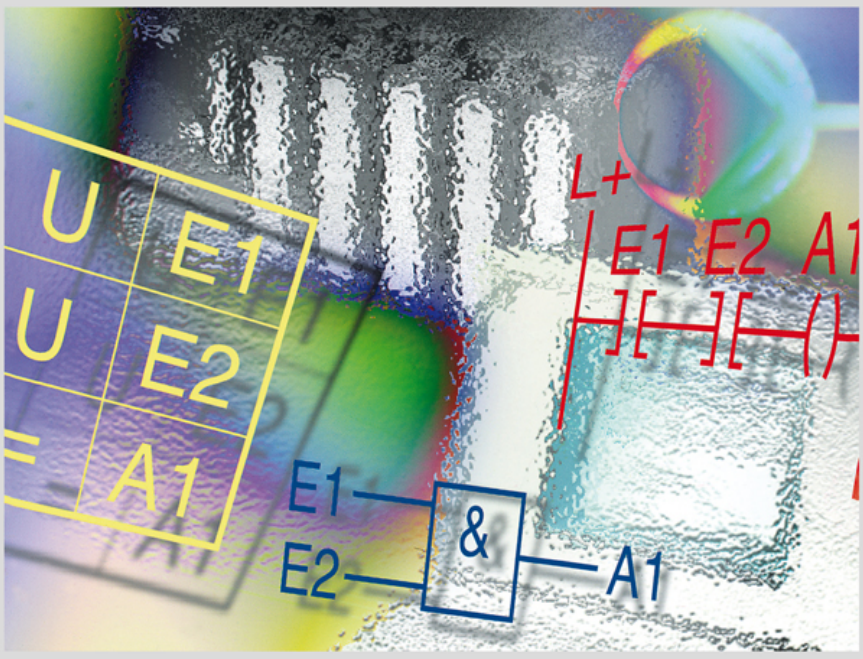
Vogel Fachbuch

Jürgen Kaftan

PLC-Basic Course with SIMATIC S7



VOGEL

Jürgen Kaftan

PLC Basic Course with SIMATIC S7

Jürgen Kaftan

PLC Basic Course with SIMATIC S7

Structure and Function of Programmable Logic Controllers, Programming with the SIMATIC S7

First Edition

JÜRGEN KAFTAN

1967-1971 Apprenticeship as an electrician
1971-1973 Skilled tradesman in the trade of electrician
1973-1975 Training as a state-certified electrical engineer
1975-1977 Employed as a technician
1977-1978 School for master craftsmen, graduated as master electrician
1979-1992 Nuremberg vocational training centre for hearing and speech impaired persons, Master Trainer
1985-1992 Course Leader for PLCs in vocational training (BFH) for hearing and speech impaired persons
1985-1992 Course Leader for PLCs at the Chamber of Crafts
1992-1995 IKH Elektrogerätebau – System Training Courses (Managing Director)
Since 1995 Manager of IHK Systemschulungen for Hardware and Software (Chamber of Industry and Commerce hardware and software system training courses) in Weißenburg and Roth (Central Franconia)

Jürgen Kaftan is the author of the following reference books published by Vogel:
PLC Basic Course 1
PLC Basic Course 2
PLC Basic Course with SIMATIC S7

The book includes a Siemens AG demonstration CD-ROM »STEP 7 Professional, Edition 2004 SR4, Trial License« including: STEP 7 V5.3 SP3, S7-GRAPH V5.3 SP2, S7-SCL V5.3 SP1, S7-PLCSIM V5.3 SP1 and can be used for test purposes for 14 days.
The software only runs under Windows 2000 Professional SP4 and above or Windows XP Professional SP1 and above.
For further information on the Internet, visit www.siemens.de/sce/promotoren.

Further information:
www.vogel-buchverlag.de

ISBN 978-3-8343-3201-1
First Edition 2011

Printed in Germany

Book jacket illustration: Michael M. Kappenstein, Frankfurt

Foreword

Programmable logic controllers, known for short as PLCs, are an integral part of automation at the lower, middle and upper performance levels. Due to the ever increasing size and complexity of the hardware and software required in automation projects, the industry has had to develop faster, more capable and effective automation systems and to simplify handling for users. All the exercises in this book have been developed and tested using Siemens' SIMATIC S7-300 PLC. The command set of this controller ranges from binary processing to 32-bit floating point arithmetic. The controller is programmed under the Windows XP operating system and it is assumed that users have an appropriate basic knowledge. All the procedures for programming the SIMATIC S7-300 are demonstrated exactly and are easy to understand for users. The topic builds up from "easy to hard" and is ideally suited for use in vocational and technical training colleges, etc. as well as for use on a teach-yourself basis.

I would like to thank the Wükro-Lehrsysteme company in Würzburg as well as everybody who has helped in the completion of this book. I am always grateful to receive feedback from readers and users.

Weißenburg/Heuberg (Central Franconia) Jürgen Kaftan

Contents

1 Introduction

The programmable logic controller (PLC) has the job of carrying out open-loop or closed-loop control of a machine or plant's individual operations in accordance with a specified function cycle depending on encoder signals.

1.1 Number system

A programmable logic controller does not use the decimal system for processing the addresses of storage locations, inputs, outputs, times, bit memories, etc.; rather, the **dual number system** is used.

The dual number system only includes the digits 0 and 1 that can easily be represented and evaluated in data processing. **It is a binary number system.**

The significances of a dual number are, as shown below, assigned to the powers of 2.

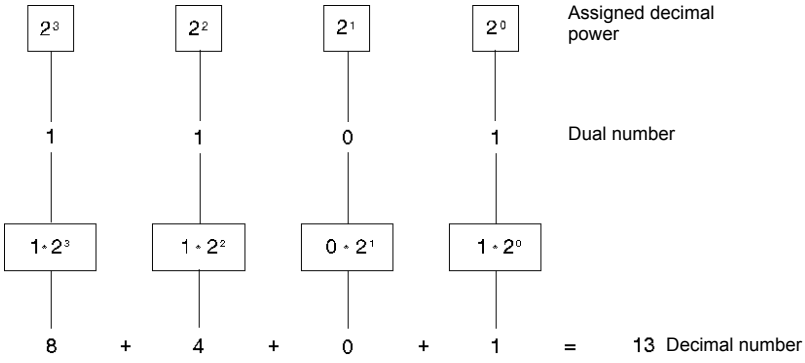


Figure 1.1 Each digit within a dual number is assigned to a power of two

1.2 Terms from computer science

In connection with programmable logic controllers, people often use terms from data or information processing like bit, byte, word and double word.

1.2.1 Bit

A bit (an abbreviation of binary digit) is the smallest binary (dual-value) unit of information.

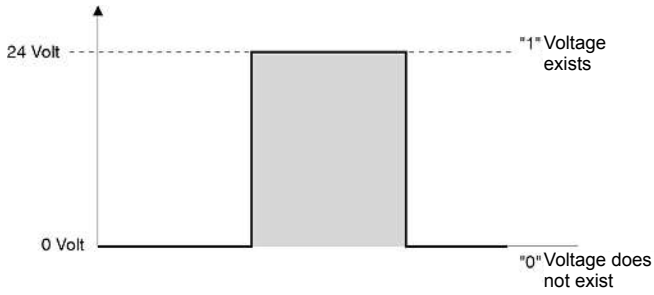


Figure 1.2
A bit can take on a signal state of "1" or "0"

1.2.2 Byte

A byte is a term for a unit of eight bits.

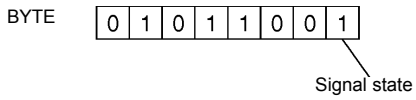


Figure 1.3
A byte has a size of 8 bits

1.2.3 Word

A word consists of two bytes or 16 bits. Using "words", you can, for example, represent:

- ❒ Dual numbers,
- ❒ Letters,
- ❒ Control instructions.

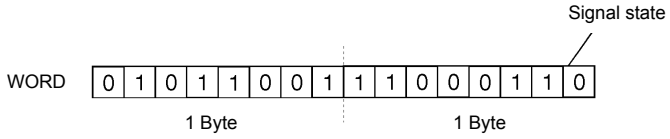
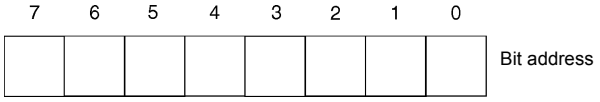


Figure 1.4
A word has a size of 2 bytes or 16 bits

1.2.4 Bit address

For the system to detect and address bits, each individual bit in a byte is assigned a digit, what is known as a bit address.

Figure 1.5
In each byte, the right-most bit is given bit address 0 and the left-most one gets the bit address 7

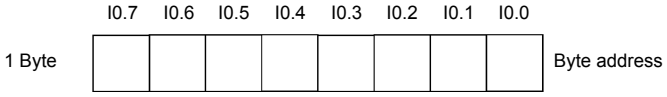


1.2.5 Byte address

The individual bytes are also given numbers, i.e. the byte addresses. Additionally, the operand is labelled. This means, for example, that IB 2 stands for input byte 2 or QB 4 represents output byte 4. Individual bits are uniquely addressed by the combination of the bit and byte addresses.

The bit address is separated from the byte address by a full stop. To the right of the full stop, there is the bit address, with the byte address being to the left of it.

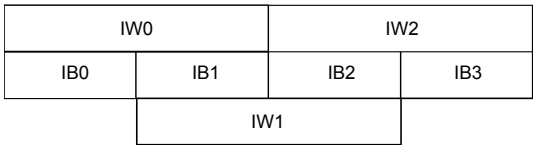
Figure 1.6
Example:
Byte address I0.0 or Q4.5



1.2.6 Word address

The numbering of words yields the word address. When using words, e.g. IW (input word), QW (output word), MW (memory word), DW (data word), the word address is always the lower byte address of the two associated bytes.

Figure 1.7
Word address



2 Arrangement of a PLC

PLCs are mass-produced. Initially, they do not yet have a task. Manufacturers integrate all the components necessary for the control engineering such as the logic elements, latching/unlatching functions, times, counters, etc. and these components are linked to form a functioning controller by means of programming. There are a large number of different control units that differ from one another by virtue of the following functional units:

- ❒ Inputs and outputs,
- ❒ Storage locations,
- ❒ Counters,
- ❒ Times,
- ❒ Bit memory functions,
- ❒ Special functions,
- ❒ Operating speed,
- ❒ Type of program processing.

Relatively large control units are assembled from individual components on a modular basis. Using a modular system like this, it is possible to start from a basic configuration and build up PLC systems that you can customize for specific applications. For relatively small control tasks, manufacturers offer compact control units. These are self-contained devices that have a fixed number of inputs and outputs.

2.1 Structure of a PLC

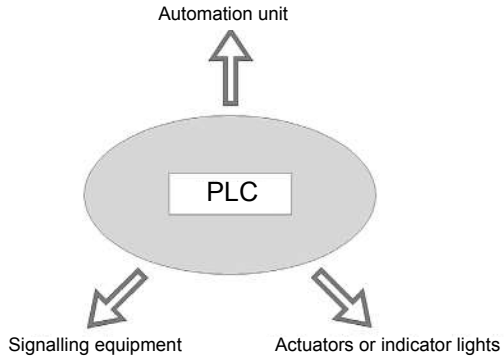


Figure 2.1
Basic structure of a programmable logic controller

2.2 Structure of an automation unit

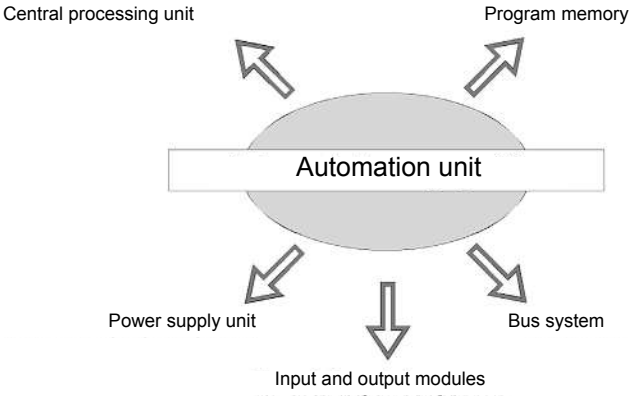


Figure 2.2
Basic structure of an automation unit

- ❒ Signalling equipment supplies digital and analog signals for further processing to the PLC,
- ❒ Actuators or indicator lights receive digital and analog signals from the PLC.

2.3 Hardware requirements

To be able to work with the examples below, you need the following hardware components.

DIN rail
The individual modules are mounted on the DIN rail.

Power supply unit (PS)
The power supply unit converts the 120/230 V AC mains supply to 24 V DC to supply the S7-300 modules.

Central processing unit (CPU)
The central processing unit is for executing the user programs. It communicates with other modules via the MPI interface.

Input and output module
The system inputs signals from the encoder via the input module into the S7-300 or outputs them to the output module. An LED display shows the signal state at the modules.

MPI cable
The MPI cable connects the computer to the central processing unit (CPU). Without this cable it is not possible to communicate with the PC.

Personal computer (PC)
The PC is for configuring, parameterizing and programming the S7-300.

Programming unit (PU)
The PU is also for configuring, parameterizing and programming the S7-300.

For this course, we will be using as the setup an S7-300 (CPU 314) with one digital input module (16 inputs) and one digital output module (16 outputs). In this connection, the input module contains addresses I 0.0...I 1.7 (IW0) and the output module contains addresses Q 4.0...Q 5.7 (QW4).

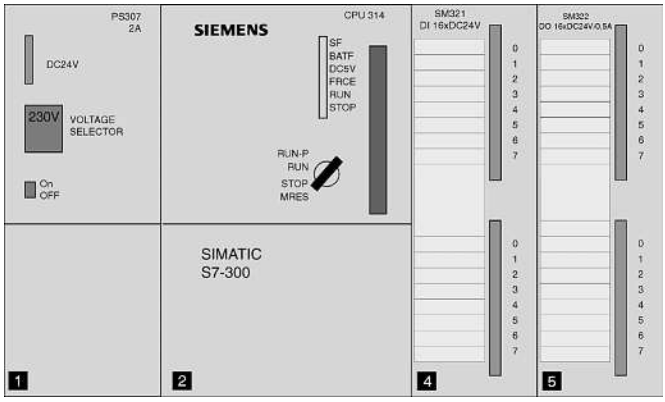


Figure 2.3
Structure of the training unit

2.3.1 Hardware structure

When setting up an S7-300, you must observe slot rules. You plug in the modules from left to right:

- ❒ You must always plug in the power supply unit (PS) as the first module on the DIN rail,
- ❒ You must plug in the central processing unit (CPU) as the second module,
- ❒ You plug in the input and output modules after the CPU,
- ❒ Next to the central processing unit (CPU), you may only plug in at a maximum 8 signal modules,
- ❒ You can plug in the modules horizontally or vertically.

2.4 Software requirements

For programming the S7-300, we use the Windows XP operating system as well as **Version 3.xx** of the **STEP 7 software package.** This version integrates the following three methods of representation: Statement List (STL), Ladder Diagram (LAD) and Function block diagram (FBD). Previous versions, e.g. 2.xx did not contain the Function block diagram (FBD) representation.

For programming the exercises in the book, we assume that Siemens' STEP 7 software package is already installed. To be able to use the STEP 7 programming language, you also need a knowledge of the Windows XP operating system (file management). You must carry out installation on the Windows XP user interface.

2.4.1 STEP 7 programming language

Following the replacement of SIMATIC S5 by SIMATIC S7 in the Autumn of 1995, a new programming language (STEP 7) was developed based on the IEC 1131 standard. This programming language provides the entire functionality for parameterizing, configuring and programming the S7-300 automation unit. STEP 7 runs under Windows XP and works on an object-oriented basis. The symbols for the objects are mapped on the graphical user interface. STEP 7 has an integrated online help system that provides valuable hints and tips.

2.4.2 Objects

On the graphical user interface, the system represents objects with symbols.
A symbol is assigned to a specific object. STEP 7 objects offer access to the following processing functions:

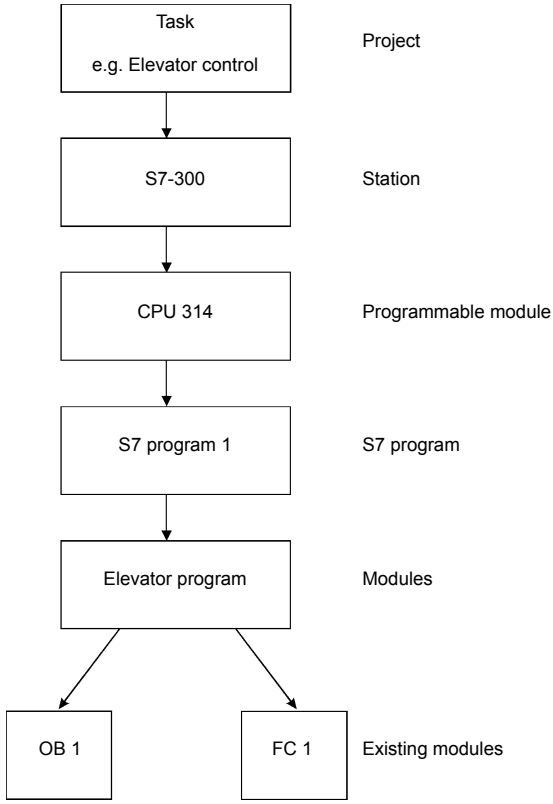
- ❒ Create and open objects,
- ❒ Edit and save objects,
- ❒ Rename objects,
- ❒ Delete objects,
- ❒ Copy and paste objects.

An object can, for example, contain as a symbol a project, a SIMATIC station, an S7 program, etc.

2.4.3 Projects

STEP 7 allows you to divide a plant into projects. A project contains all the data for an automation solution. You can consider a project to be the most important object.

Figure 2.4
Project structure with SIMATIC S7



The data for an automation solution is managed in a project. In our example, the project is a lift controller. This project contains the entire automation solution. The project includes, for example, the S7-300 station as well as the programmable controller (PC) with the CPU 314. You enter the S7 program in this PC. The program is then broken down into the corresponding blocks, such as, OB 1 and FC 1, for example.

2.4.4 Configuring an S7-300

When configuring, the S7 modules are arranged in a configuration table.
You can choose the modules from an electronic catalogue and enter them in the configuration table to match the slot. The slot in the configuration table corresponds to the slot on the subrack. An address is then automatically assigned to each module.

2.4.5 Parameterization

You can set the properties of the individual modules in various ways. One of the parameters is, for example, cycle time monitoring with the CPU S7-300. The parameter can be changed.

3 Way of Functioning of a PLC

3.1 Modules of the PLC

The various modules of a PLC are explained in Figure 3.1.

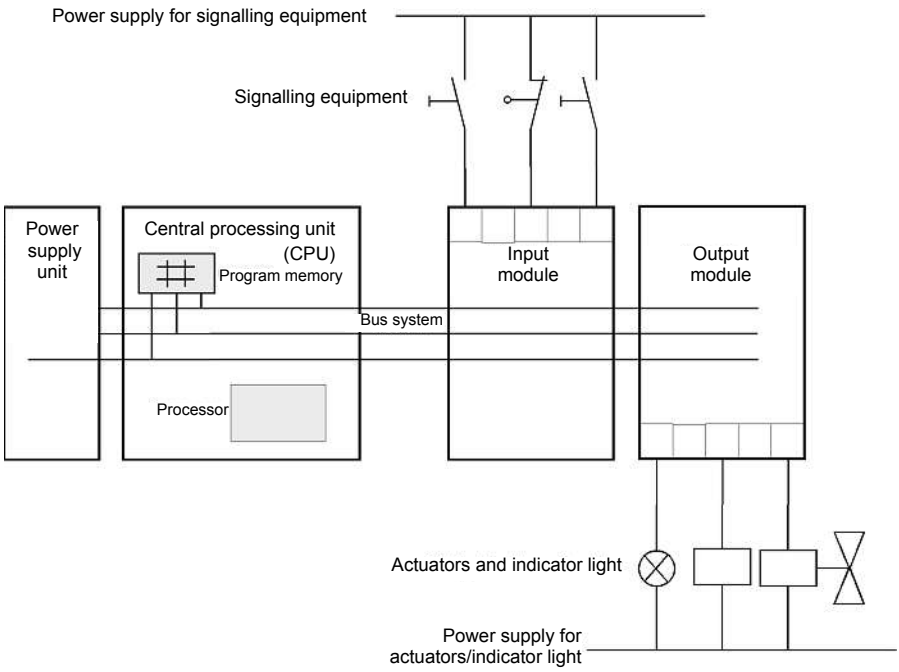


Figure 3.1 Interaction and arrangement of the modules of a PLC

3.1.1 Power supply unit

The power supply unit generates the voltage for the programmable controller's electronic modules from the mains voltage. This voltage is 24 V DC. The voltages for signalling equipment, actuators and indicator lights, which are above 24 V (24 to 220 V), are supplied by mains units or control-power transformers that are additionally provided for this purpose.

3.1.2 Program memory

Storage elements are components in which information can be stored in the form of binary signals. Semi-conductor components are mainly used as program memory. A memory comprises:

- ❒ 512,
- ❒ 1024,
- ❒ 2048 etc. storage locations.

It is normal to state the capacity of program memory (i.e. the number of storage locations) in **multiples of 1 K** (here, 1 K stands for 1024). In each storage location, you can use a programming unit to write (program) a control instruction. In this connection, each of the binary elements of a storage location can take on a signal state of "1" or "0".

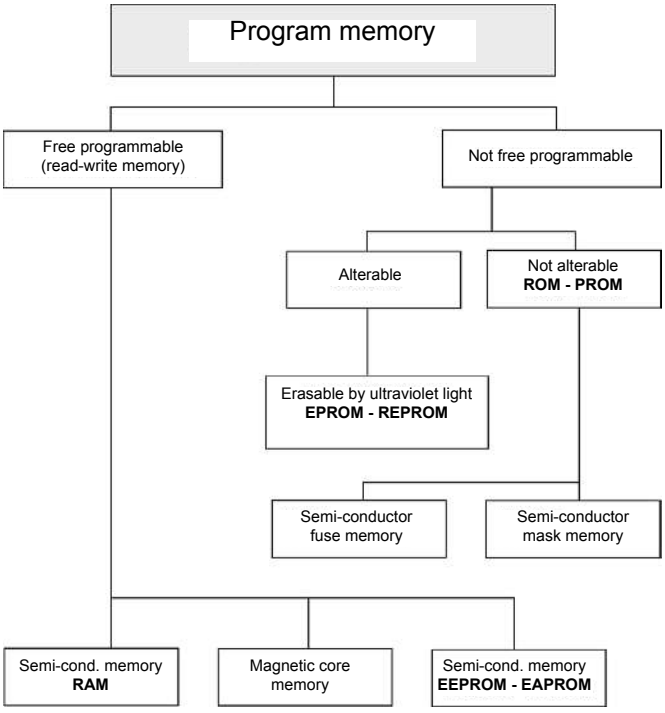


Figure 3.2
Schematic of a program memory

RAM (random-access memory)
RAM is semi-conductor Read-Write memory. The individual storage locations are labelled with addresses that are used to freely access the storage locations.

It is possible to write information to the storage locations as often as you like. The information is read out without its content being lost.

However, RAM is volatile memory, i.e., the information is lost when the power supply fails or is switched off. RAM is deleted electrically.

ROM
Read-only memory contains information that cannot be deleted or changed. ROM stands for Read-Only memory. Manufacturers enter the information and they are the only ones who can change it.

EPROM
EPROM stands for erasable programmable read-only memory. You can erase the entire content of the EPROM using UV light; after this, you can reprogram it. This makes it very suitable for transportation without data loss.

REPROM
REPROM stands for reprogrammable erasable read-only memory. You can also only erase its contents using UV light.

EEPROM
EEPROM stands for electrically erasable programmable read-only memory. In an EEPROM, it is possible to delete and write to each storage location electrically.

EAPROM
EAPROM stands for electrically alterable programmable read-only memory.

3.1.3 Central processing unit (CPU)

The voltage coming from the signals is switched to the terminal strip of the input module. In the CPU (central processing unit, see Figure 3.3), the processor processes the program in memory and when doing this polls whether the individual inputs of the unit are carrying a voltage or not. Depending on this status at the inputs and the program in memory, the processor instructs the output module to switch a voltage to the corresponding connections of the terminal strip. Again, depending on the voltage status at the connections of the output module, the system switches on or off the connected actuators or indicator lights.

The address counter polls successively (on a serial basis) the program memory instruction by instruction and brings about program-dependent transfer of information from the program memory to the statement register. All the memory in a processor is normally referred to as registers.

The processor gets its instructions from the statement register. While the processor is processing the current statement, the address counter shifts the next statement into the statement register.

The status transfer of the inputs to the process input image (PII) is followed by linking, application of the timers, counters, and accumulators and transfer of the results of logic operations (RLO) to the process output image (PIQ). If the system detects a block end (BE) after processing of the user program, the respective status is transferred from the PIQ to the outputs.

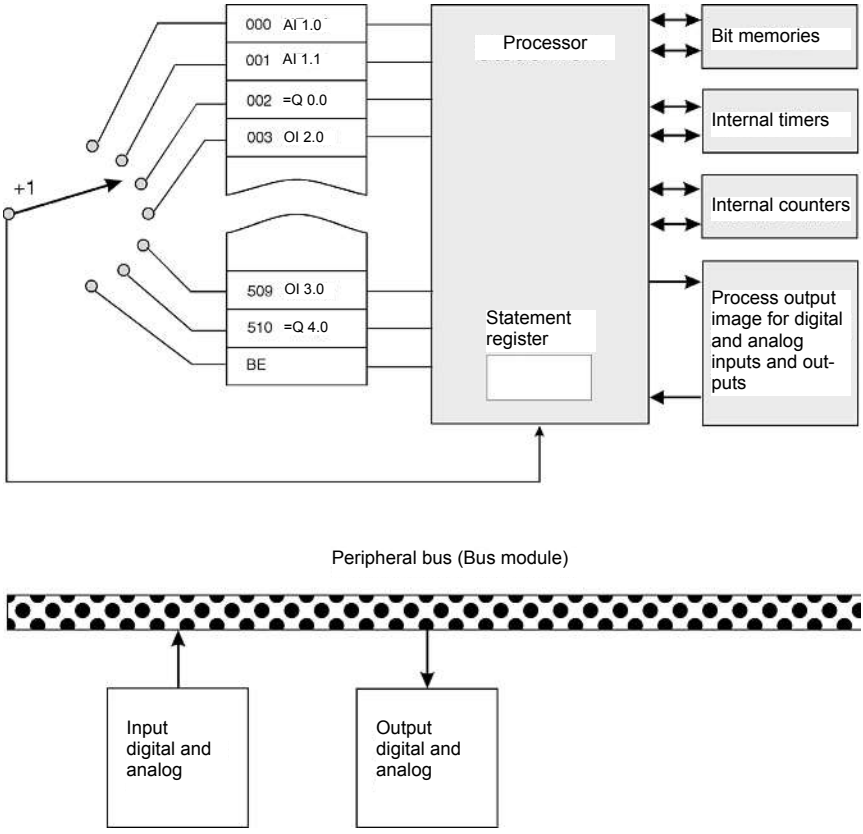


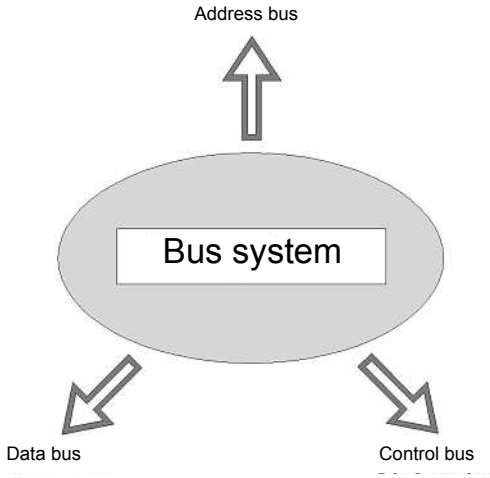
Bild 3.3 Central processing unit (CPU)

The peripheral bus handles data exchange between the central processing unit and the peripherals. Peripherals include the digital input and output modules, the analog input and output modules as well as the timer, counter and limit value modules.

3.1.4 Bus system

The bus system (see Figure 3.4) is a group line for transferring signals. The system exchanges signals in the programmable controller (PC) between the processor and the input and output modules across what is known as a process bus system. The process bus consists of three parallel signal lines:

Figure 3.4
The bus system



- ❒ Using the address bus, the system addresses the addresses on the individual modules,
- ❒ Using the data bus, the system transfers data, e.g. from the input to the output modules,
- ❒ On the control bus, the system transfers the signals for controlling and monitoring the function cycle within the programmable controller.

3.1.5 Input and output modules

Using the input and output modules, the system exchanges data with the process to be controlled.

4 Program Processing and Programming

You have the option of linear and structured program processing.

4.1 Linear program processing

Linear program processing (see Figure 4.1) is mostly used for simple, not too comprehensive controllers and can be implemented in a single organization block (OB). In this connection, the control unit processes the statements in the order in which they are stored in the program memory. When the end of program (BE) is reached, program processing starts from the beginning again. The time that a control unit needs to carry out processing of all the statements once is called the cycle time. This is why this type of processing is also referred to as cyclical processing.

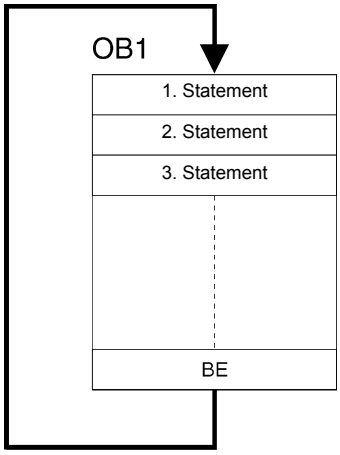


Figure 4.1
Linear program processing

4.2 Structured programming

With comprehensive control jobs, you divide the program into small manageable program blocks that are arranged by function. This has the advantage of allowing you to test program sections individually and, if they function, to combine them into an overall function. STEP 7 provides the following user blocks for this:

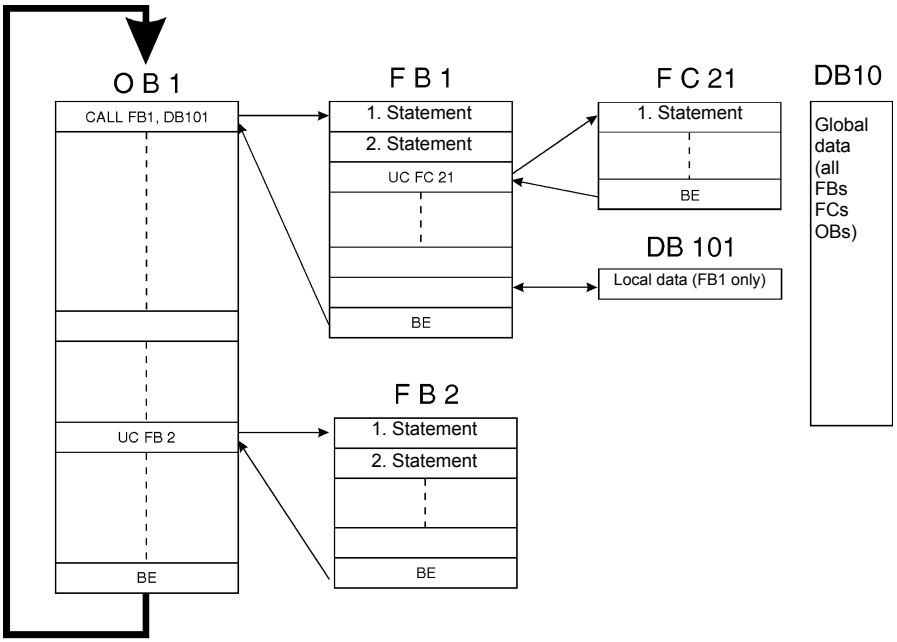


Figure 4.2 Structured programming using user blocks with: OB organization block, FB function block, FC function, DB data block

Organization bock (OB)
The operating system cyclically calls an OB which functions as the interface between the user program and the operating system. In the OB, the system informs the processor of the programmable controller (PLC) by means of block call commands about the program blocks that it has to process.

Function block (FB)
FBs have an assigned memory area. When an FB is called, it is possible to assign a data block (DB) to it. It is possible to access the data in this instance DB by means of calls from the FB. An FB can be assigned to different DBs. It is possible to call further FBs and FCs via block call commands in a function block.