SIEMENS

GRAPH 5

Ablaufsteuerungen graphisch programmieren unter dem Betriebssystem S5-DOS

Graphically Programming Sequence Controllers under S5-DOS Operating System

Programmation graphique des commandes séquentielles sous le système d'éxploitation S5-DOS

Handbuch
Manual
Manuel

6ES5 998-1SA01

Ausgabe 05
Release 05
5 ème édition

SIEMENS

SIMATIC S5

GRAPH 5

Handbuch
Manual
Manuel

Inhalt		Reg.
Wichtige Hinweise Important Notices Remarques importantes	C79000-R8563-C332	
GRAPH 5		
Handbuch	C79000-B8500-C332-01	1
Manual	C79000-B8576-C332-01	2
Manuel	C79000-B8577-C332-01	3
		4
		5

Manual
C79000-B8576-C332-01

Contents

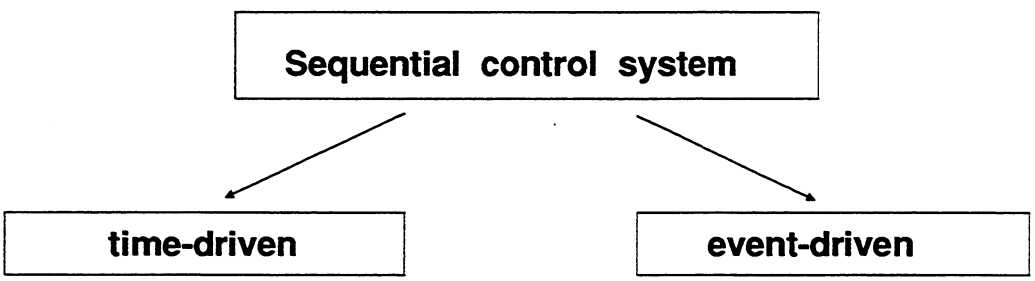
What is a Sequential Control System?

1 What is a Sequential Control System?

In control engineering, a distinction must be made between logic control systems and sequential control systems. **Logic control systems** describe the static relationships between the input and output signals of a controller. Control tasks in which the timing of inputs and outputs is important, are implemented by **sequential control systems**.

Sequential control: A mode of control, forcing step-by-step sequential operation, one step proceeding to the next programmed step dependent on step enabling conditions.

There are two different types of sequential control system:



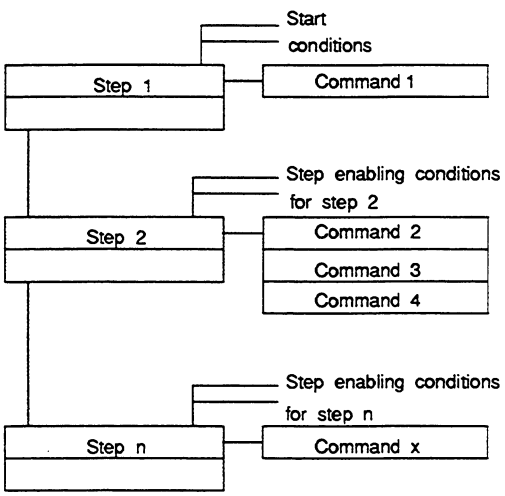
The step enabling conditions are only dependent on the time (e.g. waiting or monitoring times).

The step enabling conditions are dependent on signals from the process being controlled (e.g. on acknowledgements/feedback).

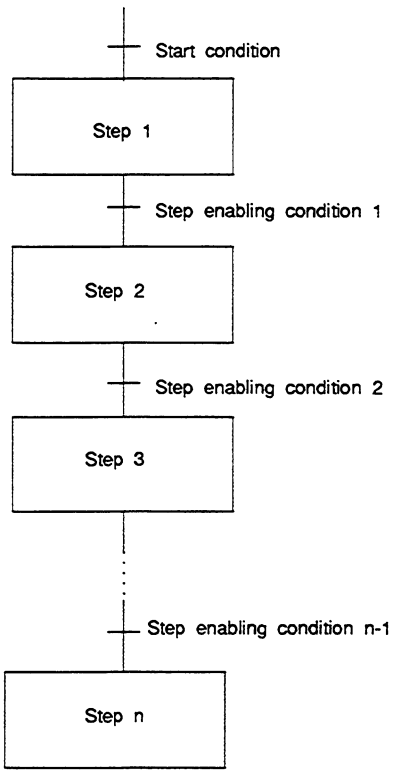
in practice, a combination of the two is usually found.

The main characteristics of sequential control systems are **steps** and **step enabling conditions**. The control task is divided into single steps whose execution is dependent on step enabling conditions.

Each step is assigned control operations and step enabling conditions.



Representation of the step-by-step sequence
using the conventional method
(acc. to DIN 40 719)



Representation of the step-by-step sequence
in GRAFCET notation
(acc. to IEC proposal SC65A/WG6)

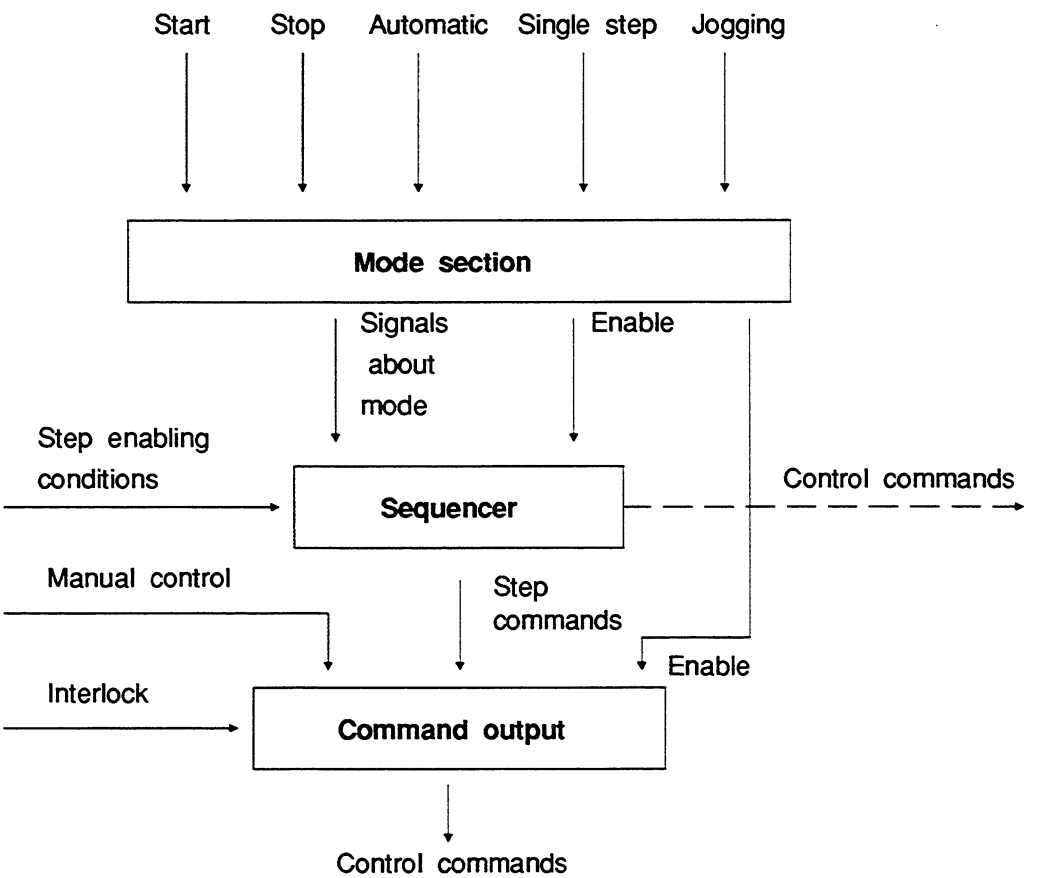
Sequential control system

The step enabling conditions allow the program to continue from one step to the next. The operations within a step consist of instructions for internal and external units (e.g. set flag, start timer, switch control elements).

Structure of a sequential control system

In general, a sequential control system consists of the following:

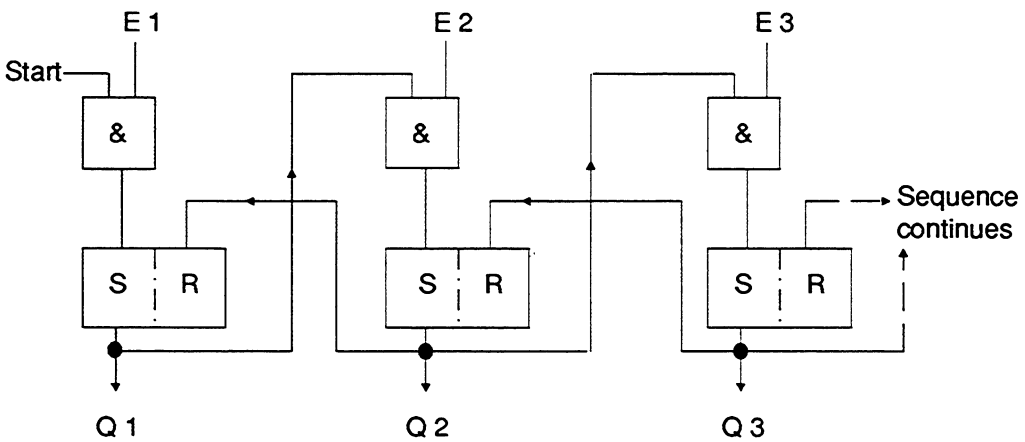
- mode section,
- sequencer,
- command output.



The preset parameters for the operating mode are processed in the **mode section**. The result is passed on to the sequencer and to command output in the form of signals (e.g. enable).

The **sequencer** ensures that the control is executed step-by-step. Depending on the step enabling conditions, the program proceeds from one step to the next.

Step enabling conditions



Basic structure of a sequencer (in conventional notation)

A step corresponds to a flip-flop. The output sends commands, initializes the next step and resets the previous step. The sequence continues depending on the step enabling conditions. The output of the control command can be directly from the step itself; however, the commands are usually sent to the control elements via the command output.

In the command **output** the step operations of the sequencer are logically linked with the signals from the mode section and the interlocks.

The outputs are commands to the control elements.

Why plan and program graphically?

Planning and programming a sequencer with the conventional methods is both time consuming and often proves difficult, particularly when dealing with more complex sequences (branches, jumps). You must establish the structure of the sequence by programming sequence blocks. With branches and jumps, the sequence is determined by load and transfer operations within the sequence blocks. Getting the timing right is just as awkward. Obtaining clear documentation is often difficult. Program tests involve a considerable amount of work.

Up to now, the only methods available for such tasks were extremely complicated, making the implementation of programmable sequential control systems difficult and demanding a relatively large investment of both time and effort.

New software was necessary for programming sequencers, that was both easier to handle and which made programming clearer and more user-friendly.

This is why a software package for graphic planning and programming of sequencers was developed:

GRAPH 5

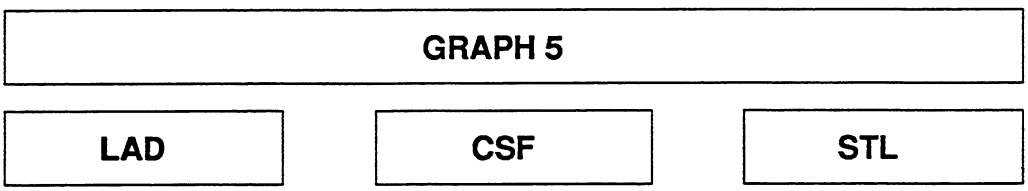
You first establish the structure of the sequencer graphically and then program the step enabling conditions (transitions) and actions (steps) at the detailed or zoom-in level in either LAD, CSF or STL. You enter the waiting or monitoring times by simply specifying a timer value in the sequence structure.

The modes are implemented using standard function blocks that are called in the user program.

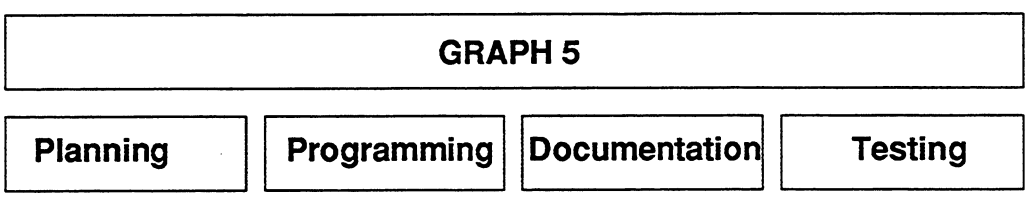
1.1 What is GRAPH 5?

GRAPH 5 is a software package for graphic planning and programming of sequential control systems and is an extension of STEP 5.

Using GRAPH 5, you can plan a program intended for step-by-step execution to match a technological sequence of events (i.e. sequential controller). You program the sequential operations in LAD, CSF or STL..



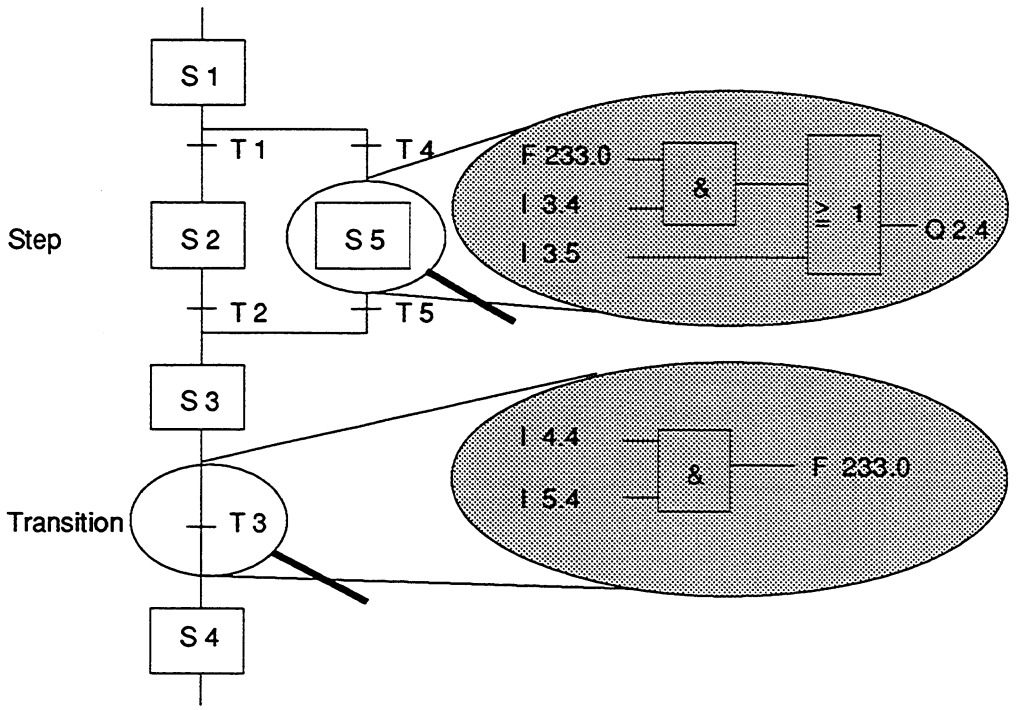
GRAPH 5
- is available on the programmer just as LAD, CSF, STL
- allows a sequencer to be structured by breaking it down into **steps** (actions) and **transitions** (step enabling conditions)
- supports planning and design, programming, documentation, testing/diagnosis.



Cyclic processing of the sequencer in the programmable controller is implemented by GRAPH 5 standard function blocks.

A sequencer is programmed with GRAPH 5 in two levels of representation:

1. the overview level
2. the detailed or "zoom-in" level



Representation at the overview level

Representation at the zoom-in level

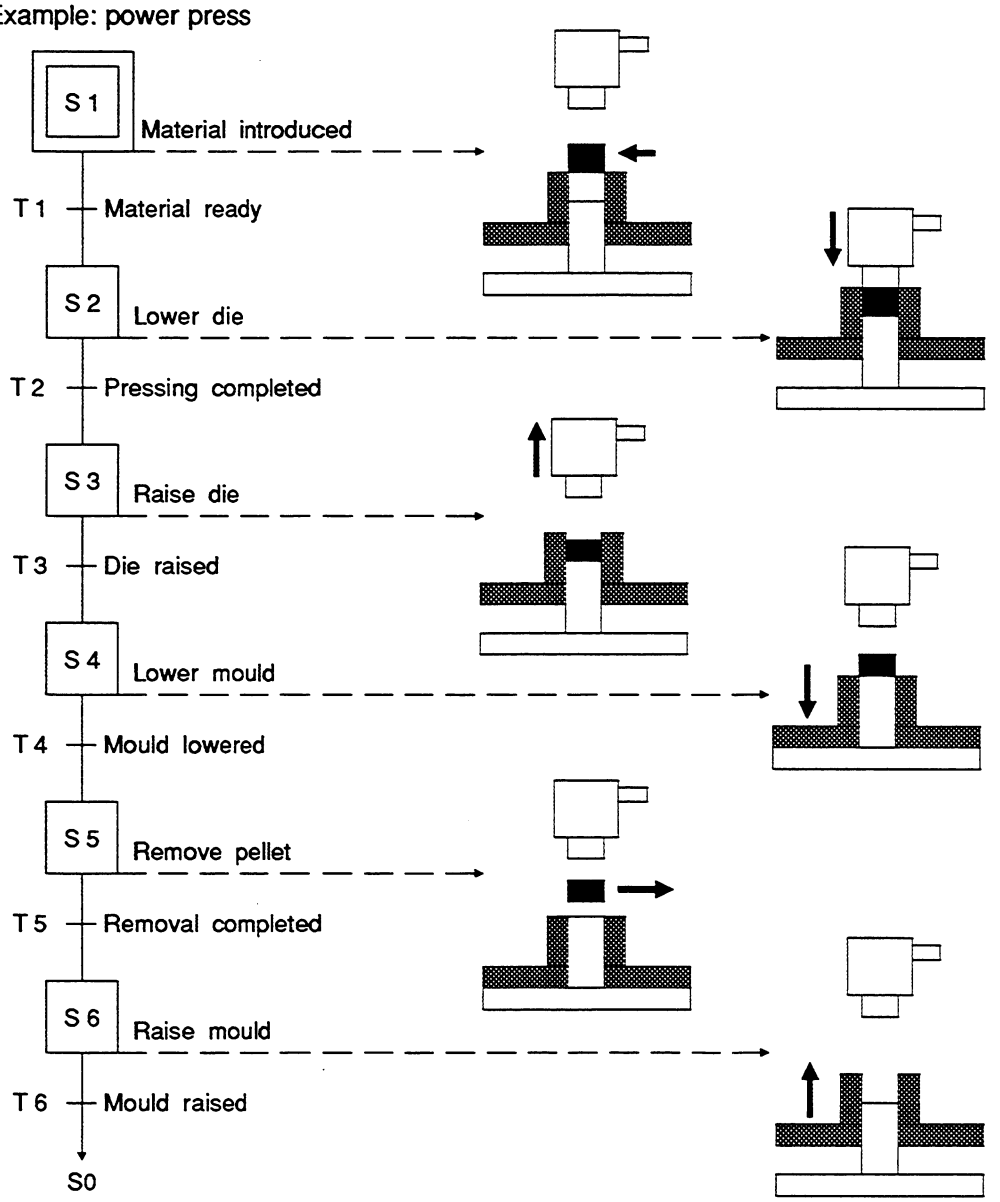
The overall structure of the sequencer is created at the **overview level**. Steps and transitions, simultaneous and alternative branches and their junctions, as well as jumps, can be programmed. Waiting and monitoring times can be entered.

At the **zoom-in level** the contents of the steps and transitions are programmed using the zoom-in function:

- the actions in the **step**
- the step enabling conditions in the **transition**

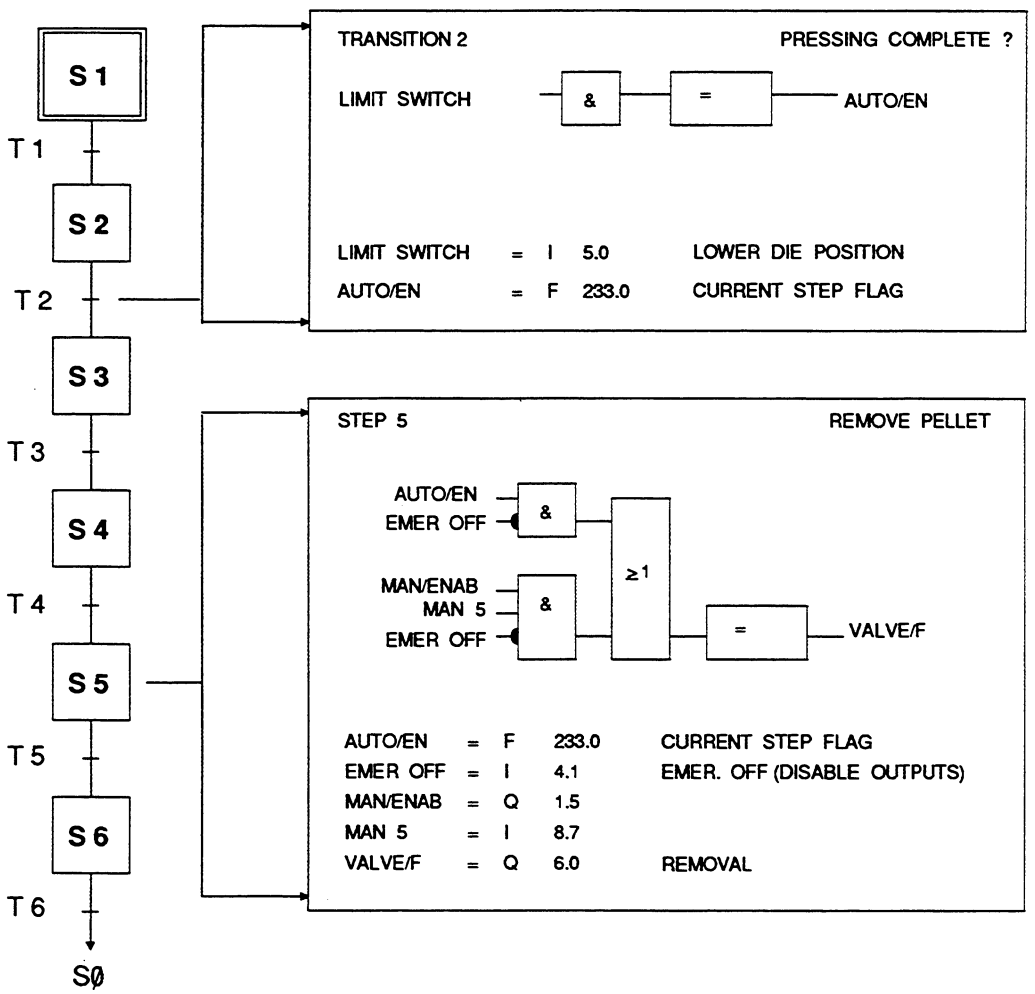
Representation at the overview level

Example: power press



Representation at the zoom-in level

Example: powder press



After **planning** the structure of the sequencer at the **overview** level and **programming** the steps and transitions at the **zoom-in** level, the sequential control program is completely established.

GRAPH 5 therefore provides you with a user interface with which you can create straightforward sequential control systems easily and quickly.

Apart from supporting planning and programming, GRAPH 5 also supports documentation and testing/diagnosis.

Documentation

During planning and programming:

- comments for the steps and transitions at the overview level,
- (step/transition) segment titles, statement and segment comments, display of the assignment list of symbols used at the zoom-in level.

Printout of the following:

- sequence identification screen form,
- overview level with all comments,
- list of all transitions and transition comments,
- list of all steps with step comments and corresponding waiting and monitoring times,
- all the transitions at the zoom-in level with transition comments (segment titles), statement comments, assignment list of the symbols used,
- all the steps at the zoom-in level with step comments (segment titles), statement comments, assignment list of the symbols used.

Testing/Diagnosis:

The current status of the sequencer is displayed in a status display, i.e. active steps are clearly marked. The status of individual steps and transitions can also be followed at the zoom-in level, with the statuses of individual operands and logic operations displayed on the screen.

If a timeout occurs, the affected sequencer is indicated. The cause of the timeout can be traced from the overview level through to the zoom-in level. If you select the sequencer involved, you can display the affected step (steps). You can find out the exact cause of the timeout at the zoom-in level.

Definition of terms

Active step

A step is active when the actions contained in it are being executed.

Valid transition

A transition is valid when the step(s) preceding it is (are) active.

Switching transitions

A transition switches when it is valid and the step enabling conditions are satisfied. Switching means that the transition deactivates the preceding step(s) and activates the next step(s).

1.2 How does GRAPH 5 Function?

A sequence control system is characterized by its steps, i.e. by the subdivision of a control task into individual sequence steps. It proceeds to the next step depending on the **step enabling conditions**.

With GRAPH 5, the structure of the sequencer is determined by the following:

1. Step: Description of the actions executed by the sequencer when a certain status exists. These actions are programmed at the zoom-in level; a flag (F 233.0) is used as a substitute for the enable signal. This flag has the value 1, when the step is active.

2. Transition: Description of the step enabling conditions with which a sequencer changes from one status to the next (i.e. proceeds from one step to the next). These step enabling conditions are programmed at the zoom-in level. The result of logic operation is not the definitive step enabling condition, it can, in some cases, still be corrected by the GRAPH 5 program, e.g. waiting times not yet elapsed etc. Flag 233.0 is used to activate the next step(s).

For **planning** and programming using the PG, the GRAPH 5 PG software is necessary. The program for a sequential control system is created offline.

To run the program on a programmable controller (PLC), the standard function blocks for GRAPH 5 are necessary. These FBs are available for specific PLCs. The FBs are used to implement the modes of the sequencer. They are called in the user program from which they also obtain the required parameters.

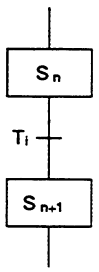
Testing and **diagnosis** are carried out online with the PLC.

1.3 The Elements of GRAPH 5

The elements result from a series of steps and transitions. The following rule applies:

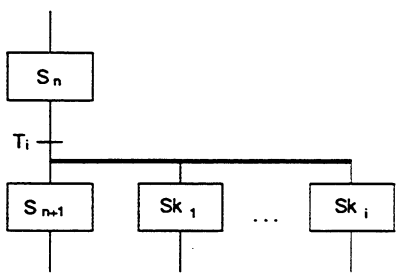
Every transition must follow a step and every step must follow a transition.

1. Linear sequence



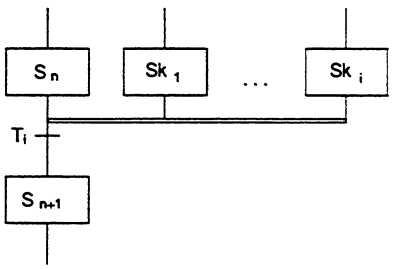
The sequence proceeds from S_n to S_{n+1}. When T_i switches, S_{n+1} is activated and S_n deactivated. If several steps follow on in a linear sequence, they are lined up one after the other.

2. Simultaneous branch



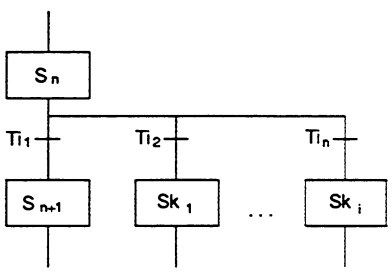
Several steps are activated simultaneously by only **one** transition. The sequence proceeds from S_n to S_{n+1} **and** Sk_1 **and** ... Sk_i.
When T_i switches, S_{n+1} to Sk_i are activated and S_n is deactivated (corresponds to an AND sequence).

3. Synchronization



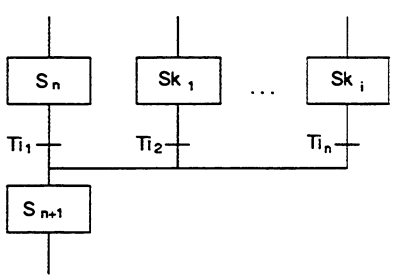
Parallel branches are joined again by means of the synchronization. The sequencer proceeds from S_n **and** Sk_1 **and**... Sk_i to S_{n+1}.
T_i becomes valid when all the preceding steps S_n to Sk_i are active. When T_i switches, these steps are deactivated, S_{n+1} is activated.

4. Alternative branch



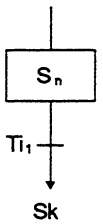
One of the branches will be run through. The sequencer proceeds from S_n to S_{n+1} **or** (exclusive) Sk_1 **or**... Sk_i.
As soon as S_n is active, all the transitions Ti_1 to Ti_n become valid. The transition with a satisfied enabling condition will switch.
Note: If possible, the step enabling conditions of the transitions Ti_1 to Ti_n should be mutually exclusive. If the conditions for several transitions are satisfied simultaneously, the transition furthest left will be enabled (corresponds to OR sequence).

5. Alternative junction



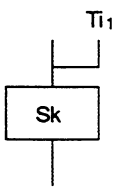
When alternative branches join again, the following step S_{n+1} will be activated when one of the previous transitions Ti_1 to Ti_n switches. The sequencer therefore proceeds from S_n or Sk_1 or... Sk_i to the next step.

6. Jump



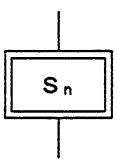
The sequencer proceeds from S_n to Sk (as with the linear sequence, however, without a graphic connection). When Ti_1 switches, S_n is deactivated and Sk is activated.

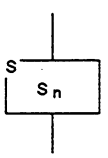
7. Junction of a jump



The target of the jump Sk is activated when T_i switches. There is no graphic connection between T_i and Sk.

8. Initial, selective stept

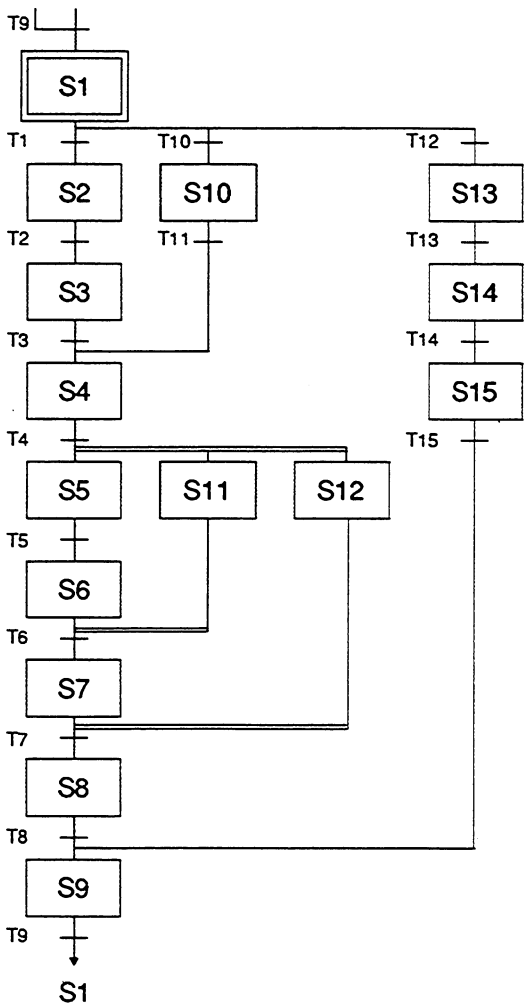


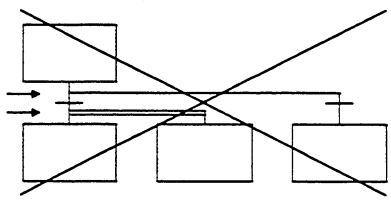


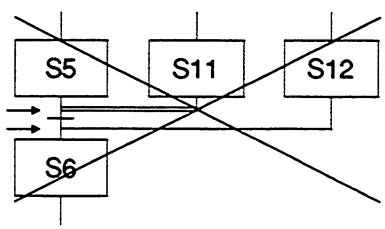
Initial step:
This is activated at the start of the sequence without the conditions being checked.

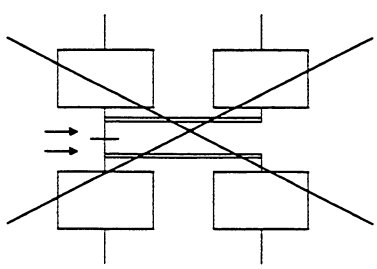
Selective step:
The action part of a step marked as selective is only processed when the step is active. Normally (without selective steps) all the steps of the sequencer are run through cyclically; if a step is not active, the actions are not carried out. With a selective step, the action part is skipped if the step is not active.
Caution:
any interlocks programmed in the step will also be skipped!

You can program the structure of your sequencer with these elements.
By nesting parallel and alternative branches, complex structures can be created, as shown below:









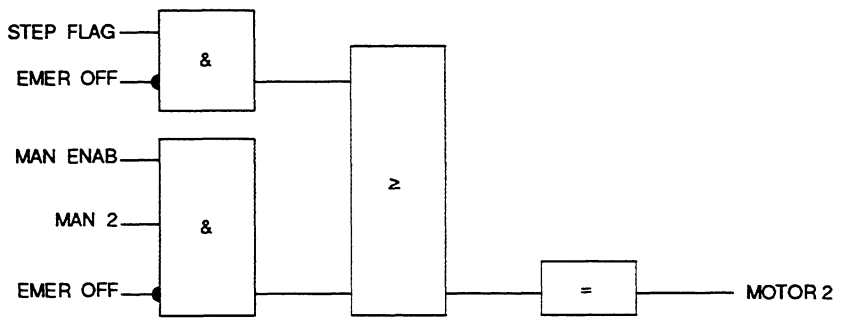
☞ It is not possible to have two branches following each other immediately without a step between them.
This also applies to the junctions of two branches.

1.4 Zoom-in Level

The steps and transitions are programmed at the zoom-in level, i.e. their content is specified in LAD, CSF or STL.

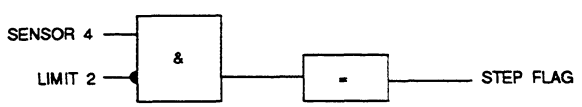
Programming the steps:

Steps are the active part of the sequencer. Commands, e.g. to actuators, load and transfer operations, starting timers and counters and FB calls are programmed in the steps. The step flag is assigned by GRAPH 5; the action part can be programmed as required for the task in hand. Interlocks are programmed at the same time (single control element).



Programming the transitions:

Transitions are the step enabling conditions for the steps. The conditions that must be satisfied to allow the next step (or steps) to be activated must be programmed.



1.5 Waiting and Monitoring Times

A waiting and a monitoring time can be assigned to every step.

Waiting time The minimum time TW, for which a step remains enabled even if the following transition has already been satisfied. The next step will only become active after the waiting time has elapsed.

Monitoring time The program checks whether or not the step enabling conditions for the next step become active within a preset time (TM). The sequencer must switch to the next step before TM elapses, otherwise a timeout will be detected.

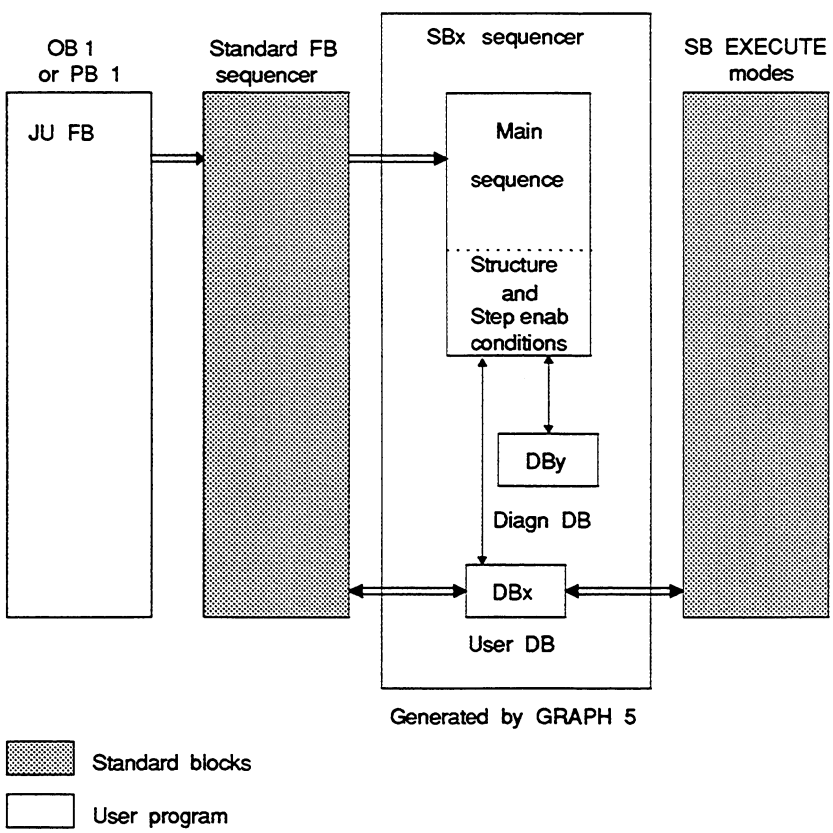
You can specify different values for TM and TW in every step. You simply need to enter time values. The time function does not need to be scanned in the next transition, but is evaluated automatically by the standard function blocks for the modes.

1.6 Comments

You can specify a comment for every step and transition. The comments are displayed both at the overview and at the zoom-in level.

1.7 Program Structure of a Sequencer with GRAPH 5

The program structure of a sequential control system is generated and managed largely by the GRAPH 5 system. The creation of the program (sequence block and data block) is supported by the programmer. To run the program on the programmable controller, you must call and assign parameters to the standard function block "sequencer". The SB "execute" is not called by the user program but simply loaded in the PLC. The user DB and diagnosis DB are not programmed, but generated with the DBGEN PG function and loaded in the PLC.



You call the standard function block "sequencer" and assign parameters for the sequencer (modes, sequence block number etc.). The structure of the sequencer and all the actions and step enabling conditions are located in the sequence block SBx.
At each transition, the "execute" SB is called to execute the modes. The DBx is the data block for SBx. The diagnosis data block DBy is available for diagnosis and is used by all the sequence blocks in the PLC.
Secondary sequences are possible.

Programming with GRAPH 5

Offline:

- In a program block (or OB 1), call the standard function block FB 70 or FB 72 or FB 73 for the sequencer and assign parameters (modes). The "execute" SB is not called but simply loaded in the PLC.
- If you require options for supplementary functions (FB 72, FB 73) call FB 74 in the PB or OB 1.
- For secondary sequencers, call FB 71 in SBx.
- Create the sequence block SBx on the programmer, structure the sequencer at the overview level and program the contents of steps and transitions at the zoom-in level.
- Create the user DB and the diagnosis DB with the DBGEN function on the programmer.
- For a fast re-translation of the sequence blocks, generate the block #SBRL with the RLGEN function.

Online:

- Transfer all the blocks required for running the program to the PLC.

Blocks required in the PLC

Standard FBs:	FB 70	for the main sequence
	FB 71	for the secondary sequence (if required)
	SB 0	execute block
	and/or	
	FB 72	for the main sequence
	FB 74	for the modes (if required)
	SB 2	execute block
	and/or	
	FB 73	for the main sequence
	FB 74	for the modes (if required)
	SB 3	execute block
Sequence and data blocks:	: SBx	sequence block (for the main sequence, if applicable, further sequence block(s) for the secondary sequence(s)
	DBx	user DB
	DBy	diagnosis DB

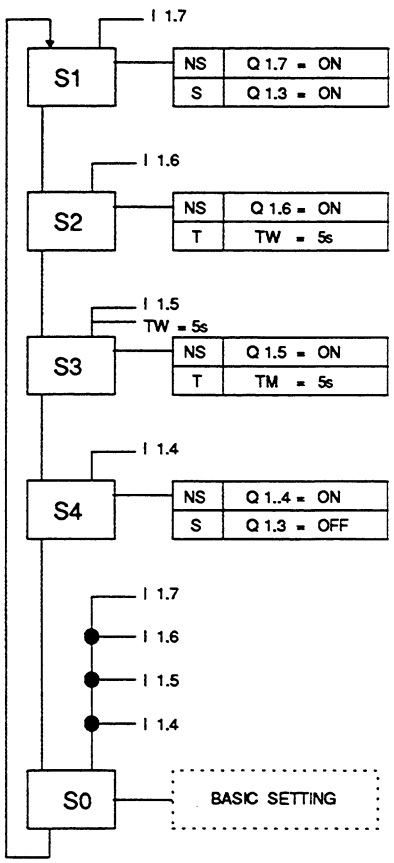
All further blocks for the user program, e.g.

OB, PB, FB, FX, DB, DX

1.8 Examples with and without GRAPH 5

Programming without GRAPH 5

Old representation acc. to DIN 40719

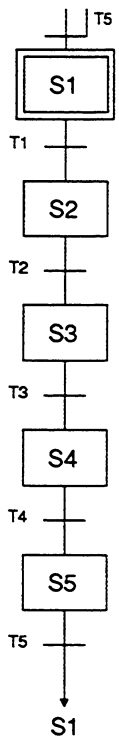


STL sequential control program

SB 16:	A	I	1.7
	BE		
SB 17:	=	Q	1.7
	S	Q	1.3
	A	I	1.6
	BE		
SB 18:	=	Q	1.6
	L	KT	50.1
	SD	T	40
	A	I	1.5
	A	T	40
	BE		
SB 19:	=	Q	1.5
	L	KT	50.1
	SD	T	41
	A	I	1.4
	BE		
SB 20:	=	Q	1.4
	R	Q	1.3
	AN	I	1.7
	AN	I	1.6
	AN	I	1.5
	AN	I	1.4
	BE		

It was previously not possible to program sequencers graphically. According to the DIN standard, the design had to be translated into an STL program.
A separate SB had to be programmed for every step, taking care to adhere to the necessary sequence of operations.
You had to make sure that a branch was executed correctly by the program by scanning the branch conditions and loading and transferring the next sequence block number.
The sequencer structure was established by programming the sequence blocks.

Programmimg with GRAPH 5



Step 1:

Step 2:
: *A step flag*
:= Q 1.7
:S Q 1.4
: *BE*

Step 3:
: *A step flag*
:= A Q 1.6
: *BE*

Step 4:
: *A step flag*
:= A Q 1.5
: *BE*

Step 5:
: *A step flag*
:= Q 1.4
:S Q 1..3
: *BE*

Transition 1:
: A I 1..7
: *BE*

Transition 2:
: A I 1.6
: *BE*

Transition 3:
: A I 1.5
: *BE*

Transition 4:
: A I 1.4
: *BE*

Transition 5:
: AN I 1..7
: AN I 1.6
: AN I 1.5
: AN I 1.4
: *BE*

Italics: *preset by the system; you do not need to make these entries*

The sequence is programmed graphically at the overview level. The zoom-in programming can be performed in LAD, CSF or STL.
The entire sequencer is located in one SB.
The structure is clearly established in the overview. You only need to program the actions and step enabling conditions, the program code for the sequencer (switching mechanism) is generated automatically by GRAPH 5.

1.9 Advantages of Programming with GRAPH 5

With GRAPH 5, you can program sequencers both easily and clearly. Compared with the conventional methods of programming sequencers, GRAPH 5 makes your job much simpler.

Previously	**New with GRAPH 5**
A sequence block had to be programmed for every step.	One sequence block contains the whole sequencer.
Manual conversion of the sequencer into LAD, CSF, STL.	Programming with GRAPH 5. **Conversion** performed **automatically**.
The sequencer structure had to be established by the program. SBs required special handling for branches, junctions and jumps. You had to make sure that the correct step was called.	The **overview representation** contains all the information about the structure, the sequence is clearly established and is automatically converted into program code.
You had to start and scan monitoring and waiting times.	You can specify **waiting** and **monitoring times**. The timers are started and evaluated automatically.

In addition to convenient programming, GRAPH 5 also has the following advantages:

- GRAPH 5 provides convenient **design and planning functions** when working on the programmer.
- GRAPH 5 is an efficient means of **structuring**, making the creation of programs more cost-effective.
- GRAPH 5 automatically converts the created sequencer structure into a program.
- GRAPH 5 makes fast **diagnosis** easier from the overview level down to the zoom-in level.
- GRAPH 5 provides you with **clear documentation**.

With GRAPH 5, you can program sequencers clearly, accurately and quickly. You can get to know GRAPH 5 easily and are soon in a position to create your own sequencers.

1.10 Characteristics of GRAPH 5

GRAPH 5 is a software package for planning and programming sequencers.

Device

Programmers with the S5-DOS (PCP/M) operating system:
PG 635 II, PG 685, PG 730, PG 750, PG 710, PG 770
Programmers with the S5-DOS/MT operating system (FlexOS): PG 730, PG 750, PG 770

Programmable controllers S5 100 with CPU 103, S5 115 U, S5 135 U with S/R processor, S5 135 U with CPU 928, S5 150 S, S5 150 U, S5 155 U, S5 155 H.

Elements/functions

Simultaneous/alternative branching,
simultaneous/alternative junctions,
jumps,
waiting/monitoring times,
zoom-in (LAD, CSF, STL),
comments.

Technical data

All sequencer steps in **one** sequence block,
max. 246 sequencers per PLC (SB 10...255)
max. 127 step/transition pairs per sequencer,
max. 8 Init steps per sequencers,
max. 8 simultaneous or alternative branches,
max. 31 branches and junctions in total,
main/secondary sequencer,
basic modes: OFF, AUTOMATIC, MANUAL,
zoom-in programming in LAD, CSF, STL,
corresponds to IEC draft standard SC65A/WG6.

Working on the Programmer

2 Working on the Programmer

This chapter describes how to use GRAPH 5 on the programmer:
- from package selection to function selection,
- GRAPH 5 functions, how to program the sequence identification and sequential control systems, and
- functions of the special keys for GRAPH 5.

Before you start working with GRAPH 5, please read the software contract and the product information carefully. Installing and using the PG, handling diskettes and hard disk drives are described in the PG manuals. Installing the GRAPH 5 software package is described in the product information.

☞ Make at least one back-up copy of all original diskettes.

How to start the S5 command interpreter (select package) is also described in the PG manuals. This manual describes the GRAPH 5 functions following package selection. To program at the zoom-in level, you must be familiar with STEP 5. To run the program in the PLC, you must call and assign parameters to the PLC-specific function blocks (FB 70, FB 71 etc.) for the required version of the sequential control system and transfer the program with the standard FB to the PLC memory. For further information about the procedures, we recommend the following literature:
- manual for the STEP 5 basic package,
- description of GRAPH 5, standard FBs for programmable controllers,
- programmable controller manuals, e.g. S5 155U. These manuals also include programming instructions.

Notation used in this manual

>	This character precedes any activity you perform on the programmer.
bold	Input, operations and keys are shown in bold face.
italics	Messages displayed by the programmer are printed in *italics*, except for command lines that are framed by an oblong box.

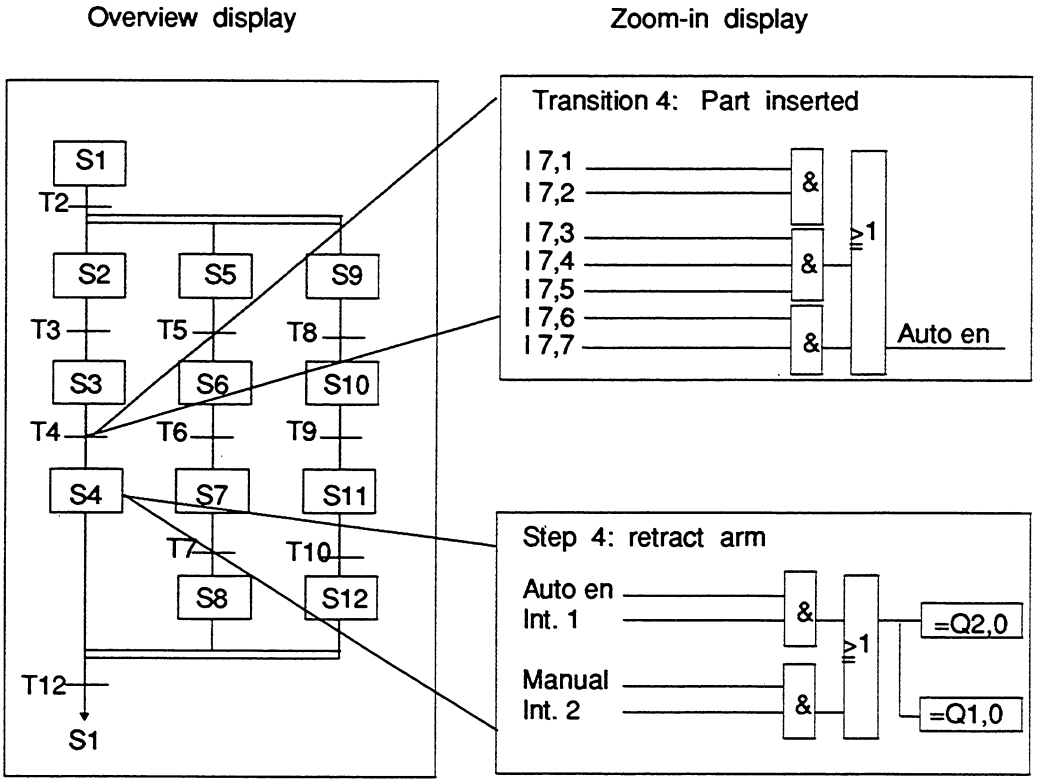


Fig. 2.1 GRAPH 5, overview and zoom-in representation

The PG supports you throughout program creation:
- planning and programming the sequencer,
- creating the user and diagnosis DB,
- assigning parameters to function blocks and
- testing the program.

The structure of the program for a sequencer (without secondary sequencers) is shown below:

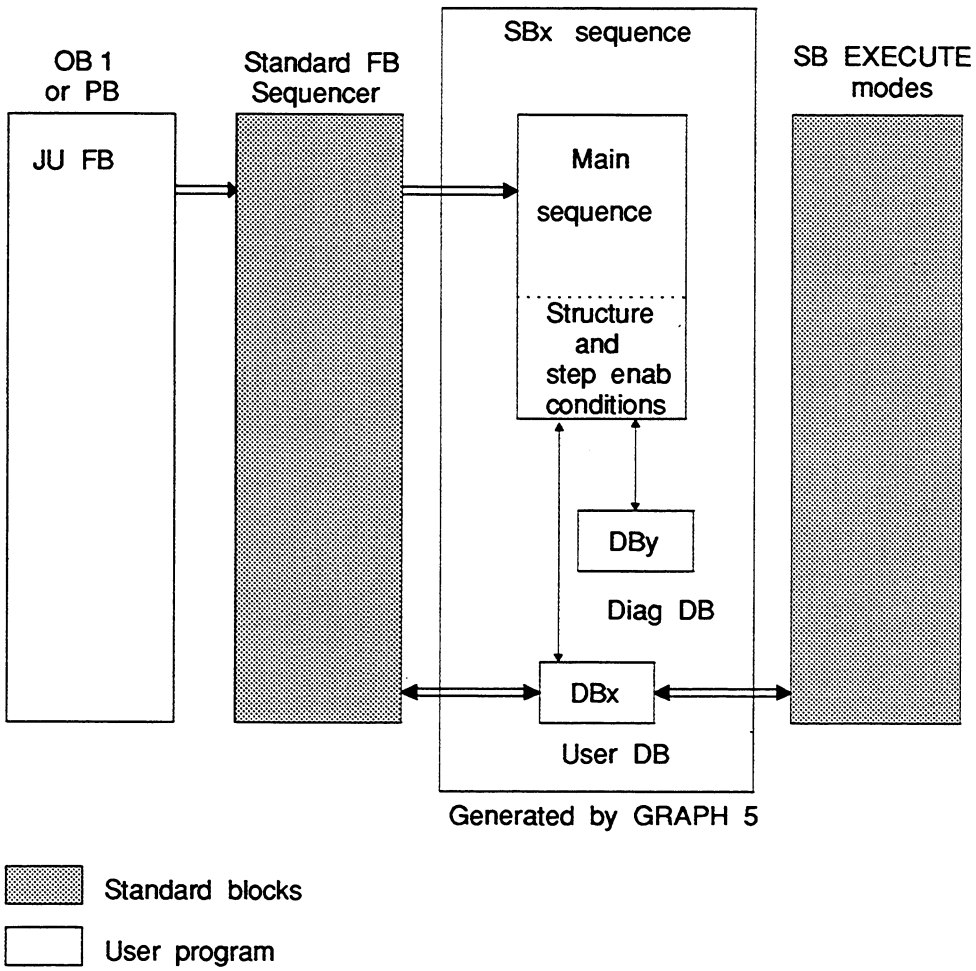


Fig. 2.2 Program structure of a sequencer with GRAPH 5

OB1 or PB Call and assign parameters to the standard FB for GRAPH 5.

Standard FB The standard function blocks FB 70 to FB 74 and the standard block SB "execute" (SB 0, SB 2, SB 3) manage the sequencer created in SBn and implement the modes (e.g. AUTOMATIC, MANUAL etc.). The assignment of parameters and incorporating the sequencer into the user program is explained in the descriptions of the software package for standard FBs.

SBx The structure of the sequencer, the step enabling conditions and actions are specified in the sequence block. An SB consists of steps and transitions. Actions are programmed in the step part and step enabling conditions in the transition part, where SB "execute" is also called.

SB execute This standard block executes the operating modes transferred to the standard FB as parameters. SB execute is only loaded in the PLC no parameters can be assigned.

User DB For each sequencer, a user DB with the same number as the sequence block is required. This contains data about the structure, initialization steps, programmed timers etc. and stores the status of the sequencer for processing in the next PLC cycle. The user DB is generated with the DBGEN PG function after you have created the sequencers.

Diagnosis DB There is a common diagnosis DB to perform diagnostic functions for all the sequencers in the PLC. This DB contains the number of a step in which a timeout occurred as required for diagnosis on the PG. The diagnosis DB is only necessary when using FB 70. If you use FB 72 or FB 73 (FB 74), the diagnosis DB does not need to exist in the PLC memory. The diagnosis DB is generated with the DBGEN PG function after you have created the sequencers.

The program is stored on diskette and/or hard disk and transferred to the PLC memory.

2.1 From Package Selection to Function Selection

After you start the GRAPH 5 S5 package, you require three steps from package selection (S5 KOMI) to calling a function.

1st step : Select package

2nd step : Enter presets

3rd step : Select function

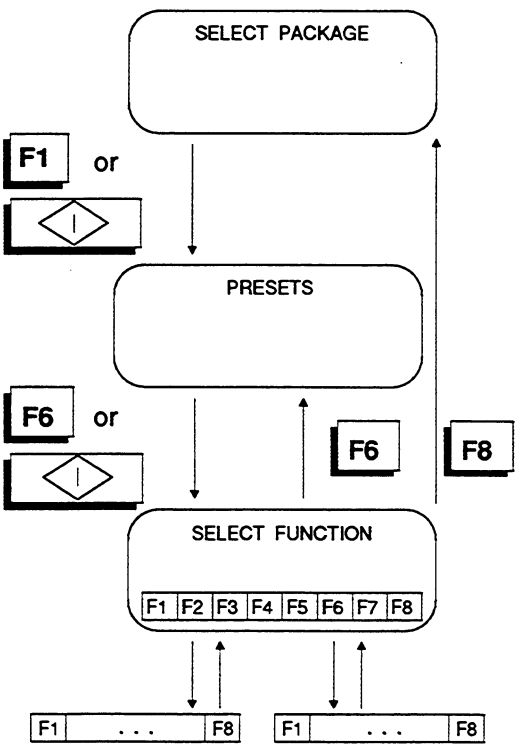


Fig. 2.3 Hierarchy within the STEP 5 packages

2.1.1 Select the GRAPH 5 Package

Ready to start?

You have made a back-up copy of the GRAPH 5 package. S5 KOMI and the GRAPH 5 package are on the hard disk or on diskette. We must assume that you are familiar with using the PG and can program in STEP 5.
You have started S5 KOMI.
The SELECT PACKAGE screen form is displayed.

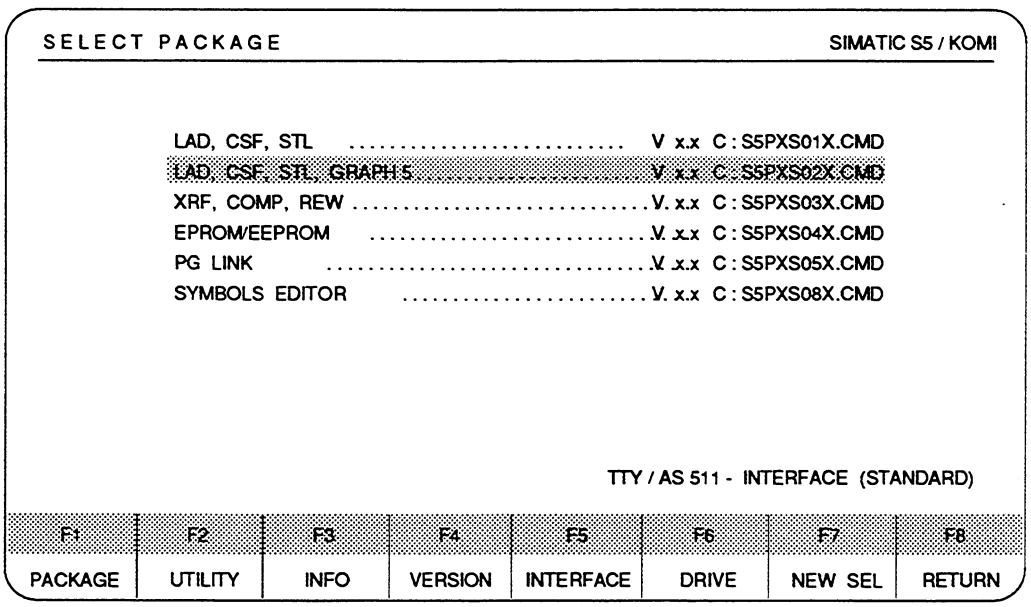


Fig. 2.4 Example of the SELECT PACKAGE screen form

> Position the cursor on the **GRAPH 5 ..... B:S5PXSO2X.CMD** package.
> Press **F1** (PACKAGE) to start the GRAPH 5 package.

☞ If you require information about the package, press **F3** (INFO).

2.1.2 Presets

The presets screen form now appears. This is the same as in the LAD, CSF, STL package.

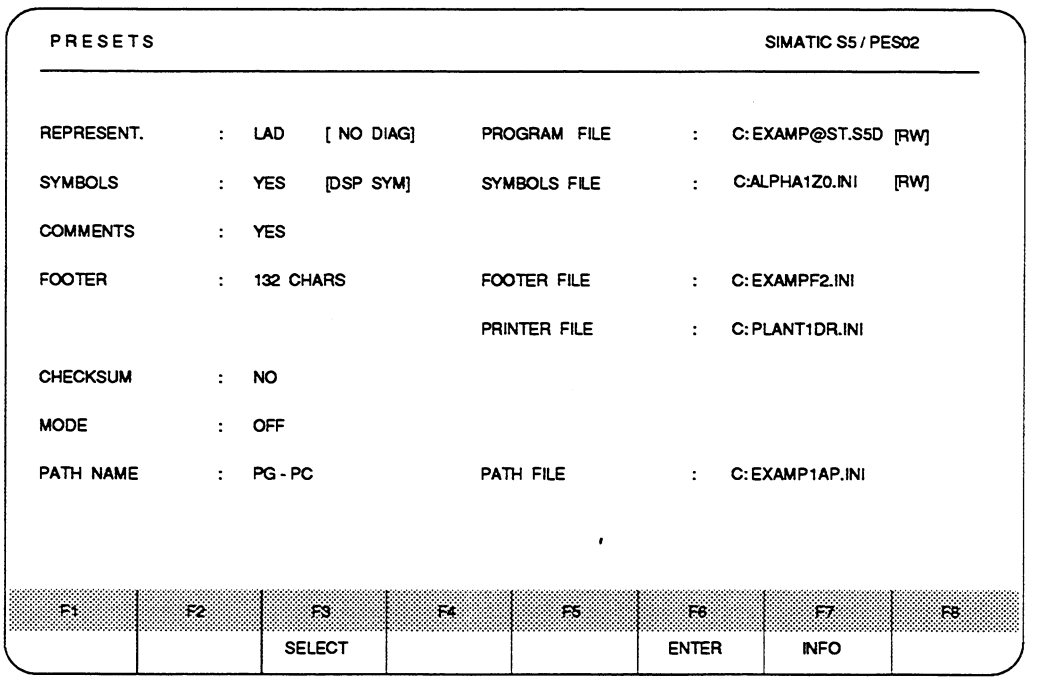


Fig. 2.5 Example of the PRESETS screen form

F3 (SELECT) Indicates the options available in the field marked by the cursor.

F6 (ENTER) Declares the selected and displayed parameters as valid (DEFAULT) and calls function selection.

F7 (INFO) This key provides you with an explanation of the field currently marked by the cursor.

Enter key The enter key has the same function as the function key **F6** (ENTER).

Break key The PG does not enter the parameters you have just input or modified.

> Position the cursor and make your selection with **F3** (SELECT).
> Enter the presets by pressing **F6** (ENTER) or the **enter key**.

2.1.3 Select Function

F1	F2	F3	F4	F5	F6	F7	F8
INPUT	OUTPUT	TEST	PC FCT	PC INFO	PRESETS	AUX FCT	RETURN

> Press **F1** (INPUT).

F1	F2	F3	F4	F5	F6	F7	F8
BLOCK	GRAPH5		SCR FORM	DBGEN	RLGEN		RETURN

> Press **F2** (GRAPH 5).

The following command line appears on the screen:

```
INPUT GRAPH 5 DEVICE:     BLOCK:
```

> Complete the command line and press the **enter key**, e.g.

```
INPUT GRAPH 5  DEVICE: FD  BLOCK: SB 10
```

☞ You must only use sequence blocks from SB 10 onwards. SBs 0 to 9 and DBs 0 to 9 are used by SIMATIC (PLC, interface modules etc.) and are not available for other purposes.

2.2 GRAPH 5 Functions

2.2.1 Sequential Control, Sequence Identification

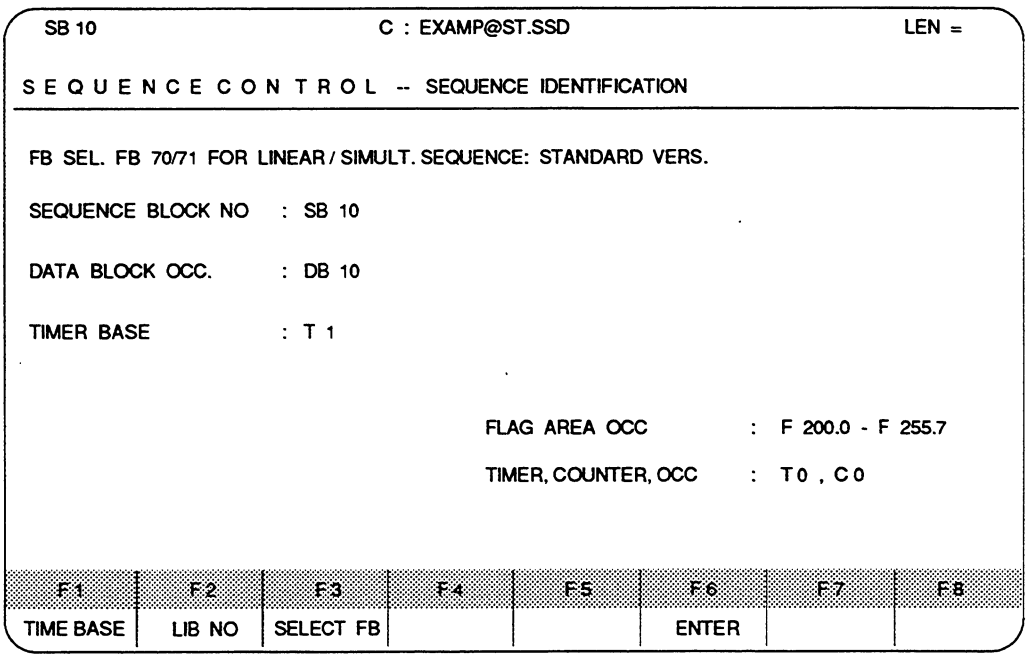


Fig. 2.6 Example of the SEQUENCE CONTROL - SEQUENCE IDENTIFICATION screen form

The data block number corresponds to the sequence block number; this 1:1 assignment cannot be altered. You can use sequence blocks from SB 10 onwards. Flags, timers and counters used by GRAPH 5 are displayed in the ID screen form.

☞ The flag area is not available within the sequencer, outside the sequencer it can be used as a scratchpad area.

TIME BASE

If you press **F1** (TIME BASE) you can enter the timer base (T 1 ... T 252). The timer base specifies the start of the area used for waiting and monitoring times. The timer T 0 is used by GRAPH 5. Two timers are required per simultaneous branch (max. 2 x 8 = 16), i.e. if a simultaneous branch is programmed, T 1 to T 252 can be used as the start address of the timer base. If 8 simultaneous branches are programmed, T 1 to T 238 are possible as the start address.

☞ These timers are occupied even if no times are entered in the branch; they must not be used outside GRAPH 5. The timers permitted depend on the particular PLC. They must not overlap for different sequence blocks loaded in the PLC.

LIB NO

If you press **F2** (LIB NO) you can enter a 1 to 5 digit library number. The library number can only be input or modified in the sequence identification screen form. When you output (display) an SB, the **F2** (LIB NO) key displayed in the softkey menu has no effect in GRAPH 5.

SELECT FB

If you press **F3** (SELECT FB), you select the FBs for the standard programs.

The following function blocks are available:

- FB 70/71 for linear/simultaneous sequence, STANDARD VERSION
- FB 72 for linear/simultaneous sequence: FAST VERSION
- FB 73 for linear sequence: FAST VERSION
- FB 78 for GRAPH 5-EDDI, this function is described in the GRAPH 5-EDDI manual.

☞ The selected FB and the corresponding "execute" SB (SB 0 for FB 70/71, SB 2 for FB 72, SB 3 for FB 73) must be available in the programmable controller. If FB 73 (for linear sequences) is selected, then no simultaneous branch can be entered in the sequencer.

2.2.2 Programming a Sequential Control System

The screen displays the minimum structure of the sequencer at the overview level, as soon as you press **F6** (ENTER) or the **enter key** in the ID screen form.

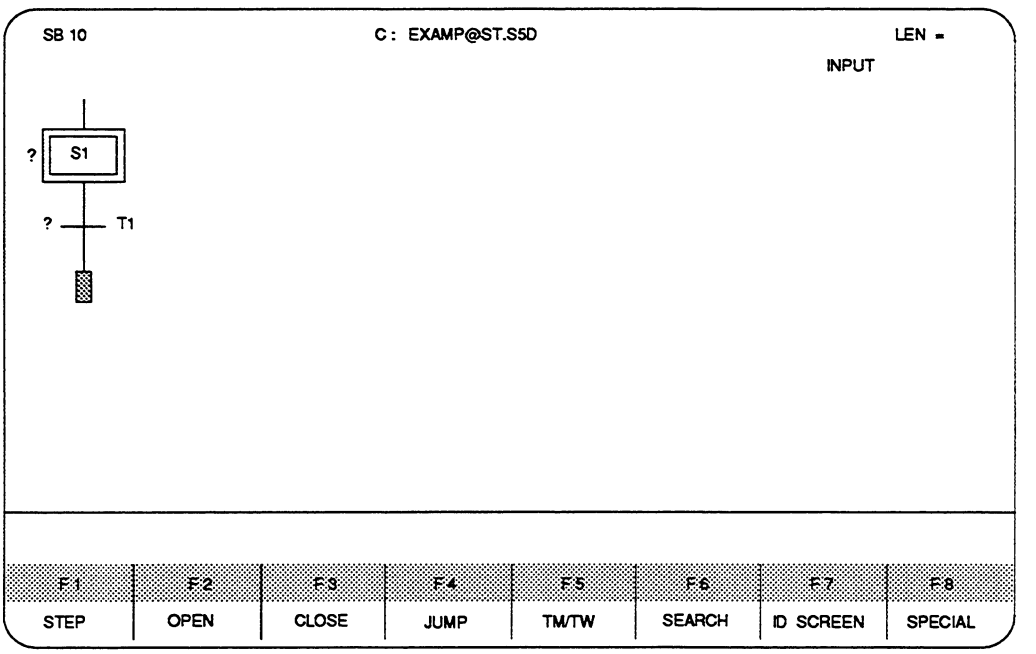
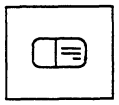


Fig. 2.7 Minimum structure of a sequencer at the overview level

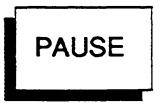
You can now program the sequencer both at the overview and zoom-in levels. The special keys for GRAPH 5 programming are now available in addition to the STEP 5 keys.

2.3 Special Keys for GRAPH 5

Display mode

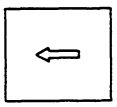


Half screen

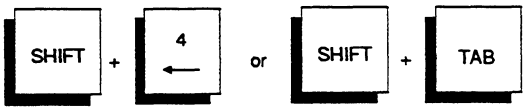


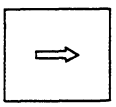
At the overview level: you switch over between "half" and "full screen".
At the zoom-in level: STL output and correction mode: changes the display between operands and statement comments for the step or transition.

Scrolling

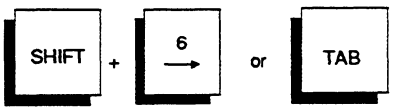


Double arrow key left



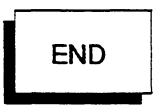


Double arrow key right



At the overview level in the "half screen" representation: move the screen contents to the right or left.
At the zoom-in level: position the cursor.

Zoom-in function
Special function in GRAPH 5



Zoom-in function

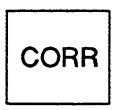
Change from the overview level to the zoom-in level.
At the zoom-in level in the **correction mode**: display of the assignment list with the assignments of absolute and symbolic operands in the segment.

Comment

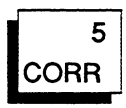
Comment key

At the overview level: comments for steps/transitions.
At the zoom-in level: 1 xCOM : segment title
2 xCOM : segment comment

Correction

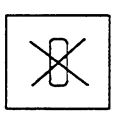


Correction mode

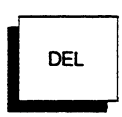


At the overview level:
During output, change to the correction mode and display softkey level 1.
At the zoom-in level:
change to the correction mode, in LAD/CSF, display of the absolute and symbolic operands with operand comments depending on the cursor position.

Delete



Delete character



At the overview level: delete step/transtition, branch or jump.
At the zoom-in level: delete the character at the cursor position.

2.3.1 Screen Mode

At the overview level

At this level, you can use the **half screen** key to switch over between the comment output options half/full screen.
In the "full screen" representation (8 parallel sequencers), the comment of the sequencer element marked by the cursor appears in the lower comment line. In the "half screen" representation (4 parallel sequencers), the right-hand half of the screen is used to list the comments of the first three colums (sequencers) on the left of the screen. By moving the left-hand screen contents to the left or right, you can display the other comments.

Ready to start?

You have selected COMMENTS: YES in the presets screen form.
A sequence block is displayed at the overview level.

Keystrokes

> Press the **half screen** key.

Each time you press this key, the display changes from "half" to "full screen" and vice-versa.

At the zoom-in level

In the output or correction mode of STL, you can switch over between the display of operand and statement comments by pressing the **half screen** key.

Ready to start?

Presets REPRESENT: STL
COMMENTS: YES
SYMBOLS: YES

Display of a step or transition at the zoom-in level.

Keystrokes

> Press the **half screen** key.

The display changes from operand comments to statement comments and vice-versa.

2.3.2 Scrolling

If there are more than four parallel sequencers and you are using the "half screen" mode, only three sequencers along with their step and transition comments can be displayed.
Using the double arrow keys right/left, you can move the display on the left-hand side of the screen horizontally. The comments on the right are then changed to match the display on the left.

Ready to start?

Presets COMMENTS: YES
A sequence block is displayed at the overview level.

Keystrokes

To move the left half of the screen to the right or left:

> Press the **double arrow key right/left**

2.3.3 The Zoom-in Function

The contents of transitions and steps are programmed at the zoom-in level in STEP 5 (LAD, CSF, STL). You position the cursor on the required step or transition. With the zoom-in key, you then display the first segment of the step or transition. When you switch over to the zoom-in level, the PG is in the OUTPUT mode. You must then press the **CORR** key to switch to the CORRECTION mode.

Ready to start?

A sequence block is displayed at the overview level

To change to the zoom-in level

> Press the **zoom-in** key.

To change to the overview level

> Enter the segment with the **enter key** and return to the overview level, or
> Abort segment processing with the **break key** and return to the overview level.

Steps and transitions are programmed at the **zoom-in level** just as other blocks in STEP 5. At the zoom-in level, a step or a transition is like a separate block. It can have segments added and can contain comments etc.
A **?** no longer appears to the left of steps and transitions at the overview level once their segments are programmed.

At the zoom-in level you can use the keys **+ (+1) or - (-1)** or **roll screen up/down** to jump to the previous or next step/transition without returning to the overview level. When the prompt *change step/transition?* appears, you can change using the **enter key**. If you do not want to change, press the **break key**.

2.3.4 Comments

At the **overview level** you can input comments for each step and each transition. Each comment can be up to 32 characters long. This comment corresponds to the segment title in a STEP 5 block.
With the **zoom-in function**, not only comments for a step or transition can be entered, but also segment comments.
In the statement list, you can also enter statement comments.

Ready to start?
Presets COMMENTS: YES

Comments at the overview level

> Position the cursor on the step or transition at the overview level.
> Press the **COM** key.
The cursor is now positioned in the comment input field.
> Type in the comment.

To enter the comment
> Press the **return** key.
To discard the comment
> Press the **break** key.

Comments at the zoom-in level

Comments for a step or transition

If you press the **COM** key once at the zoom-in level, you jump to the comment field where you can enter a segment title.

☞ The segment title of the first segment of a step or transition is the same as the comment at the overview level.

Statement comments

In STL, you can enter statement comments of up to 32 characters in length.

> Select the segment of the step or transition.
> Position the cursor on the statement and then position the cursor in the field for the statement comment using the **double arrow key right**.
> Input or modify the statement comment.
> Complete the statement comment with the **return** key.

☞ Segment and statement comments can also be entered in SC comment blocks.

Segment comments

You can enter a segment comment for each step or transition. If a step or transition consists of several segments, then only one segment comment is possible per step or transition. It is advisable to enter the segment comment only in the first segment.

The cursor is located within the segment.

> Press the **COM** key twice.
> Type in the segment comment and complete the input with the **return** key.

To enter the comment and return to the segment

> Press the **enter** key.

To discard the comment and return to the segment

> Press the **break** key.

2.3.5 Correction

In the output mode, the **CORR** key switches over to the CORRECTION mode.

At the overview level

When an SB is displayed at the overview level, the sequencer and the following softkey menu appear:

F1	F2	F3	F4	F5	F6	F7	F8
				TM/TW	SEARCH	ID SCREEN	

To modify steps, transitions, branches, jumps etc.

> Press the **CORR** key.

The first softkey level is displayed as follows:

F1	F2	F3	F4	F5	F6	F7	F8
STEP	OPEN	CLOSE	JUMP	TM/TW	SEAR CH	ID SCREEN	SPECIAL

You can now make changes at the overview level.

At the zoom-in level

At this level, the **CORR** key also switches over to the CORRECTION mode.
The softkey menu is then displayed as follows:

F1	F2	F3	F4	F5	F6	F7	F8
DISP SYMB	REFERENCE	SEARCH		ADDRESSES	LIB NO.	⟶ LAD	

> Press the **CORR** key.
Instead of the softkey menu, the command line now appears. You can now make changes in the segment.

☞ **LAD/CSF**
If you position the cursor on an operand, the assignment of absolute and symbolic operands and the operand comment are also displayed. This assumes that you selected SYMBOLS: YES in the presets screen form.

2.3.6 Deleting

You can use the **delete** key in the CORRECTION mode both at the overview and at the zoom-in level.

At the overview level

Delete step/transition, branch or jump

Depending on the cursor position, you can delete a step/transition pair or a transition/step pair. If you position the cursor on the ends of branches, these are deleted. Jumps can be deleted in the same way. You can use the **break** key to "undo" the delete function.

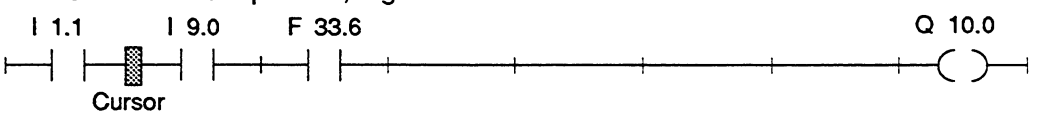
> Position the cursor on the step or transition.

> Press the **delete** key.

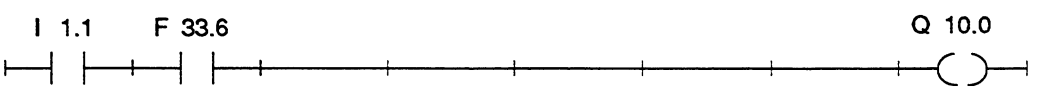
At the zoom-in level

LAD/CSF

Delete at the cursor position, e.g.



> Press the **delete** key.
Contact I 9.0 is deleted.



STL

Delete statements or part of a statement at the cursor position, e.g. you want to delete the statement A I 9.0.

```
: A  I  1.1                          : A  I  1.1
A  I  9.0  > press delete key.   ——→  : A  I  33.3
: A  I  33.3
```

To restore the deleted statement:

> Press the **break** key.

Programming at the Overview Level

3 Programming at the Overview Level

At the overview level, you program your sequencer graphically.

Ready to start?

You selected GRAPH 5 in the select package screen form and selected the presets. Once you have entered the presets by pressing **F6** (ENTER) you can select the input function as follows:

> Press **F1** (INPUT).

> Press **F2** (GRAPH5) to select GRAPH 5 input.

Fill in the command line, e.g.:

INPUT GRAPH5 DEVICE : FD BLOCK : SBn

After you have entered the device (PG, PC (for programmable controller) or FD) and the sequence block, the sequence identification screen form appears as soon as you press the **enter** key.

☞ Only sequence blocks ≥ SB 10 are permitted. SB 0 to SB 9 and DB 0 to DB 9 are used by SIMATIC S5 (PLC, interface modules etc.) and are not available for other purposes. Blocks from SB 10 to SB 255 are permitted.

> With **F1** (TIME BASE) you can enter the timer base and, if required, with **F2** (LIB NO) the library number. Complete your input with the **enter** key.

> Press **F3** (SELECT FB) to select the FBs for the standard programs:

FB 70/71 for linear/simultaneous sequences: STANDARD VERSION
FB 72 for linear/simultaneous sequences: FAST VERSION
FB 73 for linear sequences: FAST VERSION
FB 78 for GRAPH 5-EDDI (this function is described in the GRAPH 5-EDDI manual).

☞ A sequence block created with STEP 5, **F1** (INPUT), **F1** (BLOCK) is not identical with a sequence block created in GRAPH 5 **F1** (INPUT), **F2** (GRAPH 5). If you want to create a GRAPH 5 sequence block with the same block number as an already existing STEP 5 sequence block in the same program file, rename or delete the STEP 5 sequence block.

3.1 Minimum Sequencer and Softkey Levels

After you have entered the data in the sequence identification screen form by pressing **F6** (ENTER) or the **enter key**, a minimum sequencer and the first softkey menu are displayed on the screen. You can expand the sequencer using the cursor and function keys. Initially, softkey level 1 is displayed. To exit level 1 and reach level 2, press **F8** (SPECIAL).

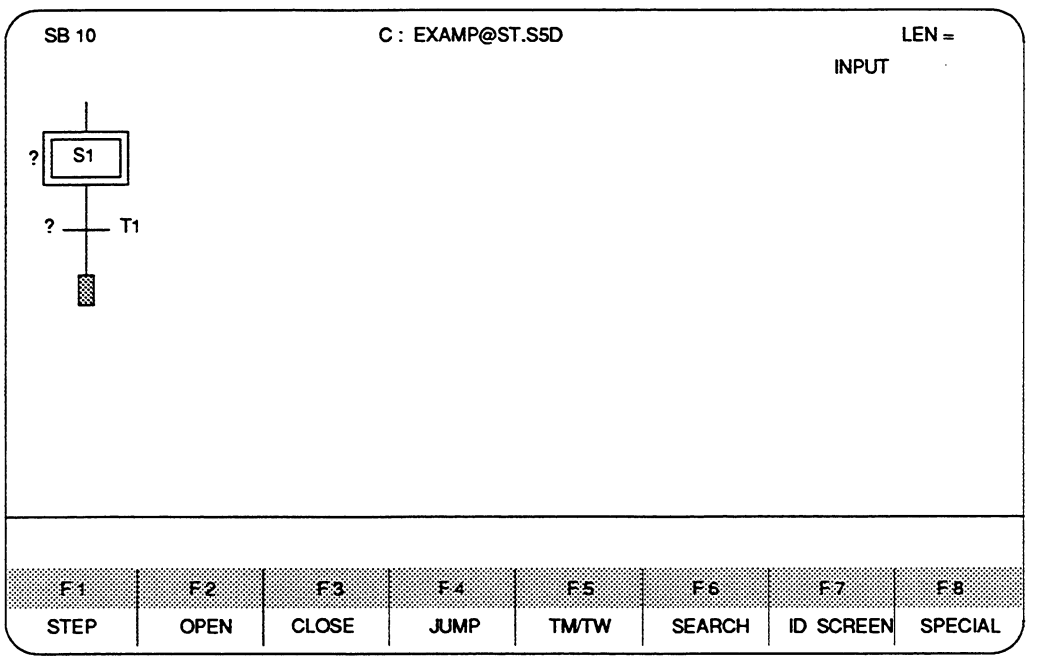


Fig. 3.1 Minimum structure of a sequencer at the overview level

☞ The question marks in the display mean that the step (transition) has not yet been programmed at the zoom-in level.

Softkey Level 1:

F1	F2	F3	F4	F5	F6	F7	F8
STEP	OPEN	CLOSE	JUMP	TM/TW	SEARCH	ID SCREEN	SPECIAL

F1 (STEP) Add a step/transition pair to the end of a sequencer or branch.
Insert a step/transition pair following a transition.
Insert a transition/step pair following a step.

F2 (OPEN) Open a simultaneous branch if the cursor is positioned on a step (only with the setting FB 70/71 and FB 72).
Open an alternative branch if the cursor is positioned on a transition.

F3 (CLOSE) Close a simultaneous or alternative branch.

F4 (JUMP) Complete an alternative branch by jumping to any step.
Complete a sequence by jumping to any step or complete sequence by jumping to step 0.

F5 (TM/TW) Enter the monitoring time (TM) and/or waiting time (TW) or a step.

F6 (SEARCH) Search for a step or transition in the overview display.

F7 (ID SCREEN) Display the sequence identification screen form;
to return to the overview level:
F6 (ENTER) or **enter key** (enter modifications) or **break key** (no modifications entered).

F8 (SPECIAL) Switch over to the 2nd softkey level.

Softkey Level 2:

F1	F2	F3	F4	F5	F6	F7	F8
INITIAL	SELECTIVE	EXCHANGE	COPY				RETURN

F1 (INITIAL) Specify initial step. An FB can contain a maximum of 8 initial steps, but they must be in different simultaneous levels. No monitoring or waiting times (TM/TW) are permitted.

F2 (SELECTIVE) Change selectivity. Change from non-selective step to selective step and vice-versa with **F2**.

F3 (EXCHANGE) Exchange the contents of the step or transition (zoom-in) marked by the cursor with the contents of the required zoom-in. Segment comments, monitoring and waiting times (TM/TW) comments and the step characteristics INITIAL/SELECTIVE are also exchanged.

F4 (COPY) Copy the contents of the step or transition (zoom-in) on which the cursor is positioned into the target step or transition. Segment comments, monitoring and waiting times (TM/TW), comments and the step characteristics INITIAL/SELECTIVE are also copied.

F8 (RETURN) Return to the 1st softkey level.

3.2 Programming a Sequencer

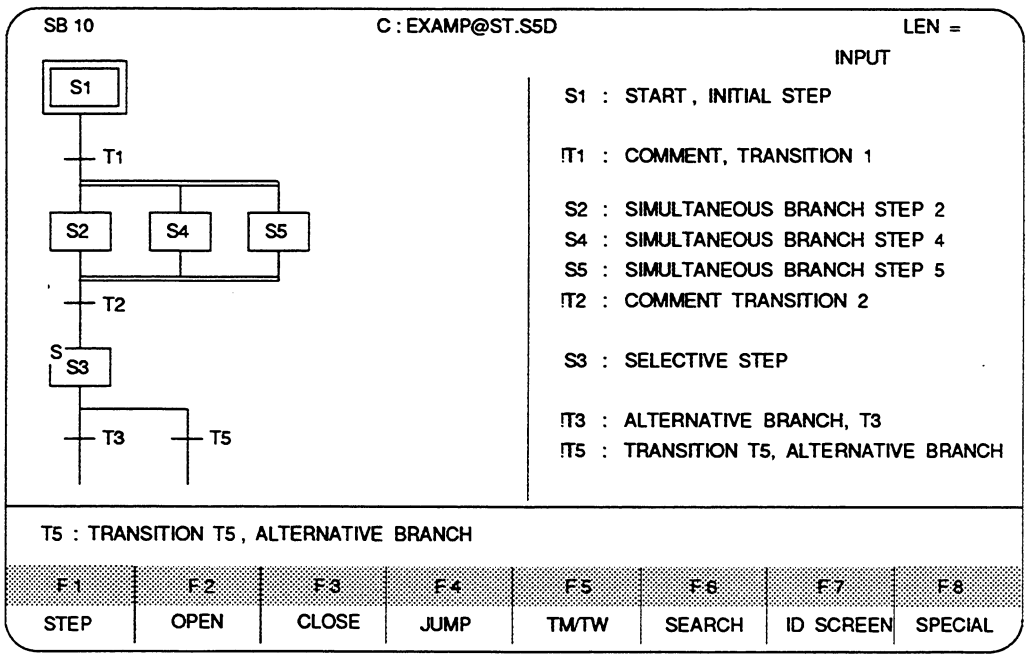


Fig. 3.2 Example of a sequencer at the overview level

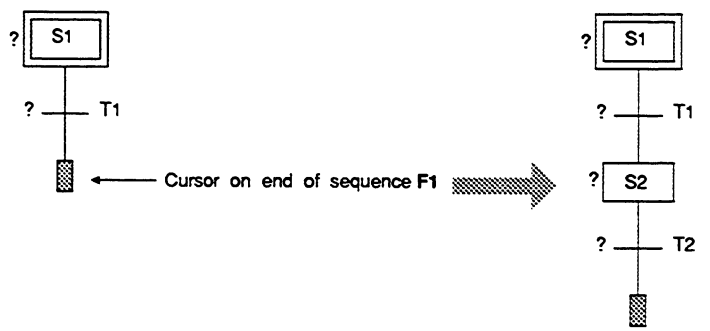
3.2.1 Step/Transition

Depending on the cursor position, you can expand the sequencer by one step/transition pair or transition/step pair (max. 127 steps). A transition always follows a step. A step or jump to a step always follows a transition.

To add a step/transition to the end of a sequence or branch

> Position the cursor at the end of the sequence or branch
> Press **F1** (STEP).

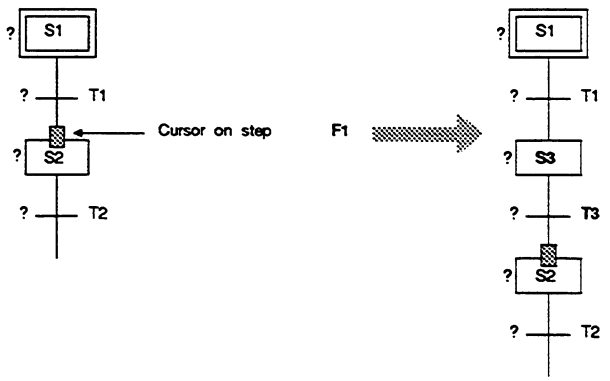
Example:



To insert a step/transition

> Position the cursor on the step.
> Press **F1** (STEP).

Example:

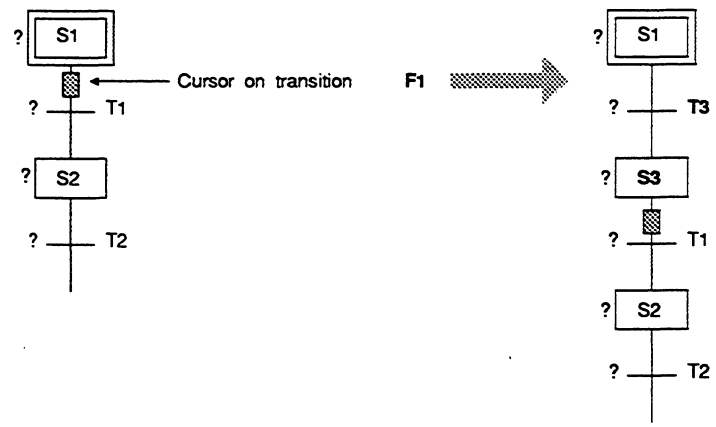


☞ If you want to make use of the maximum number of steps/transitions (127) and then want to delete or reposition individual steps in the sequencer, the PG displays the following message:
memory or internal buffer full.
Reason: the deleted steps are only taken into account when you store the modified sequencer. You must first store the sequencer and then output it again before you can enter the remaining steps.

To insert a transition/step

> Position the cursor on the transition.
> Press **F1** (STEP).

Example:



☞ The steps and transitions are numbered by the GRAPH 5 software when they are stored. The numbering is consecutive from top to bottom. If there are several parallel branches, first the branch on the extreme left will be numbered through from top to bottom, then the second from left branch once again from top to bottom.

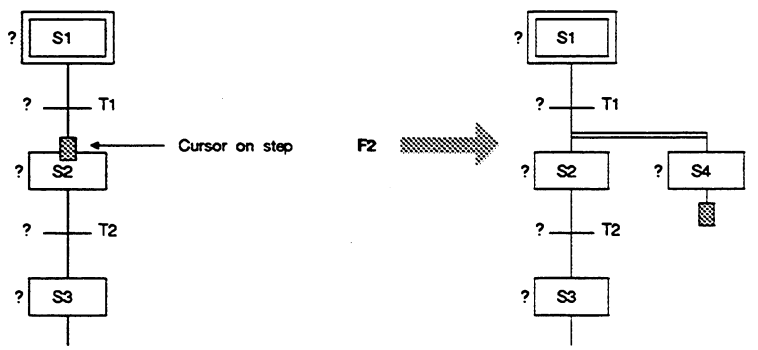
3.2.2 Simultaneous Branch

To open a simultaneous branch, the cursor must be positioned on a step.

To open a simultaneous branch

> Position the cursor on the step.
> Press **F2** (OPEN).

Example:



☞ A simultaneous branch can only be opened with the settings FB 70/71 and FB 72. With the setting FB 73 (linear sequence) the following error message is displayed: *Action not permitted at this point.*

The maximum number of branches and junctions together is 31. If this value is exceeded, the PG displays the following message:
memory or internal buffer full.

Closing a simultaneous branch

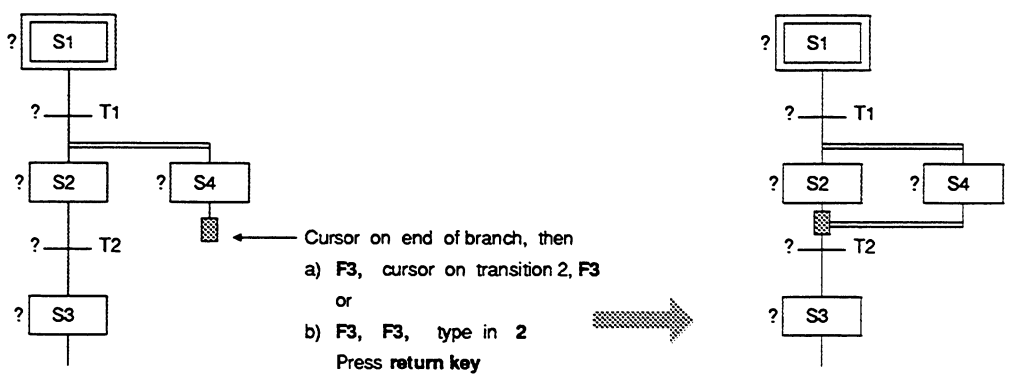
An open simultaneous branch always ends with a step and must therefore be connected to a transition. You can either specify the target transition directly using the cursor or indirectly by typing in the target transition number.

> Position the cursor on the end of the branch.
Press **F3** (CLOSE).

a) Direct:
> Position the cursor on the destination transition.
> Press **F3** (CLOSE).

b) Indirect:
> Press **F3** (CLOSE).
> Type in the number of the destination transition.
> Press the **return** key.

Example:



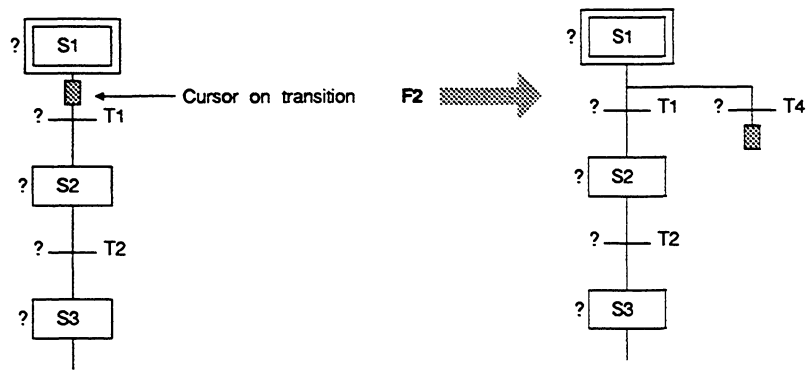
3.2.3 Alternative Branch

Before you open an alternative branch, you must position the cursor on a transition.

To open an alternative branch

> Position the cursor on the transition.
> Press **F2** (OPEN).

Example:



Closing an alternative branch

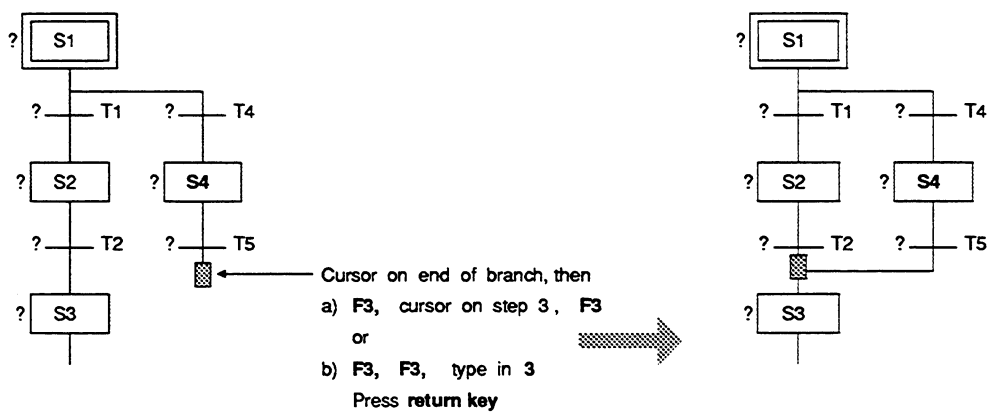
An open alternative branch always ends with a transition and must therefore be connected to a step. You can specify the target step either directly using the cursor or indirectly by typing in the target step number.

> Position the cursor on the end of the branch.
> Press **F3** (CLOSE).

a) Direct:
> Position the cursor on the destination step.
> Press **F3** (CLOSE).

b) Indirect:
> Press **F3** (CLOSE).
> Type in the number of the destination step.
> Press the **return** key.

Example:



☞ At least one step must be located between the opening and closing of a branch.

3.2.4 Jump from the End of the Branch or Sequence to a Step

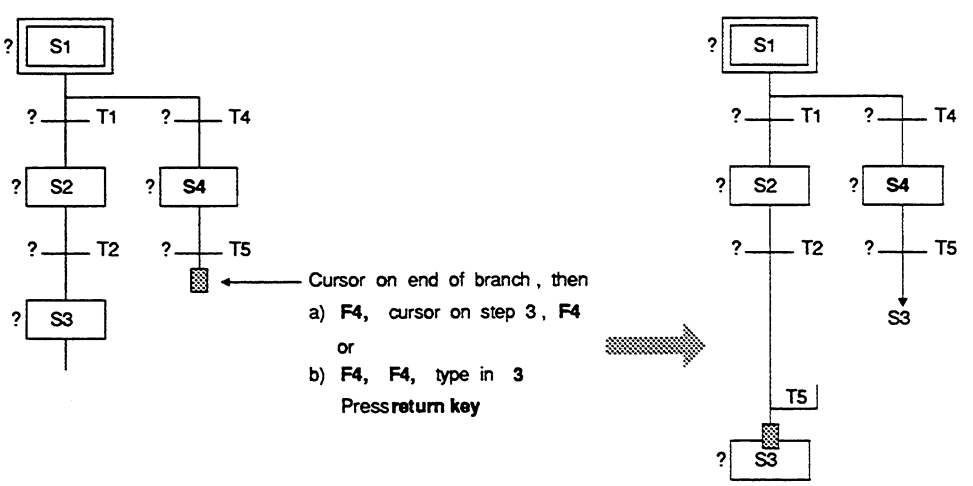
A sequence or open alternative branches can be closed by a jump. The jump can be made to any step in the sequence. Jumps can also be made to S0. This means that the sequencer is terminated at this point. You can specify the target step either directly using the cursor or indirectly by typing in the target step number.

> Position the cursor on the end of the branch or sequence.
> Press **F4** (JUMP).

a) Direct:
> Position the cursor on the destination step.
> Press **F4** (JUMP).

b) Indirect:
> Press **F4** (JUMP).
> Type in the number of the target step.
> Press the **return** key.

Example:



3.2.5 End of a Sequence

The sequence is completed by a jump to any step or to step 0 (S0).

Step 0 means that the sequencer is terminated at this point.

3.2.6 Inputting Monitoring and Waiting Times

You can assign a monitoring and/or waiting time to every step except for initial steps.

Waiting time
The waiting time (TW) is the minimum time a step remains enabled even if the follow-on transition is already satisfied before this time elapses. The follow-on step becomes active at the earliest when the waiting time TW has elapsed.

Monitoring time
The step enable conditions for the next step must be satisfied within the preset monitoring time (TM). If the follow-on step does not become active within TM, a timeout message is displayed.

Possible inputs: KT..., IW, QW, FW, PW, OW (DW not allowed!).

> Position the cursor on the step.
> Press **F5** (TM/TW).
> Type in the value for TM.
> Press the **double arrow key right**.
> Type in the value for TW.
> Press the **return** key.

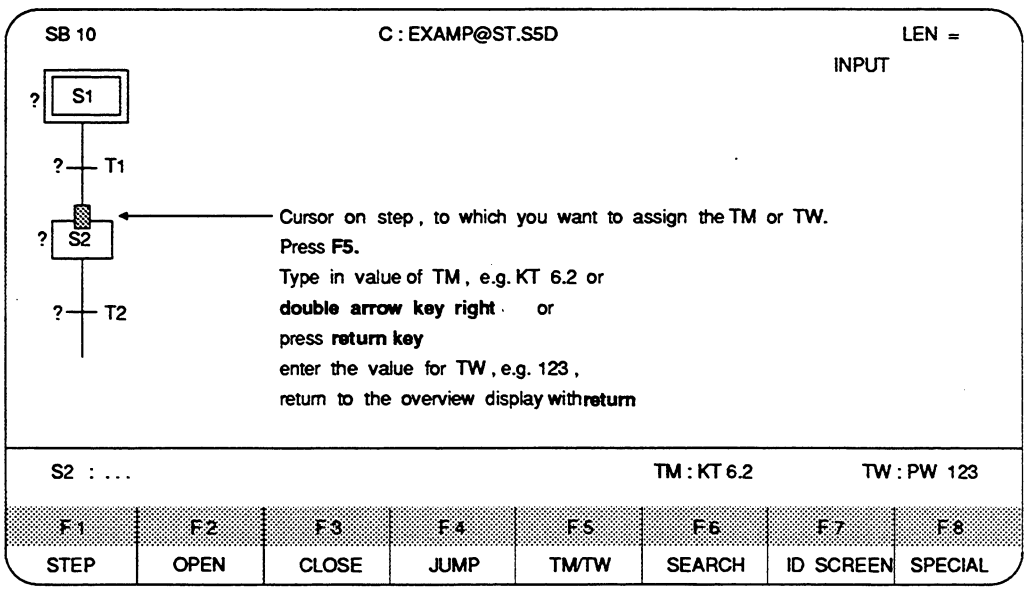
To move from input field to input field

> Press the **double arrow key right** or **left**.

To enter the value:

> Press the **return** key.

Example:



3.2.7 Searching for a Step or Transition in the Overview Display

Using the SEARCH function, you can position the cursor directly on the required step or transition.

> Press **F6** (SEARCH).
> Type in **Sn** or **Tn**.
> Press the **return** key.

3.2.8 Displaying the Sequence Identification Screen Form

If you press **F7** (ID SCREEN) you obtain the sequence identification screen form.
> Press **F7** (ID SCREEN).

To return from the ID screen form to the input level without entering changes:
> Press the **break** key and the **enter** key.

To return from the ID screen form to the input level and enter the modifications:
> Press the **enter** key or **F6** (ENTER).

3.2.9 Calling the 2nd Softkey Level

To reach the 2nd softkey level from the 1st, press **F8** (SPECIAL).

To return to the 1st softkey level from the 2nd, press **F8** (RETURN).

2nd level

F1	F2	F3	F4	F5	F6	F7	F8
INITIAL	SELECTIVE	EXCHANGE	COPY				RETURN

3.2.10 Initial Step

An initial step is activated unconditionally when the sequencer is started. The INITIAL function is used to specify the initial steps. A maximum of 8 initial steps can be defined. These steps must be located in different simultaneous branches. You cannot assign monitoring times (TM) or waiting times (TW) to initial steps. A step becomes an initial step and an initial step becomes a normal step as explained below.
Change from the 1st softkey level to the 2nd by pressing **F8** (SPECIAL).
Position the cursor on the required step (at the 1st or 2nd level). By pressing **F1** (INITIAL) you define a step as an initial step or change an initial step back to a normal step. Return to the 1st level by pressing **F8** (RETURN).

To change from the 1st softkey level to the 2nd softkey level

> Press **F8** (SPECIAL).
> Position the cursor.
> Press **F1** (INITIAL).

To return to the 1st softkey level

> Press **F8** (RETURN).

3.2.11 Selective Step

This function changes the selectivity of a step. A selective step is only processed in the cycle in which the appropriate step flag is set. Otherwise the selective step is skipped. The difference between a normal step and a selective step is as follows:

Normal step
All steps in the sequence are run through cyclically. If a step is not active, the actions are not executed.

Selective step
If a selective step is not active, the action part is skipped using a jump command at the beginning of the step.

☞ If a selective step is not active, interlocks will also be skipped!

A step becomes a selective step and a selective step becomes a normal step as follows:

To change from the 1st softkey level to the 2nd softkey level

> Press **F8** (SPECIAL).
> Position the cursor on the step.
> Press **F2** (SELECTIVE).
  By pressing **F2** (SELECTIVE) you can define a step as a selective step or a selective step as a normal step and vice-versa.

To return to the 1st softkey level

> Press **F8** (RETURN).

3.2.12 Exchanging Contents of Steps or Transitions at the Zoom-in Level

With this function you can exchange the contents of two steps or two transitions at the zoom-in level. Comments (segment titles, segment and statement comments), monitoring and waiting times (TM/TW) and the step characteristics INITIAL/SELECTIVE are also exchanged.

First, position the cursor on one of the two steps or transitions. Then press **F3** (EXCHANGE) and the step or transition is marked (invisibly). After this, specify the second step (target step) or second transition (target transition).

You can specify the target step or transition either directly using the cursor or indirectly by typing in the target step or transition number. You cannot exchange a step with a transition or vice-versa.

To change from the 1st softkey level to the 2nd softkey level

> Press **F8** (SPECIAL).
> Position the cursor on the step or transition.
> Press **F3** (EXCHANGE).

a) Exchanging directly:

> Position the cursor on the target step or target transition.
> Press **F3** (EXCHANGE).

b) Exchanging indirectly:

> Press **F3** (EXCHANGE).
> Type in the number of the target step or target transition.
> Press the **return** key.

To return to the 1st softkey level

Press **F8** (RETURN).

3.2.13 Copying the Contents of a Step or Transition at the Zoom-in Level

With this function you can copy the contents of a step or transition at the zoom-in level.

Comments (segment titles, segment and statement comments), monitoring and waiting times (TM/TW) and the step characteristics INITIAL/SELECTIVE are copied.

First position the cursor on the step or transition to the copied. Then press **F4** (COPY), the step or transition is marked (invisibly). Then specify the target step or transition to which you want to copy the content.

You can specify the target step or transition directly using the cursor or indirectly by typing in the target step or transition number. You cannot copy from a step to a transition and vice-versa.

To change from the first softkey level to the second softkey level

> Press **F8** (SPECIAL).
> Position the cursor on the step or transition.
> Press **F4** (COPY).

a) To copy directly:

> Position the cursor on the target step or transition.
> Press **F4** (COPY).

b) To copy indirectly:

> Press **F4** (COPY).
> Type in the number of the target step or transition.
> Press the **return** key.

To return to the 1st softkey level

Press **F8** (RETURN).

3.2.14 Changing from the 2nd to 1st Softkey Level

If you press **F8** (RETURN) at the 2nd softkey level, you return to the 1st softkey level.

Level 2 :

F1	F2	F3	F4	F5	F6	F7	F8
INITIAL	SELECTIVE	EXCHANGE	COPY				RETURN

> Press **F8** (RETURN).

Level 1 :

F1	F2	F3	F4	F5	F6	F7	F8
STEP	OPEN	CLOSE	JUMP	TM/TW	SEARCH	ID SCREEN	SPECIAL

3.2.15 Completing the Input

To store the block

> Press the **enter** key.
Input is completed and the block is stored in the preset program file.

To abort input without storing the block

> Press the **break** key.
The PG displays the following message: *Abort! Destroy SB in PG?*

Yes:

> Press the **enter** key.
The ID screen form is displayed.

> Press the **break** key.
The PG displays the message: *Abort!*

> Press the **enter** key.
The PG displays the SELECT FUNCTION screen form.

No:

> Press the **break** key.
The PG returns to the overview level.

Programming at the Zoom-in Level

4 Programming at the Zoom-in Level

You program the contents of the steps and transitions at the zoom-in level. Programming is in the STEP 5 programming language with the methods of representation LAD, CSF, STL. Each step and each transition can contain one or more segments. Programming in STEP 5 is described in the manual for the STEP 5 basic package.
During both input and output of a sequence block, you can switch from the overview level to the zoom-in level by positioning the cursor on the step or transition and pressing the **zoom-in** key.
The segment is displayed on the screen in the OUTPUT mode.

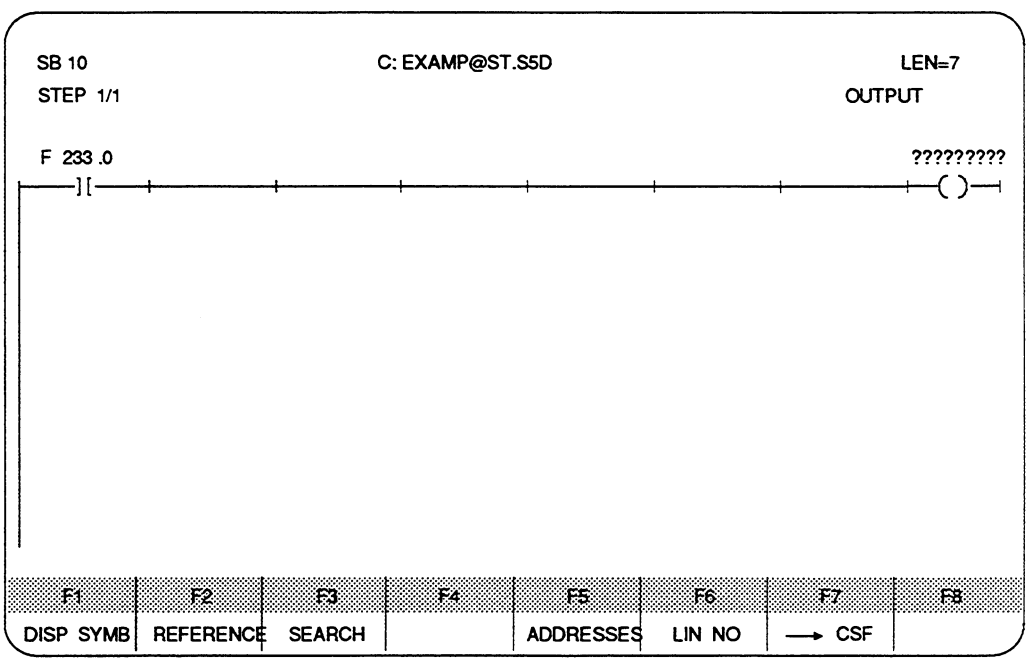


Fig. 4.1 Example of screen output at the zoom-in level, representation LAD, mode OUTPUT

☞ The question marks displayed mean that the step (transition) has not yet been programmed at the zoom-in level.

4.1 Significance of Flag 233.0

Flag 233.0 in steps

Flag 233.0 is used as a substitute for the enable signal of the displayed step. It can be scanned at any point and as often as required within the zoom-in of a step, however, it must not be changed. This enable signal has the value 1 when the corresponding step is active in automatic operation.
In the PLC, it is not this flag that is stored, but rather a different flag from the area starting at FY 234. The flag bit stored here is different for every step; in the PG, however, the same pseudo flag F 233.0 is displayed. You do not need to program the assignment of the step number to the flag bit, you simply need to know that the displayed flag value is always assigned to the displayed step.

Flag 233.0 in transitions

At the zoom-in level, the transition indicates the user section of the step enabling conditions. The result of logic operation (RLO) obtained is not the definitive step enabling condition, and may still be corrected by GRAPH 5 (waiting time not yet elapsed, UQIT, T+1 signal does not exist for conditional step control etc.).

STL at the zoom-in level

Only the user part of the step enabling condition is displayed. No flags from the flag area (F 200.0 to F 255.7) occupied by the GRAPH 5 software can be used (especially not F 233.0).

The RLO valid at :BE can, if necessary, be updated by the GRAPH 5 software and then used to continue the sequence.

LAD/CSF at the zoom-in level

LAD and CSF segments must be completed with an assignment (exception block call). For this reason, a non-existent assignment in the PG must be simulated for the display on the screen. The flag 233.0 is intended to show that the signal to activate the next step(s) will be supplied. This flag display is generated automatically by the GRAPH 5 software, there is no command sent from the PLC. There is therefore also no status display for this assignment.

☞ Flag 233.0 must not be programmed anywhere in transitions at the zoom-in level. If it is required for the graphics, it will be generated automatically.

4.2 Programming Segments in Steps/Transitions

At the zoom-in level of a step or transition, statements can be programmed in LAD, CSF and STL just as in STEP 5. This also applies to statement comments, segment titles and segment comments. At the zoom-in level, you must switch over from the output mode to the correction mode by pressing the **CORR** key (CORRECTION display). If you press the **enter** key, you return to the output mode.

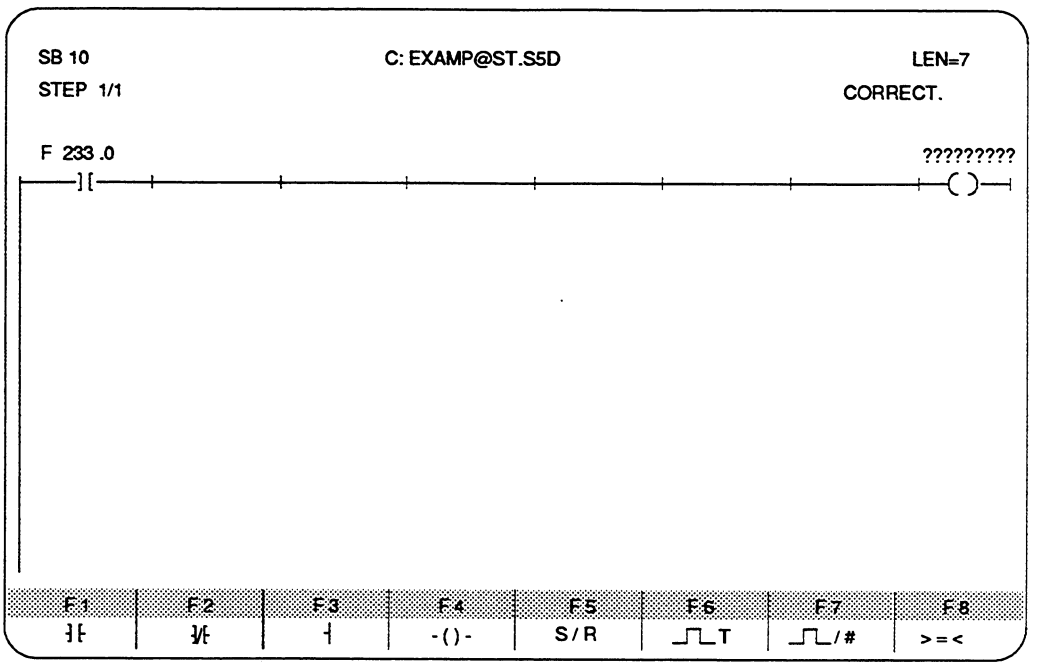


Fig. 4.2 Example of screen output at the zoom-in level, representation LAD, mode CORRECTION

☞ You can only input one segment comment per step/transition. Data blocks must not be programmed, they can only be called!

Segments must not be completed with BEC or BEU!

4.3 Inserting, Appending, Deleting Segments

You can insert, append or delete segments at the zoom-in level of a step or transition in LAD, CSF and STL just as in STEP 5. Select the segment in the output mode at the zoom-in level.

☞ Each time an SB is modified, the user data block must be generated with **F5** (DBGEN) and if it exists, the re-translation list #SBRL with **F6** (RLGEN).

Inserting a segment in a step or transition

Ready to start?

The PG is in the OUTPUT mode.

To insert a segment

> Select the segment **before** which you want to insert the segment.

> Press the **insert segment** key.
A segment is inserted, the PG is in the insert mode and the segment can be programmed as usual.

> Enter the inserted segment by pressing the **enter** key.
The PG returns to the OUTPUT mode.

Appending a segment to a step or transition

Ready to start?

The PG is in the OUTPUT mode.

To append a segment

> Select the last segment.

> Press the **segment end (***)** key.
A segment is appended, the PG is in the insert mode and you can input the segment as usual.

> Enter the appended segment by pressing the **enter** key.
The PG returns to the OUTPUT mode.

Deleting a segment in a step or transition

Ready to start?

The PG is in the OUTPUT mode.

To delete a segment

> Select the segment to be deleted.

> Press the **delete segment** key.
The PG prompts: *Delete?*

Yes:

> Press the **enter** key.
The segment is deleted.

No:

> Press the **break** key.
The segment is not deleted.

The PG is in the OUTPUT mode.

Activities after Programming

5 Activities after Programming

This chapter describes the functions you can execute after you have programmed a sequencer, e.g.:

- outputting a sequence block (sequencer) on the screen or printer,
- correcting a sequencer,
- generating diagnosis and user data blocks and
- generating a re-translation list (#RLGEN). This helps to speed up the screen display of an SB.

5.1 Outputting a Sequence Block

When you output a sequence block, you can handle it just like a block in LAD, CSF, STL.

Ready to start?

You have selected the GRAPH 5 package and entered the presets.
The PG is displaying the SELECT FUNCTION screen form.

F1	F2	F3	F4	F5	F6	F7	F8
INPUT	OUTPUT	TEST	PC FCT	PC INFO	PRESETS	AUX FCT	RETURN

> Press **F2** (OUTPUT).

F1	F2	F3	F4	F5	F6	F7	F8
	BLOCK		SCR FORM				RETURN

> Press **F2** (BLOCK).

OUTPUT DEVICE: **a)** BLOCK: **b)** SEARCH: **c)** PTR: **d)**

> Fill in the command line.

a) = PG, PC (programmable controller), FD (preset program file)
b) = SBn, n = block number
c) = Step/transition number (Sn/Tn)
d) = * : Standard print
1 : Normal print
2 : Condensed print (with filing margin)
3 : Super-condensed print (DIN A4 only)
blank: No printout

> Press the **enter** key.
The sequence is displayed at the overview level. The section displayed contains the step/transition specified as the SEARCH key.

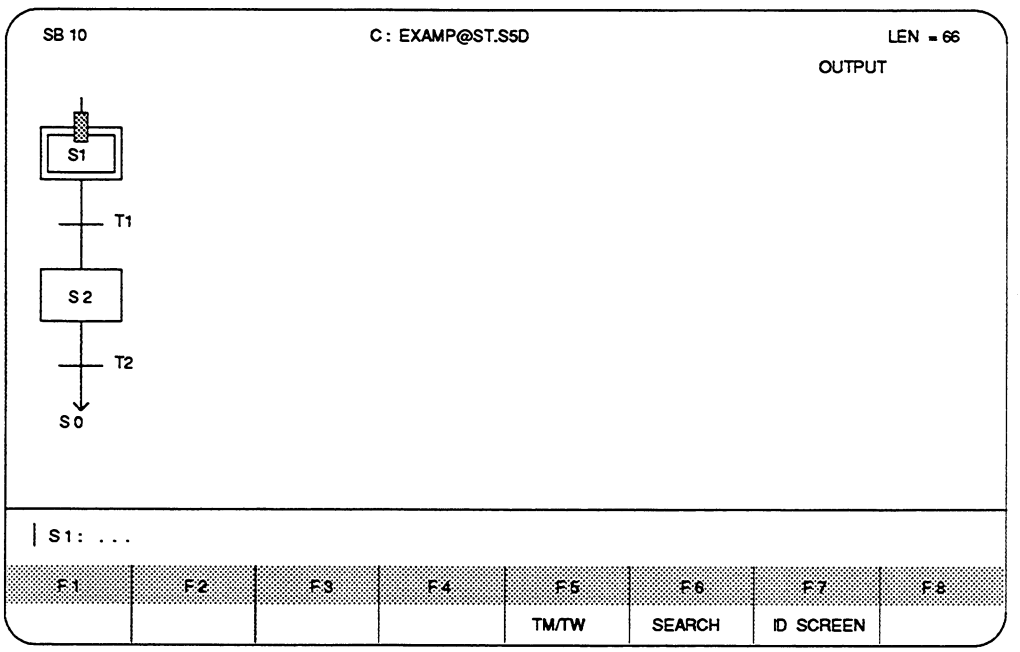


Fig. 5.1 Example of the screen display at the overview level

In the output mode, you can use the keys **F5** (TM/TW), **F6** (SEARCH) and **F7** (ID SCREEN). Their functions correspond to those in the input mode and the program can be handled in the same way as in the input mode.

☞ You can only search for steps and transitions at the overview level (by specifying the step or transition number: Sn or Tn). All other search keys are ignored.

When you switch over to the **zoom-in level** the following softkey menu is displayed in the OUTPUT mode.

F1	F2	F3	F4	F5	F6	F7	F8
DISP SYMB	REFERENCE	SEARCH		ADDRESSES	LIB NO	⟶ LAD	

F1 (DISP SYMB) Displays the symbolic operands and operand comments. You return to the LAD, CSF or STL segment by pressing the **enter** or **break** key.

F2 (REFERENCE) Generates reference list; outputs cross references for individual operands, jumps to a block, jumps back to starting block.

F3 (SEARCH) Searches for the segment within the step or transition in which the search key occurs.

F5 (ADDRESSES) In STL, addresses can be displayed as words or bytes.

F6 (LIB NO) Not relevant, you can input or modify the library number at the overview level with **F7** (ID SCREEN), **F2** (LIB NO).

F7 (--> LAD) Changes the method of representation.

5.2 Correcting Sequence Blocks.

At both the overview and zoom-in levels you change from the output to the correction mode by pressing the **CORR** key.
All the input softkey functions are available.

☞ After **any** change in an SB, you must regenerate the user data block with **F5** (DBGEN) and, if it exists, the re-translation list #SBRL with **F6** (RLGEN).

5.3 Generating the Diagnosis DB and the User DB

Each sequencer requires a user DB with a number identical to that of the sequence block. For diagnosis of the sequencers in the PLC memory, a diagnosis DB is also required that is accessed when you call the diagnosis function. The diagnosis DB is the same for all sequencers in the PLC. These DBs are generated with the DBGEN function, **F5**.
When you generate the DBs, they are generated for **all** SB-DB pairs on the FD or in the PLC. You cannot select a group of DBs to be generated.
As a check, the occupied timer area is also output.
When the sequencers are started, all DBs, including the diagnosis DB, must exist in the PLC memory.

Generating the user DBs and diagnosis DB
Ready to start?
You have selected the GRAPH 5 package and the presets.

The PG is displaying the SELECT FUNCTION screen form.

F1	F2	F3	F4	F5	F6	F7	F8
INPUT	OUTPUT	TEST	PC FCT	PC INFO	PRESETS	AUX FCT	RETURN

> Press **F1** (INPUT).

F1	F2	F3	F4	F5	F6	F7	F8
BLOCK	GRAPH5		SCR FORM	DBGEN	RLGEN		RETURN

> Press **F5** (DBGEN).
The generation of the user DBs and a diagnosis DB is started.

Fill in the command line

DBGEN DEVICE: **FD or PC** DIAGNOSIS–DB: **DBn**

n = DB number, must not be identical to the number of a user DB.
> Press the **enter** key.
The PG displays the list of existing blocks, e.g.

```
SEQUENCE BLOCK      DATA BLOCK      DB ALREADY EXISTS      OCC. TIMER

SB  10              DB  10          NO                     T  1 -  4
SB  11              DB  11          NO                     T  5 - 12
SB  33              DB  33          NO                     T 21 - 26

DBGEN DEVICE: FD  DIAGN. DB:  DB255
```

Fig. 5.1 Example of a block list with SBs and user DBs to generate a diagnosis DB

The DB is generated when you press the **enter** key. DBs you have already generated are marked with YES in the DB ALREADY EXISTS column. If a diagnosis DB already exists, this is not displayed.
If the diagnosis DB and user DBs already exist, the message: *DB with diagnosis DB no. already exists already exists, overwrite DB?*
You can abort the generation by pressing the **break** key or start the generation with the **enter** key. Each existing DB is indicated individually in the message line following which the program jumps to the next SB-DB pair.

5.4 Fast Re-translation of Sequence Blocks

To display long sequence blocks quickly, you can store re-translation information for the SB in a list, the re-translation list #SBRL.nnn. The re-translation list is generated using the PG function **F1** (INPUT), **F6** (RLGEN) and stored in the preset program file. It is not transferred to the PLC memory.
This re-translation list contains an internal version identifier that is also entered in the SB. The version identifier is used to ensure the consistency of the SB and its corresponding re-translation list #SBRL. Each time an SB is modified, the version number is incremented. This means that the corresponding re-translation list #SBRL must also be updated (RLGEN called again).
The SB with the new version identifier must be written back to the source device (FD or PLC).

5.4.1 Generating the Re-translation List

Ready to start?
You have entered the presets, the sequence block is located on FD or in the PLC.

The PG is displaying the SELECT FUNCTION screen form.

F1	F2	F3	F4	F5	F6	F7	F8
INPUT	OUTPUT	TEST	PC FCT	PC INFO	PRESETS	AUX FCT	RETURN

> Press **F1** (INPUT).

F1	F2	F3	F4	F5	F6	F7	F8
BLOCK	GRAPH5		SCR FORM	DBGEN	RLGEN		RETURN

> Press **F6** (RLGEN).
The generation of the block is initiated.

Fill in the command line

RLGEN DEVICE: **FD** or **PC** BLOCK: **SBn**

> **FD** or **PC** = source device on which the SBn is located.
> **n** = SB number
> Press the **enter** key.
> The SB must be written back to the source device with this new version number.
> The following prompt is therefore displayed:
> *SBn already in destination file, overwrite?*
> This must be acknowledged by pressing the **enter** key.
> If a corresponding #SBRL exists, the PG displays the following prompt:
> *#SBRL.nnn already in destination file, overwrite?*
> Acknowledge with the **enter** key.
> The following prompt is displayed:
> *SBn already in destination file, overwrite?*
> Acknowledge with the **enter** key.

5.4.2 Outputting an SB with Fast Re-translation

When an SB is displayed using the PG (OUTPUT, STATUS), the program checks whether or not a re-translation list #SBRL.nnn exists. If it does, the information stored in it will be used for a fast re-translation. Long blocks can be displayed much more quickly using this list.
If the version identifiers of the SB and #SBRL do not match (e.g. after modifying the SB without following it with RLGEN), the re-translation takes place as normal. Following the re-translation, you are prompted to decide whether the re-translation information should be entered in the re-translation list #SBRL:
#SBRL.nnn already in destination file, overwrite?

If you acknowledge the prompt with the **enter** key and complete processing of the SB with the **break** key, the version identifiers are once again the same.
If there is an SBn and a #SBRL.nnn with the same version identifier on the FD and if you output the SBn with a different version identifier, the following prompt is displayed:
#SBRL.nnn already in destination file, overwrite?

To avoid this, you should always transfer the SB to the other device whenever you have modified it.

5.4.3 Deleting the Re-translation List

You can delete a re-translation list #SBRL.nnn in the same way as a DOC file using the keys **F7** (AUX FCT) and **F2** (DELETE) in the preset program file.

5.5 Cross Reference Lists with GRAPH 5

You select the cross reference lists just as for STEP 5 blocks. The information is divided according to steps and transitions. In the TIMERS section of the cross reference list, the times of the steps are shown as assignments in the previous transition.
Some of the information refers to the operation part, this can be ignored.
This information consists of the following:

- DW 10 to DW 17
- SB 0 call
- Flags 230.0 255.7

5.6 Rewiring GRAPH 5 Blocks

Rewiring is described in the STEP 5 manual.

☞ The following must not be rewired:
Flags from on F 200.0 onwards and
the user DBs and diagnosis DBs.

Transferring and Testing the Program

6 Transferring and Testing the Program

Transferring blocks to the PLC or EPROM/EEPROM submodule is described in the STEP 5 manual.

6.1 Transfer to EPROM/EEPROM Submodule

Ready to start?

The EPROM/EEPROM submodule is plugged in to the PG.
You have selected the EPROM/EEPROM package in package selection.
You have entered the presets.

The PG is displaying the SELECT FUNCTION screen form.

F1	F2	F3	F4	F5	F6	F7	F8
BLOW	READ	DELETE	DUPLICATE	E INFO	PRESETS	AUX FCT	RETURN

> Press **F1** (BLOW).

Fill in the command line, e.g.

BLOW EPROM BLOCK: **SB10**

> Press the **enter** key.

☞ Data blocks must not be transferred to the EPROM/EEPROM submodule.

6.2 Transferring to the PLC Memory

You transfer blocks with **F1** (TRANSFER).

Ready to start?
The PLC is online with the PG.
You have selected the GRAPH 5 package in package selection.
You have entered the presets.

The PG is displaying the SELECT FUNCTION screen form.

F1	F2	F3	F4	F5	F6	F7	F8
INPUT	OUTPUT	TEST	PC FCT	PC INFO	PRESETS	AUX FCT	RETURN

> Press **F7** (AUX FCT).

F1	F2	F3	F4	F5	F6	F7	F8
TRANSFER	DELETE	DIR			PRG FILE		RETURN

> Press **F1** (TRANSFER).

Fill in the command line, e.g.

TRANS SOURCE: **FD** BLOCK: **SB10** TO DEST: **PC** BLOCK:

> Press the **enter** key.
The block or blocks are transferred to the PLC memory.

All the blocks required for running the program must be loaded in the PLC memory:

Standard function blocks

- FB 70 for the main sequence (STANDARD VERSION
- FB 71 for the secondary sequence (STANDARD VERSION)
- SB 0 execution block (STANDARD VERSION

or

- FB 72 main sequence (linear/simultaneous sequence: FAST VERSION)
- SB 2 execution block (linear/simultanoues sequence: FAST VERSION
- FB 74 modes (if necessary)

or

- FB 73 main sequence (linear sequence: FAST VERSION
- SB 3 execution block (linear sequence: FAST VERSION)
- FB 74 modes (if necessary)

Sequence and data blocks

- SBn sequence block(s)
- DBn user data block(s)
- DBy diagnosis data block

☞ Only transfer user DBs and the diagnosis DB if you have not already generated them using the PG function **F5** (DBGEN).

All other blocks

OB, FB, PB, DB etc., required to run the sequencer.

6.3 GRAPH 5 Test Facilities

GRAPH 5 sequencers loaded and running in the PLC can be followed online at the PG using the test functions **F3** (TEST). This is described in the PLC test functions in the STEP 5 manual.

Ready to start?

The PLC is online with the PG.
You have selected the GRAPH 5 package in package selection.
You have entered the presets.

The PG is displaying the SELECT FUNCTION screen form.

F1	F2	F3	F4	F5	F6	F7	F8
INPUT	OUTPUT	TEST	PC FCT	PC INFO	PRESETS	AUX FCT	RETURN

> Press **F3** (TEST).
The PG displays the TEST FUNCTIONS screen form.

F1	F2	F3	F4	F5	F6	F7	F8
PROG TEST	END TEST	STATUS			DIAGNOSIS		RETURN

In GRAPH 5, the functions **F3** (STATUS) and **F6** (DIAGNOSIS) are relevant.

6.3.1 Status

The status display indicates the status of the currently active sequencer.

The active steps are marked with a "*" symbol in the overview display. By positioning the cursor on a step or transition in the overview display, and then pressing the **zoom-in** key, you can display the status of the contents of the step or transition. This allows you to check the status of individual inputs and outputs at the zoom-in level.

Status display of the sequencer at the overview level

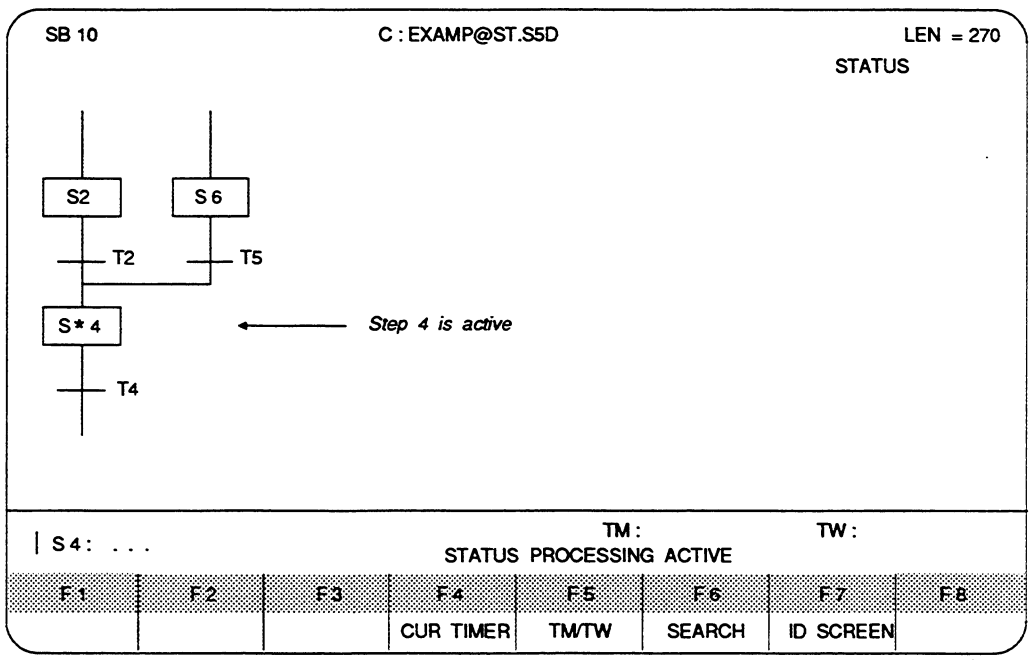


Fig. 6.1 Example of a status display at the overview level

In the status display at the overview level, the functions **F4** (CUR TIMER), **F5** (TM/TW), **F6** (SEARCH) and **F7** (ID SCREEN) are available. The functions **F5** to **F7** are described in the Sections 3.2.6 to 3.2.8.

☞ With **F4** (CUR TIMER) you can follow the monitoring and waiting times of a step.

Example: you want to display the status of SB 10.

Ready to start?

SB 10 is in the PLC memory, the PG and PLC are online.
The PG is displaying the TEST FUNCTIONS screen form.

F1	F2	F3	F4	F5	F6	F7	F8
PROG TEST	END TEST	STATUS			DIAGNOSIS		RETURN

> Press **F3** (STATUS).
> Fill in the command line, e.g.

STATUS BLOCK: **SB10** SEARCH:

The status of the sequencer is displayed at the overview level (Fig. 6.1). Active steps are marked with " * "

Status of the sequencer at the zoom-in level

You can display the status of a step or transition corresponding to the status display of a segment.

Ready to start?
Status display at the overview level.

> Position the cursor on the step or transition.
> Press the **zoom-in** key.

The segment of the selected step or transition is displayed in the STATUS mode.

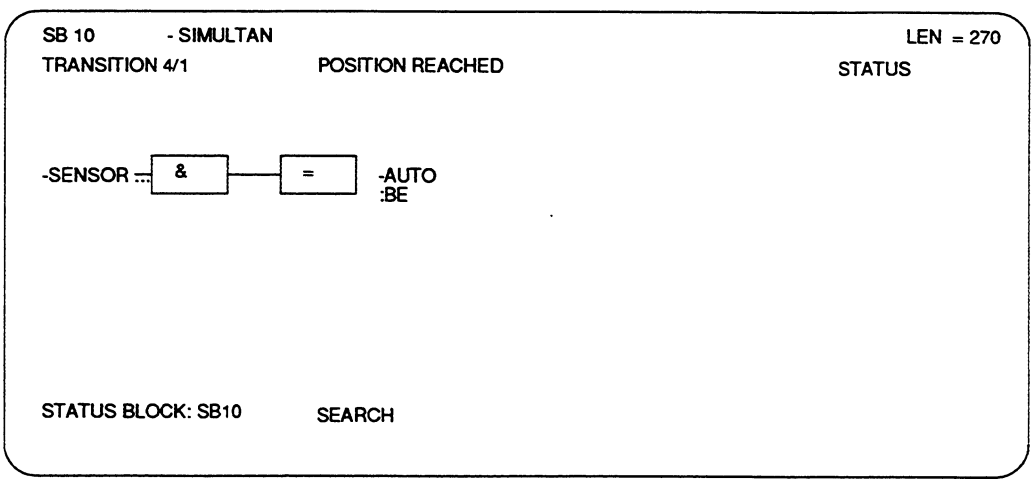


Fig. 6.2 Example of a status display at the zoom-in level

Using the keys **CORR** and **zoom-in** you can obtain the assignments from the assignment list.

Displaying waiting and monitoring times

You can display the waiting and monitoring times of a step in the status display with **F4** (CUR TIMER).

Ready to start?
Status display at the overview level.

> Position the cursor on a step with a monitoring time (TM) and/or waiting time (TW).
> Press **F4** (CUR TIMER).

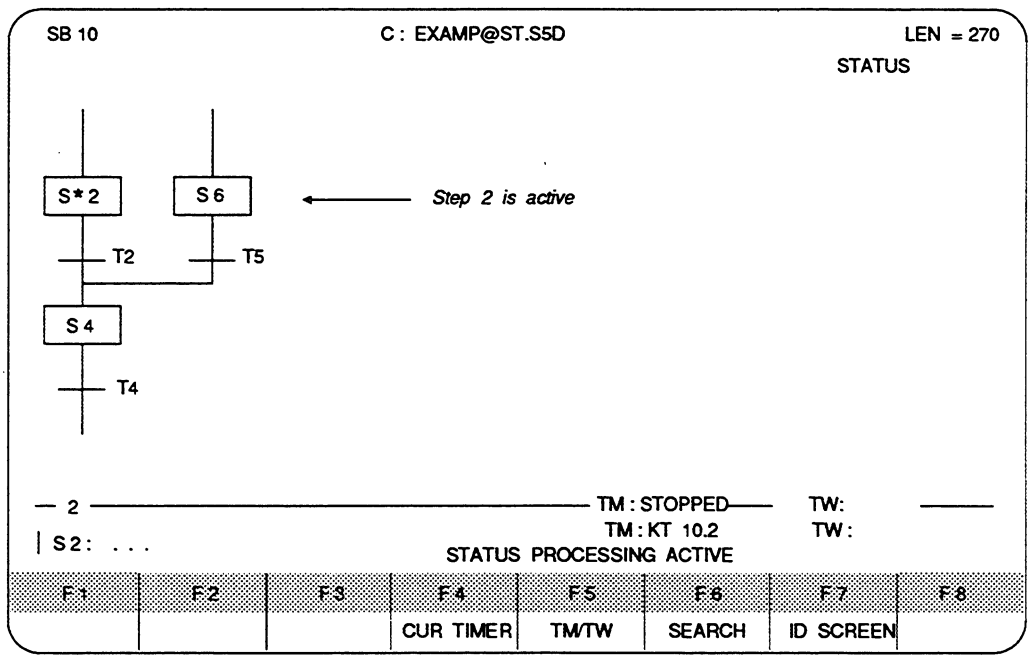


Fig. 6.3 Example of waiting and monitoring time display

If the step is active, the current timer values are displayed. Once the monitoring time has elapsed, the message **STOPPED** is displayed.

6.3.2 Diagnosis

The diagnosis function relates to the diagnosis DB created with DBGEN.
This DB must be located in the PLC memory.
The DIAGNOSIS function is an aid to troubleshooting. You can trace the cause of a timeout right down to the zoom-in level.

Ready to start?

The SB, user DBs and diagnosis DB are in the PLC memory.
The PG and PLC are online.
The PG is displaying the TEST FUNCTIONS screen form.

F1	F2	F3	F4	F5	F6	F7	F8
PROG TEST	END TEST	STATUS			DIAGNOSIS		RETURN

To call the diagnosis function

> Press **F6** (DIAGNOSIS).
> Fill in the command line, e.g.

START DIAGNOSIS GRAPH5 DIAGNOSIS DB: **DBn**

n = number of the diagnosis DB used.

A list of all the sequence blocks loaded in the PLC is displayed. If a timeout occurs, then the message **GROUP TIMEOUT** (there can be timeouts in several sequencers at the same time) will be displayed. Sequence blocks with timeouts in their sequences are marked with **TIMEOUT**.

Example:

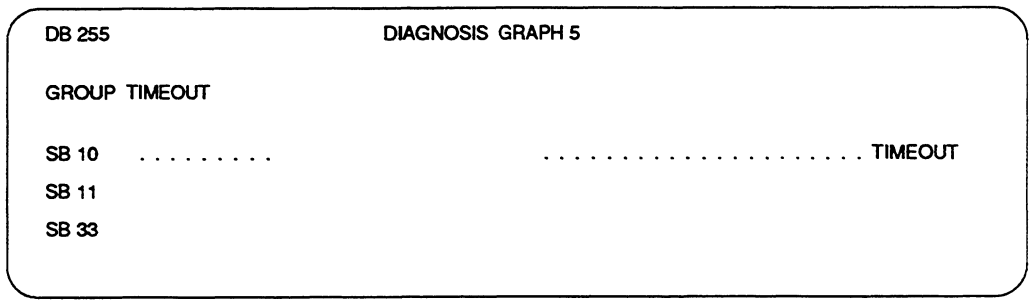


Fig. 6.4 Example of a list of SBs displayed with the DIAGNOSIS function (SB 10 has a timeout)

Diagnosis of the sequencer at the overview level

To display a sequence with a timeout and to find the cause of the error, position the cursor in the list on the appropriate SB. When you press the **zoom-in** key, the status of the sequence is displayed at the overview level. Steps at which the timeout has occurred are displayed inversely.

Ready to start?

The SB list is being displayed by the DIAGNOSIS function.

> Position the cursor on the required SB.

> Press the **zoom-in** key.
The SB is displayed at the overview level.

Diagnosis at the zoom-in level

The contents of a step or transition can be diagnosed.
Position the cursor on the step or transition in the overview display. Press the **zoom-in** key to change to the zoom-in level. The status of the step or transition is displayed.

Example of Planning and Starting Up

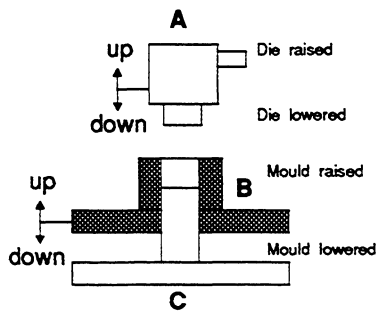
7 Example of Planning and Starting Up

Task

You want to automate a powder press for producing pellets.

Components of the press:

- moveable, upper dye A
- moveable mould B
- fixed, lower dye C
- unit to introduce material
- unit to remove the pellet



The operation runs as follows:

- When mould B and upper dye A are in their upper positions, the material can be introduced.
- Once the material has been introduced, dye A is lowered, compresses the powder in the mould and then returns to its upper position.
- Mould B is lowered until it reaches its lower position, the compressed pellet can now be removed.
- The mould then returns to its starting position and a new cycle can begin.

7.1 Planning the Program

7.1.1 Concept Phase

The operation is studied to determine which steps the sequence can be divided into and what will determine the end of one and the beginning of the next step (when is the transition? ⇒ step enabling condition, transition).

Step/action: introduce material
Transition/condition: material ready?

Step/action: lower dye A
Transition/condition: pressing completed?

Step/action: raise dye A
Transition/condition: dye A raised?

Step/action: lower mould B
Transition/condition: mould B lowered?

Step/action: remove pellet
Transition/condition: removal completed?

Step/action: raise mould B
Transition/condition: mould B raised?

The sequence of the steps and transitions can be represented graphically:

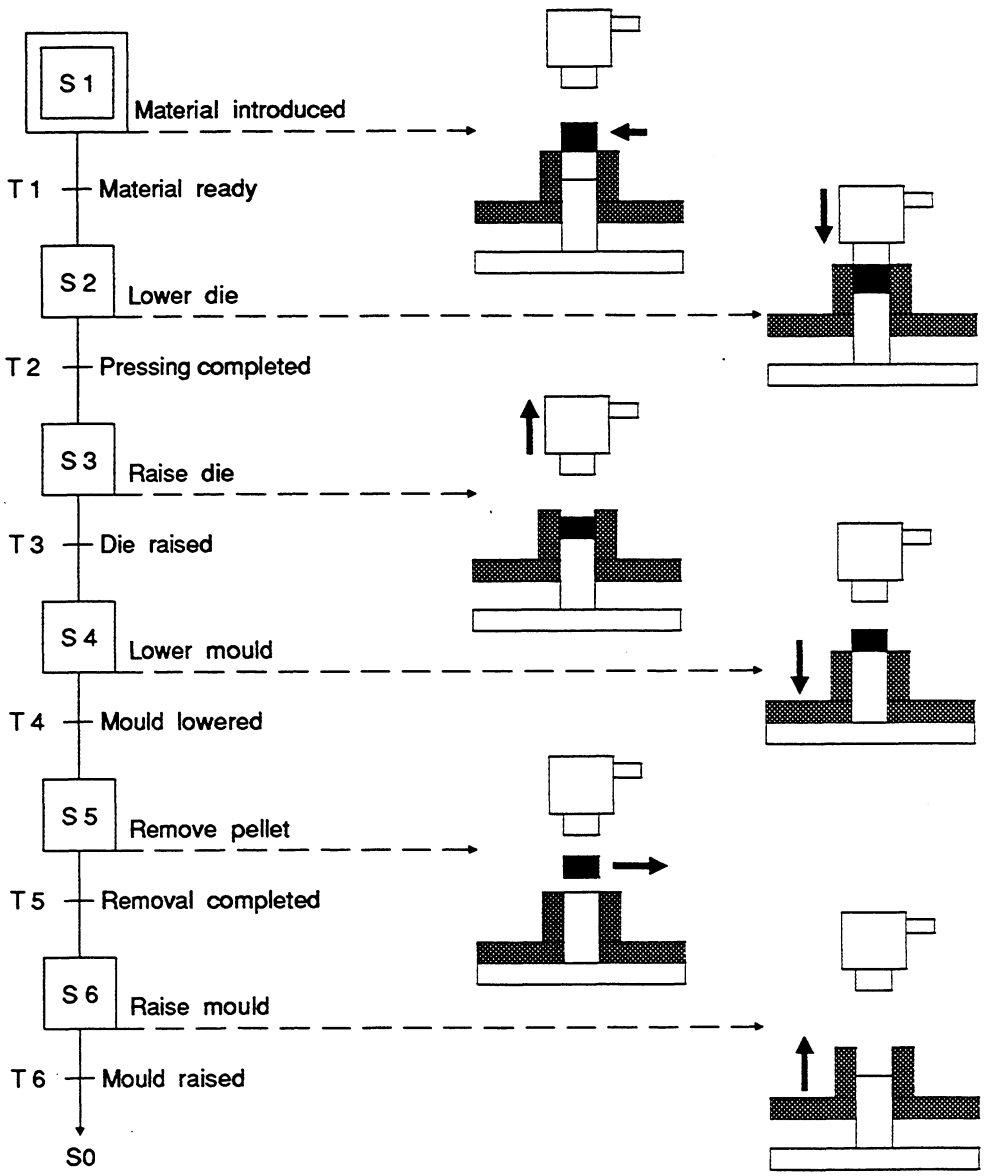


Fig. 7.1 Sequence of steps and step enabling conditions for the powder press example

7.1.2 Planning on the Programmer

Based on the created concept, you can now input the graphical solution to the task as a step-transition structure in GRAPH 5 on the PG.

- Call GRAPH 5 input:
  call the input function with **F1** (INPUT) and **F2** (GRAPH 5)
- Fill in the sequence identification screen form, for example as follows:

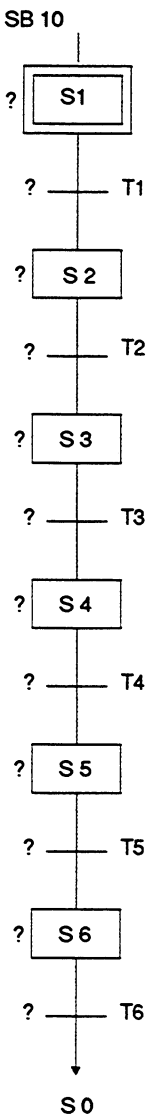
```
SB 10                          C : EXAMP@ST.SSD                    LEN =

SEQUENCE CONTROL - SEQUENCE IDENTIFICATION

FB SEL: FB 70/71 FOR LINEAR / SIMULT. SEQUENCE : STANDARD VERS.

SEQUENCE BLOCK NO   : SB 10

DATA BLOCK OCC.     : DB 10

TIMER BASE          : T 1

                                  FLAG AREA OCC        : F 200.0 - F 255.7

                                  TIMER, COUNTER OCC   : T 0 , C 0

| F1        | F2      | F3         | F4 | F5 | F6    | F7 | F8 |
| TIME BASE | LIB NO  | SELECT FB  |    |    | ENTER |    |    |
```

Fig. 7.2 ID screen form for the powder press example

SEQUENCE BLOCK NO:	SB 10	Specified SB
DATA BLOCK OCC:	DB 10	Number of the user DB for the sequencer. The numbers of the SB and DB are the same.
FLAG AREA OCC:	F 200.0 – F 255.7	This area is fixed within GRAPH 5 and must not be used anywhere else within the SB, outside the SB it can be used as a scratchpad area.
TIMER BASE:	T1	Specifies the first free timer location for the waiting and monitoring times.

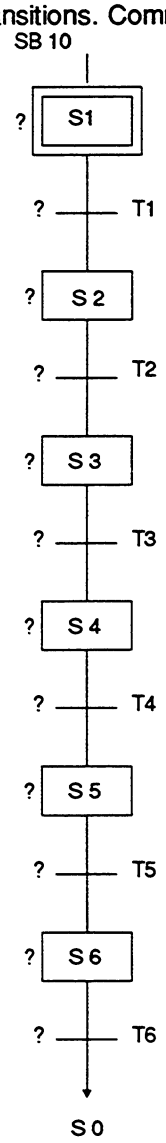
Inputting the sequencer

You input the sequence structure with **F1** to **F4**.



Inputting comments

After you have completed the overview, you can add comments to the steps and transitions. Comments can be entered in both upper and lower case characters.

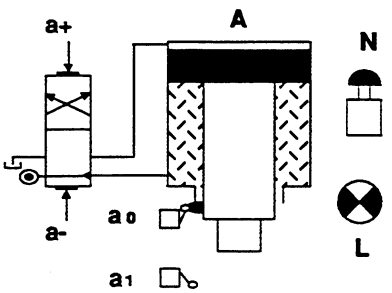


S1	: MATERIAL INTRODUCED
T1	: MATERIAL READY ?
S2	: LOWER DIE A
T2	: PRESSING COMPLETED ?
S3	: RAISE DIE A
T3	: DIE A RAISED ?
S4	: LOWER MOULD
T4	: MOULD LOWERED ?
S5	: REMOVE PELLET
T5	: PELLET REMOVED ?
S6	: RAISE MOULD
T6	: MOULD RAISED ?
S0	: GRAPH5 END

7.2 Programming

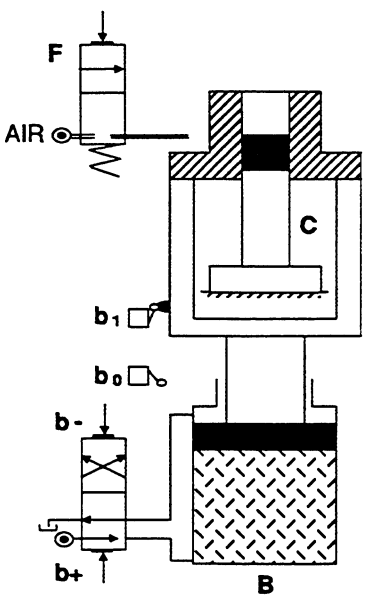
7.2.1 Preparing for Programming

To implement the sequential control, appropriate sensors, actuators etc. must be selected and the necessary commands and feedback messages determined.



Controlling the actuators

Q 3.0	-HYDRO/a+	: Lower die
Q 4.0	-HYDRO/a-	: Raise die
Q 5.0	-HYDRO/b-	: Lower mould
Q 7.0	-HYDRO/b+	: Raise mould
Q 6.0	-VALVE/F	: Remove pellet
Q 2.0	-DISPLAYL	: LED "ready"



Limit positions of the actuators

I 5.0	-LIMSWIa1	: Lower die position
I 6.0	-LIMSWIa0	: Upper die position
I 7.0	-LIMSWIb0	: Lower mould position
I 9.0	-LIMSWIb1	: Upper mould position
I 4.0	-ENABLE	: For start of cycle
I 4.1	-EMER OFF/N	:

In addition to this, other (symbolic) assignments can be made for manual interventions, releases etc.

The assignment list is stored in a symbols file, example:

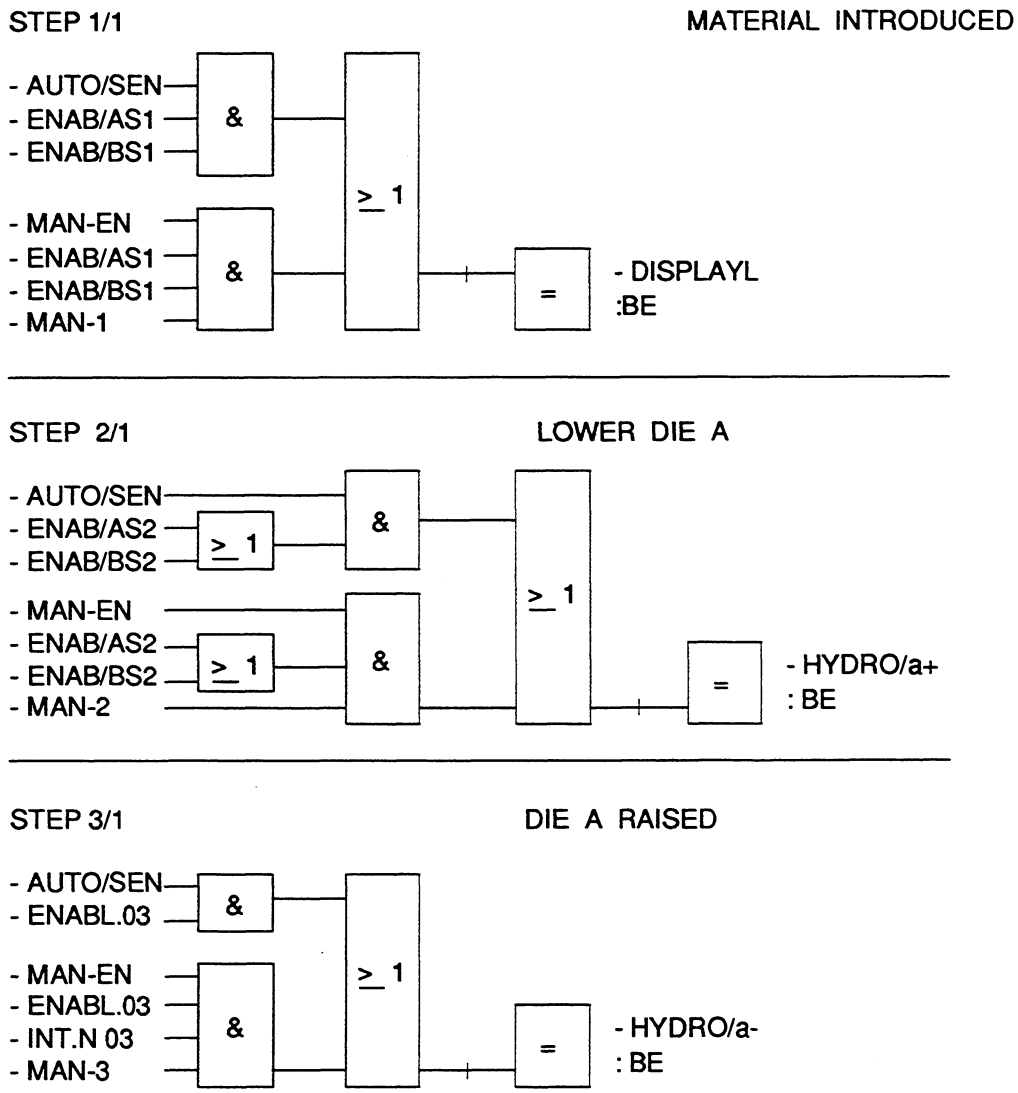
C:PRESSZ0.SEQ

OPERAND	SYMBOL	COMMENT
I 4.0	ENABLE	For cycle start
I 4.3	ENAB/AS1	First enable signal for step 01
I 4.4	ENAB/BS1	Second enable signal for step 01
I 4.7	MAN-1	Console "A" switch 4
I 5.0	LIMSWIa1	Lower dye position
I 5.3	ENAB/AS2	First enable signal for step 02
I 5.4	ENAB/BS2	Second enable signal for step 02
I 5.7	MAN-2	Console "A" switch 5
I 6.0	LIMSWIa0	Upper dye position
I 6.3	ENABL.03	Enable for step 03
I 6.6	INT.N 03	Interlock for manual switch MAN-3
I 6.7	MAN-3	Console "A" switch 6
I 7.0	LIMSWIb0	Lower mould position
I 7.3	ENAB/AS4	First enable signal for step 04
I 7.4	ENAB/BS4	Second enable signal for step 04
I 7.5	ENAB/CS4	Third enable signal for step 04
I 7.7	MAN-4	Console "B" switch 10
I 8.3	ENABL.05	Enable for step 05
I 8.7	MAN-5	Console "B" switch 11
I 9.0	LIMSWIb0	Upper mould position
I 9.3	ENABL.06	Enable for step 06
I 9.7	MAN-6	Console "C" switch 13
Q 1.5	MAN-EN	
Q 2.0	DISPLAYL	"Ready" LED
Q 3.0	HYDRO/a+	Lower dye
Q 4.0	HYDRO/a-	Raise dye
Q 5.0	HYDRO/b-	Lower mould
Q 6.0	VALVE/F	Remove pellet
Q 7.0	HYDRO/b+	Raise mould
F 233.0	AUTO/SEN	Current step flag
PB 10	START/10	Start call for powder press
SB 0	EXEC SB0	Execute block for GRAPH 5
SB 10	PRESS	Powder press program
FB 70	CONTROL	Call FB for GRAPH 5
DB 10	USRDB/03	Powder press user DB
DB 255	DIAGN.	Communication DB for diagnosis

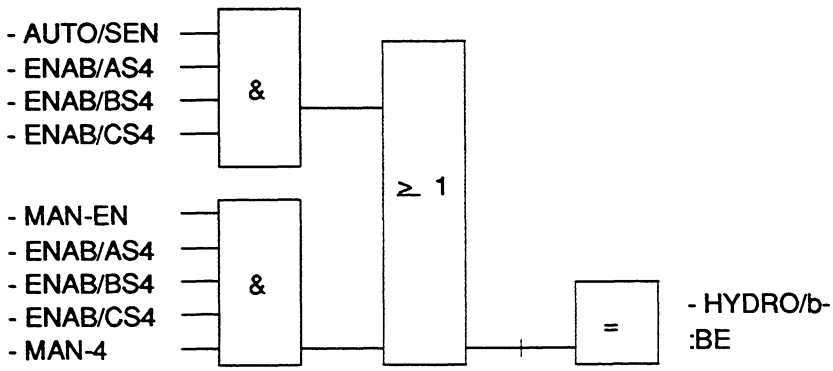
7.2.2 Programming on the PG

Steps and transitions are programmed at the zoom-in level as follows:

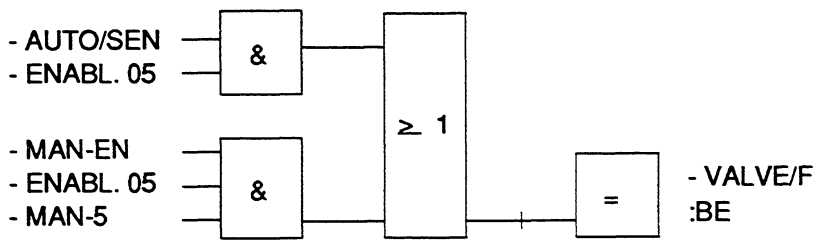
Program all steps



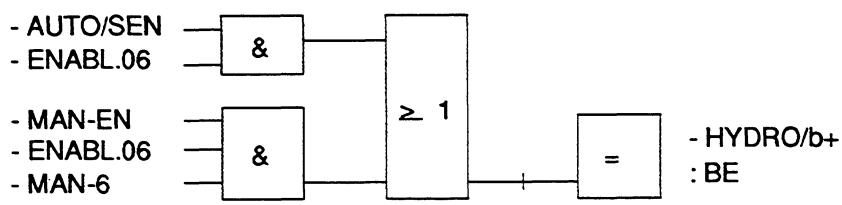
STEP 4/1 LOWER MOULD



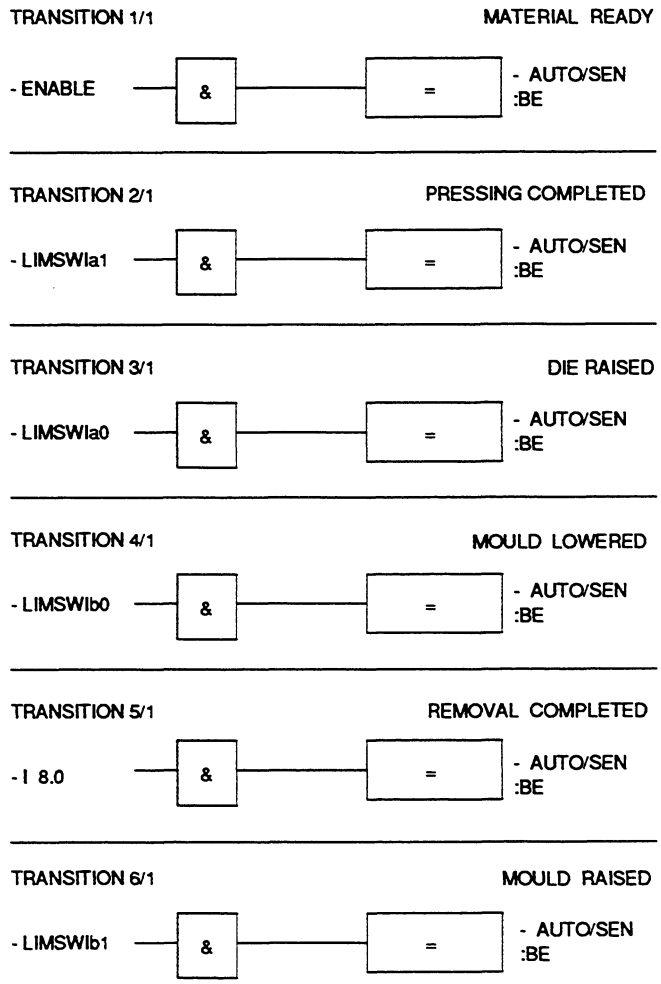
STEP 5/1 REMOVE PELLET



STEP 6/1 MOULD RAISED



Program all transitions



☞ The current step flag F 233.0 has been assigned the symbolic name -AUTO/SEN in this example.

7.2.3 Entering the Waiting and Monitoring Times

You assign waiting and monitoring times with **F5** (TM/TW). You can also specify the time using symbols.
Step 2 (LOWER DYE A) has a monitoring time. If the lower position (pressing completed) is not reached after 5 seconds, a timeout is signalled.

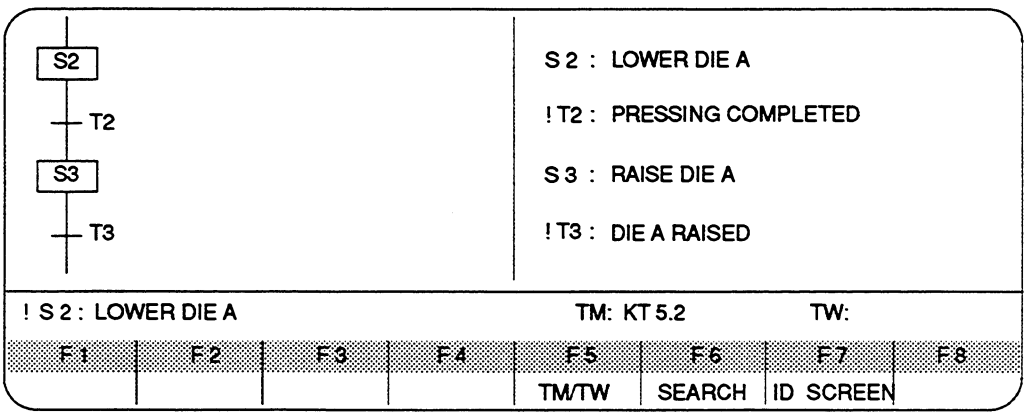


Fig. 7.3 Example of step 2 with a monitoring time of 5 seconds

Step 5 (removal) has a waiting time of 3 seconds.

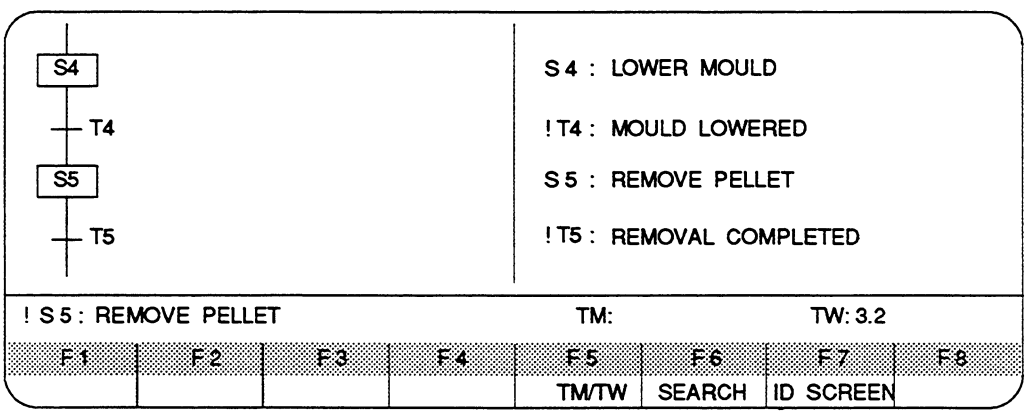


Fig. 7.4 Example of STEP 5 with a waiting time of 3 seconds

7.3 Starting Up and Testing

7.3.1 Transferring the Program to the PLC

Transfer the sequence block to the PLC using the transfer function (see programmer manuals), **F7** (AUX FCT), **F1** (TRANSFER).

Transferring the blocks required for running the sequencer in the PLC

For a sequencer, the standard function blocks SB 0 and FB 70 are required, for secondary sequences FB 71 is also required.
These blocks must be loaded in the PLC.

SB 0 is called at each transition. It executes the modes that have been transferred to FB 70 as parameters.

FB 70 calls GRAPH 5. It manages the modes and monitors the sequence.

FB 71 is responsible for calling secondary sequences.

7.3.2 Generating the Diagnosis DB and the User DB

To diagnose sequences loaded in the PLC, a diagnosis DB is required and is accessed when the DIAGNOSIS function is called.

Each sequencer also requires a user DB (with a number identical to the SB number). These data blocks are generated with **F1** (INPUT), **F5** (DBGEN); the diagnosis DB is the same for all sequencers in the PLC. If these data blocks are not generated directly in the PLC (DBGEN DEVICE: PC) they must be transferred to the PLC.

7.3.3 Calling FB 70 in the Program

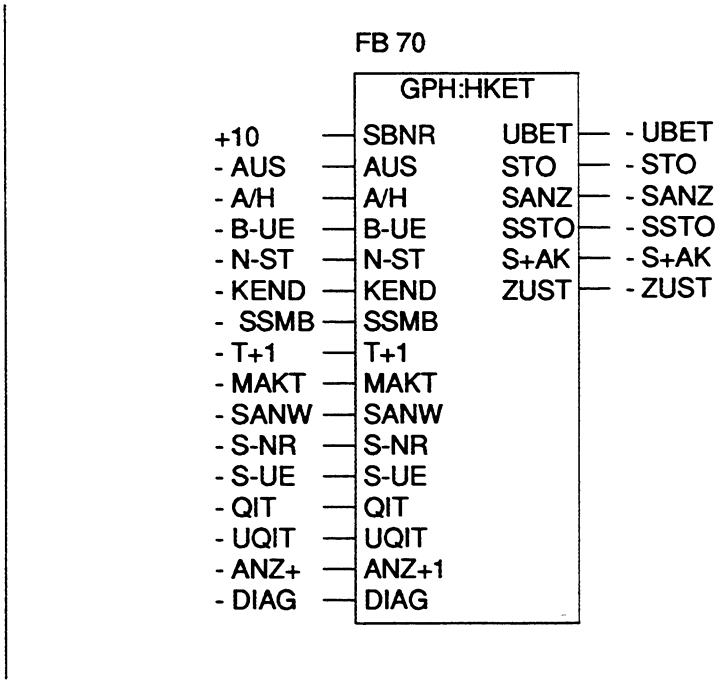
In OB 1, a program block (PB) is called that in turn calls FB 70 (with JU FB70 or JC FB70). The number of the PB must be the same as the number of the sequence block.

The number of the SB to be called (SB 10) must be transferred to FB 70 (parameter SBNR).

```
OB1
SEGMENT 1
      : JU -START/10
      : BE
```

e.g. CSF:

```
PB 10
SEGMENT 1
```



7.3.4 Testing the Program

With the status function, you can test the status of the sequencer at the overview and zoom-in levels (Section 6.3.1 Status).

7.4 Printing Out the Program

If you print out the final program, you obtain the following:

- Sequence identification screen
- Overview level with all comments
- List of all transitions with transition comments
- List of all steps with step comments and waiting and monitoring times TM/TW
- All transitions at the zoom-in level with comments and assignment lists of the symbols used
- All steps at the zoom-in level with comments and assignment list of the symbols used

7.5 Troubleshooting

If the Sequence has a Timeout

The DIAGNOSIS function is useful for troubleshooting. Sequences with a timeout are displayed. The DIAGNOSIS function allows you to follow the status of the sequence at both the overview and zoom-in level (Section 6.3.2 Diagnosis).
If a timeout occurs, the sequence involved is marked with the message **TIMEOUT**.

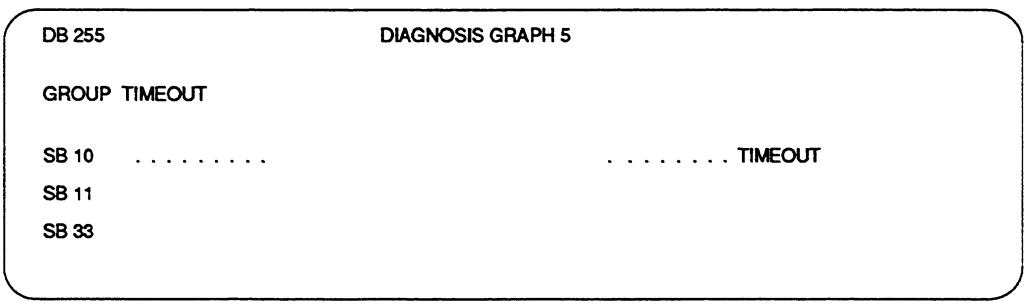


Fig. 7.5 Example of a list of SBs displayed by the DIAGNOSIS function (SB 10 has a timeout)

To display a sequence with a timeout and to find the cause of the error, position the cursor on the corresponding SB in the list. Press the **zoom-in** key to display the status of the sequence at the overview level. Steps in which a timeout has occurred are displayed inversely.

The contents of the step or transition can be diagnosed. Position the cursor on the step or transition at the overview level. Press the **zoom-in** key to change to the zoom-in level. The status of the step or transition is displayed.

Running the Program on the PLC

8 Running the Program on the PLC

Three standard programs are available for running sequencers cyclically on the PLC, as follows:

Version 1:	FB 70, FB 71, SB 0	STANDARD VERSION for complex modes with simultaneous branches and secondary sequences.
Version 2:	FB 72, SB 2	FAST VERSION for simple modes with simultaneous branches.
Version 3:	FB 73, SB 3	FAST VERSION for simple modes without simultaneous branches.
	FB 74	For supplementary functions for FB 72, FB 73

8.1 Characteristic Data of the Standard FBs

A special version of blocks FB 70 to FB 74 is required for each type of PLC. The versions can be used in different programmable controllers as shown in the following table:

PLC	Version 1 FB 70/71	Version 2, 3 FB 72/73, FB 74
S5 100 with CPU 103	X	
S5 115 U	X	X
S5 135 U	X	X
S5 150 S/U	X	X
S5 155 U/H	X	X

8.1.1 Overview of Diagnosis Options

PLC	PG (online with PLC)	CP 526 (op. and mon.) (CPU 921, 922, 928	DIMOS	GRAPH 5 WF 470 (not for CPU 921)
S5 100 U	FB 70			
S5 115 U	FB 70			FB 70 FB 72, 73
S5 135 U	FB 70	FB 70 FB 72, 73		FB 70 FB 72, 73
S5 150 S/U	FB 70	FB 70 FB 72, 73	FB 70 FB 72, 73	FB 0 FB 72, 73
S5 155 U/H	FB 70			

The diagnosis function is explained in the descriptions of the standard function blocks.

8.1.2 Characteristics of FB 70 to FB 74

Characteristics of FB 70

- Max. 127 steps
- Max. 8 simultaneous branches
- Max. 8 alternative branches
- Jumps unrestricted
- Waiting and monitoring time in every step
- Modes:
  OFF
  AUTOMATIC/MANUAL
  INITIALIZATION
  EXECUTE
  STEP SELECTION
  COLD RESTART/WARM RESTART FUNCTION
  AUTOMATIC SEQUENCE END FUNCTION
  DIAGNOSIS WITH PG
  SELF-DIAGNOSIS
  STEP DISPLAY
  TIMEOUT DISPLAY
  SECONDARY SEQUENCE WITH FB 71

Characteristics of FB 71 in conjunction with FB 70

- Max. 127 steps
- Max. 8 simultaneous branches
- Max. 8 alternative branches
- Jumps unrestricted
- Waiting and monitoring time in every step
- Modes:
  INITIALIZATION
  EXECUTE
  DIAGNOSIS WITH PG
  SELF-DIAGNOSIS
  STEP DISPLAY
  TIMEOUT DISPLAY
  SECONDARY SEQUENCE WITH FB 71 (a secondary sequence can call further secondary sequences)

Characteristics of FB 72

- Max. 127 steps
- Max. 8 simultaneous branches
- Max. 8 alternative branches
- Jumps unrestricted
- Waiting and monitoring time in every step
- Modes:
  OFF
  AUTOMATIC/MANUAL
  INITIALIZATION
  EXECUTE
  TIMEOUT DISPLAY

Characteristics of FB 73

- Max. 127 steps
- Only linear sequence
- Max. 8 alternative branches
- Jumps unrestricted
- Waiting and monitoring time in every step
- Modes:
  OFF
  AUTOMATIC/MANUAL
  INITIALIZATION
  EXECUTE
  TIMEOUT DISPLAY

Characteristics of FB 74 in conjunction with FB 72, FB 73

- Modes:
  EXECUTE without condition
  STEP SELECTION
  DELETE STEP
  SYNCHRONIZE

8.2 Program Structure in the PLC

The sequencer is created in a sequence block SBx (x = 10...255) with a sequential structure and the contents of the transitions and steps on diskette using the PG. In the programmable controller (PLC) a user data block DBx with the same number x as the sequence block is also required per sequencer. This also applies to secondary sequences.

The data block DBx contains data about the structure, init steps, program times etc. of the sequencer. The DBx must therefore be re-generated following any modification to the structure (DBGEN function). DBx also contains the status of the sequencer for processing in the next PLC cycle.

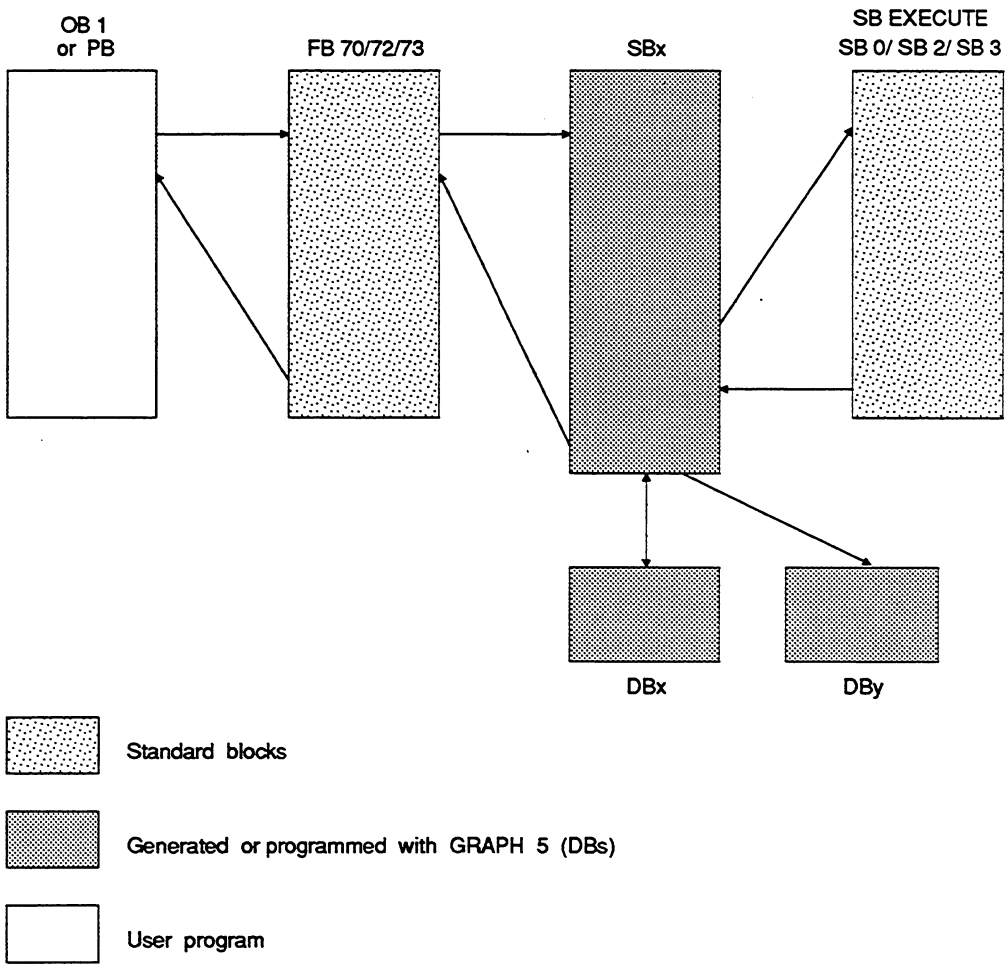
With the standard function block FB 170, a common data block DBy is required for diagnostic functions for all the sequencers. The number y of the diagnosis DB must not be the same as the number of a user DB. The diagnosis DB is created during a generation run by the programmer after the sequencers have been created on diskette (DBGEN function). DBy contains the step numbers with timeouts for diagnosis using the programmer. DBy is only necessary when using FB 70.

The standard function blocks FB 70 to FB 74 along with the standard block SB EXECUTE (SB 0, SB 2, SB 3) manage the sequencer generated in SBx and implement the modes. The blocks should be used in pairs as shown below:

FB 70 and SB 0,
FB 72 and SB 2,
FB 73 and SB 3.

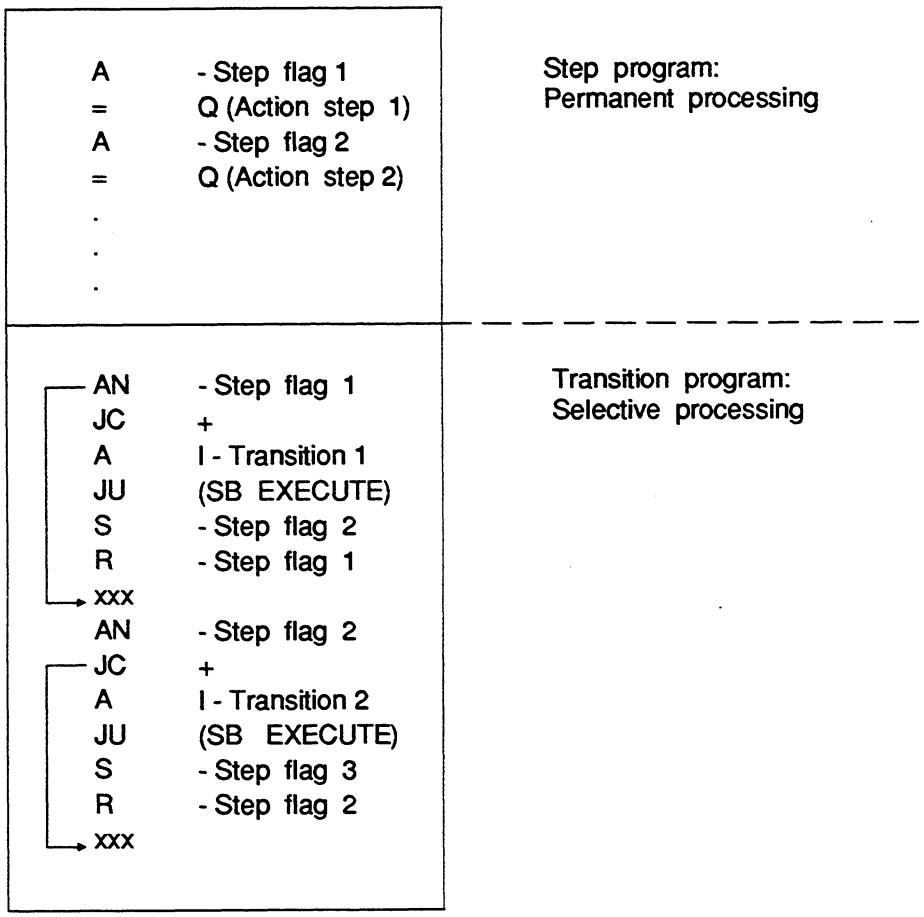
When FB 70...FB 74 are called, the input parameters for the connections to the operating console for AUTO/MANUAL etc. are specified. SB EXECUTE is only loaded and does not have parameters assigned.

The resulting program has the structure shown below (without FB 71 and FB 74):



8.2.1 Basic Structure of SBx and DBx/DBy

SBx



The step flag 1...127 is represented on the programmer as F 233.0. Depending on the step number, this flag F 233.0 stands for a special flag belonging to the sequencer. The step program is executed permanently in each mode and reacts just as normal program blocks. The dependence on a particular step number results from the scanning of the current step flag.

In any step segment, a logic operation can be performed without step flags independent of the steps, i.e. the initialization mode (submode) can also be stored in the sequential program, with the advantage that only one block needs to be checked for both the automatic and initialization modes.
If several and different functions depend on the same step, only flag F 233.0 needs to be scanned in the step.

8.2.2 Basic Program Execution in the PLC

FBs 70 to 73 recognize the selected mode.

SB EXECUTE checks the logic of the transition and enables the switchover to the next step, depending on the operating mode and status of the sequence.

For example: AUTOMATIC active, transition logical "1":
switch to the next step,
or
AUTOMATIC active, transition logical "1":
however, TIMEOUT caused by elapsed monitoring time:
switching to next step not enabled.

If the switching to the next step is recognized and is possible, the next step flag is set and the current flag reset. However, the next action will only be carried out in the next PLC cycle after the program checks that the previous action is inactive and the step flag active, since the step program is stored before the transition program in sequence block SBx.

With simultaneous branches, all the branches (levels) in a PLC cycle are processed one after the other. Seen externally, the switch to the next step is simultaneous.

If consecutive transitions at one level are satisfied simultaneously, only one step is switched further, i.e. each step becomes active at least for one cycle.

Transitions that have not been programmed are considered as satisfied.

8.2.3 Procedure for Creating Programs

1. Program sequencer(s) in sequence block(s) (SBx) with the PG and save on diskette. If necessary, program secondary sequence(s) in sequence block(s) and save on diskette. FB 71 (secondary sequence) is called in one step of the main sequence.

2. Call and assign parameters to FB 70 (if necessary, FB 71 in SBx) or
   call and assign parameters to FB 72 (if necessary, also FB 74) or
   call and assign parameters to FB 73 (if necessary, also FB 74).

3. Generate data blocks (DBx, DBy) and save on diskette.

4. Load all the blocks required, OB, PB, SB, DB, FB 70 (if necessary, FB 71) or FB 72, or FB 73 (if necessary, FB 74) and SB EXECUTE (SB 0 or SB 2, or SB3) in the PLC.

8.3 Versions of the Standard Programs

in the sequence identification screen form, you can specify the version of the standard programs you want to use, as follows:

FB 70/71 FOR LINEAR/SIMULTANEOUS SEQUENCE: STANDARD VERSION
FB 72 FOR LINEAR/SIMULTANEOUS SEQUENCE: FAST VERSION
FB 73 FOR LINEAR SEQUENCE: FAST VERSION

Depending on the FB specified, block calls are generated in the SB as follows:

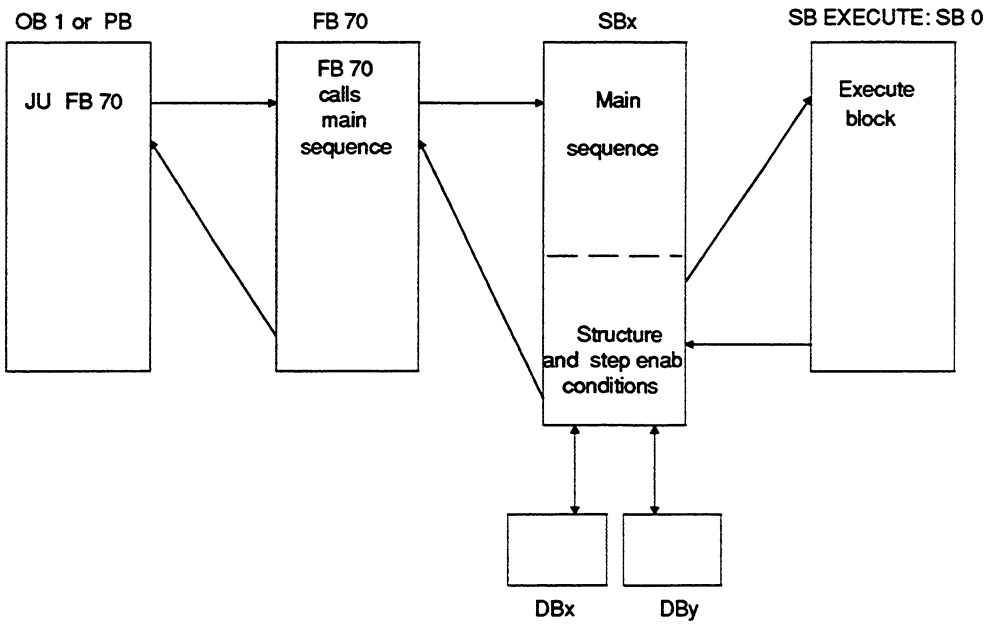
FB 70: SB 0 and DBx **and** DBy are generated in SBx
FB 72: SB 2 and DBx **and** DBy are generated in SBx *)
FB 73: SB 3 and DBx **and** DBy are generated in SBx *)

FB 70 **and** FB 72 **and** FB 73 can be used simultaneously in the same PLC.

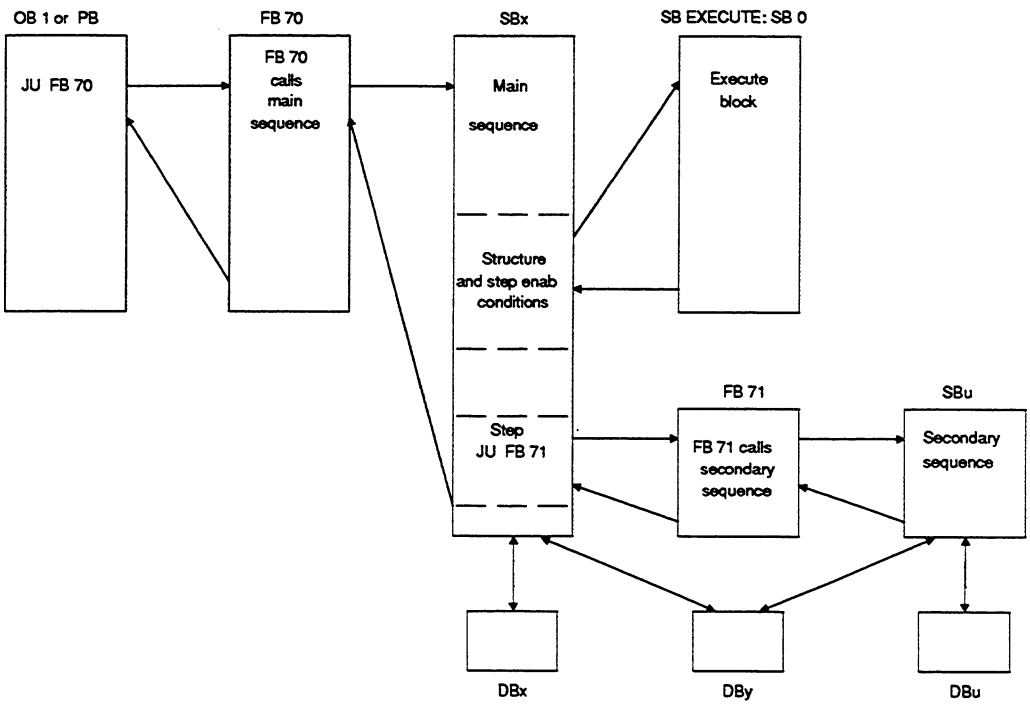
*) The DBGEN function not only generates the user DB but also DBy (diagnosis DB). This is, however, not required with FB 72 and FB 73.

Version 1-3	User DBx	Diagnosis DBy	SB EXECUTE
Version 1 FB 70 with subseq. FB 71	DBx DBu	DBy	Load SB 0 in PLC
Version 2 FB 72	DBx	not required	Load SB 2 in PLC
Version 3 FB 73	DBx	not required	Load SB 3 in PLC

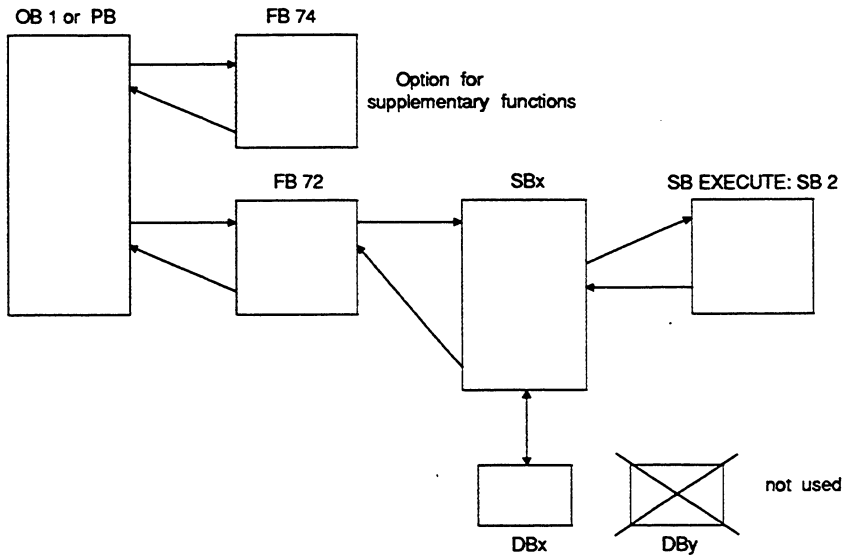
8.3.1 Program Structure with FB 70



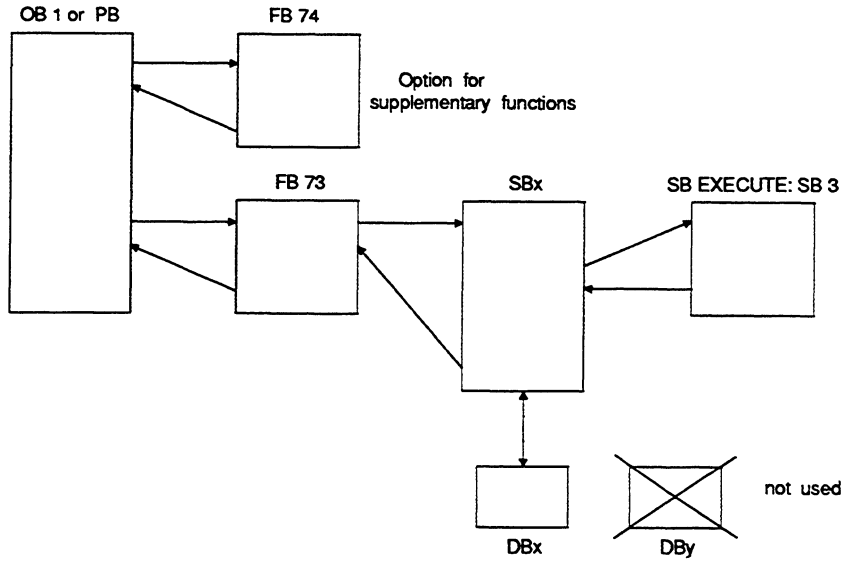
8.3.2 Program Structure with FB 70 and FB 71 Secondary Sequence



8.3.3 Program Structure with FB 72



8.3.4 Program Structure with FB 73



Appendix

This appendix contains the following:

I S5 Terminology

The following list contains the most common SIMATIC S5 abbreviations. It does not, however, contain any softkey labels, since these are explained by HELP texts within the software.

A

ABS	Absolute addressing, e.g. I 1.0
AS 511	511 interface module, interface to the PLC

B

B	Block
BE	Block end

C

COM	Comment key
CORR	Correction key
CPU	Central Processing Unit
CSF	Control System Flowchart, graphical representation of automation tasks with symbols according to DIN 40700/DIN 40719

D

D,	Data (1 bit),
DB	Data block
DBDO.nnn	Documentation block for DB data block
DC	Comment block for DB data block
DCX	Comment block for DX data block
DIR	Directory of the hard disk, diskette, PLC, EPROM and files
DOCFILE	Documentation file, e.g. for plant comments
DSP ABS	Presets screen form, display absolute operands
DSP SYM	Presets screen form, display symbolic operands
DW	Data word (16 bits)
DX	Extended data block
DXDO.nnn	Documentation block for DX data block

E

EEPROM	Electrically erasable programmable read-only memory
EPROM	Erasable programmable read-only memory

F

F, FY, FW, FD	Flag bit, flag byte, flag word, flag double word
FB	Function block
FBDO.nnn	Documentation block for FB function block
FC	Comment block for FB function block
FCX	Comment block for FX function block
FD	Preset program file on floppy disk (also stands for hard disk)
FlexOS	Operating system
FV	Block preheader for FB
FVX	Block preheader for FX
FX	Extended data block
FXDO.nnn	Documentation block for FX function block

G

GRAPH 5	Software package for planning and programming sequential control systems in a clear graphical representation

I

ID screen	Sequence identification screen form

K

KOMI	Command interpreter

L

LAD	Ladder Diagram, graphic representation of automation tasks with the symbols of circuit diagrams according to DIN 19239
LEN	Length of a block
LIB	Library number

O

OB	Organization block
OBDO.nnn	Documentation block for organization block
OC	Comment block for organization block
OY, OW	Byte, word from the extended periphery

P

PB	Program block
PBDO.nnn	Documentation block for program block
PC	Comment block for program block
PG	Programmer
PLC	Programmable controller (only two characters are available in the command line and message line for the device field, in this case, programmable controller is abbreviated to PC)
PW	Peripheral word
PY	Peripheral byte

R

RAM	Random Access Memory
REW	Rewiring, renaming inputs and outputs in the user program (package XRF, COMP, REW)
RLO	Result of logic operation (bit condition code)

S

S	S flag, extended flag area
SAC	Step address counter
SB	Sequence block
SBDO.nnn	Documentation block for sequence block
SC	Comment block for sequence block
SINEC H1	Bus system, network for industrial applications
STA	Status (bit condition code)
STEP 5	Programming langugae for programming SIMATIC S5 programmable controllers
STL	Statement List, STEP 5 method of representation as a sequence of abbreviations of PLC operations (complying with DIN 19239)
SYM	Symbolic addressing, e.g. -INPUT
SYSID	Block for system identification
S5-KOMI	S5 command interpreter
S5-DOS/MT	S5 operating system (multi-tasking capability)

T

T	Timer
TM	Monitoring time
TW	Waiting time

X

XRF	Cross reference list (XRF, COMP, REW package)

II Glossary

Active step A step in the sequencer is active when its actions are being executed.

Monitoring time The monitoring time (TM) is the time within which the step enabling conditions for the next step must be active and before which the sequencer must move on to the next step. If the next step is not activated within this time, a timeout is indicated. The monitoring time is evaluated automatically by the standard function blocks for the modes.

Network Link between several computers (PC, PG, PLC) by means of interface modules, physical lines and appropriate software to be able to exchange data between the computers.

Overview level The structure of the sequential control system (steps, transitions, simultaneous and alternative branches) is programmed at the overview level.

Presets Screen form with parameters for the current STEP 5 software package. The STEP 5 software package uses the parameters entered in the presets screen form.

Process variable A process variable also simply known as a variable, is an operand to which a process-dependent value is assigned. These values can either be variable or constant. These operands have a signal state, known as their status.

Segment Division of a SIMATIC S5 block.

Sequence block A sequence block (SB) is a STEP 5 block. There are two types of sequence blocks:
1. Sequence blocks in the LAD, CSF, STL package. These contain all or part of the user program in the form of STEP 5 operations (basic operations) and if required comments. They extend the range of program blocks.
2. Sequence blocks in the GRAPH 5 package. These are special sequence blocks for sequential control systems. They contain the sequences in the form of steps, step enable conditions (transitions) and branches. The steps and transitions in these blocks contain the user program in the form of STEP 5 operations and if required comments.

Sequential control system A control system with a sequence of steps, one step following on from the previous step depending on step enabling conditions.

Softkey A key assignment displayed as a menu at the lower edge of the screen. This indicates the function currently assigned to the keys F1 to F8.

Status Display of the signal states of operands on the programmer. The status function is an online function between the PG and PLC.

Step A step is part of a sequencer that describes the actions to be executed by the control system when a certain status exists. The actions of a step are programmed at the zoom-in level in STEP 5.

Step enabling condition See transition

Symbols file Assignment list stored in a file.

S5 KOMI The S5 command interpreter manages and coordinates the S5 packages, utilities and overlays. When you start the basic package, you enter the SIMATIC environment with its terminology, data structures (blocks), screen forms and function keys.

S5 packages The whole programmer software cannot be loaded simultaneously in the user memory (RAM). For this reason, it is divided into packages. These packages are displayed by the S5 command interpreter and stored in the user memory when they are selected. Apart from the STEP 5 packages (LAD, CSF, STL, symbols editor...) there are also further packages such as GRAPH 5, KOMDOK, PG-NET and the COM packages.

Transition A transition is part of the sequencer and contains the step enabling conditions with which the control system changes from one status to the next. The step enabling conditions are programmed at the zoom-in level in STEP 5.

A transition is valid when its previous step(s) is (are) active. A transition switches when it is valid and the step enabling conditions are satisfied. Switching means that the transition terminates the previous step(s) and activates the next step(s).

Valid transition A transition is valid when its previous step(s) is (are) active.

Waiting time The waiting time (TW) is a minimum time for which a step remains active even if the next transition has already been satisfied. The next step can only become active when the waiting time has elapsed. The waiting time is evaluated automatically by the standard function blocks for the modes.

Zoom-in level The contents of the steps and transitions are programmed using the zoom-in function, their status is also displayed at this level.

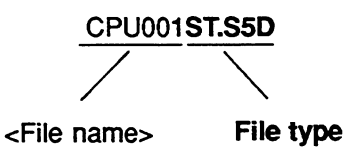
III S5 File Types

S5 files

The basic package sets up S5 files. There are several types of S5 files. Different types of blocks are stored in different file types, as follows:

- blocks in PROGRAM FILES of the type ST.S5D,
- assignment lists in SYMBOLS FILES of the types Z0.INI and Z1.INI,
- footers in FOOTER FILES of the types F1.INI and F2.INI,
- printer parameters in PRINTER FILES of the type DR.INI etc.

The file types are distinguished by the last two characters of the name and the three-character extension. The whole file name of an S5 file has a maximum of 8 plus 3 characters, of which you can only select the first 6 characters. These 6 characters are known as the file **name**. Example of a PROGRAM FILE:



The following table lists the file types used in STEP 5.

File name	Content	Remarks
@@@@@@ ST.S5D	S5 program file	
@@@@@@ AP.INI	Selection path (for bus selection)	
@@@@@@ D0.INI	File for DB source language	
@@@@@@ D#.INI	DB source language (index file) # = 1..9	
@@@@@@ DR.INI	Printer parameters	
@@@@@@ F1.INI	User footer (80 characters)	
@@@@@@ F2.INI	User footer (132 characters)	
@@@@@@ LS.INI	Printer protocol in file	
@@@@@@ SD.INI	SYSID data	
@@@@@@ SU.INI	Embedded commands (submits), KOMDOK	
@@@@@@ SF.INI	SUBMIT error list, KOMDOK	
@@@@@@ TP.INI	Key macros	
@@@@@@ Z0.INI	Assignment list (symbols file)	
@@@@@@ Z#.INI	ASSLI index files	
@@@@@@ XR.INI	Reference list (XRF file)	
@@@@@@ A0.INI	STL source file	
@@@@@@ A1.INI	Intermediate file (independent of language)	
@@@@@@ AF.INI	Error list and compiler	
@@@@@@ AT.INI	Function key assignment	
@@@@@@Z0.SEQ	Assignment list source file	
@@@@@@ ZF.INI	Error list ASSLI (following translation SEQ -> INI)	

IV List of Documentation

Siemens' specialist literature

Berger, Hans
Automating with the SIMATIC S5-155 U
ISBN 3-8009-1562-6

Berger, Hans
Automating with the SIMATIC S5-135 U
ISBN 3-8009-1561-8

Berger, Hans
Automating with the SIMATIC S5-115 U
ISBN 3-8009-1530-8

Siemens' manuals

STEP 5 Basic Package
Order no. 6ES5 998-0SC21

STEP 5/MT Basic Package
Order no. 6ES5 998-0FC21

GRAPH 5
Standard Function Blocks
Programmable Controllers S5 115U, 135U, 150U, 155U
Order no. C79000-G8563-C587

GRAPH 5-EDDI
Standard Function Blocks
Programmable Controllers S5 115U, 135U, 150U, 155U
Order no. C79000-G8563-C679

FlexOS (English)
Order no. 6EA9200-0AA10-0AB0

PCP/M-86 (English)
Order no. 6ES5 998-2SA21

Further programming examples and instructions can be found in the programmable controller manuals and the documentation for the other packages.

V Index

A

B

C

D

E

F

G

I

J

K

L

M

O

P

R

S

T

U

V

W

Z