To allow archiving, an archive tag is created in WinCC when you compile the OS in the SIMATIC Manager so that the value is entered in the archive.

Test

This selects the characteristic intended for the manual test (manufacturing instruction) in the batch report in SIMATIC BATCH.

Setpoint ID / Process value ID

The ID number specifies whether the characteristic value is saved for logging. If the ID is greater than 0, the setpoint or process value is also made available for an external program. By assigning unique ID numbers to the characteristic values, the processing is made easier in an external application. ID numbers from 0 through 32767 can be entered.

Material

With the PI and PO data types, a material identifier can be entered here.

Batch ID

With the PI and PO data types, a numeric batch identifier can be entered here.

Block

This attribute contains the name of the block type represented by the block contact. When generating the interface I/Os, the interface I/Os for which the system attribute "S7_contact=true" is set are adopted in the SFC type from the block type. The names of the interface I/Os of the block type are appended to the I/O name of the block contact as postfixes. Inputs and inouts of the block type become outputs of the SFC type and outputs of the block type become inputs of the SFC type.

2.3 The SFC Instance

An SFC instance is derived from an SFC type. The SFC type is first inserted into a CFC chart in the same way as a function block type in CFC. The SFC instances are therefore always assigned to a CFC chart and are addressed using the chart. SFC instances are displayed like CFC instances; in other words, their interface is visible in the CFC chart.

SFC instances are not displayed in the SIMATIC Manager since they can only be addressed here via the CFC chart. With the assignment of the CFC chart to the plant hierarchy, the SFC instances contained are also indirectly assigned to the plant hierarchy.

2.3.1 Configuring the SFC Instance

Compared with SFC types, the configuration of SFC instances in SFC is limited. The interface and the sequencers cannot be changed in the SFC instance, since they are identical in all instances of the SFC type. The parameters (initial value, comment, and system attribute) of the interface I/Os, on the other hand, can be modified for each specific instance.

Changes to the Configuration

Due to the restrictions relating to configuring SFC instances, all the changes you make directly in the SFC instances can be downloaded at any time, even if the SFC instances are currently being processed on the CPU.

After completing the changes, the OS must be compiled and downloaded to ensure that the current data is available on the OS station.

2.4 Working with Charts, Types, and Instances

2.4.1 Creating an SFC Chart

- SIMATIC Manager

  You create an SFC chart with the SIMATIC Manager by opening the **chart folder** in the **component view** or the **hierarchy folder** of the project in the **plant view** and inserting the chart there ("Insert > S7 Software > SFC" or "Insert > Technological Objects > SFC"). The chart is given a standard name by the system, for example SFC(1) that you can change. The name must be unique on the CPU. This is checked by the system.

- SFC Editor

  Open the “New" dialog box in SFC using the menu command “SFC > New...". Select the project and the chart folder in the component view.

  In the **plant view** or the **process object view**, open the hierarchy folder of the project in which you want to create the chart.

  In the "Object type" box, select "SFC" from the drop-down list box and enter a chart name in the "Object name" box. The name must be unique in the chart folder; this is checked by the system. When you click “OK", a new window is opened with the SFC chart (initial status).

2.4.2 Creating an SFC Type

You create an SFC type as follows:

- In the component view of the SIMATIC Manager with the chart folder selected, right-click to open the context-sensitive menu ("Insert New Object > SFC Type") or select the menu command "Insert > S7 Software > SFC Type".
- In the SFC Editor with the menu command "SFC > New...". In the "New" dialog, select "SFC type" from the drop-down list box for the "Object type" box.

The next free FB number is automatically reserved for the SFC type and this is copied to the block folder as a type template. This allows you to configure messages and instances of the type once it is created without having to compile the type. The FB number can be modified later in the Object Properties dialog.

When you first create an SFC type, the blocks required for compiling are copied to the current program and then managed on the ES. The blocks are included in the supplied block library.

Note: SFC types cannot be assigned to a hierarchy folder in the plant view since they themselves are not relevant to execution (from the perspective of the process to be automated).

2.4.3 Creating an SFC Instance

You create an SFC instance by dragging the SFC type from the block catalog or from a library to the CFC chart in CFC.
The SFC types in the chart folder are displayed in the CFC block catalog (in "All blocks" and in the folder of the family if they are assigned to a family, otherwise in the "Other blocks" folder).
You can also insert SFC types in the CFC chart from the libraries (for example SFC library, master data library, ...).

The SFC instance is represented like a CFC instance block. If there is not enough space to position the SFC instance; in other words, it overlaps one or more existing objects, it is displayed as an "overlapping block" (light gray and without visible I/Os). After moving them to a free position in the chart, the overlapping blocks are displayed as "normal" blocks again.

You can assign parameter values to the SFC instance in the CFC chart and interconnect it.

2.4.4 Opening a Chart or Type

- SIMATIC Manager:
  You can open a chart or type using the SIMATIC Manager by double-clicking the icon in the chart folder of the S7 program in your project. The SFC editor is then started and the selected chart or type is opened.
- SFC Editor:
  In the "Open" dialog of the SFC editor, you select "SFC" or "SFC type" from the drop-down list box in the "Object type" box and then select the required object.

  In the “SFC" menu of the SFC editor, you will see the last SFCs to be edited shown as menu entries. If you select one of these names, the relevant chart / type / instance is opened, or if the SFC is already open, displayed in the foreground.

2.4.5 Opening an SFC Instance

You can open SFC instances in the CFC chart. The SFC is started with the topology of the SFC type. This topology can only be viewed and cannot be modified. The properties of the SFC instance and parameters and properties of interface I/Os can be modified.

2.4.6 Copying Charts

With the SIMATIC Manager, you can copy charts; in other words, you can transfer tested substructures or even entire structures from one CPU to another CPU of the same type or copy them within the same CPU. Existing references are not lost if the relevant charts are copied together at one time.

For more detailed information on copying, refer to the SFC online help.

2.4.7 Copying and Moving SFC Types

SFC types are **copied** in the SIMATIC Manager. The runtime objects belonging to the SFC type are also copied. If the SFC type is not up to date (time stamp of the FB is older than the time stamp of the SFC type), a message will be displayed. If the SFC type already exists at the destination (SFC type with the same name), this is overwritten following a prompt for confirmation and any properties that differ from the previous type are passed on to the SFC instances.

SFC types are **moved** in the SIMATIC Manager. SFC types can only be moved when there are no SFC instances of the SFC type in the source. The runtime objects belonging to the SFC type are also moved. If the SFC type already exists at the destination (SFC type with the same name), this is overwritten following a prompt for confirmation and any differences compared with the previous type are passed on to the SFC instances.

2.4.8 Copying and Moving SFC Instances

SFC instances can be copied or moved within the CFC chart, between CFC charts, or indirectly by copying/moving the CFC chart in the SIMATIC Manager. The runtime objects belonging to the SFC instance are also copied/moved.

If you **copy** an SFC instance within a CFC chart or between CFC charts of the same chart folder or copy a CFC chart within a chart folder, the SFC instance is also copied. The runtime objects belonging to the SFC instance are also copied. If you copy an SFC instance between CFC charts from different chart folders or copy a CFC chart to a different chart folder, the SFC type is also copied.

If you move an SFC instance within a CFC chart, only the position of the SFC instance changes. If you move an SFC instance between CFC charts of the same chart folder, the SFC instance is moved. The runtime objects belonging to the SFC instance are retained. If you move an SFC instance between CFC charts from different chart folders or move a CFC chart to a different chart folder, the SFC type is also copied or moved.

2.4.9 Deleting Charts and Types

You delete SFC charts and SFC types only in the SIMATIC Manager.

- You delete SFC charts in the same way as other objects (hierarchy folders, OS pictures, ...) by selecting the object and pressing the DEL key.
- SFC types can only be deleted when no SFC instances of the SFC type exist. If instances of an SFC type exist, a message to this effect is displayed.

  The runtime objects belonging to the SFC type are also deleted.

You cannot delete charts or types in the SFC editor.

2.4.10 Deleting SFC Instances

You delete SFC instances in the CFC chart or indirectly by deleting the CFC chart in the SIMATIC Manager. The runtime objects belonging to the SFC instance are also deleted.

2.5 Properties of Charts and Types

2.5.1 Adapting Chart Properties

You can modify the chart properties for the active chart. With the menu command “SFC > Properties", you open the properties dialog box.

You can modify the chart properties in the following tabs.

- **General**
  This tab is used to enter or modify the chart name, the author and the comment.
- **Operating parameters CPU**
  Here you can
  - set the defaults for the initial state of the chart. These are: "Step control mode", "Operating mode", "Instruction output", "Cyclic execution" and "Time monitoring".
  - set the options for starting the chart: "Autostart" and "Use default operating parameters when SFC starts".
- **OS**
  If the “Transfer chart to OS for visualization" option is set, the SFC chart is transferred to the OS automatically with the next OS compilation.
- **Version**
  In the "Version:" box, you can see the current version of the chart. When you create a chart, Version 0.1 is entered here. To identify working versions, you can change this version number (0.0 through 15.15 are permitted).

  With "Data version", you obtain information on the software version used to create the chart.

For a description of the operating parameters, refer to Section 3.2, "How the SFC Runs", and the SFC online help.

2.5.2 Adapting Type Properties

You can display and modify the properties for the active SFC type. The "SFC > Properties..." menu command opens a dialog box. You can modify the properties in the following three tabs.

- **General**
  This tab includes the entry or modification of the type name, the author, the version, the family, the FB number, and the comment.

- **Operating Parameters CPU**
  In this tab, you can set the defaults for the initial status of the SFC instances created from this type. These are: "Step control mode", "Operating mode", "Instruction output", "Cyclic operation" and "Time monitoring" and the options for starting the SFC instance: "Autostart" and "Use default operating parameters when SFC starts".

- **Options**
  In this tab, you can make the classification of the SFC type for SIMATIC BATCH:

  - The category
    "None" → there is no classification
    "EOP" → the SFC type is classified as an "operation type"
    "EPH" → the SFC type is classified as a "phase type".

  - Allow operator instructions on the OS; in other words, allow input of values in the operator dialog.

  - Control strategy selection
    This field lists all the control strategies configured for the SFC type (maximum 32).

    By suitable selection of the selection criteria, you can decide which control strategies of the SFC type are enabled as default in the SFC instances. This applies to all SFC instances that already exist in the project - as long as the default value has not been changed in the instance - and for all the SFC instances yet to be generated. The control strategy selection can be modified for the individual instances.

- **Version**
  In the "Version:" box, you can see the current version of the SFC type. When you create an SFC type, Version 0.1 is entered here. To identify working versions, you can change this version number (0.0 through 15.15 are permitted).

  With "Data version", you obtain information on the software version used to create the SFC type.

2.5.3 Adapting Instance Properties

You can display and modify the properties for the SFC instance opened in the CFC. The "SFC > Properties..." menu command opens a dialog box with the three following tabs:

- **General**
  This tab is used to enter or modify the instance name and the comment. All other properties (see SFC type) can only be read and not modified.
- **Operating Parameters CPU**
  In this tab, you can change the SFC instance operating parameters (see SFC Type).
- **Options**
  In this tab, you can view the options set for the SFC type for SIMATIC BATCH and change some of them:
  - The category (cannot be changed)
    "None" → there is no classification
    "EOP" → the SFC type is classified as an "operation type"
    "EPH" → the SFC type is classified as a "phase type".
  - Allow operator instructions on the OS (cannot be changed)
    If the option is set, the entry of values in the operator dialog is permitted.
  - Control strategy selection
    This field lists all the control strategies configured for the SFC type (maximum 32).

    The control strategies adopted from the SFC type are shown as selected. You can change the selection for each SFC instance; in other words, you can select a new control strategy or deselect one that was previously selected.

    Note: If you change the control strategy selection for an SFC instance, all later changes to the control strategy selection in the SFC type are no longer adopted automatically for this SFC instance.
- **Version**
  In the "Version:" box, you always see the version of the associated SFC type. The version cannot be changed in the SFC instance.

  With "Data version", you obtain information on the software version used to create the SFC instance.

2.6 Configuring Messages in SFC

You can start configuring messages in SFC with the menu command SFC **>** Message....

You can configure specific message texts for each SFC chart/SFC type/SFC instance. You can modify the message texts in a dialog.

For an **SFC chart**, you can configure seven messages that require an acknowledgment and no messages that do not require acknowledgment.

For an **SFC type** or an **SFC instance**, you can configure seven messages that require an acknowledgment and five that do not. The SFC type itself requires the remaining available messages (one per message type and 10 status messages for SIMATIC BATCH).

The reserved message events have default texts:

- "Run time exceeded”
- "Operator prompt"
- 10 status messages for SIMATIC BATCH (only with SFC type/instance)

Run Time Exceeded

The "step error" is a process control message that must be acknowledged and is signaled along with 3 associated values using the message block ALARM_8P. The remaining seven free messages (SFC type/instance only) and associated values can be assigned as required.

The standard interface therefore has I/Os via which messages can be triggered (SIG_2 ... SIG_8) including the associated values AUX_PR04 ... AUX_PR10 (with an SFC type/instance).
These I/Os can be used by having interconnections in the actions of the steps or by having direct block interconnections.

Operator Prompt

The operator prompt is a message that does not require acknowledgment and that is signaled in an SFC chart with the NOTIFY message block and in an SFC type with NOTIFY_8P.

You will find a table of the I/Os used for messages in the interface in the SFC online help in the topic "**Messages**".

Note:

When configuring messages, please remember that the messages of ALARM_8P must only be assigned message classes with mandatory acknowledgment.

2.7 The Runtime Properties

The runtime properties of an SFC chart or SFC instance determine how the SFC is included in the execution of the entire structure on the CPU. These properties are vital to the performance of the target system in terms of reaction times, dead times, or the stability of time-dependent structures, for example control loops.

An SFC type does not have any runtime properties, since it is not relevant to execution of the program. An SFC type cannot be installed in the run sequence.

You edit the Run Sequence with the run sequence editor of CFC. This is started with the menu command "Edit > Run Sequence..." or with the button in the toolbar.

2.7.1 Run Sequence

Each SFC is installed in a run sequence

Each SFC must be installed in at least two tasks; in the

- task for the startup behavior (OB100)
- task for normal processing (for example OB32).

Note:
Installing an SFC in the same task or in several cyclic tasks is not permitted since this can lead to an unpredictable response. Installation in tasks other than OB100 and in a cyclic task (for example OB32) is of no practical value since the processing of the SFC is not intended there.

Changing the Run Sequence

To change the run sequence, select the icon of the SFC, select “Cut", then select the required task and then “Paste". If you have selected a task, the SFC is installed at the beginning of the task. If you have selected an object within the task, the SFC chart is installed after it.

As an alternative to cut / paste, you can also drag an SFC from an open task to another task with the mouse.

Removing an SFC from a Task

To remove an SFC from a task, select the chart and delete it with the “Delete" function or with the DEL key. Before the chart is deleted, you are prompted to confirm your intention.

2.7.1.1 Runtime Groups

SFCs can be installed in runtime groups if you want them to have the attributes "scan rate" and/or "phase offset". The attributes can be set only with the object properties of the runtime group; in other words, all charts of the runtime group have the same “scan rate" and “phase offset".

By using SFCs in runtime groups, technologically-oriented groups can be formed in conjunction with CFC charts. From a technological point of view, a better structuring of the project can be achieved that promises a considerable improvement in performance when configurations are changed (among other things due to shorter compilation times).

Installing SFCs in a Runtime Group

You create a runtime group with the menu command “Insert Runtime Group..." for the selected task (in the “Edit" menu or in the context-sensitive menu). In the dialog box, you enter the name and any comment you require and the runtime attributes for the scan rate and phase offset.

Install the SFC chart in the runtime group as usual (same procedure as installation in a task).

You will find information on the runtime attributes "scan rate" and "phase offset" of runtime groups in the online help "CFC Basic Knowledge".

2.8 Configuring Sequential Control Systems

Prerequisites

Before you can configure sequential control systems, you must first create the required basic automation functions with CFC and/or STEP 7 tools. At the same time, the AS blocks to be used in the SFC charts or SFC instances are also inserted. Automation functions that do not yet exist can be added later and then used in an SFC.

Procedure

When you configure a sequential control system (SFC chart or SFC type),

- create the sequence topology with the required number of sequencers and the required arrangement of SFC elements.
- configure the start condition, the preprocessing , and the postprocessing in the Properties dialog of the sequencers.
- configure the steps and transitions, the actions and conditions in the Properties dialog of the steps and transitions. These procedures are dealt with in greater detail in the following topics.

Color Settings

The objects of a chart are displayed in different colors depending on their current state. The elements of an unselected sequential control system, for example, are displayed in “white" (parameters not set) or “gray" (parameters set) and in “blue" when they are selected.

With the “Options > Customize > Colors..." menu command, you can select your own color scheme for certain elements.

Note:

If CFC blocks are deleted that access the SFC, these accesses will be set up as **textual interconnections** (text of the address is marked in yellow). Textual interconnections can be close (they become real interconnections) as soon as the real interconnection partner exists again in the chart folder. You can exit this explicitly with the menu command **Options > Make Textual Interconnections** or in the compilation dialog by setting the check box "Make textual interconnections".

2.8.1 Creating the Sequencer Topology

Appearance of the Sequencer

In its initial state, the new SFC consists of one sequencer; this can, however, be extended to up to 8 (SFC chart) or 32 (SFC type) sequencers. Each sequencer is created in its own working window. You can change from one sequencer to another using the tabs at the bottom edge of the window.

A newly created sequencer ("Insert > Sequence > ..." menu command) is inserted in its initial status consisting of an initial step, transition and final step, at a selected position in the SFC and a tab is added at the lower edge of the window. Each tab contains the name of the sequencer (RUN, SEQ1, ...).

See also Section 2.8.2, Configuring Several Sequencers.

If you insert or delete SFC elements in the sequencer, its layout is changed automatically according to predefined rules. These determine the spacing between elements, the size of steps and transitions, the alignment of alternative sequences etc. You can change the display/layout rules at any time (Options > Customize > Display... menu command).

You can center the sequencer topology in the window. With the zoom functions, you can increase or reduce the size of the display (in percentage steps determined by the zoom factor).

Adding Elements

To add further elements to the SFC, select the icon of the required element in the element bar.

The mouse pointer changes its appearance from an arrow to the selected icon with a positioning crosshair. To insert the sequencer elements, position the cross and the required point (the installation position is indicated by a green line) and left-click. The inserted elements are selected and displayed in color.

Syntax Rules

The sequencer topology is formed by the sequences of steps and transitions. The fundamental rule of the sequencer topology is that a step (S) must be followed by a transition (T) and a transition must always be followed by a step (sequence: **S-T-S** or **T-S-T**). The editor automatically adheres to the rules.

Example:

If you insert a simultaneous sequence in a sequencer following a transition and before a step, a transition is created automatically before the step since the syntax rules require a transition before and after a simultaneous sequence.

2.8.2 Configuring Several Sequencers

An SFC can contain several sequencers that can be used for different applications. By specifying different start conditions, you can arrange for a specific sequencer to start when a particular event occurs. You can, for example, configure a separate sequencer for every operating state (idle, active, error, ...) or for every control strategy (heating, cooling, tempering, ...).

Note:

Predefined sequencers are already available for various standard scenarios. These sequencer templates are available in the "SFC Library". You can copy these templates and adapt them to your own purposes.

Summary of the Execution of a Sequencer

If you want to put together the processing of the "active" state and the "holding", "held", and "resuming" states as an alternative sequence in a sequencer, the RUNHOLD input of the SFC must be set to TRUE. If RUNHOLD is set to FALSE, the mechanism for the sequencer change would not function (a sequencer cannot be held and then (re)started and then resumed). If the runtime system detects that a sequencer should be resumed that was previously aborted, this is displayed as an execution error (EXEC_ERR). The configuration must then be corrected according to the rule above.

Start Conditions of the Sequencers

The first sequencer of an SFC has the condition "<SFCName>.RUN=Run" (chart) or "RUN=Run" (type); the start condition of each sequencer added subsequently is empty and therefore not satisfied; in other words, it is never processed. In contrast to newly created transitions that are always satisfied, a new sequencer must always be given a defined start condition (Sequencer Properties > Start Condition tab).

Since it is possible that several start conditions can be satisfied at the same time, you can assign different priorities to the individual sequencers (Sequencer Properties > General tan, Priority: 1 through 255). 1 is the lowest and 255 the highest priority.

You can formulate the start condition of a sequencer so that the status of the operating state logic is checked and the appropriate sequencer executes when the SFC is in a particular state. You can also formulate any other conditions to suit your purposes.

Examples

Example 1:
You configure a sequencer with a start condition that queries one of the control strategies of the SFC. The formulation is, for example, "QCS=Heat". If the SFC is set to this control strategy, the sequencer executes and is not dependent on the operating state of the SFC.

Example 2:
You configure a sequencer with the start condition "<SFCName>.IDLE=Idle". This sequencer executes when the operating state is "Idle".

Example 3:
Any process status can also be queried as the start condition. You interconnect it with the external signal "LOCKERROR" (input of the SFC) →, the SFC changes to the "Error" state if the error signal is applied. You also configure a sequencer for handling the problem whose start condition is, for example "<SFCName>.ERROR=Error AND <process status>=1".

Example 4:
As an alternative to example 3, error handling is also possible without a state change. To achieve this, you configure a sequencer with the start condition "<process status>=1" and assign it high priority. This sequencer always executes when the problem occurs and the sequencer currently executing has a lower priority than the sequencer for handling the problem. In this case, you do not interconnect the process status with the "LOCKERROR" input otherwise this would mean a change to the "Error" state.

Note: With a chart, the address "<SFCName>.I/O", and with a type only "I/O" is specified.

Notes on Configuration

- You can insert a new sequencer consisting of an initial step, a transition, and a final step with the menu commands "Insert > Sequence > Before Current Sequence" or "Insert > Sequence > At End".
- You can copy sequencers and paste them again or move them to change their order.
- The names of the steps and transitions must be unique within a sequencer; the same names can be used in different sequencers.
- You can also configure an additional action for each sequencer. The action consists of a part known as the **preprocessing** that executes after the start of the sequencer in every cycle before the steps and transitions are processed, and a part that executes after processing of the steps and transitions known as the **postprocessing**. This, for example, allows you to make prior settings or pass on the results of sequencer execution.

  You configure both parts in the "Sequencer Properties" dialog box. The dialog box provides the "Preprocessing" and "Postprocessing" tabs whose structure corresponds to the processing phases of steps.

2.9 Creating SFC Elements

2.9.1 Creating a Sequence

When you create a sequence , depending on the position, a step-transition sequence (ST) or transition-step sequence (TS) is created (see Fig. 2-1). You create a sequence by clicking a vertical link in the chart between a step and transition or between a transition and step.

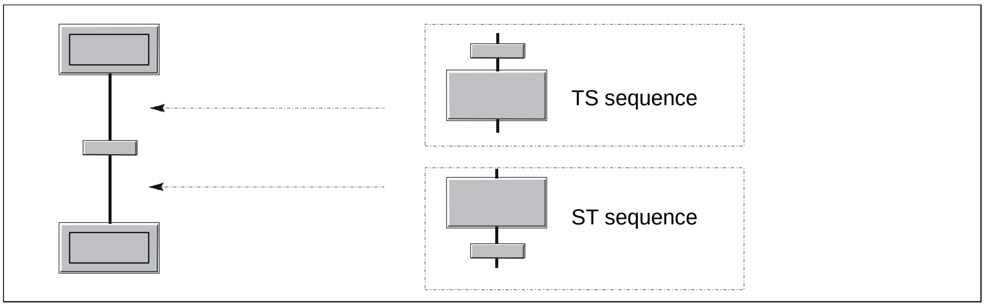


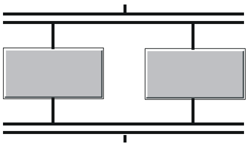
Fig. 2-1: Creating a TS or ST Sequence Depending on the Location

If you move the mouse pointer to a permitted insertion point in the existing sequence, you will see a horizontal green line.

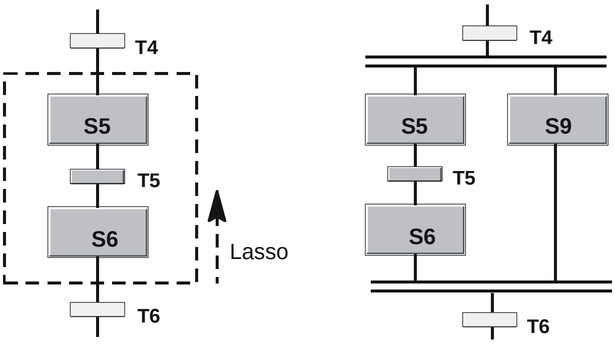
By holding down the left mouse button and dragging the pointer vertically, you can select the length of the sequence you are inserting. The current length that will be inserted (number of ST/TS pairs) is displayed as a number at the insertion point.

2.9.2 Creating and Extending a Simultaneous Sequence

When you create a simultaneous sequence , two sequences are generated, each consisting of a step. Depending on the point of insertion, a further transition is added automatically before or after the simultaneous sequence to ensure that the syntax is maintained.



If you open a lasso (in the Edit mode) around the elements of a sequence path, the enclosed elements become part of the left sequence path of the generated simultaneous sequence.



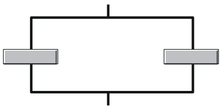
You can add further sequence paths to a simultaneous sequence or delete paths and also insert them in a different sequence. You can move a sequence path within a simultaneous sequence or to any other position in the chart (except in the return branch of a loop). If you delete the second last path, the remaining path is simply integrated in the surrounding structure and the simultaneous sequence is eliminated.

To add further sequence paths, simultaneous sequences, or alternative sequences to a simultaneous sequence, change to the required insert mode and click the mouse with the positioning cross located on the upper or lower double line.

If you move the mouse pointer to a permitted insertion point in the existing sequencer, you will see a horizontal green line. Within the simultaneous sequence (in the vicinity of the upper simultaneous divergence or lower simultaneous convergence), the vertical green line indicates that you are inserting a further sequence path. If, for example, you insert an alternative sequence beside a sequence path, an extra step is created before and after the sequence to maintain the correct syntax.

2.9.3 Creating and Extending an Alternative Sequence

When you create an alternative sequence , two sequences are generated, each consisting of a transition. Depending on the point of insertion, a further step is added automatically before or after the alternative sequence to ensure that the syntax is maintained.



If you open a lasso (in the insert mode) around the elements of a sequence path, the enclosed elements become components of the left sequence path of the generated alternative sequence (refer to the description “Creating a Simultaneous Sequence").

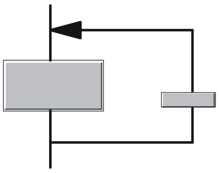
You can add further sequence paths to an alternative sequence or delete paths and also insert them in a different sequence. You can move sequence paths within the alternative sequence or to any other position in the sequencer. If you delete the second last path, the remaining path is simply integrated in the surrounding structure and the alternative sequence is eliminated.

To add further sequence paths, simultaneous sequences, or alternative sequences to an alternative sequence, change to the required insert mode and click the mouse with the positioning crosshair located on the divergence/convergence line.

If you move the mouse pointer to a permitted insertion point in the existing sequencer, you will see a horizontal green line.
Within the alternative sequence (in the vicinity of the upper divergence or lower convergence beside the sequence path), the vertical green line indicates that you are inserting a further sequence path. If, for example, you insert a simultaneous sequence beside a sequence path, an extra transition is created before and after the sequence to maintain the correct syntax.

2.9.4 Creating a Loop

When you create a loop, a sequence path (that can consist of a single step) and a return path with a transition are generated.



You can create loops around existing sequences. You select the beginning and end of the loop by positioning the mouse pointer on the vertical link, holding down the left mouse button and dragging vertically to the required position and releasing the button. The syntax is maintained by adding whatever elements are necessary. If, for example, you create a loop around a transition, the enclosed sequence then consists of this transition and a step before and after it. Below the loop, a further transition is added.

To move the loop, you can select it (including the elements enclosed within the sequence in the loop) and drag it to any position on the vertical line of the sequencer while holding down the left mouse button.

Notes

- Loops leading into or out of alternative or simultaneous sequences are not permitted. Due to the block-oriented sequencer topography, only entire chart elements, including alternative and simultaneous sequences, can exist within a loop.
- The return path of a loop can and must contain only one transition.

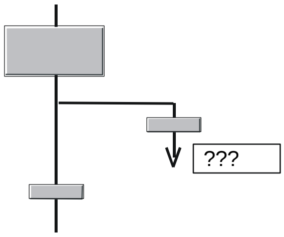
Changing the Return Jump Convergence

To change the return jump convergence of an existing loop, select the horizontal line of the arrow tip and drag it to a different, syntactically correct position in the sequencer (above the starting point).

If you drag the arrow tip below the starting point of the loop, this does not move the return jump convergence but the starting point.

2.9.5 Creating a Jump

When you insert a jump, a transition is created with an arrow and information about the destination of the jump.



To insert a jump, click on the vertical link of the sequence immediately below a step. A simple click creates a jump with an undefined destination. The destination is displayed as question marks (???).

If you require several jumps from one step, repeat the above procedure. The sequence path with the jumps is extended by the number of jumps inserted.

When you insert the jump, you can also select the jump destination directly. Drag the mouse from the point of origin of the jump directly to the destination step and then release the mouse button. Instead of the question mark, the name of the step is now entered as the jump destination.

Note:
If you use jumps into or out off a sequence of a simultaneous sequence, consider the possible consequences for program execution on the CPU.
For more detailed information, refer to the SFC online help in the notes in the topic "Creating a Jump".

Changing the Jump Destination

You specify the destination of the jump by changing the name (???) in the object properties of the destination. Double-click the destination to open a dialog box. All the existing steps of the chart are listed and can be sorted. From this list, you select the step name for the destination of the jump.

If the step name of a jump destination is changed later, all the jumps to the step are automatically adapted.

Deleting a Jump

The jump is deleted if you select the jump transition and delete it.

If the jump destination is deleted, all the jumps to this step become undefined.

2.9.6 Creating and Editing Text Objects

You can insert, delete, copy, and move (to other charts not with Drag&Drop) a text object at any (free) position in the chart.

After inserting a text object using the button in the toolbar ab| or selecting the "Insert > Text" menu command, an open text box is displayed in the window. The text cursor is active and you can begin editing immediately. A line break is added automatically at the right edge of the box. If you enter more text than can be displayed in the box, the size of the box is not increased automatically and the text is moved out of the visible area. You can make the entire text visible by increasing the size of the box manually.

To change the size of a box, click on the box handles and drag with the mouse until the required size is reached. If you change the width of the box, the length of the text lines is automatically adapted. You can pick up the box using the handles of the frame and move it to any position in the window.

When you open a text box, the cursor is positioned at the point in the text at which you clicked with the mouse. You can exit the editing mode and close the text box by clicking outside the text box with the mouse.

Note:

If elements of the sequence topology are covered by the text object, the text object is displayed as a frame with a transparent surface (no content). The chart elements below it remain visible.

Copying, Moving, Deleting

With the mouse pointer over the selected text box (displayed in a frame), open the context-sensitive menu with the right mouse button. Here, you will find the menu commands: "Cut text object", "Copy text object", "Delete text object".

To paste, click on a free position in the chart and then select the “Paste" menu command (context-sensitive menu or “Edit" menu). Click again at the required position to insert the text object (the mouse pointer is displayed as a symbol for “pasting" or “copying").

2.10 Editing SFC Elements

In the “Edit" menu (and in the context-sensitive menu), you will find functions for editing the SFC further.

Copy:
You can copy the selected elements of a sequence path that form a syntactical unit (sequence of elements without gaps) and position them at a different, syntactically correct position within the sequence path or in a different sequence path on the same or another CPU. If necessary, new names may be assigned to the copied elements automatically. The copied elements contain the same actions or conditions as the originals.
The following applies to jumps: If you copy a sequence that contains the step of the jump destination, the copy of the jump destination is adapted accordingly.
If you copy a sequence that contains a jump but the destination of the jump is not within the copied object, the jump destination is undefined (???).

Cutting and **Pasting:** You can move the selected elements of a sequence forming a syntactical unit (without gaps) to another syntactically correct position within the sequencer or into another sequencer of the same or a different CPU (cut and paste).

Delete: The selected elements are deleted from the sequencer topology following a prompt for confirmation ("Do you really want to delete the selected objects?"). If you delete only one element from a syntactical unit, the syntax is immediately restored by entering a new element to suit the syntax (this has no parameters assigned). This means that you have only deleted the parameter settings for the object.

The last step of a sequence in a simultaneous sequence cannot be deleted. To delete a sequence consisting of only one last step, you must select the sequence path by clicking on the vertical link.
The same principle applies to alternative sequences.

If you select the transition in a jump, only its content is deleted. If you select the jump destination (transition and jump destination selected), the jump is deleted.

2.11 Editing in the Properties Dialog

The sequencers and the steps and transitions entered in the sequence topology must now be linked to the “block world" or basic control. You do this in the "Properties" dialog for the sequencers (Section 2.11.1), the steps (Section 2.11.2) and transitions (Section 2.11.3).

The permitted entries for addresses are described in detail in the online help in the topic "Step: Editing addresses".

2.11.1 Editing Sequencer Properties

You can open the sequencer properties with the "Edit > Sequencer Properties..." menu command or with the mouse pointer on the sequencer name (tab at lower edge of window) by right-clicking and selecting the "Sequencer Properties" menu command. A dialog box is displayed in which you specify the properties or formulate the actions and conditions. To edit the sequencer, the Properties dialog is divided into the following tabs:

2.11.1.1 Tab: General

In the “General" tab, you can enter or change the name and comment of the sequencer. In the "Number" box, you can see the number generated by SFC. This cannot be changed. In The "Priority" combo box, you can assign the sequencer a priority of 1 through 255. 1 is the lowest and 255 the highest priority. The priority decides which sequencer of an SFC is started when the start conditions of several sequencers are met simultaneously.

2.11.1.2 Tab: Start Condition

In this tab, you can define the conditions for the SFC chart/type that will start the sequencer (for example, "<SFCName>.RUN = Active" the sequencer when the SFC chart is in the "Active" operating state).

The other editing steps in this tab are identical to those for transition conditions. See Section 2.11.3.2.

2.11.1.3 Tab: OS Comment

The editing steps in this tab are identical to those for OS comments for the transitions. See Section 2.11.3.3.

2.11.1.4 Tab: Preprocessing and Tab: Postprocessing

In this tab you can define the following actions for the SFC chart/type:

- To be executed in every cycle after the sequencer has started before processing the steps and transitions (preprocessing).
- To be executed in every cycle after processing the steps and transitions (postprocessing).

The editing steps in this tab are identical to those for formulating actions of the steps. See Section 2.11.2.2.

2.11.2 Editing Object Properties: Step

You open the object properties by double-clicking the step you want to edit or using the menu command “Object Properties..." of the “Edit" menu or the context-sensitive menu if a step is already selected. A dialog box is displayed in which you specify the properties or formulate the actions.

To edit the step, the Properties dialog is divided into four tabs:

2.11.2.1 Tab: General

In the "General" tab, you can enter or modify the step name, minimum and maximum execution time, the comment for the step and the OS comment. In the "Number" box, you can see the unique number of the step in the sequencer as assigned by SFC.

By clicking the “Confirmation" check box, you can assign a flag to the step. This flag determines the behavior of the step when the chart is run on the CPU in the “T / T and C" mode in other words, transitions following a step with this flag only become active and pass control to successor steps when they are satisfied and confirmed by the operator (as in the “T and C" mode). Without these flags, the successor transitions become active once their conditions are satisfied (as in the “T" mode).

With **Run times minimum**, you can set the minimum time that a step should remain active regardless of whether the successor transition is already satisfied.

With "Run times maximum", you can specify a time for the time monitoring limiting the maximum time that the step can be active.

In the **Comment** and **OS comment** boxes, you can enter comment text, for example, describing the actions to be executed by the step (maximum characters for step: 80, for OS: 512). The OS comment is used in process control for visualizing the step.

2.11.2.2 Tab: Initialization, Processing, Termination

The tabs for the actions (processing phases) **Initialization**, **Processing** and **Termination** have an identical structure. Here, you configure the statements that will control the process during the initial, normal, and final processing of the step (see Section 3.3.6, Phases of a Step).

The statements are adopted as OS comments if the check box is selected in the relevant line.

Formulating Actions

You enter the statements for actions in a formatted dialog. For each step, you can formulate up to 50 statements per action. Ten of these are visible in the dialog box. You can scroll the visible list with the scroll bar.

If you position the mouse pointer on an input box, the complete entry is displayed as fast information including the data type and object type. Example:

3. DP666\Reactor1\Motor3\\CFC8.Ctrllr2.P_SEL [BOOL] CFC

A double backslash (\\ ) is displayed between the path and the CFC name.

In unselected lines, you can use the “Copy/Paste Action" functions in the context-sensitive menu to copy the statements of a complete action and paste them into another action. Using this function, you can, for example, copy the statements from the Initialization" action to the "Termination" action and then change the settings "TRUE" to "FALSE".

You will find the formulation options in the online help for SFC.

Texts on a yellow background reference objects that do not yet or no longer really exist (block I/Os, charts, runtime groups).

This reference is a textual interconnection that can be closed automatically during compiling (option) or explicitly using the **Options > Close Textual Interconnection** menu command if the interconnection partner actually exists. The textual interconnection is displayed on a yellow background in the line of the address. The same highlighting is used if a previously real reference becomes a textual interconnection after deleting the interconnection partner.

Textual interconnections that cannot be closed are ignored during compilation. A warning is entered in the log indicating which I/Os still have a textual interconnection. One unclosed textual interconnection is also tolerated in a download.

Note:

Interconnections cannot be set up unless the target of the interconnection is unique; in other words, exists more than once with the same name. The interconnection is then also displayed as a textual interconnection and cannot be made. Example: The object name of an SFC is the same as the symbolic name of a DB.

Entering Addresses

Using the "Browse" dialog:
The ”Browse” dialog has four different tabs: "Plant view", "Component view", "Runtime groups" and "Symbols". When you browse (for example through CFC charts in the plant or component view), all the available objects of the chart folder are found and displayed.

Using the "Filter" button, you can limit the list of I/Os; in other words, display only those currently relevant at the present stage of configuration.

You can then select the chart you require and then the block and I/O. The dialog box remains open until you close it explicitly with the "Close" button or you close the "Object Properties" dialog.

Note on SFC Types

With an SFC type, you can only access the I/Os of its own interface. Although all other I/Os are displayed in the "Browse" dialog, they cannot be adopted.

From the Interface editor or Characteristics editor:
You can also drag the I/Os of the SFC directly from the window of the I/Os to the address box of the Properties dialog.

Note: Several I/Os can exist for a characteristic, for example for setpoints, timers, and block contacts. In this case, a list of available I/Os is displayed before you insert an address and you can select the required I/O.

From a CFC chart:
From the open CFC chart, you can select the block I/Os and drag them to the address box of the Properties dialog.

If you insert a block I/O with a **value identifier** in the address box, the value identifier is also entered in the right (empty) address box. If there is more than one value identifier for this I/O, you can open the combo box with the value identifiers in the right address box In the drop-down list box, you can select the required value identifier for this address.

To allow the value identifiers to be displayed in SFC, the option Parameter: Value identifier" must be selected in “Options > Customize > Layout...".

By editing:
When making textual entries, make sure that names are consistent. A symbol that does not exist in the symbol list (or a referenced symbol that is subsequently renamed) cannot be checked in the editor and is set up as a textual interconnection.

In the addresses, you can enter CFC block I/Os whose block does not yet really exist in the CFC chart. These instructions are displayed in the statement list on a yellow background and are set up as textual interconnections.

Example: Turn SFC chart on and off
With the statement "<SFCchart>.INTONOFF := TRUE" an SFC chart is turned on. With the statement "<SFCchart>.INTONOFF:= FALSE" an SFC chart is turned off; in other words, its final step is processed before the chart is completed.

Note:
When you enter statements, they are checked so that correct syntactic and semantic formulation is guaranteed. The program, for example, also checks whether the data types of the logically combined addresses are compatible.

With SFC access to CFC blocks, please remember the following:

- Being able to modify a type centrally means that it is possible to replace or modify blocks of which CFC block instances have already been generated. The type changes are also made to the CFC blocks. If there are SFC accesses to modified blocks, these modifications must also be made to the addresses in actions and transitions.
- Access to CFC blocks can also be modified in the relevant CFC chart. This modification is, however, restricted to "rewiring" SFC accesses; in other words moving the access to another block I/O (with ALT + Drag&Drop).

Access to Structures

In the **step properties**, you can configure structure assignments. Structures, substructures, or structure elements are possible according to the following syntax:

Structure:	<I/O name>
Substructures:	<I/O name>.[<Substructure>...]<Substructure>
Structure element:	<I/O name>.[<Substructure>...]<Element>

You can enter this information directly in the statement lines or by using the "Browse" dialog. You apply substructures or structure elements in the "Structure Dialog".

Note on SFC Types

The use of structures of the I/O type IN_OUT is not permitted. Although SFC tolerates the assignment, this is detected as an error when the S7 program is compiled.

You will find more detailed information on the use of structure access and the structure dialog in the SFC online help in the topic "Access to structures".

Documenting the Configuration

You can document the configured actions of the step.
With the "Print" button in the Object Properties dialog, you obtain a log of the step with information about the properties and assignments for initialization, processing and termination.

2.11.3 Editing Object Properties: Transition

You open the object properties by double-clicking the transition you want to edit or using the menu command “Object Properties..." of the “Edit" menu or the context-sensitive menu if a transition is already selected. A dialog box with three tabs is opened in which you can specify the properties, the conditions and the OS comments.

2.11.3.1 Tab: General

In the “General" tab, you can enter or change the name and comment. In the "Number" box, you can see the unique number of the transition in the sequencer as assigned by SFC. There are no gaps in the sequence of numbers; in other words, if a transition is deleted and a new transition inserted later, the new transition is given the number of the previously deleted transition.

2.11.3.2 Tab: Condition

In the "Condition" dialog box, you select the control passing conditions for the selected transition.

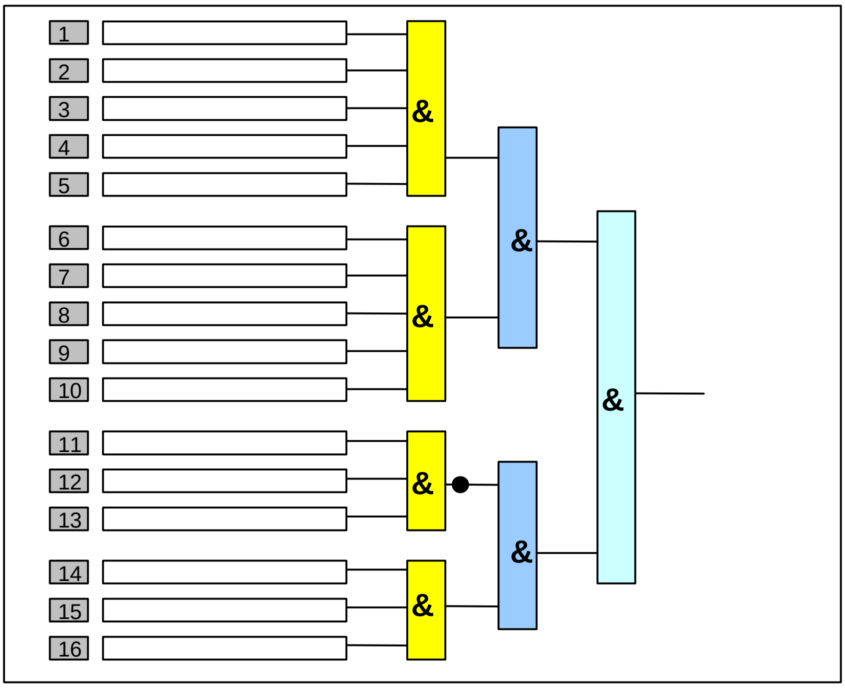


Fig. 2-2: Logical Combination of Conditions with Three-Stage Transition Logic

Formulating the Conditions of a Transition

You formulate a transition as a Boolean expression, that can consist of 2 x 5 and 2 x 3 conditions. The conditions are combined using three-stage transition logic.

The Boolean operators are designed as buttons. With a simple mouse click on the operator, you can change it from "AND (&)" to "OR (≥1)". To make a "NAND" from an "AND" and a "NOR" from an "OR", click the output of the operator. The negation is displayed by a period in bold print on the output line.

You will find the formulation options in the online help for SFC.

If you position the mouse pointer on an input box, the complete entry is displayed as fast information including the data type and object type.

You should whenever possible enter the addresses using the "Browse" selection dialog or from the open CFC chart since this means that the addresses are unique.

Note on SFC Types

With an SFC type, you can only access the I/Os of its own interface. Although all other I/Os are displayed in the "Browse" dialog, they cannot be adopted.

As when selecting addresses for the steps, you can also enter the value identifiers of the I/Os for the transitions.

The following applies to access to structures: In the transitions, it is only possible to compare structure elements.

Texts on a yellow background reference objects that do not yet or no longer really exist (block I/Os, charts, runtime groups).

This reference is a textual interconnection that can be closed automatically during compiling (option) or explicitly using the **Options > Close Textual Interconnection** menu command if the interconnection partner actually exists. The textual interconnection is displayed on a yellow background in the line of the address. The same highlighting is used if a previously real reference becomes a textual interconnection after deleting the interconnection partner.

Textual interconnections that cannot be closed are ignored during compilation. A warning is entered in the log indicating which I/Os still have a textual interconnection. One unclosed textual interconnection is also tolerated in a download.

Notes

- Interconnections cannot be set up unless the target of the interconnection is unique; in other words, exists more than once with the same name. The interconnection is then displayed as a textual interconnection and cannot be made. Example: The object name of an SFC is the same as the symbolic name of a DB.
- When you enter conditions, they are checked for consistency so that correct syntactic and semantic formulation of the subexpression is guaranteed. The program also checks whether the data types of the logically combined addresses are compatible.
- The predefined three-stage logic is normally adequate. If you require more complex formulations, you can create them as a CFC chart and enter the results calculated in the chart as the address for transition conditions.

2.11.3.3 "OS Comment" Tab

Here, you can enter a comment text for every condition that is displayed during process control on the OS.

When you first open the “OS Comment" dialog, the formulated condition is entered as an OS comment and can then be changed as required.

- The OS comment for the SFC type and SFC instance can be different. If the SFC type has an I/O that is interconnected in the SFC instance, interconnection partner is indicated in the instance. If this is an output with multiple interconnections only one of the interconnections is displayed.
  If the I/Os are not interconnected, the OS comments of the SFC type and SFC instance are identical.

  Example:

	SFC type	**SFC instance**
interconnected	RUN = TRUE	chart\block.io = TRUE
not interconnected	RUN = TRUE	chart\sfc_instance.RUN = TRUE

- Unless you edit and modify the OS comment, the OS comment is adapted automatically whenever the compare condition is changed. This automatic updating also applies to renaming, copying, moving or deleting the referenced CFC blocks.
- If you change the condition belonging to a comment text that you have edited, this is not automatically adapted. In this case, a dialog box is opened telling you that you should check the comment text. You can cancel the change in this dialog box by clicking the “Cancel" button.
- You can initialize an individual OS comment in the SFC editor by deleting the OS comment. The text from the compare condition is then used again as the OS comment and from this time onwards is automatically adapted whenever the condition is changed providing you do not edit the OS comment.

  You can enter the original OS comment (text of the condition) again using the “Use Default" context-sensitive menu command.
- You can edit the OS comments later. With the "Options > Edit OS Comments..." menu command, you open a dialog in which you specify whether you want the OS comment for the transitions to remain unchanged or want to use all conditions. In this dialog, you can also specify whether the editing of the OS comments applies to the entire chart folder, the current chart, or the selected steps/transitions.

Documenting the Configuration

You can document the configured conditions of the transition. With the "Print" button in the Object Properties dialog, you obtain a log of the transition with information about the properties and parameters of the conditions.

2.12 Compiling and Downloading

General Aspects of Compiling

When compiling (scope: entire program) all charts, SFC types, and SFC instances of the current chart folder are converted to source language and then compiled. After modifying the SFC chart (SFC type, SFC instance), you can compile so that only the changes are compiled.

When you compile, first all SFC types then all SFC instances and then all SFC charts are compiled. Finally the CFC charts are compiled.

The SFC type is compiled only when a modification has been made to the type since the last compilation that is relevant for compilation.

After providing the SFC type with an interface, a function block type is generated containing the I/Os defined in the interface and the configured sequencers. As with the SFC chart, an action and transition FC are also generated.

Saving Settings without Compiling/Downloading

You can save the settings in the dialog box without starting compilation or a download (“Apply" button). This can, for example, be useful when you want to use the "Compile and download function" function in the SIMATIC Manager. Refer to the online help of the SIMATIC Manager.

2.12.1 Compile

Compilation Settings

With the “Customize > Compilation..." command in the “Options" menu, you open a dialog box in which you will see information about the resources used in conjunction with compiling charts. Here, you can

- decide which warning limits will apply so that possible dangers are detected before you download.
- decide which resources should remain unused during compilation of the charts of the current chart folder.
- view the statistics showing how many resources (DBs, FCs) are available for compiling and how many are already being used.

Compile

The "SFC > Compile..." menu command opens a dialog box in which you can select one of the option buttons "Entire program" (all objects are compiled) or "Changes only" (only the objects changed since the last compilation are compiled).

- **Delete empty runtime groups** option: If this option is set, the empty runtime groups are deleted prior to compiling.
  The empty runtime groups can occur as a result of copying when branching and merging projects. When these empty runtime groups are created, the original names of the runtime groups are lost due to implicit incrementation of the numbers.
- **Make Textual Interconnections** tab;: If this option is set, all textual interconnections for which the referenced interconnection partner exists are made and turned into real interconnections.

  Note: Note: If the option is not set or if the textual interconnections cannot be made, they are ignored during compilation. A warning is entered in the log indicating which I/Os still have a textual interconnection. One unclosed textual interconnection is also tolerated in a download.
- The options "**generate module drivers**" and "**update sampling time**" are used in CFC and are relevant only when CFC charts have been modified that are also compiled in this S7 program.

Select the type of compilation you require ("Entire program" or "Changes only") and start the compilation with "OK".
The charts of the current program (chart folder) are checked for consistency and then compiled.

Consistency Check

During compilation, a consistency check is run automatically. The messages are written to a log file.

If errors and warnings occur, you can jump to the relevant SFC by double-clicking the message.

You can also run the consistency check without a full compilation by selecting “SFC > Consistency Check...".

Logs

Following the consistency check or compilation, you can display and print out the messages of both activities using the menu command “Options > Logs".

You can display and print the result of the consistency check and any messages resulting from compilation in the "Logs" dialog box (Options > Logs ... menu command).

Comparing the CPU

If you want to compare the time stamp of the last modifications before downloading, you can start the “Compare" function in the “CPU" menu. A dialog box with the date and time of the following is displayed:

- The last download-relevant change
- The last offline program change
- The last online program change

If the time stamp of the last download-relevant change is older than the last offline program change, this has no effect on the running of the program on the CPU; you do not need to download the program again.

If the time stamp of the offline program modification is older than the download-relevant modification, you must compile the charts and download them to the CPU so that they match.

If the time stamp of the online program modification is older than the time stamp of the offline program modification, you must download the user program from the PC/PG to the CPU so that they match.

Note on H CPUs: If the H CPU is in solo mode, for example after the failure of a CPU and there was a CPU failover, an online access results in a dialog being displayed. In this dialog, you can select the required CPU. In redundant mode, this dialog does not appear.

2.12.2 Download

After you have compiled, you can download the user program to the CPU.

The following requirements must be met before you can download:

- There must be a connection between the CPU and your PG/PC.
- The Edit mode is set.

With the programs created in SFC, you must always download to the CPU from SFC (or CFC), since only this download function guarantees the consistency of the configuration data with the CPU data. The same download function is also used if you select the "PLC > Compile and Download Objects" menu command in the SIMATIC Manager.

On the other hand, copying blocks in the "Offline Block Folder" to insert them in the "Online Block Folder" is not permitted.

If these requirements are met, you can start the download with the “CPU > Download..." menu command.
If you have made a download-relevant change in the configuration and have not compiled since you made the changes, you will be prompted to compile before you download. If the compilation is free of errors, the download will be started automatically when compilation is completed.

In the "Download" dialog box, you can choose between "Entire program" (CPU in STOP) and "Changes" (CPU can be in RUN-P) and "Download to test CPU (entire program).

Downloading to the CPU, just like working in test mode, is a protected function in S7 that must be logged. Prerequisite: The SIMATIC Logon Service is installed. The protected actions are logged in the **change log** if the current chart folder was activated for the change log (Chart folder > Object properties... > Change log tab).

Downloading Entire Program

If you select "Download entire program", all the charts of the active chart folder are downloaded to the CPU. After prompting you for confirmation, the CPU is set to "STOP" and all the blocks on the CPU are deleted.

Note on H CPUs: If the H CPU is in solo mode, for example after the failure of a CPU and there was a CPU failover, an online access results in a dialog being displayed. In this dialog, you can select the required CPU. In redundant mode, this dialog does not appear.

Downloading a Changed Program to a Test CPU

With the option "Download to test CPU", you can download a changed program for testing to a different CPU or to PLCSIM without losing the ability to download changes to the original CPU.

When you download as a test, the entire program is downloaded to the test CPU (or to PLCSIM) without losing the download identifiers and without the comparison stamp being written to the ES data management.

Downloading Changes Only

If you select "Download changes only" in the "RUN-P" CPU mode, you can download changes to your configuration to the CPU without having to change the CPU to STOP mode. With this type of download, you only download changes that have been made since the last download.

Notes on Downloading Changes

- If the sequencer topology of SFC charts/SFC types has been changed (steps or transitions have been added, deleted, copied, moved, jump destination changed...), and if the modified sequencer is active, these charts or all instances of the type must be deactivated. Deactivating before the download and reactivating after the download is handled by the downloader following positive acknowledgment of a query. Otherwise, the download is aborted.
- Modifications to the interface of the SFC type are transferred to the SFC instances immediately. The SFC instances must therefore be deactivated during downloads and execution stopped on the CPU. Deactivating before the download and reactivating after the download is handled by the downloader following positive acknowledgment of a query. Otherwise, the download is aborted.
- If SFC charts/types/instances have been modified (chart properties, object properties of the steps/transitions) without changing their structure, you can download the charts after they have been compiled while the CPU is in RUN without needing to deactivate the modified SFC chart.
- If you have not changed the SFC itself, but only the objects that are accessed (for example a symbol in the symbol table, runtime groups, block I/O), you do not need to deactivate the SFC before downloading changes.
- Following a download of changes, an SFC that is off and has the "Autostart: on" property is not started automatically but must be restarted by the operator using the external view/SFC instance.

- **F Systems:** Downloading changes to programs with a modified F part requires the entry of an F password. Without this legitimization, downloading is aborted.
- **H systems:** If you download program changes to a CPU operating in the solo mode and then "switchover with modified configuration" (menu command **CPU > Operating Mode...**), these changes are lost and you then have to download the entire program.
  Remedy: Downloading in redundant mode. In this case, you must make sure that the operating mode remains unchanged until the download is completed.

⚠ **Caution**
Read the information about the causes of stoppages when downloading changes in the online help.

2.13 Parameter-Controlled Operation

Below, you will find brief information about parameter-controlled operation and how it is configured. For a more detailed description with examples, refer to the online help of SFC.

What does parameter control mean?

Apart from fixed sequential control systems, parameter-controlled systems are used in BATCH processes. Parameter-controlled sequential control systems have variable parameters.

These variable parameters are located in a shared data block (recipe data block). You can give the recipe data block a symbolic name, for example "RecParDB".

The variables of the recipe data block are values assigned to the parameters of the basic automation during configuration.

Configuring Parameter-Controlled Operation

You configure parameter-controlled operation with the SFC editor. The procedure is basically the same as when configuring the sequential control system. During processing in the Object Properties dialog, you assign the values from the recipe data block to the parameters of the basic control.

Modification on the OS

By changing the content of the recipe data block, parameter-controlled operation is possible with different parameter sets. The parameters are changed on the OS.

3 Sequential Control System on the CPU

Introduction

This chapter describes the functions of sequential control systems on the CPU, when steps and transitions are processed, and the effects of a statement in the processing phase (action) of a step.

3.1 General

Prerequisites

The SFC created with the SFC editor is defined by the sequencers (start condition and pre/postprocessing) and their topologies, by the actions of the steps, the conditions of the transitions, and by the runtime properties; the program has been compiled and downloaded to the CPU.

State of the Sequential Control System

After downloading to the CPU, the sequential control system is in a defined state. This defined state is specified with the SFC editor (default: "Idle"). By assigning start conditions and processing sequences for the sequencers, different sequences are executed depending of the SFC status.

You can modify how a chart is executed either by changing the operating parameters using the SFC editor during testing and commissioning or on the OS, for example by changing the step control mode “step control with transition only (T)" to “step control confirmation by operator (C)" or changing the time monitoring from “Off" to “On".

To achieve a required operating state, the SFC is influenced, for example by the operator commands or controlled by the program.

Interaction with the Basic Control

On the CPU, the sequential control system has relationships to the basic automation over the action and transition functions. If the parameter-controlled operation is involved, there are also relations to the parameter data.

Each SFC has a certain run behavior assigned to it. Basic control with the blocks located in CFC charts may have a different runtime behavior from that of the SFC chart. This can be controlled by placing the external view of an SFC chart in the CFC chart. In the same way, an SFC instance is controlled by the blocks placed in the CFC chart.

The structure of the runtime system allows the sequential control system and the blocks of the basic automation to run in different cycles to reduce the cycle load. In the same way, SFCs can be installed in runtime groups and can then have different scan rates/phase offsets.

3.2 How an SFC Executes

3.2.1 Mode

The mode decides whether the running of the program is controlled by the operator or is automatically controlled by the CPU program.

The following modes are possible for an SFC chart:

- AUTO (process mode)
  Execution is controlled automatically. Control by the CPU program is implemented in the CFC chart by suitable assignment of parameter values or interconnection of inputs of the external view of the SFC chart or SFC instance.
- MANUAL (operator mode)
  Execution of the chart is controlled manually by an operator (for example during commissioning or with SFV). All step control modes are permitted.

The enable for changing to AUTO can also be set in the external view or in the SFC instance; in other words the explicit enable by the operator can be made per SFC with a central operator command or by the program (ENAUT I/O). The also makes a change from MAN to AUTO possible controlled by the program.

3.2.2 Step Control Mode

The various step control modes change the response of the SFC when passing control from one step to the successor(s). It is possible to change the step control mode in all operating states. The individual step control modes are mutually exclusive.

Table 3-1: Step Control Modes

Acronym	Step control mode...
T	Transition
C	Confirmation by operator
T and C	Transition **and** confirmation by operator
T or C	Transition **or** confirmation by operator
T / T and C	Step-specific confirmation by operator

For a more detailed description of the step control modes, refer to the online help of SFC.

Note:

In the step control modes "C" and "T or C", the minimum run time of the step can be overridden by the operator.

3.2.3 Execution Options

With the execution options (for example cyclic operation, time monitoring etc.), you can specify how the sequential control system executes. The individual execution options can be combined: The following execution options exist that you can change in the "SFC > Properties" dialog, "Operating parameters CPU" tab:

- If "instruction output: *on*" is set, the actions of active steps are processed; If "instruction output: *off*" is set, they are not processed.

  During commissioning, or if errors occur, blocking command output in conjunction with the suitable step control mode ("C", "T or C") can bring the sequential control system to a defined state without influencing the process.

  - If instruction output is off, the minimum run time of a step is not effective.
  - A change to instruction output has no effect on the steps currently to be processed, but only on those to be processed in the future, in other words, following a step change. This means that either all actions of a step are processed or no action is processed.

- If "Cyclic execution: *on*" is set, execution automatically continues with "Starting" from the "Completed" state.
  A sequencer to be processed in the "Completed" state is exited immediately in cyclic operation; in other words only the initial and final step are processed.

  In "Cyclic execution: *off*", the sequential control system remains in the "Completed" state.
  As long as no command to exit the state is received, "Completed" continues to be processed cyclically.
  This applies to all states that can only be exited as a result of a command.

- With "Time monitoring: *on*", after a step has been activated, the active time of the step is continuously compared with the monitoring time (object properties of the step, "General" tab, option "Run times maximum") and a step error is reported if the time is exceeded.

  With "Time monitoring: *off*", there is no comparison of the active and monitoring time.

- An SFC with the ”Autostart: *on* ” property starts up without any further operator input after the CPU restarts; in other words, it changes to the "Starting" state. After downloading changes (CPU does not change to STOP), there is no automatic start and the SFC must be started again by the operator or the program (for example, external view in CFC).

  The setting for "Autostart" can only be modified in the "Chart > Properties dialog, "Operating parameters CPU" tab.

- If "Use default operating parameters when SFC starts :*on*" is set, all the defaults / options set in the runtime properties that may, for example, have been changed in test mode are effective again when the SFC chart starts.

  The defaults for the execution options are listed in Section 3.3.

3.3 The Behavior of a Sequential Control System

What Determines How a Chart Behaves?

The behavior of a sequential control system depends on the operating parameters: Operating mode, Mode, Step control mode and the Execution options.

You can set the operating parameters when testing and commissioning or at the OS with SFC Visualization (except for the "Autostart" parameter that can only be set in the Properties dialog of the SFC chart or SFC type in the "Operating Parameters CPU" tab).

The **Default settings** for the operating parameters are:

Table 3-2: Defaults of the SFC Operating States

Defaults		
Step Control Mode	T	transition (process controlled)
Mode	MAN	Operator control mode
Instruction output	on	
Cyclic execution	off	
Time monitoring	off	
Start options		
Autostart	off	
Use default operating parameters when SFC starts	off	

The following operating mode is set after the CPU has started up:

Idle	(if Autostart = off)
Starting	(if Autostart = on)

3.3.1 The Operating States

How is the operating state influenced?

- In operator control mode (MAN) using commands in test mode or SFC Visualization.
- In process mode (AUTO) using interconnections with the external view of the SFC chart or with the SFC instance.

What does the operating state indicate?

The operating state of the sequential control system indicates the current state and, for example, whether operator intervention is necessary for continued operation or which commands are possible to change to a different operating state.

What are the tasks of the operating state logic?

The operating state logic of SFC (**SFC-OSL**) describes

- the states that can be adopted by an SFC chart or an SFC instance.
- the state changes possible in a particular state.
- the events that bring about a state change.

In addition to the SFC-OSL, there is also a separate simpler operating state logic for the sequencers configured in an SFC, the **sequencer OSL**. This describes

- the states that can be adopted by a sequencer.
- the state changes possible in a particular state.
- the events that bring about a state change.

The relationship between SFC-OSL and sequencer OSL results from the possibility of being able to execute one (of several) sequencers of the SFC in every state of the SFC-OSL.

3.3.2 Operating State Logic for SFC (SFC-OSL)

The current operating state of the SFC-OSL can be changed by the following events:

- Commands (Start, Resume, Hold, ...) in the "MANUAL" or "AUTO" modes.
- External signals (inputs of the SFC, commands from another SFC, ...).
- Internal signals (commands from own sequencers, from the test mode or SFC Visualization).
- Implicit state change.

The operating state logic of an SFC is defined by the diagram of state changes:

The SFC-OSL contains several state changes that were retained to ensure compatibility with older projects. In the diagram, these are shown as **broken blue lines**.

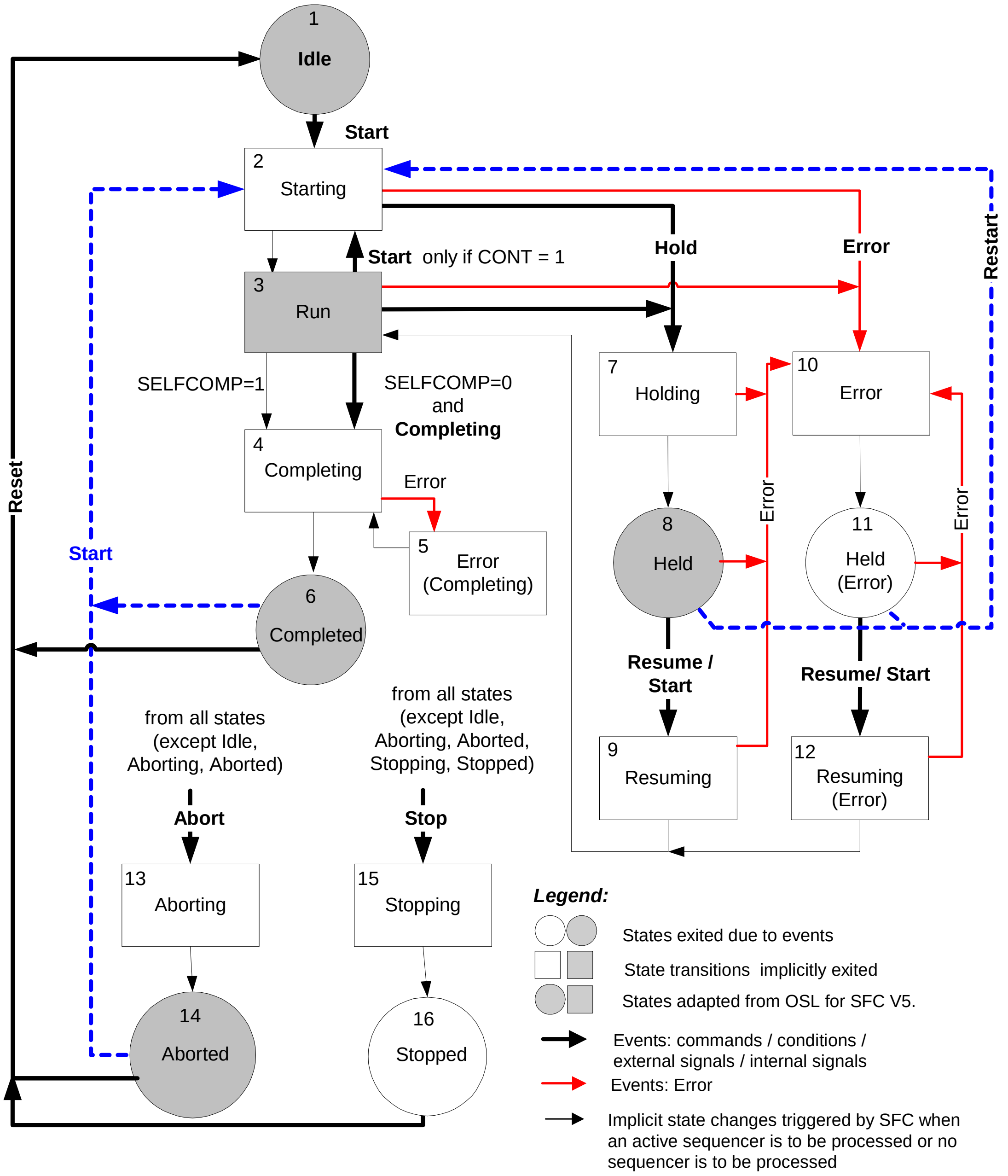


Fig. 3-1: Diagram of State Changes for SFC-OSL

The **numbers** in the diagram are identifiers for the individual states whose significance is explained in the following Table 3-3.

Operating States (SFC-OSL)

Table 3-3: Operating Sates SFC

No.	Status	Meaning
1	Idle	Basic state; Waiting for start command.
2	Starting	Processing start after start command.
3	Active	Normal execution after completing startup.
4	Completing	Completing execution after the complete command or implicit completion.
5	Error (completing)	Error processing during completion.
6	Completed	Completion finished; Waiting for reset or start command.
7	Holding	Processing hold after hold command.
8	Held	Holding finished; Waiting for resume command.
9	Resuming	Processing resume after resume command.
10	Error	Error processing after error.
11	Held (error)	Error processing finished and no further error pending; Waiting for resume command.
12	Resuming (error)	Processing resume after resume command.
13	Aborting	Processing abort after abort command.
14	Aborted	Abort finished; Waiting for reset or start command.
15	Stopping	Processing stop after stop command.
16	Stopped	Stop processing finished; Waiting for reset command.

The following tables describe the changeovers between the states (source state no. / target state no.) and what triggers them.
X = possible from several states.

State Changes due to Commands (SFC-OSL)

Source/ Target	**Command** MAN (AUTO)	**Meaning**
X/2	START	Trigger start processing by changing to the "Starting" state.
3/4	COMPLETE	Trigger complete processing by changing to the "Completing" state.
2/7 3/7	HOLD	Trigger hold processing by changing to the "Holding" state.
8/9 11/12	RESUME	Trigger resume processing by changing to the "Resuming" or "Resuming (error)" state.
X/10 4/5	ERROR	Trigger error processing by changing to the "Error" or "Error (completing)" state.
X/13	ABORT	Trigger abort processing by changing to the "Aborting" state.
X/15	STOP	Trigger stop processing by changing to the "Stopping" state.
X/2	RESTART	Trigger start processing by changing to the "Starting" state.
X/1	RESET	Change to the "Idle" state.

State Changes due to External Signals (SFC-OSL)

The external signals are inputs of the SFC with which changes between states of the OSL can be influenced by other blocks (for example, by interlock blocks).

The external signals work, however, only within the limits of the OSL. There can therefore be no state change that is not specified in the state logic.

Source/ Target	**External Signal**	**Meaning**
3/4	LOCKCOMPLETE	Trigger complete processing by changing to the "Completing" state.
2/7 3/7	LOCKHOLD	Trigger hold processing by changing to the "Holding" state.
X/10 4/5	LOCKERROR	Trigger error processing by changing to the "Error" or "Error (completing)" state. The error processing is executed completely and the state then changes to "Held (error)". If an error is pending, the state changes back to "Error" immediately.
X/13	LOCKABORT	Trigger abort processing by changing to the "Aborting" state.
X/15	LOCKSTOP	Trigger stop processing by changing to the "Stopping" state.

State Changes due to Internal Signals (SFC-OSL)

Internal signals are SFC-internal commands that are set in steps to cause a state change in the SFC. The signals are checked by the OSL and reset automatically by this after a state change.

The internal signals work, however, only within the limits of the OSL. There can therefore be no state change that is not specified in the state logic.

Source/ Target	Command	Meaning
X/2	INTSTART	Trigger start processing by changing to the "Starting" state.
3/4	INTCOMPLET E	Trigger complete processing by changing to the "Completing" state.
2/7 3/7	INTHOLD	Trigger hold processing by changing to the "Holding" state.
8/9 11/12	INTRESUME	Trigger resume processing by changing to the "Resuming" or "Resuming (error)" state.
X/10 4/5	INTERROR	Trigger error processing by changing to the "Error" or "Error (completing)" state.
X/13	INTABORT	Trigger abort processing by changing to the "Aborting" state.
X/15	INTSTOP	Trigger stop processing by changing to the "Stopping" state.
X/2	INTRESTART	Trigger start processing by changing to the "Starting" state.
X/1	INTRESET	Change to the "Idle" state.

Implicit State Changes (SFC-OSL)

The states "Starting", "Completing", "Holding", "Resuming", "Error", "Resuming (error)", "Aborting", "Stopping" are transitional states. Once a transitional state has been executed completely; in other words, there is no start condition satisfied for a sequencer or a started sequencer has been completely executed, the state changes to the next state defined in the OSL (implicit change).

The "Active" state is also treated as a transitional state when the SFC input SELFCOMP=1 is set (default for chart and type). Otherwise, the "Active" state can only be exited explicitly with the "Complete" command.

Regardless of this, transitional states can, of course, also be exited due to an external problem or due to the commands "Abort" and "Stop". In this case, the appropriate change is made and not the implicit change.

Source/ Target	SFC Status	Meaning
2/3	Sequencer idle	Start processing finished; change to "Active"
3/4	Sequencer idle	Normal processing finished; Change to "Completed" (prerequisite: external signal SELFCOMP=1)
4/6	Sequencer idle	Completion processing finished; change to "Completed"
7/8	Sequencer idle	Hold processing finished; change to "Held"
x/3	Sequencer idle	Resume processing finished; change to "Active"
13/14	Sequencer idle	Abort processing finished; change to "Aborted"
15/16	Sequencer idle	Stop processing finished; change to "stopped"

State Control using SFC I/Os (SFC-OSL)

SELFCOMP, **SELFRESET**, **RUNHOLD** and **CONT** are SFC inputs that can have parameters set and influence the behavior of the OSL.

SELFCOMP is effective in the "MAN" and "AUTO" modes.

Source/ Target	External Signal	Meaning
3/4	SELFCOMP	SELFCOMP=1 (SFC self terminating): SFC changes implicitly from the "Active" state to the "Completing" state after normal processing is completed. SELFCOMP=0 (SFC not self terminating): SFC remains in the "Active" state until the "Complete" command is received. If there is a state change due to the "Complete" command, normal execution is aborted if it was not yet completed.

SELFRESET effective only in "MAN" mode.

Source/ Target	External Signal	Meaning
X/1	SELFRESET	SELFRESET=1: In the "Completed", "Aborted" and "Stopped" states, SFC makes an automatic and immediate change to the "Idle" state; in other words, in the above states, a sequencer is started and immediately aborted. SELFRESET=0: If the "Completed", "Aborted" and "Stopped" states have been executed completely, the SFC remains in the corresponding state until the "Reset" or "Start" command is sent ("Start" not when "Stopped"). Note: In cyclic operation, SELFRESET must not be set to 1. This causes an error (LI_ERR).

RUNHOLD is effective in the "MAN" and "AUTO" modes.

Source/ Target	External Signal	Meaning
3/7	RUNHOLD	RUNHOLD=1: When there is a state change from "Active" to "Holding", the previous sequencer is **aborted** and the new sequencer started. RUNHOLD=0: When there is a state change from "Active" to "Holding", the previous sequencer is **Held** and the new sequencer started.

3.3.2.1 Continuous Mode

In AUTO mode, the additional CONT status flag can be used to allow cooperation between a user controller (or SIMATIC BATCH) and SFC. The purpose of this flag is to avoid an SFC needing to be turned off temporarily when the SFC is started again. The OSL of the SFC remains in the "Active" state and sets the QCONT output to 1. Commands are executed regardless of the status memory bit.

The SFC can then only be restarted when READY_TC = 1 is set. This is set by the SFC after processing the final step of a sequencer to be processed in the "Active" state. If the same SFC is restarted or aborted, the state flag QCONT is set to 0.

If the SFC is not started within a selectable time (CONT_TC # 0), the error output QCONT_T_ERR is set to 1. This output allows individual reactions to the error.

See also "Starting an SFC", Section 3.3.6.

3.3.3 Operating State Logic for Sequencers (sequencer OSL)

The execution of sequencers is controlled by the sequencer-OSL.

The sequencer OSL is run through independent of the SFC-OSL when a sequencer executes. This means that the sequencer has a state that must be distinguished from the state of the SFC. The SFC-OSL can, for example, be in the "holding" state, whereas (due to the processing of the sequencer for the "holding" state), the sequencer OSL is in the "active" state. The processing of the sequencer OSL is subject to the SFC OSL so that state changes in the SFC OSL usually mean a state change in the sequencer OSL.

The operating state logic of the sequencer is defined by the diagram of state changes for the sequencer OSL:

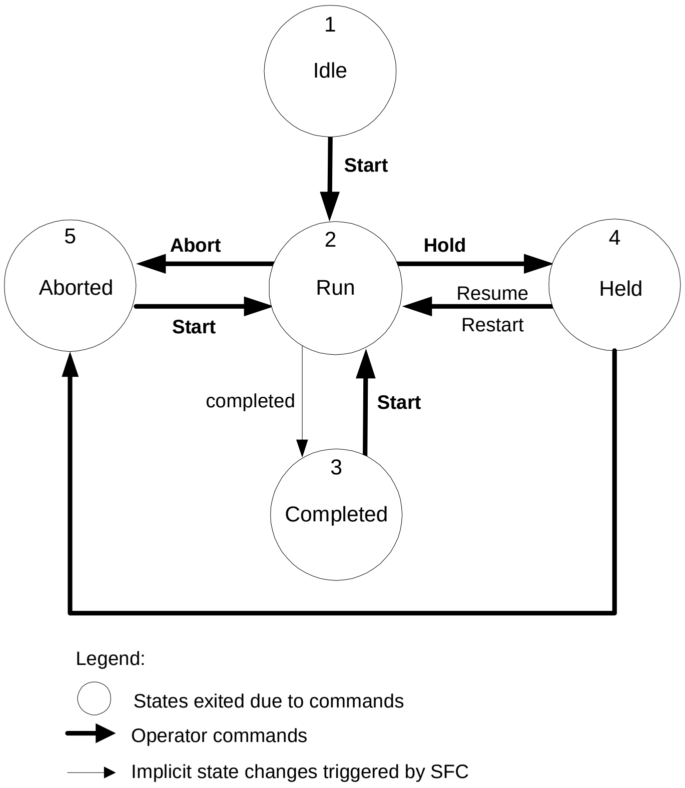


Fig. 3-2 : Diagram of State Changes for Sequencer OSL

The States of the Sequencer OSL

Table 3-4: Operating Sates Sequencers

No.	**Status**	Meaning
1	Idle	Basic state; Waiting for start command.
2	Active	Normal execution.
3	Completed	Normal processing finished; Waiting for reset or start command.
4	Held	Hold processing finished; Waiting for resume command.
5	Aborted	Abort processing finished; Waiting for reset or start command.

3.3.4 Commands

In "MAN" mode, you can set or modify the operating states using the buttons of the operator interface in SFC test mode or in SFC Visualization (refer to the table "MAN Commands for the SFC"). Operator control depends on the operating state and the command enables "EN......".

In "AUTO" mode, depending on the command enables "EN....", the commands are made over the interface inputs "START, ..." (by assigning parameters or interconnecting to a higher-level automatic controller).

Regardless of the mode and the command enables, interlock commands can be made over the interface inputs "LOCKCOMPLETE, ...".

Again regardless of the mode and the command enables, the commands can be made over the interface inputs (IN_OUT) "INSTART, ..." in the object properties of the steps. These commands are reset after they have been executed; in other words, when the state changes.

You will find more detailed information in the online help in the topic "Commands and Operating States".

Note:

The commands arranged as buttons in the operator control section of the online window (test mode) are effective for the SFC chart or SFC instance and not for sequencers.

Table 3-5: MAN Commands for the SFC

Button	Command	Meaning
▶	Start	Trigger start processing by changing to the "Starting" state.
⏸	Hold	Trigger hold processing by changing to the "Holding" state.
⏯	Resume	Trigger resume processing by changing to the "Resuming" or "Resuming (error)" state.
☒	Abort	Trigger abort processing by changing to the "Aborting" state.
✓	Complete	Trigger complete processing by changing to the "Completing" state.
■	Stop	Trigger stop processing by changing to the "Stopping" state.
↻	Restart	Trigger start processing by changing to the "Starting" state.
↶	Reset	Change to the "Idle" state.
⚡	Error	Trigger error processing by changing to the "Error" or "Error (completing)" state.

3.3.4.1 Priority of the Commands

If several commands are pending at the same time (for example external and internal commands) an interlock error is displayed (LI_ERR=1) and the commands are processed as follows:

- The commands: "Start", "Restart", "Resume", "Reset" and "Complete" are ignored, in other words, not executed.
- Of the commands: "Abort", "Stop", "Error" and "Hold", the command with the highest priority will be executed; in the list, the priority is from left to right in descending order.

Examples:

-	"Start" and "Reset":	No command is executed
-	"Complete" and "Hold":	"Hold" is executed
-	"Abort" and "Hold":	"Abort" is executed

Note: The origin of the commands is not relevant (operator input, LOCKxx, INTxx, BA_CONTROL, ...). Even if only a single command of the pending commands is permitted in the current state, this is nevertheless not executed if it belongs to the group of commands to be ignored.

3.3.5 Executing an SFC

In the following description of the basic principles of cyclic execution of an SFC, it is assumed that the SFC contains several sequencers with freely defined start conditions.

Sequence

The execution of the SFC depends on the input signals applied to the SFC interface and the internal status of the SFC. After downloading an entire chart folder to the automation system, all the SFCs are in the "Idle" state (or "Starting" if autostart is set to 1).

- The mode, commands, and external and internal signals are queried to establish how the SFC will execute, in other words, which state the SFC should adopt based on the input signals.
- All the start conditions of the sequencers are checked to determine which sequencer will execute on the basis of the checks made up to now.

SFC State Unchanged

If the SFC status is unchanged, the sequencer must be executed whose start condition is satisfied and that has the highest priority of all the sequencers with a satisfied start condition. If there are several sequencers with a satisfied start condition and the same priority, then (analogous to the execution of alternative sequences) the sequencer with the tab furthest left in the editor will execute. If the sequencer that was executing is different from the newly selected sequencer, the previous sequencer aborts and the new sequencer starts.

The start condition triggers the start of a sequencer. While the sequencer executes, it is not necessary for the start condition to remain satisfied. The sequencer executes until it is completed or until a sequencer with a higher priority and satisfied start condition needs to execute or until the sequencer is aborted or held due to a status change.

In the "Ready", "Completed", "Aborted", "Stopped", "Held", "Held (error)", "Error" and "Active" states (when SELFCOMP=0), sequencers execute until the state is exited due to a command. As a result, a sequencer is repeated any number of times if the start condition is satisfied and there is no command applied. If you want to avoid this, the last transition of the sequencer can, for example, be configured with an unsatisfied condition. As a result, the sequencer "hangs" at this transition and this state is exited only with a command.

SFC State Changed

If the SFC state has changed, the SFC executes the previous and the new sequencer depending on the state change that occurred, as follows:

If the state change is from "active" to "holding", the previous sequencer is held or aborted (depending on RUNHOLD) and the new sequencer starts.

There is a state change from "Resuming" or "Resuming (error)" to "Active" when the previous sequencer executed to completion. The new sequencer resumes or starts at the change from "resuming" (depending on RUNHOLD) and starts at the change from "resuming (error)".

If there is an implicit state change, the change of sequencer takes place when the first sequencer is completed. If there is no sequencer with a satisfied start condition, the implicit change is made immediately and the new sequencer is started.

With all other state changes resulting from commands, external or internal signals, the previous sequencer aborts if it has not yet completed and the new sequencer starts.

Notes

- When a sequencer aborts, the active steps always execute completely and then the final step executes. The completion of the active steps and the initialization or execution of the final step take place in one cycle.
- In an SFC, a sequencer can execute even if the SFC has not received a start command. This is the situation, for example, when the start condition of one or more sequencers is satisfied in the "Ready" state or when the start condition is not dependent on states.

Coordinating Sequencers and Target Steps

The start behavior of a sequencer to become active in future can be influenced by setting a target step for this sequence in step actions of the current sequencer (SFCName.TARGETSEQ:=2; SFCName.TARGETSTEP:=5). This corresponds to setting a target step manually during testing and commissioning, the difference in this case is, however, that all other target steps including those set manually are reset. This means that the required initial step is specified for the sequencer that will start later.
Please note that target steps cannot be set in simultaneous sequences.

A target step is taken into account when a sequencer starts or resumes and is then deleted.

Since the required flow and the and the target step generally depend on the previous execution of the SFC, the last active sequencer and the last active step can be queried in a transition condition (SFCName.LASTSEQ:=3; <SFCName.LASTSTEP:=2) to be able to set different target steps depending on the result of the query.

Sequencers and steps are identified by their numbers. These are displayed in the properties dialogs and can be used to configure target steps.

3.3.6 Starting an SFC (Chart or Instance)

An SFC starts only when the conditions are satisfied. The **start enable** must be set (ENSTART=1) and the SFC must be in a **state** in which starting is permitted. The following conditions must also be met:

- There must be no **interlock error** (LI_ERR must be set to 0).
- None of the **signals** INTERROR, LOCKERROR, LOCKCOMPLETE, LOCKHOLD, LOCKABORT or LOCKSTOP is active at the same time.
- There is no **operator error** in the MAN mode (OP_ERR must be set to 0).

For an SFC instance, when using control strategies, one of the defined control strategies (CS=<defined control strategy>) must be selected. If CS=0 or CS > CS_HL, it is not possible to start the SFC instance. If no control strategies are configured, the I/Os CS, CS_LL, CS_HL are assigned the value 0 during compilation and must not be modified.

When using setpoints, these must also be within the relevant limits. Otherwise it is once again not possible to start the SFC instance.

For an SFC instance, prior to starting in the AUTO mode, **parameter transfer** (PARAM=1) must be set. The transferred control strategies and setpoints are then checked and the **start disable** reset (QDIS_START=0) if the parameters are permissible. The start disable remains set as long as no parameters have been transferred by setting PARAM. Following a successful start, the start disable is set until the next parameter transfer.

If one of the conditions listed above is not satisfied, the **start disable** (QDIS_START=1) is set and the start is not executed and the "Start" button is inoperable.

Starting in the "Active" State

To be able to start in the "Active" state, the **additional start enable** must be set (ENASTART=1) and self termination must be deactivated (SELFCOMP=0).
In the AUTO mode, **continuous operation** must also be activated (CONT=1).

The start condition of the sequencer to be executed in the "Active" state must also make sure that the sequencer has not already be executed to completion (READY_TC=0). This prevents the sequencer being repeated cyclically when the start condition is satisfied.

An example of a start condition could therefore be:
RUN = TRUE AND READY_TC = FALSE.

When an SFC starts, the ready identifier is reset (READY_TC=0) and is set again (READY_TC=1) on completion of the sequencer executing in the "active" state.

3.3.7 Modifying the Control Strategy and Setpoints for an SFC Instance

Modifications to Control Strategies and/or Setpoints

You can make changes to control strategies and/or setpoints before restarting an SFC instance or change setpoints and adopt them immediately.

How are changes made to the control strategies/setpoints?

To change the control strategy and/or setpoints, there are basically two mechanisms:

1. **Changing the control strategy and/or setpoints and applying them at the next start**

   In **AUTO mode**, follow the steps outlined below:

   Here, you use the PARAM and START I/Os. The CS inputs for the control strategy and <setpoint> for the setpoints are given the new values and the PARAM input is then set to 1. The QPARAM output signals that a change of control strategy and/or setpoints has been requested at the next start; it is reset after the start. The control strategy and/or the setpoints are applied when the start is executed.

   In **MAN mode**, follow the steps outlined below:

   To make a change, the appropriate releases for changing the control strategy ENCSP or setpoints <setpoint>_ENPOP must be set. This enables the corresponding input options in the faceplate for SFC instances.

   The inputs CSP_OP for the control strategy and <setpoint>_OPP for the setpoints are assigned the new values. You can do this in the CFC test mode directly in the SFC instance or in the faceplate of the SFC instance on the "Prepared values" tab. The control strategy and/or the setpoints are applied when the start is executed.

   Assuming the values are valid, they are displayed at the outputs QCSP for die control strategy and <setpoint>_QP for the setpoints in **both modes**. If an error occurs that would violate the limits for the control strategy or setpoints, this is displayed at the outputs <setpoint>_ERROP and bad values are not applied to the outputs QCSP or <setpoint>_QP. Following a start, the current control strategy is displayed at the QCS output and the current setpoints at the <setpoint>_Q outputs (QCS = QCSP, "sw"_Q = "sw"_QP).

2. **Changing setpoints and applying them immediately**

   Using this mechanism, it is not possible to change the control strategy.

   In **AUTO mode**, follow the steps outlined below:

   Here, you use the TAKESP input. The inputs <setpoint> for the setpoints are given the new value and the TAKESP input is set. The setpoints are adopted immediately as long as the TAKESP input is set and no error has been found in the setpoints (high/low limit). The QTAKESP output signals that an immediate change of setpoints was requested.

   The PARAM signal must always be set when setpoint changes are made because the setpoints are checked and adopted in the prepared values only when PARAM is set. Only checked values are adopted at START or TAKESP.

   In the **MAN mode**, follow the steps outlined below:

   To make a change, the appropriate releases for changing the setpoints <setpoint>_ENOP must be set. This enables the corresponding input options in the faceplate for SFC instances.

   The <setpoint>_OP inputs for the setpoints are assigned the new values. You can do this in the CFC test mode directly in the SFC instance or in the faceplate of the SFC instance on the "Current values" tab. The setpoints are applied immediately.

   Note on Working in the Faceplate: Control of setpoints must be enabled on the "Current values“ page. To do this, the property '"Current setpoints" page changes enabled' must be selected in the faceplate picture "@pg_@sfc_type_actualsp.pdl".
   For more detailed information, refer to the online help of "SFC Visualization".

   Assuming the values are valid, they are displayed at the <setpoint>_Q outputs for the setpoints in **both modes**. If an error occurs that would violate the limits for the control strategy or setpoints, this is displayed at the outputs <setpoint>_ERROP and bad values are not applied to the outputs <setpoint>_Q.

Summary

In both mechanisms, the output CSSPACCEPT is used to display whether or not the requested changes were applied. This output is set after applying the control strategy and/or setpoints by the SFC instance at the relevant outputs. Output CSSPACCEPT is set for one processing cycle (at 1.) or remains set until the TAKESP request is reset (at 2.).

Output LI_ERR indicates whether or not an error occurred when attempting to apply value changes with the result that the values could not be completely adopted. If LI_ERR is set, QDIS_START is not reset and a start and adoption of the control strategy and/or setpoints is prevented (at 1.).

3.3.8 Executing a Sequential Control System

3.3.8.1 Phases of a Step

Each step is divided into three phases:

- Initialization is the action for first-time execution
- Execution is the action for cyclic execution
- Termination is the action for last execution

The figure shows the phases in the execution of a step in conjunction with a successor transition: On the left, and elements of the sequencer topology and on the right the corresponding phases in execution.

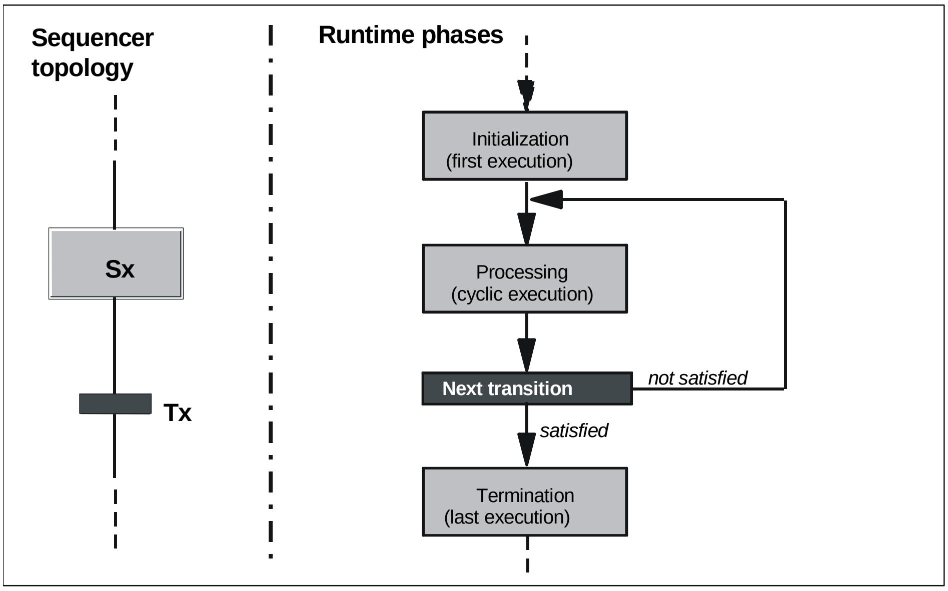

Sequencer topology
Sx
Tx
Runtime phases
Initialization (first execution)
Processing (cyclic execution)
Next transition
not satisfied
satisfied
Termination (last execution)

Fig. 3-3: Phases of a Step in Conjunction with a Successor Transition

3.3.8.2 Executing Step and Transition

The **initial step** is activated when the chart is started without querying conditions and its actions are executed.

A (normal) **step** can have the states “active" and “inactive". A step becomes active after the preceding transition passes control to it. The actions are then triggered and executed. A step becomes inactive after the successor transition is satisfied.

If there is an abort, the termination processing of the previously active step is executed and the initialization of the final step is started (overlapping in the same cycle).

A **transition** has the states “FALSE" and “TRUE". The state of the successor transition of the active step is queried. If the successor transition is true (the condition is true), the previous step is deactivated and the next step activated. If a minimum run time is configured, the transition is queried depending on the step control mode only after a this time has elapsed.

The actions of the **final step** are executed once only.

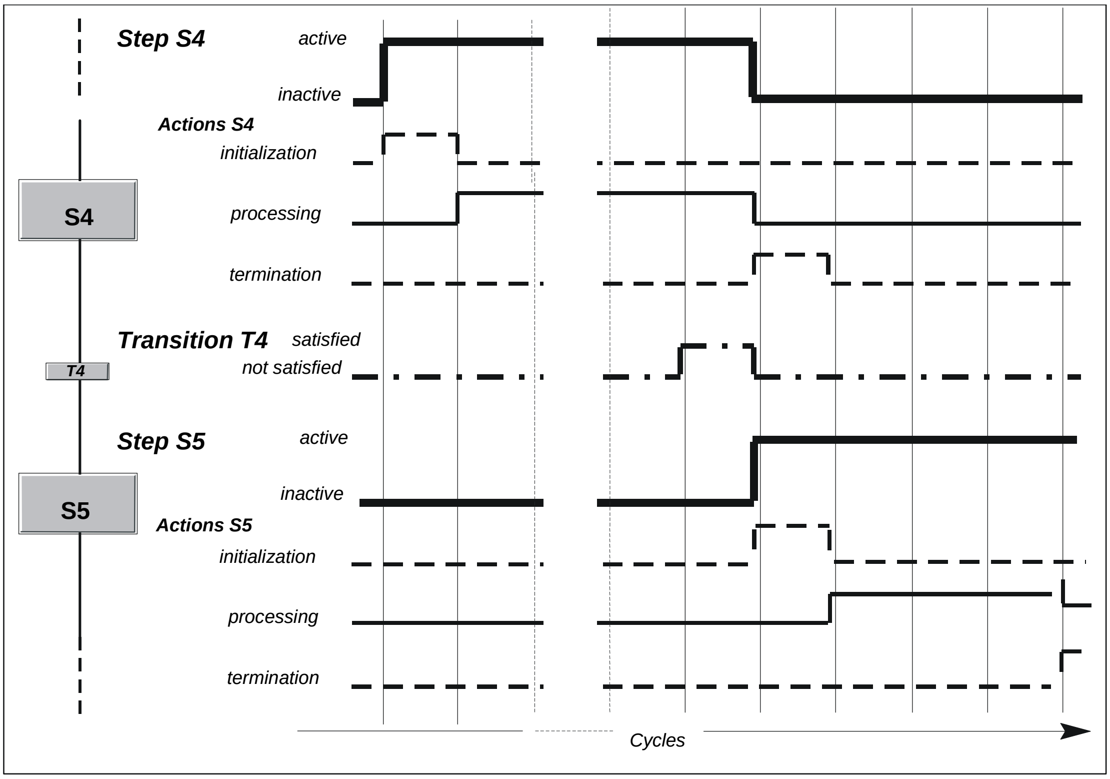


Fig. 3-4: Time Response in Executing Steps and Transitions in a Sequential Control System

When the chart progresses from one step to the next, the predecessor step is terminated in the same cycle as the first action (initialization or execution) of the next step. This allows the **non-latching behavior**" specified in **IEC 1131 - 3** to be achieved.

Example: In step S4, the execution opens a valve and in the termination, this is closed again. If the same valve is opened again in the first action of the next step (S5), the overlapping of the two actions (both in one cycle) means that the valve is not closed.

Special Situations

The example in Fig. 3-4 shows the response when all three actions of a step are configured.

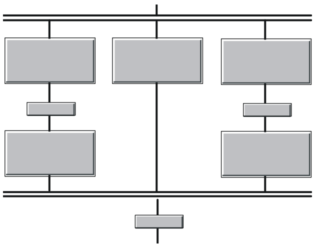
Other combinations are also possible:

- If no "initialization" is configured, the execution begins immediately when the step is activated.
- If no “termination" is configured, the step is deactivated immediately when the transition is true.

The minimum time in which a step is active depends on the number of configured actions (for normal steps 1 to 2 processing phases, with the final step up to 3). If a minimum run time is set for a step, the step remains active for at least this time even if the transition condition is satisfied earlier.

3.3.8.3 Executing a Simultaneous Sequence

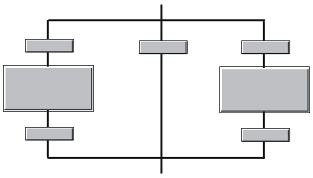
The paths in simultaneous sequences are executed more or less simultaneously in one cycle (in other words, the entire simultaneous sequence). The simultaneous sequence paths are executed independent of each other.



The transition after the simultaneous sequence becomes true when all steps at the end of the sequence paths are active and the conditions are satisfied.

3.3.8.4 Executing an Alternative Sequence

The path of an alternative sequence that is executed is the sequence with the transition whose condition is true first.



If several conditions are true at the same time, the transition furthest left in the chart topology is activated.

Note:
In an alternative sequence, there must not be no transition without parameters at the start of a sequence path.
Reason: Transitions without parameters are always TRUE and are therefore automatically satisfied. This means that they are always true **before** a transition with parameters.

3.3.8.5 Executing a Loop

Fig. 3-5 illustrates the phases in the execution of a loop: On the left, and elements of the chart topology and on the right the corresponding phases in execution.

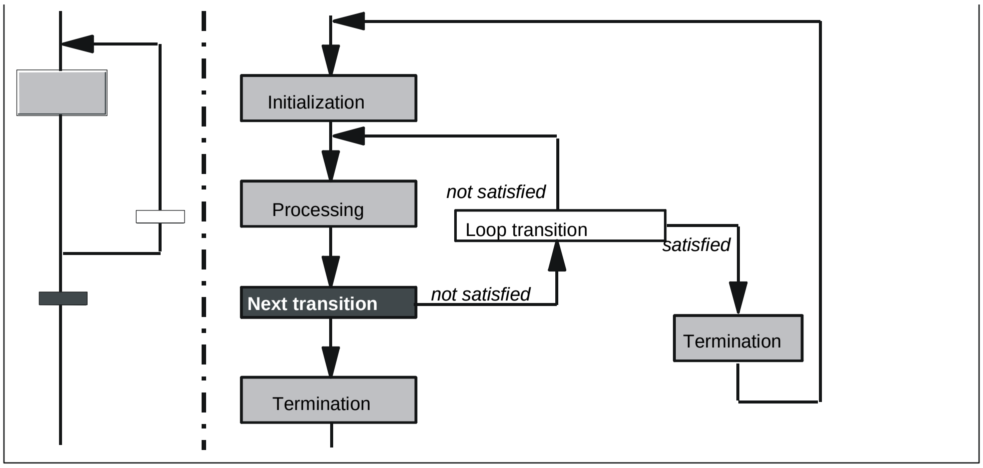


Fig. 3-5: Phases of a Loop

3.3.8.6 Executing a Jump

The jump is executed when the transition of the jump is satisfied.

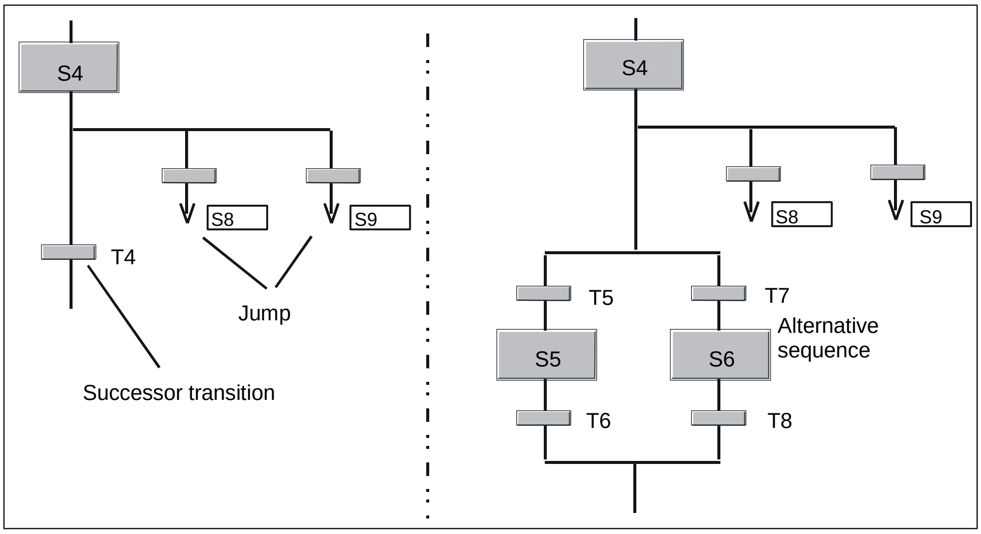


Fig. 3-6: Example: Jumps from a Successor Transition and an Alternative Sequence

If there is more than one jump following the origin of a jump (S4), then (just as in alternative sequences), the jump whose transition is satisfied first is executed. If several transitions are true at the same time, the transition furthest left is activated. If the next element in the sequence is not a successor transition (such as T4 on the left in Fig. 3-6) but an alternative sequence, all the transitions of the alternative paths (T5 and T7) are queried before the transitions of the jumps.

4 Test and Startup

Overview

To support you when you put a new program into operation, the SFC editor provides test functions that allow you to monitor the operation of the sequential control system on the CPU, to control modes, and , if necessary, to change setpoints.

This chapter describes how to monitor and control the sequential control system when testing and commissioning.

4.1 Activating Test Mode

Prerequisites for Testing

The sequential control system (chart or instance) to be tested and the basic control functions (CFC charts) must be loaded on the CPU.

The chart is opened in SFC or the SIMATIC Manager, the SFC instance in the CFC chart.

The Test Modes

Before you switch over to testing, you can select between the "process mode" and the "laboratory mode".

In **process mode**, the communication for online dynamic display of the SFCs is restricted and causes only limited extra load on the CP and bus. If overload occurs in this test mode, a message is displayed to indicate the limit of bus load has been reached. In this case, you should stop the testing of the SFCs that are not absolutely necessary for the test.

The **laboratory mode** allows convenient and efficient testing and commissioning. In the laboratory mode, in contrast to the process mode, communication for online dynamic display of SFCs is unrestricted.

You set the mode for testing with the menu commands “Debug > Process Mode" or “Debug > Laboratory Mode".

Working in test mode is a protected function in S7 that must be logged. Prerequisite: The SIMATIC Logon Service is installed. The protected actions are logged in the **change log** if the current chart folder was activated for the change log (Chart folder > Object properties... > Change log tab).

Switching over to Test Mode

You change from the edit mode to the test mode by clicking the button in the toolbar or by selecting the "Debug > Test Mode" menu command. During the Test mode, you can change to the Edit mode at any time.

The mode change relates to the currently active SFC. This SFC is displayed dynamically in its overview and is cyclically updated (you can set the watching cycle with the “Debug > Test Settings..." menu command.)

Note:
If you change the test settings, you change the watching cycle for **all** charts of this CPU.

After you change to the Test mode, the current state of the chart is displayed. This means that a sequential control system that has already started cannot always be monitored or controlled from the start. This is, for example, the case with charts that are started immediately after they are downloaded to the CPU without any operator command being required (autostart).

Note on H CPUs: If the H CPU is in solo mode, for example after the failure of a CPU and there was a CPU failover, an online access (here: download) results in a dialog being displayed. In this dialog, you can select the required CPU. In redundant mode, this dialog does not appear.

4.2 Display in the Test Mode

The window of the SFC chart or SFC instance in the test mode has an extra operator input and display section at the bottom edge of the window compared with the edit mode. All operator input relates to the chart or the instance (not to the sequencer).

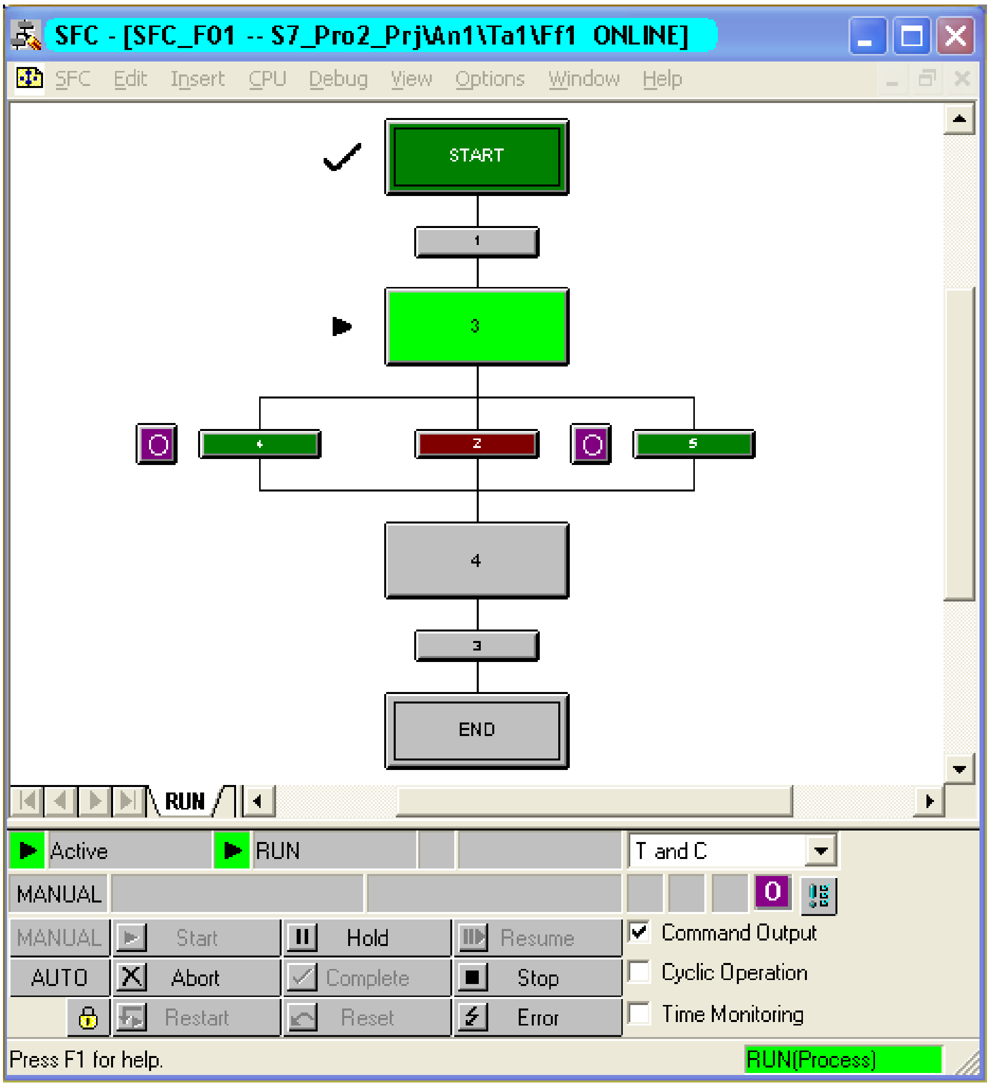


Fig. 4-1 : SFC Window in Test Mode (without Element Bar and Toolbar)

The following is displayed in the operator input and display sections (from left to right):

- in the fields of the first row:
  - Symbol and name of the SFC state
  - State display and name of the active sequencer
  - State display and name of the held sequencer
  - Field with drop-down list box for displaying and modifying the step control modes
- in the fields of the second row:
  - The current mode (MAN / AUTO)
  - State flag "CONT" (continuous operation) for smooth changeover in AUTO mode (for example to avoid an SFC having to be turned off temporarily when it is restarted). The display is made when output QCONT = 1.
  - Status display "READY T.C." (ready to complete), when the SFC is not self terminating (SELFCOMP = 0) and in the active state waiting for the "Complete" command (display, when READY_TC=1).
  - The display H for an interconnection error (or empty field)
  - The display ✋ for an operator error (or empty field)
  - The display S for a step error (or empty field)
  - The display O for an operator prompt: (or empty field)
  - The ✓ button for Confirm All
- The Buttons
  - for selecting the "MANUAL" or "AUTO" mode
  - for enabling the changeover to "AUTO" 🔒.

    After an enable changeover, the icon changes to 🔓.
- The buttons for the commands:

▶ Start	⏸ Hold	⏵ Resume
☒ Abort	✓ Complete	■ Stop
↻ Restart	↶ Reset	ϟ Error

- The check boxes for activating and deactivating the execution options "command output", "cyclic operation", "time monitoring".

Operator Prompt

The operator prompt (not with step control mode “T") is displayed in a box accessible to the operator beside the transition icon.

After clicking on the field (or button) and continuing execution, the operator prompt disappears again.

Displaying the States

In the test mode of the SFC, both the SFC and the sequencers with their start conditions are made dynamic. The current states of the SFC and sequencer are displayed in the operator input and display section of the SFC (see above).

The various states of the sequential control system, the steps, the transitions are visualized by different colors and icons.

In addition to the color frame of the steps, a status indicator also appears. The icon of the status display is therefore an additional indicator of the current states in the chart in case the colors cannot be clearly distinguished. The colors of the state displays cannot be changed.

Table 4-1: Appearance of the Status Indicator

Step state	**Step color**	**Symbol**
inactive (not executed)	gray	
inactive (executed)	dark green	✓
active	light green	▶
held	yellow	⏸
Error	red	↯
Transition State	**Transition Color**	
Inactive	gray	
Satisfied	dark green	
Not satisfied	dark red	

Display of the CPU mode

The operating mode of the CPU is displayed in the status bar (right information box) in color (green = RUN, red = STOP). The test mode is also displayed: (Laboratory) or (Process).

4.3 Operator Control and Monitoring of the Sequential Control System

Setting the Test Environment

With the "Debug > Test Settings... menu command, you can open a dialog box in which you can change the monitoring cycle for the current program (default value: 2 s). You can set cycles from 1 to 9 and in steps of ten from 10 to 90 seconds.

The same cycle time as for CFC is stored in the chart folder. This means that the same monitoring cycle is valid for both applications (SFC and CFC).

Operator Control and Monitoring

You operate and monitor the SFC in the overview display. Here, you can change the statuses, modes, step control modes, and the chart options as required.

If you have set the "Debug > Trace" menu command (check mark), you change automatically to the active sequencer. Otherwise (the menu command is not set), the sequencer that you set manually using the tab at the lower edge of the window is displayed.

If you double-click on a step or a transition, a dialog box appears resembling the object properties dialog in the Edit mode. You can also display the object properties for step and transition at the same time by selecting the required transition and double-clicking on a step to open both dialog boxes (or in the opposite order: by selecting a step and double-clicking on a transition). To be able to see both dialog boxes at the same time, the step and transition do not need to belong together.

A selected element is indicated by a blue background.

Confirming in the Sequencer and in the Object Properties Dialog

If a button is displayed for operator intervention or to confirm an error for the monitored step or transition in the sequence path, the dialog box with the object properties also has the corresponding button or buttons added to it.

Note: After a step runtime error has been acknowledged, the appearance of the step returns to the state before the error occurred (for example active = “light green").

Changes during a Test Run (SFC chart only, not SFC instance)

The step attributes (confirmation, minimum/maximum run time, constant in assignments) and transition attributes (constant in conditions) that can be modified in the Test mode are adopted on the CPU and in the ES data management when the change is made and do not require recompilation and download.

Note: If you want to modify the existing SFC instances, you must exit the test mode and then modify the corresponding SFC type. After compiling the changes and downloading them, all the instances are automatically adapted.

Trace

In Test mode, you can use the **Debug > Trace** menu command so that the active sequencer is displayed. If this menu command is not set, the explicitly selected sequencer is displayed.

4.3.1 The Object Properties of a Sequencer during Testing

Tabs of the Object Properties Dialog

The Object Properties dialog of a sequencer is divided into four parts. These are as follows:

- **General**
  The "Name" box in the "General" dialog is displayed in a frame, the frame color corresponds to the state of the transition and is updated continuously (colors: see "Default Colors" table in the online help).

  The "Comment" field displays the configured comment for this sequencer.

  In the "Priority" field, you can see the priority of the sequencer. The priority decides which sequencer of a chart is started when the start conditions of several sequencers are met simultaneously.

- **Start condition**
  Each line represents a condition. Of the maximum 16 conditions, 2 x 5 are visible on the first page and 2 x 3 on the second page. You can move to the second page by clicking the "Arrow" button on the last operator.

  To the left of the first address and to the right of the second address, there is a box with the current value of the address. The values can only be changed for an SFC chart.

- **Preprocessing**
  In this tab you can view the actions for preprocessing of the current sequencer. The values can only be changed for an SFC chart.

  Each line represents a statement. Up to 50 statements are possible. The section of statements displayed can be moved with the scroll bar at the right-hand edge of the window.

  Each statement consists of a left address, an operator, and a right address.

  To the left of the first. address and to the right of the second address, there is a box with the current value of the address. If you click one of the boxes, the “Change Value" dialog box is displayed in which you can enter a new value for the address. After you close the dialog box, the changed value is written to the CPU and becomes effective in the next processing cycle. A constant in the right address field is also written to the ES data management.

- **Postprocessing**
  In this tab you can view the actions for postprocessing of the current sequencer. The values can only be changed for an SFC chart. The structure of the tab is identical to the "Preprocessing" tab (see relevant section).

4.3.2 The Object Properties of a Step during Testing

The Properties dialog is divided into four parts. These are as follows: "General", "Initialization", "Processing" and "Termination".

- **"General" tab**

  The "Name" box in the "General" tab is displayed in a frame, the frame color corresponds to the state of the step and is updated continuously (colors: see "Default Colors" table in the online help).

  You can activate or deactivate the “Confirmation" option which sets or resets a flag that is evaluated in the “T / T and C" step control mode (step-specific confirmation by user). The sequential control system operates as follows:

  - **process-controlled** for steps without the “Confirmation" option.
    Each satisfied successor transition of a step without the “Confirmation" option passes control without operator intervention (corresponds to "T").
  - **operator-controlled** for steps with the “Confirmation" option.
    If the successor transition of an active step with the "Confirmation" option is satisfied, an operator prompt is set and control passes to the next step or steps after the prompt has been confirmed (corresponds to "T and C").

  By setting the option, "**Target step**", the current step is selected as the target step. This means that

  - The stopped sequencer starts at the selected target step instead of at the initial step with the next "Start" command.
  - the held sequencer resumes at the target step after correct processing of the interrupted steps when the “Continue" command is set.

  The target step marker is valid only for the next "Start" or "Resume" command. When you restart the CPU, the target step marker is deleted.

  You can also select several steps as the target steps. The user is responsible for selecting the target steps so that the processing is feasible; in other words, that no blockages or endless loops result. See also Section 3.3.5, Editing Sequencers.

  If you use "programmed target steps", the target steps sent by the operator in the corresponding sequencers are deleted.

  You can change the values of “**Run times minimum**" and “**Run times maximum**". By clicking in the input box, you open a dialog in which you can change the current setting. Any changes you make are entered in the ES data management when you confirm with "OK" and take effect in the next processing cycle.

  With the boxes “**Run times - Current**", “**Time to minimum**" and “**Time to maximum**" you can monitor the run time.
  If no values were configured for the run times (time = 0), then “- - -" is displayed in the boxes.

- **"Initialization/Processing/Termination" Tabs**

  To the left of the first address there is a box displaying the current value of the address. The box to the right of the second address contains the configured value that you can change (only in the SFC chart). If you click the box, you open the “Change Value" dialog box, in which you can enter the new value.

  After you close the dialog box, the changed value is written to the ES data management (and to the CPU) and becomes effective in the next processing cycle.

If the monitoring of the step detects a timeout and if the button for step error S is displayed beside the step, the confirmation button S is added to the other buttons. This allows you to acknowledge the error from the dialog box.

After a step runtime error has been acknowledged, the appearance of the step returns to the state before the error occurred (for example active = “light green").

Using the “Go To" button, you can jump from the current field of an operand to its point of use (block in the CFC chart, SFC chart, I/O address in HW Config.).

4.3.3 The Object Properties of a Transition during Testing

The Properties dialog of a transition is divided into four parts. These are as follows:

- **General**
  The "Name" box in the "General" dialog is displayed in a frame, the frame color corresponds to the state of the transition and is updated continuously (colors: see "Default Colors" table in the online help).
- **Current condition ("Current Cond.")**
  shows the current state of the conditions.
- **Last condition ("Current Cond.")**
  shows the state of the condition of the previous processing cycle.
- **The condition after error ("Cond. after Error")**
  shows the state of the condition that led to an error.

Changing Addresses

To the left of the first address and to the right of the second address, there is a box with the current value of the address. In the SFC chart, you can change the content of both fields (not possible with an SFC instance). If you click one of the boxes, the “Change Value" dialog box is displayed in which you can enter a new value for the address.

After you close the dialog box, the changed value is written to the ES data management (and to the CPU) and becomes effective in the next processing cycle.

If there is an operator prompt for the monitored transition in the sequencer and the button displayed beside the transition ○, the dialog box with the object properties also has this button 0 added to it. This allows you to acknowledge the operator prompt from the dialog box.

Using the “Go To" button, you can jump from the current field of an operand to its point of use (block in the CFC chart, SFC chart, I/O address in HW Config.).

The results of the logic operations of the conditions using Boolean operators are visualized as colored links of varying thickness. A broad, green line means “satisfied", a thin, red line means “not satisfied" and a thin, black line means “inactive".

Update

The content is updated continuously in the “Current Cond." tab. In the “Current Cond. -1" and “Cond. after Error" the “Update" button is also displayed. The state of the transition is entered here as it was at the time when the object properties were opened. Using the “Update" function, you can display the current state in a permanently open dialog box.

5 Documentation

Overview

This chapter describes how you create the documentation for your SFC charts/types/instances. This includes printing the SFC in various views and printing the parameter assignment, the properties, and chart reference data.

5.1 Printing an SFC

You can print the current chart or type by clicking the button in the toolbar 🖨 or using the "**SFC > Print...**" menu command. In the dialog box that opens, you can make settings relating to the scope of the printout (Print What) and the layout (Options).

- Print What: properties, I/Os, external view (with SFC chart) or alternatively characteristics (with SFC type/instance). For the sequencers: Properties, Normal size, Overview, Steps/Transitions. You can select either normal size or overview, but not both.
- Options (only with "Normal size"): Alternative sequence aligned left, Comment/text.

Note:
The selected range is always printed for the entire SFC. If you only require a printout of a step or a transition, use the print functions of the in the “Object Properties" dialog of the selected element.

5.2 Defining Footers

With the **SFC > Footers...** menu command, you can open a dialog box in which you enter the text printed in the footers of every printed page.

With the DOCPRO optional package, you can print the SFC chart/type along with footer data. In the footer data, a distinction is made between global data and local data specific to an SFC.

You can enter the global data for the project using DOCPRO or the SIMATIC Manager; the local specific data must be entered with the SFC editor. Remember that the data specific to an SFC overwrites the entries of the global data for the particular SFC.

You can also enter specific data even when you have not installed the DOCPRO optional package. This data is saved and can be printed later when DOCPRO is available for print jobs.

You can enter the **SFC-specific** footer data in the active fields of the tabs "Part 1" to "Part 4" and "Free fields". These include document type, date created, document number, dates of changes, free texts etc.

In the **global footers**, you can enter keywords that will be replaced by the actual texts when you print out. The keywords that can be used and their meaning (origin) are explained in the online help of SFC.

5.3 Chart Reference Data

With the "Options > Chart Reference Data..." menu command or by clicking , you start an application that creates comprehensive documentation listing, for example, the run sequence, the cross-references of the addresses etc.

With a search function, you can search for specified addresses, symbols, I/Os etc.

This helps you to check your configuration structure. With the “Cross-References Chart Element -> Address" list, you can, for example, check which addresses are used and how often and whether the write access is synchronized.

You do not need to close the window with the chart reference data if you want to continue working with the SFC editor. This means that you can view the lists while working with the SFC editor.

For a detailed description of the “Chart Reference Data", refer to the CFC online help.

5.4 Logs

The Options > Logs... menu command displays a dialog box with several tabs. The dialog displays only the tabs for the functions that were used already in the current chart folder. The tabs relevant to SFC charts are listed below:

- **Compile**

  List of messages occurring during compilation (including compiler messages).

- **Check Consistency**

  List of the messages that occurred during the consistency check.

- **Download**

  List of messages occurring during a download, for example, when the download was free of errors: "0 error(s) and 0 warning(s) found".

- Change log

  All protected actions are logged here (download, test mode). This is only possible when the SIMATIC Logon Service is installed.

- **Step Processing**

  After checking the SFC runtime behavior with the menu command "Options > Check Step Processing", this tab lists the steps that use the same address in the terminating action of one step and in the initializing action or processing action of the next step. The log shows how many SFC charts were checked and how many accesses were found and the SFC charts with an unchanged runtime behavior.

- **Convert Format**

  After converting charts of older versions to V5.x or V6.x, the charts that no longer have the same properties are listed here. These include, for example, the SFC charts that had the attributes “scan rate" and “phase offset" in the old version. The information includes the installation location (task) and the values for the scan rate and phase offset for every chart affected.
  Example: "SFC1: Task OB32   scan rate 4   phase offset 2"

- **Make Textual Interconnections**

  With the "Options > Close Textual Interconnections" menu command or the option with the same name in the Compile dialog, all open textual interconnections are closed if they have a concrete interconnection partner in the current chart folder. The textual interconnections closed with this action are displayed along with error messages indicating the textual interconnections that could not be closed.

The “Go To" button is activated when an object occurs in a selected message that can be displayed. If you click the button, the relevant chart is opened and the object selected and displayed centrally.

A Technical Specifications

A.1 Technical Specifications

Hardware Requirements

- PG or PC with:
  - Pentium Processor
  - RAM 256 Mbytes (or more)
  - Hard disk 500 Mbytes (less RAM)
  - Graphics card VGA 640 x 480
    (recommended: SVGA 1024 x 768 or higher)
- SIMATIC S7-400

Software Requirements

- Microsoft Windows NT (SP 5) / Windows 2000 (SP 3)
- STEP 7
- SCL Compiler
- CFC

An **authorization** for STEP 7, SCL, CFC and SFC must be installed.

General Numeric Data

Object	**Number**
Steps per chart	2 - 255
Transitions per chart	1 - 255
Statements per step and action	≤ 50
Conditions per transition	≤ 16
Sequencers per SFC chart	≤ 8
Sequencers per SFC type	≤ 32
Steps per sequencer	2 - 255
Transitions per sequencer	1 - 255

B Abbreviations

C	Step control mode: confirmation by operator
C / C++	High-level language for programming computers
CFC	Continuous Function Chart
CPU	Central processing unit
DB	Data block
ES	Engineering system
FB	Function block (function block with memory)
FC	Function (function block without memory)
HID	Higher level designation
IEA	Import-Export Assistant
OB	Organization block
OCM	Operator control and monitoring (WinCC)
OS	Operator station
PC	Personal computer
PCS 7	Process control system (SIMATIC)

PG	Programming device
PH	Plant hierarchy
PLC	Programmable controller
PLC	Programmable logic controller
SFB	System function block
SFC	Sequential Function Chart
SFV	SFC Visualization
STEP 7	Software development environment for SIMATIC S7 / M7
T	Step control mode: transition only
T / T and C	Step control mode: step-specific confirmation by operator
T and C	Step control mode: transition and confirmation by operator
T or C	Step control mode: transition or confirmation by operator

Glossary

A

Access

SFC elements or block I/Os can access addresses, charts, block I/Os, or run-time groups. A distinction is made between read and write access.

Action

Actions allow the activation or deactivation of run-time group and SFC charts, as well as modifications to blocks and shared resources and their input parameters. The actions are formulated in SFC in the Properties dialog.

Address

An address is the identifier of the location of data or an area containing data, examples: input I12.1; memory word MW25; data block DB3. Addresses can be specified in absolute or symbolic form.

An address is part of a STEP 7 instruction and tells the processor what it should perform an operation on.

In SFC, the address is part of a statement (step) or condition (transition).

Addressing, absolute

If data is addressed in absolute form, the absolute ↑address is used to access the value with which the operation will be performed. Example: The address Q4.0 identifies bit 0 in byte 4 of the process output image (PIQ).

Addressing, symbolic

If data is addressed using symbols, the ↑ address to be processed is specified as a symbol (not as an absolute address). Symbols are assigned to addresses in the symbol table.

Alternative sequence

A structural element of an SFC, that consists of at least two ↑sequence paths. Only the path whose transition condition is satisfied first is processed by the PLC.

B

Basic control

Basic control is the configuration of the programmable controller (PLC). The configuration tools available are the CFC and SFC editors.

Block

Blocks are separate parts of a user program that are distinguished by their function, their structure, or purpose.

CFC works with ready-made block types that are placed (inserted) in a CFC chart. When a block type is inserted in the chart, a block instance is created. These block instances and their graphic representation are blocks in the sense of CFC.

Block category

The block category identifies different forms of blocks. Block categories include, for example, data blocks (DBs), function blocks (FBs), and functions (FCs).

Block icon

Graphic representation of the most important information of a technological, block that can be controlled and monitored on the PLC. The block icon is typically placed in an OS overview picture. The corresponding faceplate can be called up using the block icon.

Block input

Block I/O that can be interconnected with block outputs and ↑ addresses of the same data type or can have parameters assigned to it.

Block instance

A block instance is the implementation of a block type. A block type inserted in a CFC chart becomes an instance. After it is inserted, the block instance has run-time properties and name that is unique within the chart.

Block I/O

Block input or block output

Block output

Block I/O that can be interconnected with block inputs and ↑ addresses of the same data type.

C

Catalog

Window in CFC, that can be switched between the catalogs of **blocks**, **charts**, **libraries**, and **unplaced blocks** (if unplaced blocks exist). You can insert blocks, charts and text boxes from the catalog into a chart.

CFC

Continuous Function Chart.

1. Function chart (CFC chart) with the graphic interconnection of technological functions (blocks).
2. An editor for plant/oriented, graphic configuration of automation tasks. Using CFC, entire software structures are created (CFC charts) from ready-made blocks.

Chart

Depending on the context, a chart in the ES is

- a CFC chart consisting of 1 to 26 chart partitions each with 6 sheets and possibly also overflow pages.
- a nested chart that results when one chart is inserted in another (chart-in-chart).
- an SFC chart consisting of between 1 and a maximum of 8 tabs. Each tab contains a sequencer.

Chart element

Chart elements of the SFC chart are the basic elements (step, transition, text) and structure elements (sequence, simultaneous sequence, alternative sequence, loop and jump).

Chart folder

Folder in the project structure containing charts of a user program.

Chart reference data

Chart reference data is data available to the user in addition to the graphic chart display in the form of lists, for example the list of accesses to shared addresses.

Component view

Device-oriented view in the SIMATIC Manager. The project is displayed with its components (station, module, program ...);
alternative to the ↑ plant view or ↑ process object view.

Consistency check

Checks the consistency of block types, shared addresses etc. of the chart folder.

Condition

Conditions in a transition allow you to do the following:

- to read values of block I/Os or shared addresses
- To logically combine the read values with a constant or another read value using Boolean operators (=, >, <, ...)

The result of a condition is a Boolean variable that can be logically combined with the results of other conditions.

Cycle time

The cycle time is the time required by the CPU to execute the user program once.

D

Data block (DB)

Data blocks are data areas in the user program containing user data. There are shared data blocks that can be accessed by all logic blocks and instance data blocks that are assigned to a particular FB call. In contrast to all other blocks, they contain no instructions.

Data type

A data type specifies how the value of a variable or constant is used at a block I/O. "BOOL", for example, defines a binary variable; "INT" defines a 16-bit integer variable.

Drag-and-drop

With the drag-and-drop function, you can move, copy and insert objects using the mouse.

Using drag and drop:

1. Select the objects by clicking them or by drawing a lasso round them.
2. Position the mouse pointer on an object, press the left mouse button and hold it pressed.
3. If you want to copy, hold down the CTRL key as well (the mouse pointer has the "+" symbol added to it).
4. Move the mouse pointer to the required position and release the button - the objects are inserted.

Dynamic display

In the dynamic display, input and output values of a block in a CFC chart or address values in an SFC chart are updated from the CPU in the Test mode.

E

Edit mode

(Alternative to the ↑ test mode)
In **CFC** blocks can be inserted, copied, moved, deleted, renamed, have parameter values assigned, or be interconnected.
In **SFC** the sequential control system is created in this mode. Chart elements can be inserted, copied, moved, deleted, renamed, and assigned parameters.

ES

Abbreviation for "Engineering System". This is a user-friendly, visual configuration system in which the process control system can be created and adapted to different situations.

External view

The external view is the graphic representation of a SFC chart as block (with interface) that allows external interconnection of the I/Os. The external view contains the standard interface that is derived from the SFC run-time system.

The external view of the SFC can be interconnected with any objects (CFC blocks, nested charts, run-time groups, textual references, shared addresses). This means that the SFC chart can also be controlled directly by CFC interconnections.

The external view is displayed in a separate window in the CFC. Other objects cannot be placed in this window. Interconnections to other CFC objects are implemented solely via the sheet bar.

F

Forced manual switchover

Switchover by the operator from the "AUTO" mode to the "MANUAL" mode without a prior enable.

H

Hierarchy folder

The hierarchy folder is used to structure the plant in a hierarchy. It can contain further hierarchy folders and objects such as CFC / SFC charts, process pictures, reports, additional documents (Excel, Word, ...). The HID of an object is obtained from the name of the hierarchy folder (path) and the object name.

Higher level designation (HID)

The HID is used to identify parts of a plant according to their functions. This is made up of the hierarchical path of the plant hierarchy.

I

Installation and startup

The CFC/SFC editors provide test functions to allow you to monitor, modify, and change parameter settings on the PLC during installation and startup.

Installing

Procedure with which a block/SFC chart is placed in a ↑ run-time group so that it is registered with an ↑ organization block (OB).

Instance

When you insert an SFC type or a block type in a CFC chart, an instance is created. Instance in this sense means that it is an instance or usage of the selected SFC type or block type.

Interconnection

In CFC, the connection of an I/O to another element. The value of the interconnected input is fetched from the other end of the interconnection during run time. Read or write access from an SFC chart to a block I/O in a CFC chart is also known as an interconnection.

I/O

The input or output of a block or chart. I/Os of the same data type can be interconnected or connected to shared addresses. An I/O is parameter, that accepts data for further processing (input) and transfers the resulting data (output).

J

Jump

A jump is a structure element of SFC, with which the execution of an SFC can be continued at a different step in the same SFC depending on a transition condition.

L

Laboratory mode

One of the test modes.
The laboratory mode allows convenient and efficient testing and commissioning. In the laboratory mode communication for online dynamic display of SFCs is unrestricted.

As an alternative to the laboratory mode, you can also select the ↑ process mode (restricted test mode). It is not possible to switch over between the process and laboratory modes when testing is active (only in the edit mode).

Library

A folder for objects that can be used again and again. A library is not project-related. Blocks are made available in block libraries sorted according to certain criteria (block families, alphabetical order etc.). Different block libraries are used depending on the PLC or situation.

Loop

Structure element in SFC consisting of a ↑ sequence and a return path; the sequence encloses and contains exactly one transition.

M

Master project

A master project is a project that has had parts of it (for example, charts) distributed into other projects (branch projects) to allow more efficient engineering. After they have been worked on, the parts are merged back into the original project = master project.

Messages, configuring

Creating messages with their attributes and texts. Messages can be configured from within SFC.

Mode

1. CPU:
Using the mode selector of the CPU, you can set the following operating modes:

- RUN with access to the STEP 7 user program,
  for example, with programming device ("RUN-P"),
- RUN with access protection ("RUN"),
- STOP and
- Memory Reset ("MRES").

2. SFC:
The mode decides how the execution of an SFC is controlled.

- AUTO (process mode): Execution is controlled automatically, for example, via the external view of an SFC chart..
- MANUAL (operator mode): Execution of the chart is controlled manually by an operator, for example during commissioning or with SFV (SFC Visualization in WinCC).

O

Online/offline

In the SIMATIC Manager, objects of the programmable controller are displayed in the online view and the objects of the ES in the offline view. Online, there is a data connection between the PLC and the programming device/PG, offline there is no connection.

Operating mode

1. The programmable controllers of SIMATIC S7 can adopt the following operating modes: STOP, STARTUP, RUN and HOLD.
2. SFC has the following operating states Ready, Starting, Active, Completing, Error (completing), Completed, Holding, Held, Continuing, Error, Held (error), Resuming (error), Aborting, Aborted, Stopping, Stopped.

Operating system

General term for all the functions that control and monitor the execution of the user programs, the distribution of resources to the individual user programs, and the maintenance of the operating mode in conjunction with the hardware (for example standard operating system MS-WINDOWS).

Operating state logic (OSL)

The operating state logic of SFC describes

- the states that can be adopted by an SFC chart or an SFC instance.
- the state changes possible in a particular state.
- the events that bring about a state change.

In addition to the SFC-OSL, there is also a separate simpler operating state logic for the sequencers, the sequencer OSL. This describes the states that can occur in a sequencer.

Organization block (OB)

Organization blocks form the interface between the operating system of the CPU and the user program. The order of execution of the user program is specified in organization blocks.

An organization block corresponds to a ↑ task.

OS

Operator Station. A station for controlling and monitoring the process. In PCS 7, the WinCC software system is used for the OS with which all the process monitoring and control functions can be implemented.

P

Phase offset

The phase offset shifts the point of activation of the run-time group within a task by a defined time compared with the basic cycle. Phase offset allows a uniform distribution of load within the CPU. See also ↑ scan rate.

Plant Hierarchy (PH)

A hierarchy structured according to technological aspects.

Plant view

View in the SIMATIC Manager based on technological aspects (plant (process cell), unit, function ...); alternative to the ↑ component view and ↑ process object view.

Postprocessing

A cyclic action can also be configured for each sequencer. The cyclic action consists of a part that is executed before the cyclic sequencer processing, the **preprocessing** and a part that is executed after the cyclic sequencer processing, the **postprocessing**.

Preprocessing

See ↑ Postprocessing.

Process

Sequence of chemical, physical, or biological processes for extraction, production or elimination of substances or products.

Process image

Reserved areas in the RAM of the CPU. Signal states of the input and output modules are entered here.

Process mode

One of the test modes. In the process mode, the communication for online dynamic display of the SFC charts is restricted and causes only limited extra load on the CP and bus.

As an alternative to the process mode, you can also select the ↑ laboratory mode (unrestricted test mode).

Processing phase

A step is divided into three processing phases: initialization, (cyclic) processing and termination. Each processing phase corresponds to an action with statements.

Process object view

View in the SIMATIC Manager. With the process object view, all the data of the basic control of an entire project can be displayed in process control-oriented view.

Process tag

A neutral object in terms of resources. It is used to connect the PLC configuration world (STEP 7, CFC ...) with the OS configuration world (WinCC). It contains information about its location during run time (for example the network address and the memory area on the PLC) as well as information on specific OS-relevant properties.

Programmable (Logic) controller (PLC)

A programmable controller a SIMATIC S7 PLC, a complete device (PLC with integrated operator panel) in SIMATIC C7 or an automation computer in SIMATIC M7. It is used for closed and open-loop control of processes in process engineering and manufacturing.

PLC is used as a general term for the unit containing the CPUs on which the user program runs. In this case, it includes S7 programmable controllers and M7 automation computers.

Project

A folder for all the automation solutions regardless of the number of stations or modules and how they are networked.

R

Recipe

A recipe is the procedure for manufacturing a product according to a specific procedure.

Resources

Resources are pools of objects (FBs, FCs, DBs, OBs, bit memory, counters, timers etc.) that can be accessed when configuring and setting parameters for a CFC/SFC chart.

Run-time attribute

Run-time groups have run-time attributes that control their activation. All the blocks and/or SFC charts in a run-time group inherit these attributes from it.

In S7 the tasks are implemented as ↑ organization blocks (OBs).

Run-time group

Run-time groups are used to structure tasks. The blocks and/or SFC charts are installed in the run-time groups. Run-time groups have run-time attributes and can be activated and deactivated separately by interconnecting them with a block output or by a statement. If a run-time group is deactivated, none of the objects it contains will be activated any longer.

Run-time properties

The run-time properties of block or SFC chart determine how the block/chart is included in the execution of the entire structure on the PLC. These properties are vital to the performance of the PLC in terms of reaction times, dead times, or the stability of time-dependent structures, for example control loops.

Run sequence

This is a program sequence with which the blocks and/or SFC charts can be installed in the order in which they are executed by the CPU. The run sequence is displayed as a separate window in CFC and contains the OBs of the relevant PLC.

S

S7 program

A folder for the ↑ symbol table, the blocks, the source files, and the charts for the programmable S7 modules.

Scan rate

The scan rate is a ↑ run-time attribute. It specifies whether a ↑ run-time group is executed every time a task is executed or only every nth time. See also ↑ phase offset.

SCL

A high-level language similar to Pascal complying with IEC 1131-3 for programming complex solutions on a PLC, for example algorithms and data processing jobs.

Sequence

Structure element in SFC containing a sequence of ↑ steps and ↑ transitions.

Sequence path

In SFC, a sequence path is a series of elements (in the topology seen as a vertical path). A simultaneous sequence or an alternative sequence consists of at least two sequence paths arranged side-by-side and containing at least one element.

In SFC, an entire ↑ sequencer can also be understood as a sequence path.

Sequencer

A sequencer is a functionally self-contained unit within a sequential control system. It consists of a series of steps and transitions and begins with an initial step and ends with a final step. Within the SFC, a maximum of 8 (chart) or 32 (type) sequencers can be created and selected using tabs.

Sequencer topology

In the sequencer topology of the SFC, the elements are displayed according to fixed syntax rules (for example, order, spacing, extent and alignment of the elements). When the topology is created, the syntax rules are adhered to automatically by the editor.

Sequential control system

A sequential control system switches from one to the next step dependent on conditions. Sequential control systems are implemented in PCS 7 with SFC charts (with up to 8 sequencers) or SFC types (with up to 32 sequencers).

Shared address

Shared addresses are I/O signals (input/output bits, bytes, words, double words; peripheral input/output bits, bytes, words, double words), memory bits, timers, counters, data block cells, functions, or function blocks. They can be accessed in absolute and symbolic form by SFC and CFC.

Symbolic access uses names (the symbol) of the shared address in the symbol table. For absolute access, the symbol table is not required. The absolute form of the shared address is entered. Symbolic addressing has the advantage that the connection is retained if the address is changed and the symbol stays the same.

When accessing addresses, make sure that the shared address has the same data type as the block I/O in the CFC chart or statement / condition in the SFC chart.

SFC

An SFC chart is a ↑ sequential control system that runs as a separate control system within the programmable controller.

SFC instance

See ↑ Instance.

SFC type

See ↑ Type.

SIMATIC Manager

Graphic user interface for SIMATIC users under Windows 95/98/NT/2000/XP. The SIMATIC Manager is used, for example, to create projects and access libraries.

Start condition

The start condition triggers the start of a sequencer. While the sequencer executes, it is not necessary for the start condition to remain satisfied. The sequencer executes until it is completed or until a sequencer with a higher priority and satisfied start condition needs to execute or until the sequencer is aborted or held due to a status change.

Statement

Within the action of a step, a statement can be used for the following:

- to write to CFC block I/Os
- to assign shared addresses
- to activate and deactivate SFC charts
- to activate and deactivate run-time groups

A statement always consists of a left (first) address, an operator and a right (second) address.

Step

The step is an element of a ↑ sequential control system and is the control instance for the execution of the associated ↑ actions. A step is divided into three processing phases: initialization, (cyclic) processing and termination.

Step control mode

An SFC can be executed in various step control modes. These affect the way in which the execution of the steps is enabled. You can select the following modes:

- C = confirmation by operator The sequential control system runs controlled by the operator.
- T = transition. The sequential control system runs controlled by the process.
- T or C = transition or confirmation by operator The sequential control system runs controlled by the process or by the operator.
- T and C = transition and confirmation by operator The sequential control system runs controlled by the process and by the operator.
- T / T and C = step-specific confirmation by operator. The sequential control system runs process-controlled in steps without the "confirmation" flag (as in "T") and must be confirmed by the operator at steps with the flag (as in "T and C").

Step types

An SFC has different types of steps: initial step, normal step, final step.

Structure

A structure is a structured ↑ data type made up of various elements. An element can be an elementary or a structured data type.

Structure element

Structure elements in SFC consist of an arrangement of basic elements. Structure elements are: "Sequence", "Simultaneous sequence", "Alternative sequence", "Loop" and "Jump".

Symbol

A symbol is a name defined by the user according to certain syntactical rules. After its purpose has been defined (for example to represent a variable, data type, jump label, block), it can be used in programming and in operator monitoring and control.

Example: address : I5.0, data type: BOOL, symbol: Emer stop.

Symbol table

Table for assigning symbols (= name) to addresses for shared data and blocks.

Examples:	**Symbol**	**Address**
	Emerstop	I1.7,
	Ctrller	FB24

T

Target step

In the test mode (or in SFC visualization on the OS), a step can be selected as the target step. This means that the deactivated ↑ sequencer begins again at the selected step instead of the initial step at the next "Start" command and a halted sequencer resumes at the target step after completing execution of the interrupted step(s) when the "resume" command is received.

Task

Tasks form the interface between the operating system of the CPU and the user program. The order of execution of the user program is specified in tasks. A task corresponds to an organization block in S7.

Test mode

(Alternative to the ↑ Edit mode) Mode of CFC/SFC for testing and optimizing user program running online on the CPU. Values of block I/Os and the execution of SFC charts on the CPU can be monitored and modified.
For testing, you can select either the ↑ process mode or the ↑ laboratory mode.

Top chart

A CFC chart that is not nested in another chart and can be displayed in the SIMATIC Manager (↑ nested charts).

Transition

A transition is a basic element of SFC and contains the conditions according to which a sequential control system passes control from one step to the successor step.

Type

In SFC, there is not only the object type "SFC chart" but also the object type "SFC type". The SFC type allows the definition of ↑sequential control systems including an interface.

The SFC type cannot run alone. Like a function block type, an SFC type must be placed in a CFC chart to obtain a runnable object, in this case, an SFC instance (↑ instance). To run an SFC instance, both the SFC type and the SFC instance are loaded on the PLC.

U

Update cycle

In the Test mode, this specifies the intervals at which the watched values of the block I/Os are updated.

User program

The user program contains all the instructions and declarations and the data for processing the signals allowing a plant or process to be controlled. It is assigned to a programmable module (for example CPU, FM) and can be structured in smaller units.

In S7, a user program on the ES consists of the ↑ symbol table, the source files, the blocks, and the charts.

V

Value identifier

Symbolic substitute (text) for defined values of block I/Os of the data types BOOL, BYTE, INT, DINT, WORD and DWORD.

Index