SIEMENS

# Nachtrag/Supplement/Additif

C79000-Q8563-C886-01

## Grundfunktionen
## Standard Funktionsbausteine

## Basic Functions
## Standard Function Blocks

## Fonctions de base
## Blocs fonctionnels standard

## C79000-G8563-C886-01

**Deutsch**
Dieser Nachtrag enthält Ergänzungen zur Beschreibung der Standard-Funktionsbausteine der Grundfunktionen. Es werden die Standard-Funktionsbausteine für die CPU 945 und die CPU 948 beschrieben.

**English**
This supplement contains additional information about the standard function blocks for basic functions. The standard function blocks for the CPU 945 and CPU 948 are described.

**Français**
Cet additif au manuel contient des informations complémentaires sur les blocs fonctionnels standard des fonctions de base pour les CPU 945 et CPU 948.

# 1 General

## 1.1 CPU 945

The characteristics described in the manual under the heading "S5-115U Programmable Controller" apply to the CPUs 941 to 944. The essential differences compared with the CPU 945 are as follows:

> There are new function blocks for the CPU 945 which cannot be used in the previous processors CPU 941 to CPU 944. These are therefore stored in their own file (see below).

> The extended I/O area "O" can also be addressed with the function blocks for the CPU 945. The modules addressed in the O area must be operated in an extension unit addressed in this way. *Addressing the O area is not possible with the CC 115U.*

> DX blocks can now be addressed with the CPU 945.

## 1.2 CPU 948

The function blocks for the CPU 946/947 are also suitable for operation in the CPU 948. The functions and handling remain the same.

## 1.3 Diskettes Supplied

The new release of the diskettes is A02.

Order no. for S5-DOS: 6ES5 848-8AA02
Order no. for MS-DOS, S5-DOS/MT: 6ES5 848-7AA02

Changes compared with release 01:

The function blocks for the S5-115U, CPU 945 have been added.

| | File |
|---|---|
| S5-95U/100U, CPU 103 | S5AA01ST.S5D |
| S5-115U, CPU 941 to 944 | S5AA52ST.S5D |
| S5-115U, CPU 945 | S5AA55ST.S5D |
| S5-135U, CPU 922, CPU 928, CPU 928B [1)] | S5AA20ST.S5D |
| S5-155U, CPU 946/947, CPU 948 [1)] | S5AA60ST.S5D |

1) With the CPU 922, CPU 928 or CPU 928B in the S5-155U, the file S5AA20ST.S5D must be used.

# 2 Function Blocks for the S5-115U, CPU 945

## 2.1 Overview

| Block | Name | Significance, |
|---|---|---|
| FB 3 | MUL:32 | 32 bit binary multiplier |
| FB 4 | DIV:32 | 32 bit binary divider |
| FB 5 | RAD:16 | 16 bit binary root extractor |
| FB 6 | RAD:GP | Floating point root extractor |
| FB 10 | REG:SCHB | Shift register (bit) |
| FB 11 | REG:SCHW | Shift register (word) |
| FB 12 | REG:FIFO | FIFO buffer |
| FB 13 | REG:LIFO | LIFO buffer |
| FB 38 | RETTEN | Save scratchpad flags |
| FB 39 | LADEN | Load scratchpad flags |
| FB 44 | DB-COPY | Copy data areas |
| FB 45 | DB-COPY | Copy data areas |
| FB 196 | PER:ET | Read and write for extended I/Os |
| FB 197 | PER:ET | Read and write for extended I/Os |

The calls for function blocks FB 3 to FB 13 and their parameter assignment can be found in the description.

## 2.2 Function Blocks FB 38 and FB 39

The function block FB 38:RETTEN saves data from the "scratchpad flag area" in the data block assigned to it when cyclic program execution is interrupted by timed interrupts or event-driven interrupts.

The function block FB 39:LADEN transfers the data from the data block assigned to it back to the "scratchpad flag area" following an interruption in the cyclic program due to timed interrupts or event-driven interrupts.

The function blocks FB 38:RETTEN and FB 39:LADEN must always be used as pairs for all time-controlled or interrupt-driven programs and must use the same data block. They can be called in a maximum of 10 different interrupt levels and save the scratchpad flag area (FY 200 to FY 255) and certain system data.

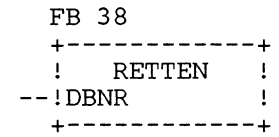
The data block, in which the saved data are stored must be initialized up to and including DW 440.

### Calling function block FB 38

in STL:

```
       JU  FB 38
NAME : RETTEN
DBNR :
```

in LAD/CSF:



### Calling function block FB 39

in STL:

```
       JU  FB 39
NAME : LADEN
DBNR :
```

in LAD/CSF:

```
FB 39
+------------+
!   LADEN    !
--!DBNR       !
+------------+
```

### Explanation of the parameters

| NAME | PARA TYPE | DATA TYPE | MEANING |
|---|---|---|---|
| DBNR | B | - | Data block, in which the data will be saved (1 <= DBNR <= 255) |

***Note:*** *In case of error, the CPU changes to STOP.* The error number is then located in accumulator 1:

$F001_H$ DB for saving/loading does not exist
$F002_H$ Stack overflow
$F003_H$ DB for saving/loading too short

## 2.3 Function Blocks FB 44 and FB 45

### Calling function block FB 44

in STL:

```
       JU  FB 44
NAME : DB-COPY
QTYP :
QANF :
LAEN :
ZTYP :
ZANF :
FEHL :
     :
```

in LAD/CSF:

```
FB 44
+------------+
!  DB-COPY   !
--!QTYP    FEHL!--
--!QANF        !
--!LAEN        !
--!ZTYP        !
--!ZANF        !
+------------+
```

### Calling function block FB 45

```
     :L   KY   x,y
     :L   KF   +z
     :JU  FB 45
NAME :DB-COPY
     :T   FY   a
```

Assignment of the accumulators 2 and 1 before calling function block FB 45:

Accu 2: KY = x,y
x = block, in which the parameter field is located:
x = 1: DB block
x = 2: DX block
y = DB/DX number: 1 <= y <= 255

Accu 1: KF = +z
address of the first parameter (number of the 1st data word) within the specified block.
0 <= z <= 251

The parameter field must assigned as shown below before calling the function block:

| | |
|---|---|
| DW z | QTYP |
| DW (z+1) | QANF |
| DW (z+2) | LAEN |
| DW (z+3) | ZTYP |
| DW (z+4) | ZANF |

**Explanation of the parameters**

| NAME | PARA TYPE | DATA TYPE | MEANING |
|---|---|---|---|
| QTYP | D | KY | Specifies the source DB type and DB number |
| QANF | D | KF | Source start data word |
| LAEN | D | KF | Length of the data area to be copied (number of data words) |
| ZTYP | D | KY | Specifies destination DB type and DB number |
| ZANF | D | KF | Destination start data word |
| FEHL | Q | BY | Error byte (not for FB 45) |

**Assignment of the parameters**

QTYP: KY = x,y x = block type x = 1: DB block
ZTYP: KY = x,y x = 2: DX block
y = DB/DX number: 1 <= y <= 255

QANF: KF = 0 to 2042
ZANF: KF = 0 to 2042

LAEN: KF = 0 to 255 QANF + LAEN <= 2043
ZANF + LAEN <= 2043

FEHL: contains the error number if the parameter assignment is illegal

With the function block FB 45, a parameter assignment error is indicated by RLO = "1" after exiting the block. The error number can then be read from the content of accu 1. Refer to the description for the error numbers.

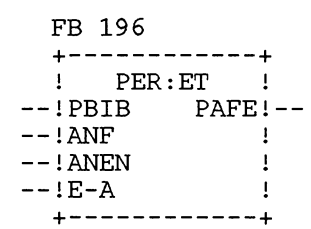
## 2.4 Function Blocks FB 196 and FB 197

**Calling function block FB 196**

in STL:

```
      JU  FB 196
NAME : PER:ET
PBIB :
ANF  :
ANEN :
E-A  :
PAFE :
```

in LAD/CSF:



### Calling function block FB 197

```
      :L    KY    x,y
      :L    KF    +z
      :JU   FB    197
NAME  :PER:ET
      :T    FY    a
```

Assignment of the accumulators 2 and 1 before calling function block FB 197:

Accu 2: KY = x,y

| x = area in which the parameter field is located: | | y = DB/DX number: |
|---|---|---|
| x = 00: | DB block | 1 <= y <= 255 |
| x = 01: | DX block | 1 <= y <= 255 |
| x = 02: | Flags | irrelevant |
| x = 03: | S flags | irrelevant |

If the parameter field is in a DB/DX block for indirect parameter assignment, this area must be initialized and values assigned before calling FB 197.

Accu 1: KF = +z address of the first parameter within the specified area:

| area at | | |
|---|---|---|
| | DW: | 0 <= z <= 2039 |
| | FY: | 0 <= z <= 193 |
| | SY: | 0 <= z <= 4089 |

e.g. z = 5: parameter field begins at
DW 5 (where x = 00 or x = 01)
FY 5 (where x = 02)
SY 5 (where x = 03)

The pre-assignment of the parameter field before calling the function block can be found in the description.

### Explanation of the parameters

| NAME | PARA TYPE | DATA TYPE | MEANING |
|---|---|---|---|
| PBIB | D | KY | Specifies the areas to be processed |
| ANF | D | KF | Start of internal area |
| ANEN | D | KY | Start and end of the field on the interface module |
| E-A | I | BI | Specifies the transfer direction |
| PAFE | Q | BI | Parameter assignment error |

### Assignment of the parameters

PBIB: KY = x,y x = specifies the types of data area between which the data transfer will take place

| x = | Data block DB | Data block DX | Flag area F | Flag area S |
|---|---|---|---|---|
| P area | 16 | 17 | 18 | 19 |
| Q area | 32 | 33 | 34 | 35 |
| IM3 area | 48 | 49 | 50 | 51 |
| IM4 area | 64 | 65 | 66 | 67 |

y = DB/DX number
1 <= y <= 255 (when transferring from/to a data block)
y = irrelevant (when transferring from/to the flag area)

ANF: Start of the internal area (DW number or flag byte number):
The parameter specifies from which address onwards the filed to be transferred begins in the data or flag area.

| | | |
|---|---|---|
| KF = +x | 0 <= x <= 2042 | DW number with data blocks |
| | 0 <= x <= 199 | byte number with F flags |
| | 0 <= x <= 4095 | byte number with S flags |

ANEN: KY = x,y x = number of the start byte 0 <= x <= 255
y = number of the end byte 0 <= y <= 255
y - x <= 254 (a maximum of 255 bytes can be transferred at once)

E-A: Transfer direction: signal state "0": read inputs
signal state "1": write outputs

PAFE: Parameter assignment error: if the parameter assignment for FB 196 is illegal, the PAFE parameter has signal state "1". The error number can then be read from the content of accu 1.

With the function block FB 197 a parameter assignment error is indicated by RLO = "1" after exiting the block. The error number can then also be read from the content of accu 1.

Error number:

1: illegal area ID
2: illegal DB/DX number
3: DB/DX does not exist/too short
4: parameter ANF illegal
5: parameter ANEN illegal
6: Area exceeded
7: illegal area ID in accu 2 (only with FB 197)
8: illegal DB/DX number in accu 2 (only with FB 197)
9: start of the parameter field (accu 1) illegal (only with FB 197)
10: DB/DX does not exist/too short (only with FB 197)

## 2.5 Technical Data

| Block name | LIB no. P71200-S... | Call length (words) | Block length (words) | Data area occupied |
|---|---|---|---|---|
| FB 3 MUL:32 | 3003-A-1 | 11 | 209 | - |
| FB 4 DIV:32 | 3004-A-1 | 14 | 203 | - |
| FB 5 RAD:16 | 3005-A-1 | 6 | 126 | - |
| FB 6 RAD:GP | 3006-A-1 | 5 | 127 | - |
| FB 10 REG:SCHB | 3010-A-1 | 14 | 250 | DW 0 and DW 1 of the open data block |
| FB 11 REG:SCHW | 3011-A-1 | 10 | 120 | DW 0 and DW 1 DB of the data block specified in parameter DB; shift register located in the data block up to DW (k+1) |
| FB 12 REG:FIFO | 3012-A-1 | 11 | 148 | DW 0 to DW 4 of the data block specified in parameter DB; FIFO register located in the data block up to DW (k+4) |
| FB 13 REG:LIFO | 3013-A-1 | 11 | 110 | DW 0 and DW 1 of the data block specified in parameter DB; LIFO register located in the data block up to DW (k+1) |
| FB 38 RETTEN | 3038-A-1 | 3 | 93 | Assigned data block up to and including DW 440 |
| FB 39 LADEN | 3039-A-1 | 3 | 86 | |
| FB 44 DB-COPY | 3044-A-1 | 8 | 247 | - |
| FB 45 DB-COPY | 3045-A-1 | 2 | 318 | DW z to DW (z+4) in specified data block |
| FB 196 PER:ET | 3196-A-1 | 7 | 214 | DB/DX block according to parameter assignment |
| FB 197 PER:ET | 3197-A-1 | 2 | 282 | |

- Nesting depth: 0
- Secondary blocks: none
- Flag area occupied: FY 200 to FY 255
- System area occupied: -
- Misc.: -

## 2.6 Processing times

| | | | | | |
|---|---|---|---|---|---|
| FB 3 | 10 µs - 35 µs | | | | |
| FB 4 | 15 µs - 120 µs | | | | |
| FB 5 | 45 µs | | | | |
| FB 6 | 40 µs | | | | |
| FB 10 | 45 µs | with 8 Bits | 280 µs | with 128 Bits | |
| FB 11 | 145 µs | for SR | 180 µs | for SL | if k = 64 |
| | 535 µs | for SR | 675 µs | for SL | if k = 254 |
| FB 12 | 15 µs | for S, R | 16 µs | for LO | |
| FB 13 | 10 µs | for S, R | 12 µs | for LO | |
| FB 38 | 133 µs | | | | |
| FB 39 | 133 µs | | | | |
| FB 44 | 25 µs - 115 µs [1)] | | | | |
| FB 45 | 35 µs - 125 µs [1)] | | | | |
| FB 196 | 25 µs - 115 µs [1)] | | | | |
| FB 197 | 35 µs - 125 µs [1)] | | | | |

[1)] The processing times depend on the number of bytes to be transferred.

# 3 Function Blocks for the S5-155U, CPU 946/947 and CPU 948

## 3.1 Technical Data

| Block name | LIB no. P71200-S... | Call length (words) | Block length (words) | Data area occupied |
|---|---|---|---|---|
| FB 3 ADD:32 | 6003-A-1 | 10 | 69 | - |
| FB 7 SUB:32 | 6007-A-1 | 10 | 69 | - |
| FB 11 MUL:32 | 6011-A-1 | 11 | 197 | - |
| FB 15 DIV:32 | 6015-A-1 | 14 | 203 | - |
| FB 18 RAD:16 | 6018-A-1 | 6 | 128 | - |
| FB 19 RAD:GP | 6019-A-1 | 5 | 129 | - |
| FB 24 REG:SCHB | 6024-A-1 | 14 | 252 | DW 0 and DW 1 of the open data block |
| FB 25 REG:SCHW | 6025-A-2 | 10 | 117 | DW 0 and DW 1 of the data block specified in parameter DB; shift register located in the data block up to DW (k+1) |
| FB 26 REG:FIFO | 6026-A-1 | 11 | 162 | DW 0 to DW 4 of the data block specified in parameter DB; FIFO register located in the data block up to DW (k+4) |
| FB 27 REG:LIFO | 6027-A-1 | 11 | 119 | DW 0 and DW 1 of the data block specified in parameter DB; LIFO register located in the data block up to DW (k+1) |
| FB 30 AE:464 | 6030-A-1 | 11 | 386 | - |
| FB 31 AE:460 | 6031-A-1 | 13 | 305 | - |
| FB 32 AE:463 | 6032-A-1 | 11 | 221 | - |
| FB 33 AE:466 | 6033-A-1 | 11 | 254 | - |
| FB 38 RETTEN | 6038-B-1 | 3 | 105 | Assigned data block up to and including DW 816 |
| FB 39 LADEN | 6039-B-1 | 3 | 96 | |
| FB 41 RLG:AA | 6041-B-1 | 12 | 105 | - |
| FB 44 DB-COPY | 6044-A-1 | 8 | 233 | - |
| FB 45 DB-COPY | 6045-A-1 | 2 | 303 | DW z to DW (z+4) in specified data block |
| FB 129 SST:UHR | 6129-B-1 | 2 | 53 | DB 5: DW 0 to DW 11 |
| FB 196 PER:ET | 6196-B-3 | 7 | 301 | DB/DX block according to parameter assignment |
| FB 197 PER:ET | 6197-B-1 | 2 | 369 | |

- Nesting depth: 0
  FB 38/39 and FB 129: nesting depth: 1 (special operating system functions are called which count as normal block calls)
- Secondary blocks: none
- Flag area occupied: FY 200 to FY 255
- System area occupied: -
- Misc.: -

## 3.2 Processing times for the CPU 948

| | | | | | |
|---|---|---|---|---|---|
| FB 3 | 30 µs | | | | |
| FB 7 | 30 µs | | | | |
| FB 11 | 110 µs | | | | |
| FB 15 | 200 µs | | | | |
| FB 18 | 120 µs | | | | |
| FB 19 | 90 µs | | | | |
| FB 24 | 40 µs- 1490 µs | | | | |
| FB 25 | 360 µs | for SR | 435 µs | for SL | if k = 64 |
| | 1370 µs | for SR | 1640 µs | for SL | if k = 254 |
| FB 26 | 35 µs | | | | |
| FB 27 | 30 µs | | | | |
| FB 30 | 95 µs | | | | |
| FB 31 | 95 µs | | | | |
| FB 32 | 80 µs | | | | |
| FB 33 | 90 µs | | | | |
| FB 38 | 110 µs | | | | |
| FB 39 | 105 µs | | | | |
| FB 41 | 53 µs | | | | |
| FB 44 | 65 µs - 315 µs [1)] | | | | |
| FB 45 | 90 µs - 340 µs [1)] | | | | |
| FB 196 | 70 µs - 1200 µs [1)] | | | | |
| FB 197 | 90 µs - 1400 µs [1)] | | | | |

[1)] The processing times depend on the number of bytes to be transferred.

SIEMENS

# SIMATIC S5
# Grundfunktionen
# Standard-Funktionsbausteine

**Automatisierungsgeräte**
**S5-95U, -100U, -115U, -135U, -155U**

**Beschreibung**

# Basic Functions
# Standard Function Blocks

**S5-95U, -100U, -115U, -135U, -155U**
**Programmable Controllers**

**Description**

# Fonctions de base
# Blocs fonctionnels standards

**Automates programmables**
**S5-95U, -100U, -115U, -135U, 155U**

**Description**

C79000-G8563-C886-01

C79000-H8563-C886-01

***Copyright***

Copyright © Siemens AG 1993 All Rights Reserved
Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung Ihres Inhalts ist nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadensersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patentverteilung oder GM-Eintragung.

***Haftungsausschluß***

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so daß wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden jedoch regelmäßig überprüft und notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten. Für Verbesserungsvorschläge sind wir dankbar.

Technische Änderungen bleiben vorbehalten.

***Copyright***

Copyright © Siemens AG 1993 All Rights Reserved
The reproduction, transmission or use of this document or its contents is not permitted without express written authority.
Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

***Disclaimer of Liability***

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.

Technical data subject to change.

***Copyright***

Copyright © Siemens AG 1993 Tous droits réservés
Toute communication ou reproduction de ce support d'information, toute exploitation ou communication de son contenu sont interdites, sauf autorisation expresse. Tout manquement à cette règle est illicite et expose son auteur au versement de dommages et intérêts. Tous nos droits sont réservés, notamment pour le cas de la délivrance d'un brevet ou celui de l'enregistrement d'un modèle d'utilité.

***Exclusion de responsabilité***

Nous avons vérifié la conformité du contenu du présent manuel avec le matériel et le logiciel qui y sont décrits. Or des divergences n'étant pas exclues, nous ne pouvons pas nous porter garants pour la conformité intégrale. Si l'usage du manuel devait révéler des erreurs, nous en tiendrons compte et apporterons les corrections nécessaires dès la prochaine édition. Veuillez nous faire part de vos suggestions.

Nous nous réservons le droit de modifier les caractéristiques techniques.

SIEMENS

# SIMATIC S5

## Basic Functions
## Standard Function Blocks

Programmable controllers S5-95U, -100U, -115U, -135U, -155U

**Description**

Bestell-Nr. C79000-B8576-C886-01

# Contents


| | | Page |
|---|---|---|
| **1** | **Introduction** | **1- 1** |
| **2** | **Arithmetic functions** | **2- 1** |
| 2.1 | Overview | 2- 1 |
| 2.2 | Description | 2- 2 |
| | - 32-bit binary adder ... FB1/FB3 | 2- 2 |
| | - 32-bit binary subtractor ... FB2/FB7 | 2- 4 |
| | - 32-bit binary multiplier ... FB3/FB11 | 2- 6 |
| | - 32-bit binary divider ... FB4/FB15 | 2- 9 |
| | - 16-bit binary root extractor ... FB5/FB18 | 2-12 |
| | - Floating point root extractor ... FB6/FB19 | 2-14 |
| **3** | **Registers** | **3- 1** |
| 3.1 | Overview | 3- 1 |
| 3.2 | Description | 3- 2 |
| | - Shift register (bit) ... FB10/FB24 | 3- 2 |
| | - Shift register(word) ... FB11/FB25 | 3- 7 |
| | - Buffer (FIFO) ... FB12/FB26 | 3-10 |
| | - Stack (LIFO) ... FB13/FB27 | 3-14 |
| **4** | **Analog input/output** | **4- 1** |
| 4.1 | Overview | 4- 1 |
| 4.2 | Description | 4- 2 |
| | - Reading in the analog value: ... AE:464 FB 30 | 4- 2 |
| | - Reading in the analog value: ... AE:460 FB 31 | 4- 8 |
| | - Reading in the analog value: ... AE:463 FB 32 | 4-12 |
| | - Reading in the analog value: ... AE:466 FB 33 | 4-17 |
| | - Outputting an analog value ... FB41 | 4- 22 |

**5 Special functions ..... 5- 1**

5.1 Overview ..... 5- 1

5.2 Description ..... 5- 2

- Date and time ..... FB129 ..... 5- 2
- Read/write extended peripherals ... FB196/FB197 (PER:ET) ..... 5- 4
- Save/load ..... 5-14
- Code converter ..... 5-16
- Copying date areas ..... 5-18

**6 Technical data ..... 6- 1**

# 1 Introduction

The standard function blocks for basic functions described below include the following:

- Arithmetic functions
  (addition, subtraction, multiplication, division and root extraction)

- Registers
  (shift register, buffer and stack)

- Analog value processing

- Special functions

The standard function blocks use flag bytes as "scratchpad flags" i.e. the flags are used to both generate and store intermediate results. These flags can be used by other (user-generated) function blocks and by the remaining program only as "scratchpad flags", however, they must not be used as "static" flags.

# 2 Arithmetic functions

The standard function blocks for arithmetic functions replace functions (operations) which cannot be carried out directly by the programmable controller, i.e. these operations are not contained in the operation set of the programmable controller. This includes all arithmetic operations with double word operands (width of the values: 32 bits) as well as root extraction (width of the values 16 and 32 bits).

## 2.1 Overview

**32-bit binary adder**

This function block adds two fixed point binary numbers (31 bits + sign) the result is also a fixed point binary number.

**32-bit binary subtractor**

This function block subtracts two fixed point binary numbers (31 bits + sign). The result is also a fixed point binary number.

**32-bit binary multiplier**

This function block multiplies two fixed point binary numbers (31 bits + sign). The result is also a fixed point binary number (63 bits + sign).

**32-bit binary divider**

This function block divides two fixed point binary numbers (31 bits + sign). The result is also a fixed point binary number.

**32-bit binary root extractor**

This function block extracts the square root of a fixed point binary number (15 bits + sign). The result is also a fixed point binary number (8 bit root, 16-bit remainder).

**Floating point root extractor**

This function block extracts the square root of a floating point number (7-bit exponent + sign, 23-bit mantissa + sign). The result is also a floating point number.

## 2.2 Description

## 32-bit binary adder

### Functional description ADD : 32

The function block ADD : 32 adds two fixed point binary numbers (31 bits + sign). The result is also a fixed point binary number (31 bits + sign).

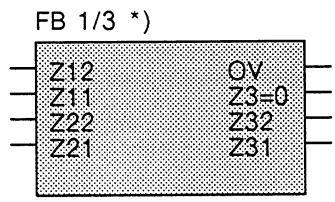
The function block ADD : 32 sets the following identifiers for further processing: overflow of the numerical range; result of addition is zero.

### Calling the function block

Representation in STL

```
      : JU FB 1/3
NAME  : ADD:32
Z12   :
Z11   :
Z22   :
Z21   :
0V    :
Z3=0  :
Z32   :
Z31   :
```

Graphic representation

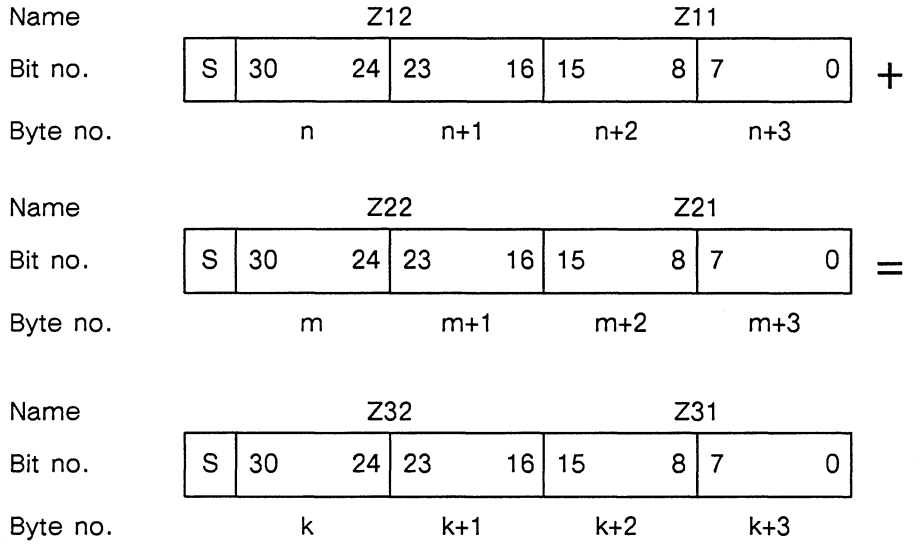



### Explanation of the inputs and outputs (parameters)

| Name | Para | Data | Designation | Notes |
|---|---|---|---|---|
| Z12 | I | W | 1st addend byte no. n, n+1 | Numerical range **) |
| Z11 | I | W | 1st addend byte no. n+2, n+3 | Numerical range **) |
| Z22 | I | W | 2nd addend byte no. m, m+1 | Numerical range **) |
| Z21 | I | W | 2nd addend byte no. m+2, m+3 | Numerical range **) |
| OV | Q | BI | Overflow | Overflow***) |
| Z3=0 | Q | BI | sum equals zero | - |
| Z32 | Q | W | sum byte no. k, k+1 | Numerical range **) |
| Z31 | Q | W | sum byte no. k+2, k+3 | Numerical range **) |

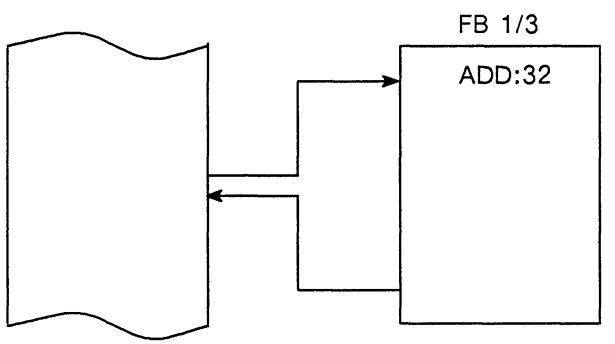
*) FB number dependent on the CPU used; see technical data
**) Numerical range of Z1, Z2 and Z3: $-2^{31}$ to $+2^{31}-1$
***) If overflow occurs, Z3 is cleared!

### Assignment of the parameters




S=Sign

### Program structure



## 32-bit binary subtractor

### Functional description SUB : 32

The function block SUB : 32 subtracts two fixed point binary numbers (31 bits + sign). The result is also a fixed point binary number (31 bits + sign).

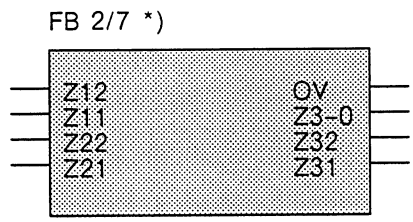
The function block SUB : 32 sets the following identifiers for further processing: overflow of the numerical range; result of subtraction is zero.

### Calling the function block

Representation in STL

```
      :  JU FB 2/7
NAME  :  SUB:32
Z12   :
Z11   :
Z22   :
Z21   :
OV    :
Z3=0  :
Z32   :
Z31   :
```

Graphic representation

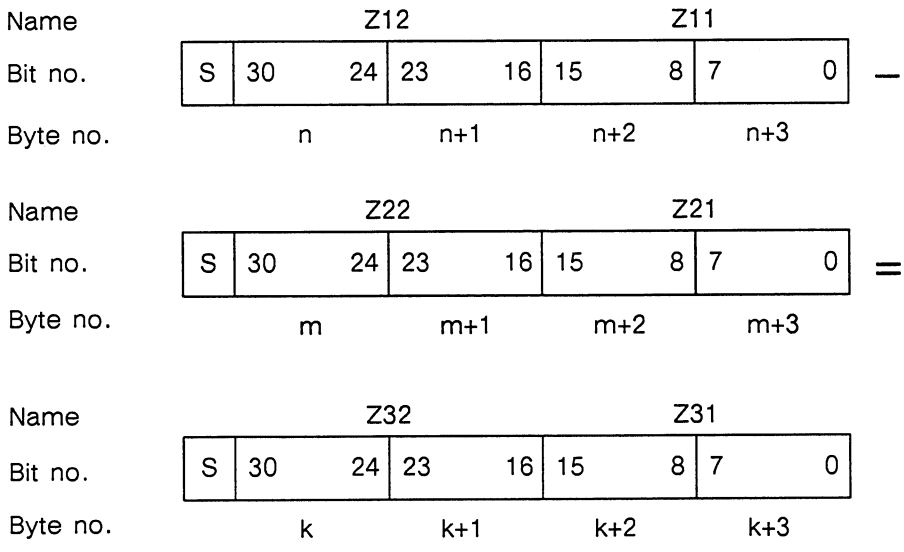



### Explanation of the inputs and outputs (parameters)

| Name | Para | Data | Designation | Notes |
|---|---|---|---|---|
| Z12 | I | W | Minuend byte no. n, n+1 | Numerical range **) |
| Z11 | I | W | Minuend byte no. n+2, n+3 | Numerical range **) |
| Z22 | I | W | Subtrahend byte no. m, m+1 | Numerical range **) |
| Z21 | I | W | Subtrahend byte no. m+2, m+3 | Numerical range **) |
| OV | Q | BI | Overflow | Overflow***) |
| Z3=0 | Q | BI | Difference equals 0 | - |
| Z32 | Q | W | Difference byte no. k, k+1 | Numerical range **) |
| Z31 | Q | W | Difference byte no. k+2, k+3 | Numerical range **) |

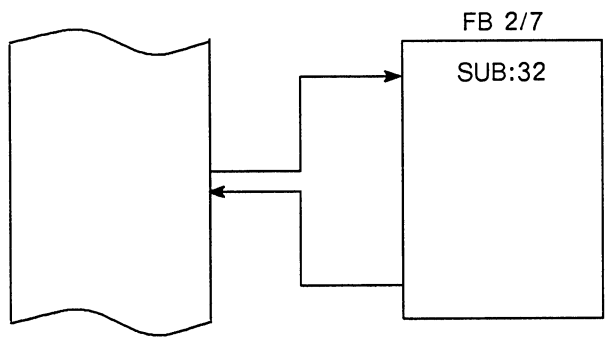
*) FB number dependent on the CPU used; see technical data
**) Numerical range of Z1, Z2 and Z3: $-2^{31}$ to $+2^{31}-1$
***) If overflow occurs, Z3 is cleared!

**Assignment of the parameters**




S = Sign

**Program structure**



## 32-bit binary multiplier

### Functional description MUL : 32

The function block MUL : 32 multiplies two fixed point binary numbers (31 bits + sign). The result is also a fixed point binary number (63 bits + sign).

The function block MUL : 32 sets the following identifiers for further processing: result of multiplication is zero.

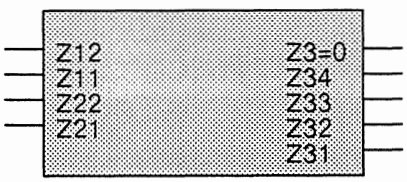
### Calling the function block

Representation in STL

```
      :  JU FB 3/11
NAME  :  MUL:32
Z12   :
Z11   :
Z22   :
Z21   :
Z3=0  :
Z34   :
Z33   :
Z32   :
Z31   :
```

Graphic representation

FB 3/11 *)




*) FB number dependent on the CPU used; see technical data

**Explanation of the inputs and outputs (parameters)**

| Name | Para | Data | Designation | Notes |
|---|---|---|---|---|
| Z12 | I | W | Multiplicand byte no. n, n+1 | Numerical range *) |
| Z11 | I | W | Multiplicand byte no. n+2, n+3 | Numerical range *) |
| Z22 | I | W | Multiplicator byte no. m, m+1 | Numerical range *) |
| Z21 | I | W | Multiplicator byte no. m+2, m+3 | Numerical range *) |
| Z3=0 | Q | BI | Product equals zero | - |
| Z34 | Q | W | Product byte no. k, k+1 | Numerical range **) |
| Z33 | Q | W | Product byte no. k+2, k+3 | Numerical range **) |
| Z32 | Q | W | Product byte no. k+4, k+5 | Numerical range **) |
| Z31 | Q | W | Product byte no. k+6, k+7 | Numerical range **) |

*) Numerical range of Z1, Z2 : $-2^{31}$ to $+2^{31}-1$
**) Numerical range of Z3 : $-2^{63}$ to $+2^{63}-1$

**Assignment of the parameters**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Name | | | Z12 | | | | Z11 | | |
| Bit no. | S | 30 | 24 | 23 | 16 | 15 | 8 | 7 | 0 |
| Byte no. | | n | | n+1 | | n+2 | | n+3 | |

X

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Name | | | Z22 | | | | Z21 | | |
| Bit no. | S | 30 | 24 | 23 | 16 | 15 | 8 | 7 | 0 |
| Byte no. | | m | | m+1 | | m+2 | | m+3 | |

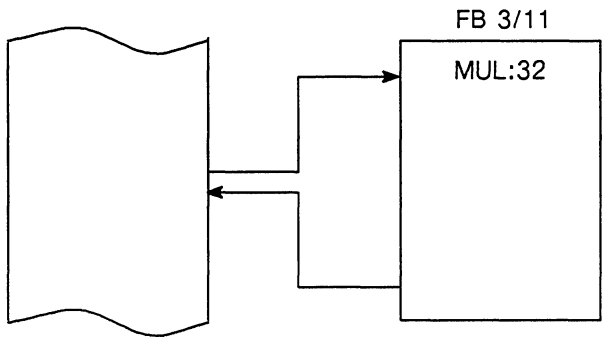
=

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Name | | | Z34 | | | | Z33 | | |
| Bit no. | S | 62 | 54 | 53 | 48 | 47 | 40 | 39 | 32 |
| Byte no. | | k | | k+1 | | k+2 | | k+3 | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Name | | | Z32 | | | | Z31 | | |
| Bit no. | 31 | | 24 | 23 | 16 | 15 | 8 | 7 | 0 |
| Byte no. | | k+4 | | k+5 | | k+6 | | k+7 | |

S = Sign

**Program structure**



## 32-bit binary divider

### Functional description DIV : 32

The function block DIV : 32 divides two fixed point binary numbers (31 bits + sign). The result is also a fixed point binary number (31 bits + sign).

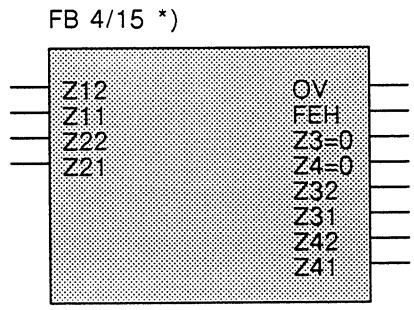
The function block DIV : 32 sets the following identifiers for further processing: overflow of the numerical range; quotient equals zero; remainder equals zero; error dividing by zero.

### Calling the function block

Representation in STL

```
      :  JU FB 4/15
NAME  :  DIV:32
Z12   :
Z11   :
Z22   :
Z21   :
OV    :
FEH   :
Z3=0  :
Z4=0  :
Z32   :
Z31   :
Z42   :
Z41   :
```

Graphic representation




*) FB number dependent on the CPU used; see technical data

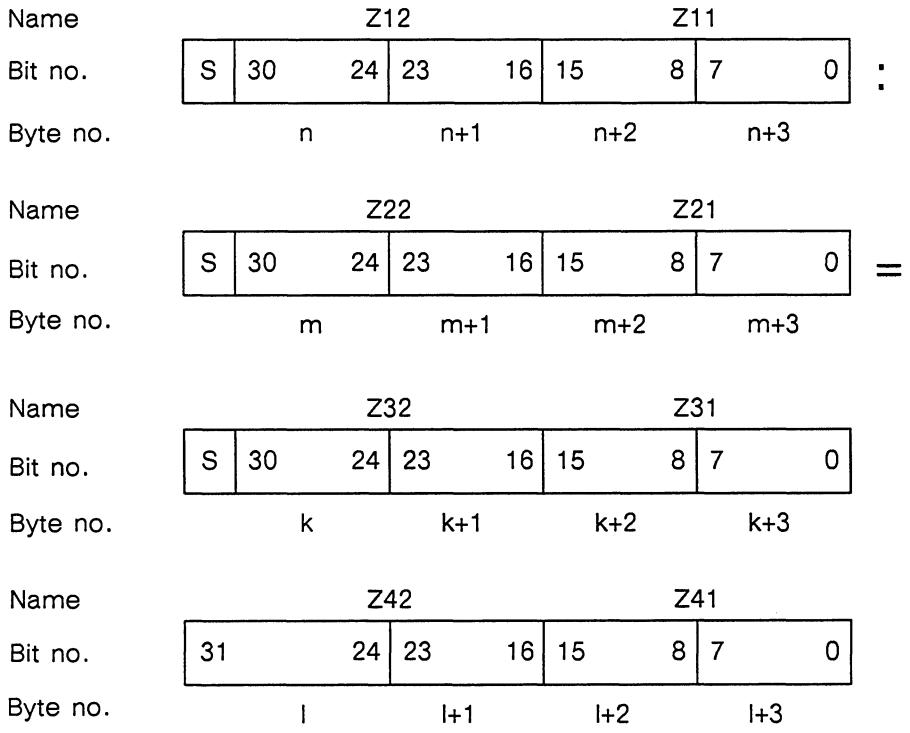
**Explanation of the inputs and outputs (parameters)**

| Name | Para | Data | Designation | Notes |
|---|---|---|---|---|
| Z12 | I | W | Dividend byte no. n, n+1 | Numerical range *) |
| Z11 | I | W | Dividend byte no. n+2, n+3 | Numerical range *) |
| Z22 | I | W | Divisor byte no. m, m+1 | Numerical range *) |
| Z21 | I | W | Divisor byte no. m+2, m+3 | Numerical range *) |
| OV | Q | BI | Overflow | (Overflow**) |
| FEH | Q | BI | Error | When dividing by zero |
| Z3=0 | | BI | Quotient equals zero | - |
| Z4=0 | Q | BI | Remainder equals zero | - |
| Z32 | Q | W | Quotient byte no. k, k+1 | Numerical range *) |
| Z31 | Q | W | Quotient byte no. k+2, k+3 | Numerical range *) |
| Z42 | Q | W | Remainder byte no. l, l+1 | Numerical range *) |
| Z41 | Q | W | Remainder byte no. l+2, l+3 | Numerical range *) |

*) Numerical range of Z1, Z2, Z3 and Z4: $-2^{31}$ to $+2^{31}-1$

**) If overflow occurs, Z3 and Z4 are cleared!

**Assignment of the parameters**




S = Sign

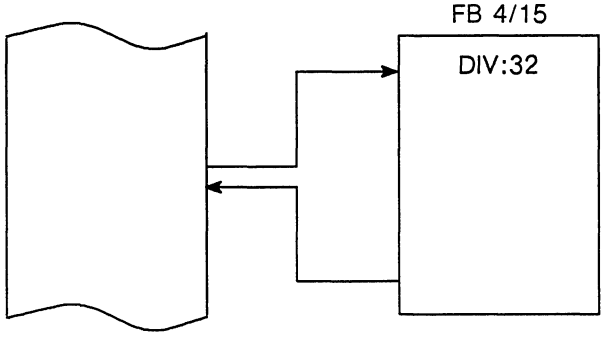
The remainder of Z4 is the non-divisible part of the dividend Z1.

Example: 11 : 2 = 5 remainder 1

↓ ↓ ↓ ↓

Z1 Z2 Z3 Z4

**Program structure**



## 16-bit binary root extractor

### Functional description RAD : 16

The function block RAD : 16 extracts the square root of a fixed point binary number (15 bits + sign). The result is also a fixed point binary number (8-bit root, 16-bit remainder).

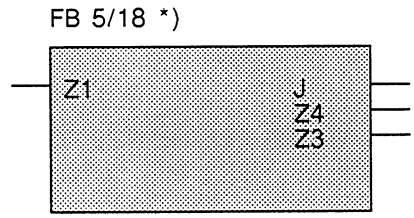
The function block RAD : 16 sets the following identifier for further processing: negative radicand.

### Calling the function block

Representation in STL

```
      :  JU FB 5/18
NAME  :  RAD:16
Z1    :
J     :
Z4    :
Z3    :
```

Graphic representation

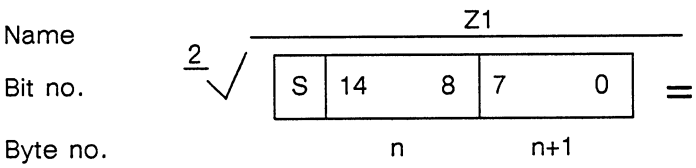


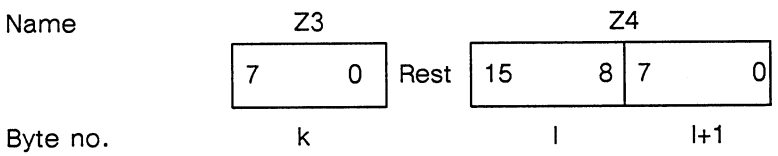

### Explanation of the inputs and outputs (parameters)

| Name | Para | Data | Designation | Notes |
|---|---|---|---|---|
| Z1 | I | W | Radicand byte no. n, n+1 | $0 \leq Z1 \leq +2^{15}-1$ |
| J | Q | BI | Radicand negative | "J" = negative |
| Z3 | Q | BY | Root byte no. k | $0 \leq Z3 \leq +181$ |
| Z4 | Q | W | Remainder byte no. l, l+1 | $0 \leq Z4 \leq +361$ |

*) FB number dependent on the CPU used; see technical data

**Assignment of the parameters**



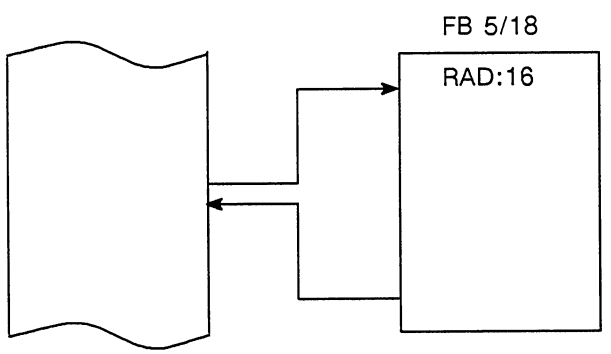

S = Sign

The remainder Z4 is the non-extractable part of the radicand Z1.

Example: $\sqrt{10} = 3$ remainder 1

Z1 Z3 Z4

**Program structure**



## Floating point root extractor

### Functional description RAD : GP

The RAD : GP function block extracts the square root of a floating point number (7 bits + sign exponent, 23 bits + sign mantissa). The result is also a floating point number, in which the least significant bit of the mantissa is not rounded.

### Function

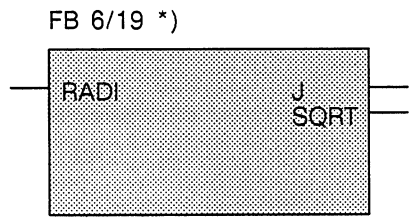
$Y = \sqrt{A}$
Y = SQRT
A = RADI

### Calling the function block

Representation in STL

```
      :  JU FB 6/19
NAME  :  RAD:GP
RADI  :
J     :
SQRT  :
```

Graphic representation

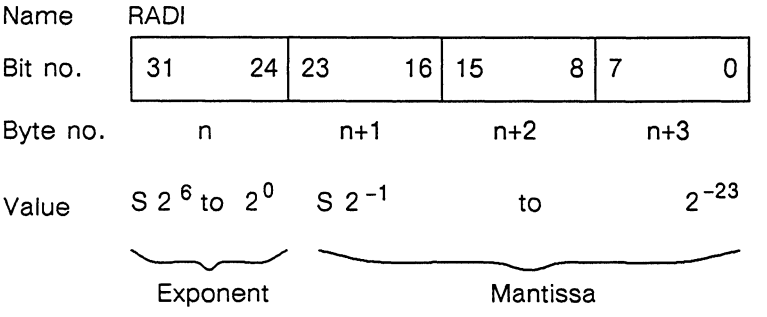


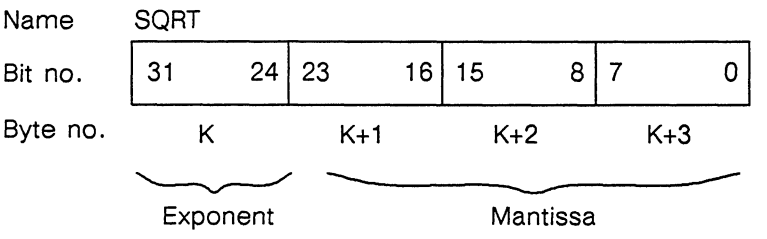

### Explanation of the inputs and outputs (parameters)

| Name | Para | Data | Designation | Notes |
|---|---|---|---|---|
| RADI | I | D | Radicand (byte no. n to n+3) | ±0,1469368E-39 ≤ RADI ≤ ±0,1701412E+39 |
| J | Q | BI | Radicand negative | "J" = negative |
| SQRT | Q | D | Root (byte no. k to k+3) | +0,3833234E-19 ≤ SQRT ≤ +0,1304382E+20 |

*) FB number dependent on the CPU used; see technical data

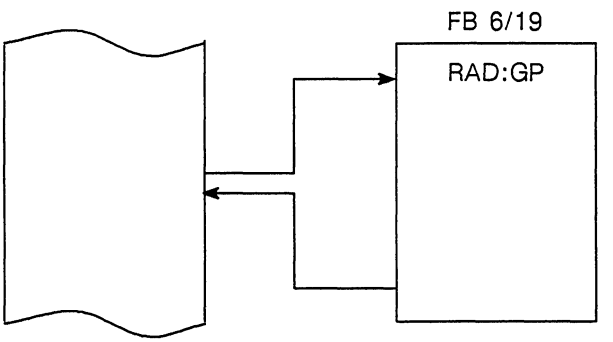
**Assignment of the parameters**




S = Sign

**Program structure**



# 3 Registers

The standard function blocks for registers (complex memory functions) handle the organization (control) of these register functions.

The function blocks operate in conjunction with data areas which contain the register information to be "managed". You can select these data areas. The bit shift register operates with a flag area, the other registers operate in conjunction with data blocks.

## 3.1 Overview

**Shift register (bit)**

The function block REG SCHB contains a one bit right-left shift register of variable length. The shift register is located in the flag area; the first and last flag of this shift register are specified when the function block is called.

The function block REG : SCHB has inputs for "shift right" and "shift left" as well as outputs for the carry.

**Shift register (word)**

The function block REG : SCHW contains a one word (16-bit) wide right-left shift register of variable length. The shift register is located in a data block which is assigned parameters and defined in length when the function block is called.

The function block REG : SCHW has inputs for "shift right" and "shift left" as well as outputs for the carry.

**Buffer (FIFO)**

The function block REG : FIFO contains a one word (16-bit) wide FIFO (first in/first out) of variable length. The FIFO is located in a data block which is assigned parameters including the length of the FIFO when the function block is called.

**LIFO STACK**

The function block REG : LIFO contains a one word (16-bit) stack of variable length (LIFO = last in/first out). The stack is located in a data block for which the parameters and the depth of the stack can be specified when the function block is called.

The function block REG : LIFO has inputs for writing and reading 16-bit words and clearing the stack.

The function block REG : LIFO generates the messages "stack full" and "stack empty".

## 3.2 Description

## Shift register (bit)

### Functional description REG : SCHB

The function block REG : SCHB contains a one bit wide right-left shift register of variable length. The shift register is in the flag area; the first and last flags in this shift register are fixed when the function block is called.

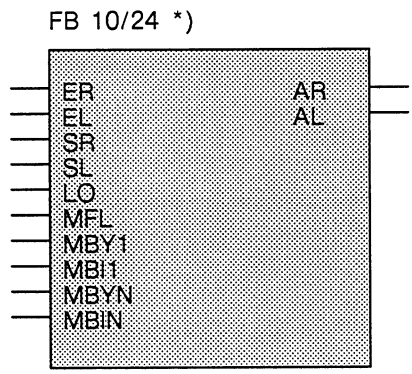
The function block REG : SCHB has inputs for "shift right" and "shift left" as well as outputs for the carry.

### Calling the function block

Representation in STL

```
      :  JU FB 10/24
NAME  :  REG: SCHB
ER    :
EL    :
SR    :
SL    :
LO    :
MFL   :  KF
MBY1  :  KF
MBI1  :  KF
MBYN  :  KF
MBIN  :  KF
AR    :
AL    :
```

Graphic representation



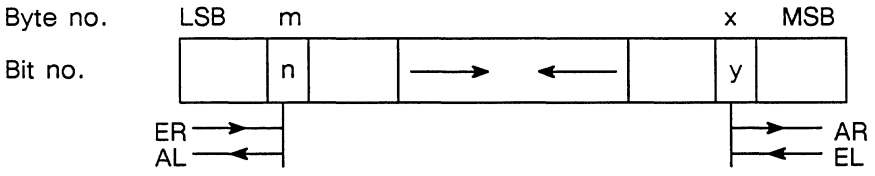

*) FB number dependent on the CPU used; see technical data

**Explanation of the inputs and outputs (parameters)**

| Name | Para | Data | Designation | Notes |
|---|---|---|---|---|
| ER | I | BI | Serial input shift right | Input data |
| EL | I | BI | Serial input shift left | Input data |
| SR | I | BI | Command shift right | On a signal change from "0" to "1" |
| SL | I | BI | Command shift left | On a signal change from "0" to "1" |
| LO | I | BI | Command clear register | On a signal change from "0" to "1" |
| MFL | D | KF | Flag byte for status bits | - |
| MBY1 | D | KF | 1st flag byte of the shift register | - |
| MBI1 | D | KF | 1st flag bit of the shift register | - |
| MBYN | D | KF | Last flag byte of the shift register | - |
| MBIN | D | KF | Last flag bit of the shift register | - |
| AR | Q | BI | Serial output shift right | Output data |
| AL | Q | BI | Serial output shift left | Output data |

The commands to the inputs SR, SL and LO are executed on a positive-going edge, i.e. change of signal status from "0" to "1".

**Structure of the shift register**

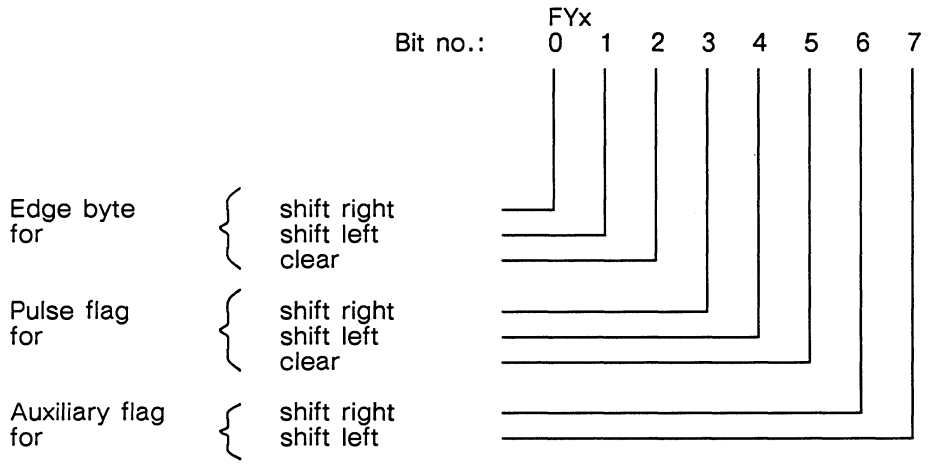



m 1st flag byte of the register/parameter : MBY1)
x last flag byte of the register/parameter : MBYN)
n 1st flag bit of the register/parameter : MBI1)
y last flag bit of the register/parameter : MBIN)

Numerical range m : 0 to 199)
x : 0 to 199) (x > = m)
n : 0 to 7
y : 0 to 7 (y > n when x = m)

**Assignment of the status bit**

The function block REG : SCHB requires the status bits located in a flag byte to organize the shift register. The number of the flag byte is specified at the input MFL. When processing the function block, the status bits are loaded in flag byte FY255.




Numerical range x : 0 to 199)

**Shift register functions**

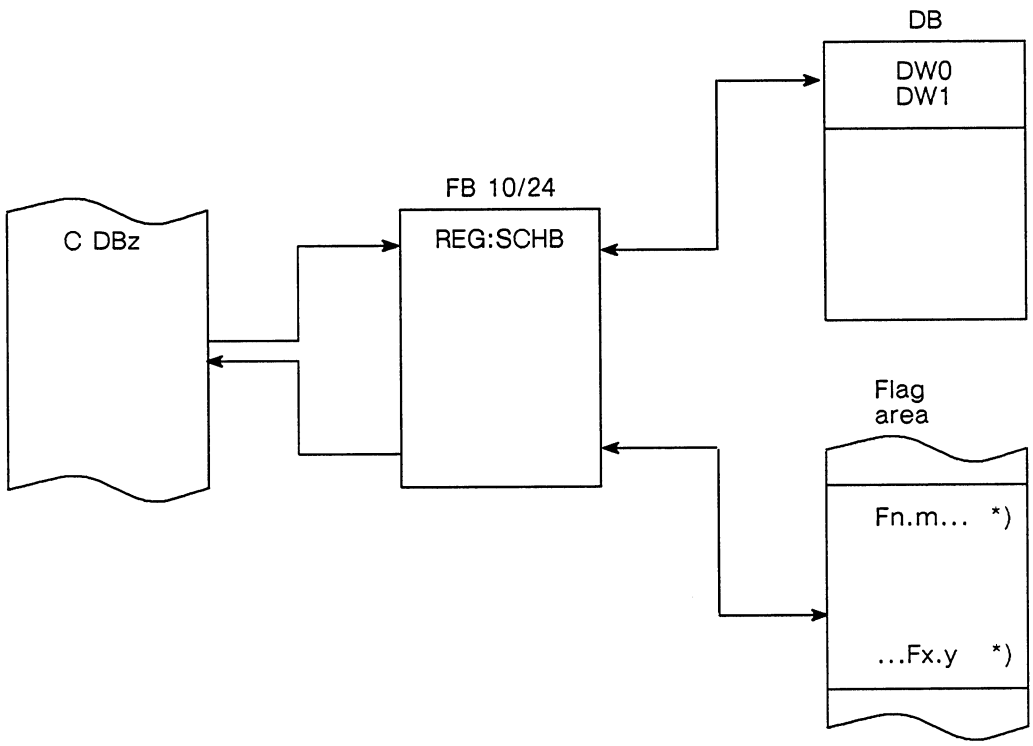
With a positive-going edge at input "shift right" (SR) the information at input ER is written to the register. The information in the last bit of the register is shifted to the right. The information of the last bit of the register is read out to output AR.

With a positive-going edge at input "shift left" (SL) the information at input EL is written to the register. The information in the last bit of the register is shifted to the left. The information of the last bit of the register is read out to output AL.

For internal processing, the function block REG : SCHB uses the data words DW 0 and DW 1 of a data block. Before the function block is called a data block which is long enough must be opend. The data block area which is used must be erased in the cold restart branch.

When clearing the shift register, all the flags of the register and the outputs AR and AL are set to signal status "0".

**Program structure**




*) The variables n, m, x and y are determined by the user when the function block is called.

## Shift register (word)

### Functional description REG : SCHW

The function block REG : SCHW contains a one word (16-bit) wide right-left shift register of variable length. The shift register is located in a data block and is assigned parameters including the length of the shift register when the function block is called. Before the function block is colled a data block which is long enough must be opend. The data block area which is used must be erased in the cold restart branch.

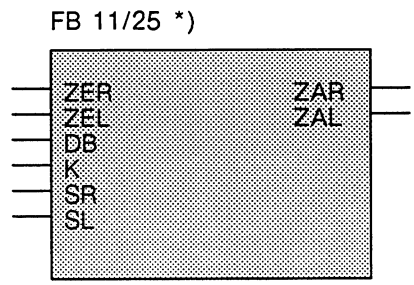
The function block REG : SCHW has inputs for "shift right" and "shift left" as well as outputs for the carry.

### Calling the function block

Representation in STL

```
      :  JU FB 11/25
NAME  :  REG: SCHW
ZER   :
ZEL   :
DB    :
K     :  KF
SR    :
SL    :
ZAR   :
ZAL   :
```

Graphic representation



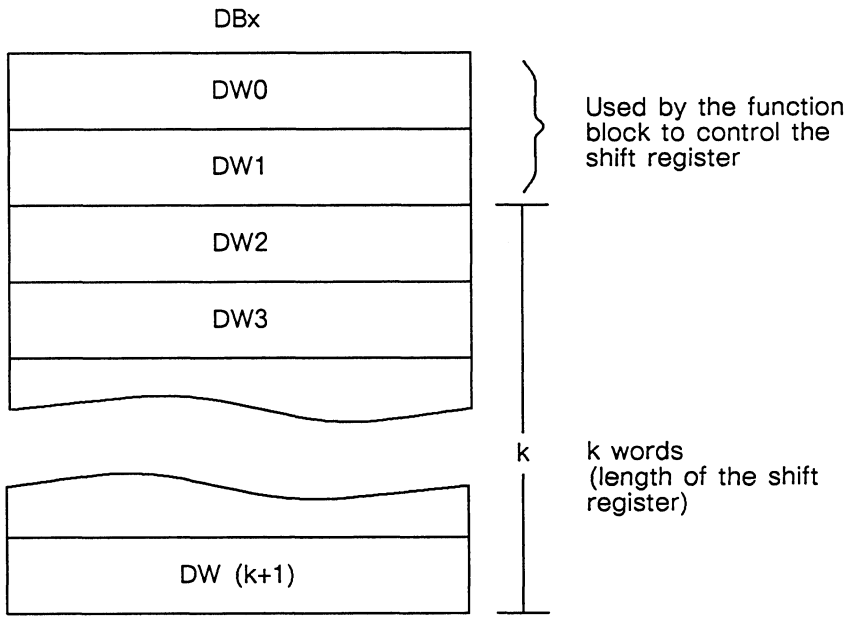

### Explanation of the inputs and outputs (parameters)

| Name | Para | Data | Designation | Notes |
|---|---|---|---|---|
| ZER | I | W | Serial input shift right | Input data |
| ZEL | I | W | Serial input shift left | Input data |
| DB | B | - | Specifies the data block | - |
| K | D | KF | Specifies the shift register length | K=1 to 254 **) |
| SR | I | BI | Command shift right | On a signal change from "0" to "1" |
| SL | I | BI | Command shift left | On a signal change from "0" to "1" |
| ZAR | Q | W | Serial output shift right | Output data |
| ZAL | Q | W | Serial output shift left | Output data |

*) Dependent on the CPU used: see technical data

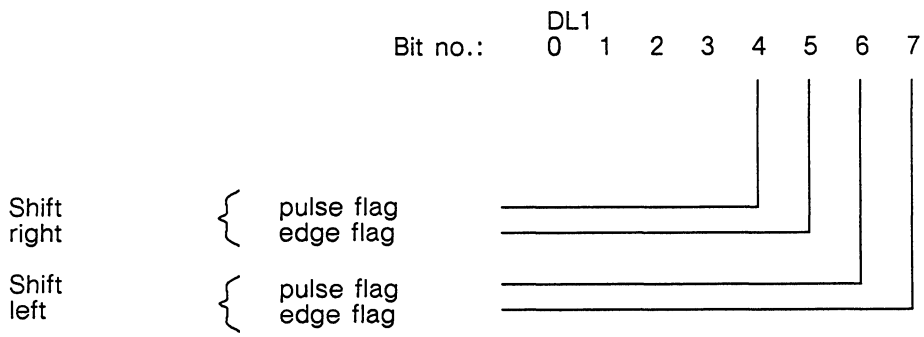
**) FB number dependent on the CPU used; see technical data

**Structure of the shift register**




Numerical range x : 2 to 255
k : 1 to 254*)

**Assignment of the status bit**




*) Depending on the type of CPU used.
If k = 0 the function block is not processed

**Shift register function**

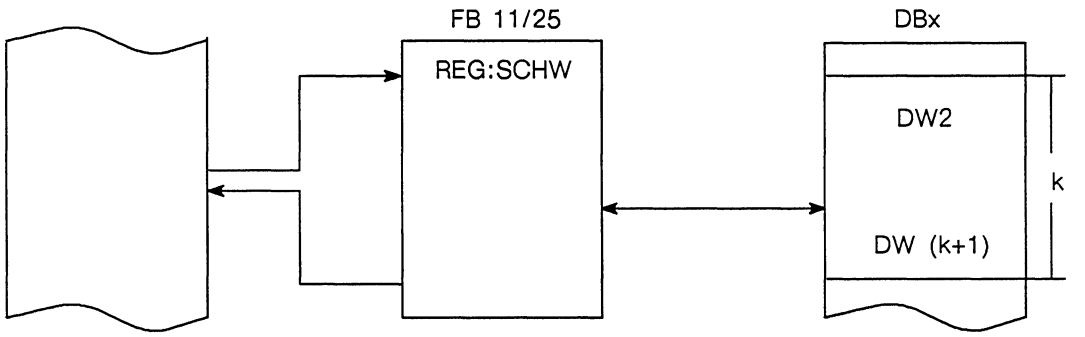
The binary inputs (shift right" (SR) and "shift left" (SL) only react to a positive-going edge. The required status bits are stored in the left byte of the data word (DW 1). When the function block is processed, they are written to flag byte FY255. The data word (DW 1) then functions as a buffer for volatile data.

On a positive-going edge of the signal at "shift right" (SR) the information at input ZER is written to the register (DW 2). At the same time, the contents of the register are shifted to the right (in the direction of data words with higher numbers). The information in the last data word (DW k+1) is supplied to output ZAR.

On a positive-going edge of the signal at "shift left" (SL) the information at input ZEL is written to the register (DW k+1). At the same time, the contents of the register are shifted to the left (in the direction of data words with lower numbers). The information in the first data word (DW 2) is supplied to output ZAL.

If data words are used for the parameters ZER, ZEL, ZAR and ZAL, the data words are in the data block specified in the parameter DB.

**Program structure**




The variables k and x are specified by the user when the function block is called.

## Buffer

### Functional description REG : FIFO

The function block REG : FIFO contains a one word (16-bit) wide FIFO (first in/first out) of variable length. The FIFO is located in a data block which is assigned parameters including the length of the FIFO when the function block is called. Before the function block is colled a data block which is long enough must be opened. The data block area which is used must be erased in the cold restart branch.

The function block REG : FIFO has inputs for writing and reading 16-bit wide words and for clearing the buffer.

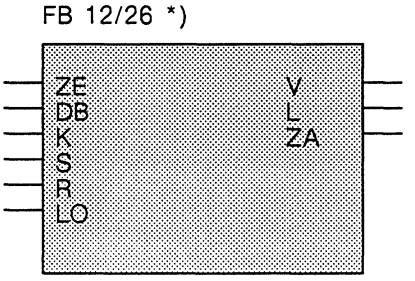
The function block REG : FIFO outputs the statuses "buffer full" and "buffer empty".

### Calling the function block

Representation in STL

```
      :  JU  FB 12/26
NAME  :  REG: FIFO
ZE    :
DB    :
K     :  KF
S     :
R     :
LO    :
V     :
L     :
ZA    :
```

Graphic representation



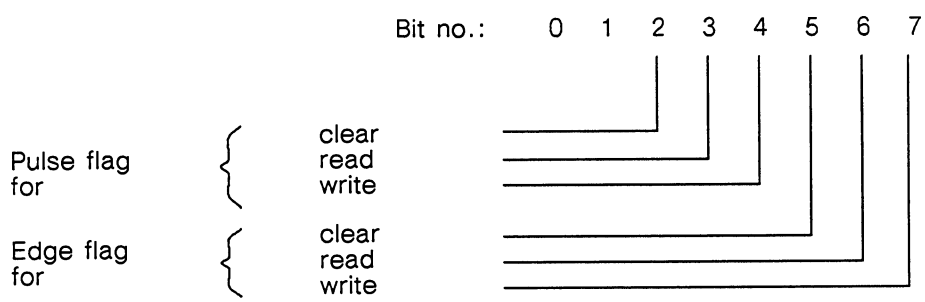

*) FB number dependent on the CPU used; see technical data

**Explanation of the inputs and outputs (parameters)**

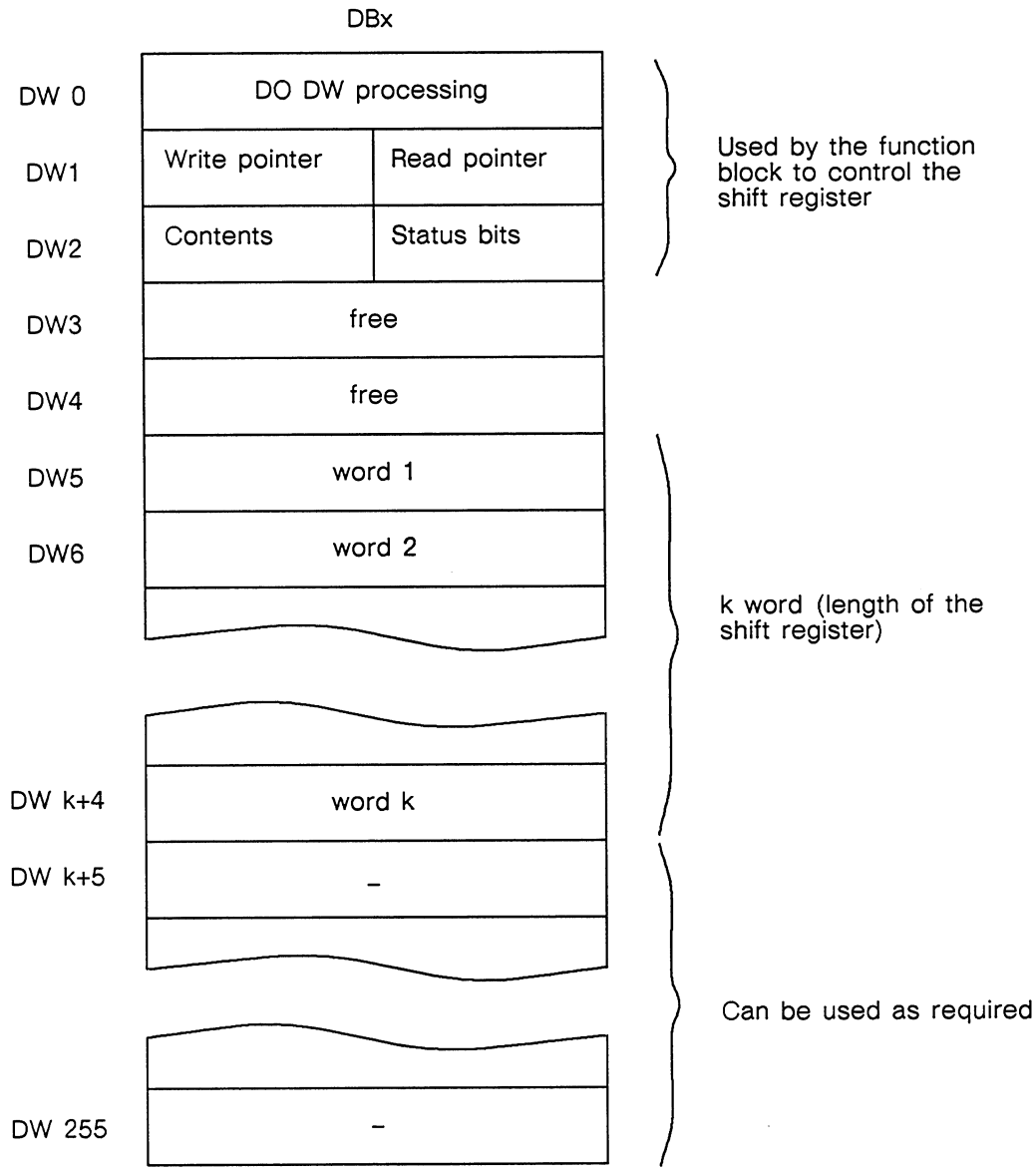
| Name | Para | Data | Designation | Notes |
|---|---|---|---|---|
| ZE | I | W | Input word | Input data |
| DB | B | – | Specifies the data block | – |
| K | D | KF | Specifies the buffer length | (KF = 1 to 250) |
| S | I | BI | Command write to register | On a signal change from "0" to "1" |
| R | I | BI | Command read from register | On a signal change from "0" to "1" |
| LO | I | BI | Command clear register | On a signal change from "0" to "1" |
| V | Q | BI | Buffer is full | If signal status "1" |
| L | Q | BI | Buffer is empty | If signal status "1" |
| ZA | Q | W | Output word | Output data |

The binary inputs S, R and LO react to a positive-going edge, i.e. the register is only written to, read from or cleared when the status signal changes from "0" to "1".

**Assignment of the status bits**



**Assignment of the buffer**




Numerical range x : 2 to 255
k : 1 to 250

**Function of the buffer**

The binary inputs "write" (S), "read" (R) and "clear" (LO) only react to a positive-going edge. The required status bits are stored in the right byte of the data word (DW 2). They are loaded in flag byte FY 255 when the function block is processed.

When clearing the buffer, both pointers are set to the value 5. The data byte "contents" is loaded with zero. The value in one of the two pointers corresponds to the parameter of the data word to be written to or read from.

On a positive-going edge at input S the word at input ZE is written in. Writing begins at data word 5. The write pointer and the contents are then incremented by one. The next word is written into data word 6.

On a positive-going edge at input R the word at output ZA is output. The read counter is then incremented by 1; the contents are decremented by 1. The next word is read out from data word 6, etc.

If one of the pointers reaches the upper limit K + 5 and the contents do not have the value K, the pointer is set to 5. This means that reading or writing starts from the beginning again.

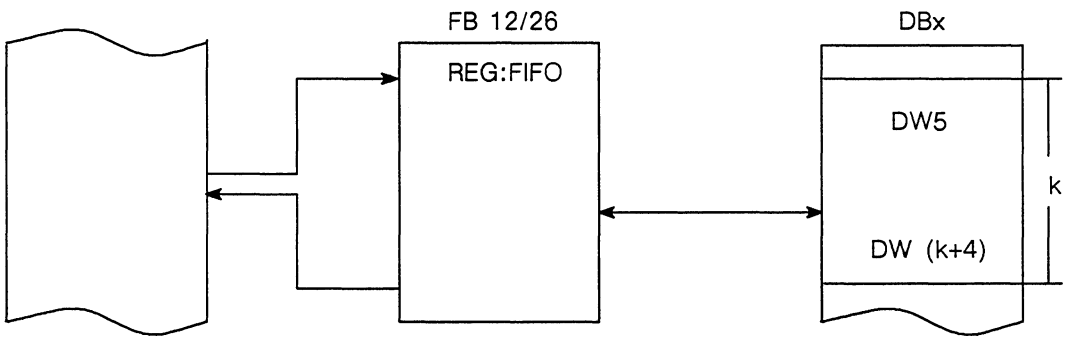
If the read counter catches up with the write counter (when the contents are zero) both counters are set to 5.

If "contents" reaches the value K, the buffer is full. If an attempt is made to write into a full buffer, the output "buffer full" (V) is set. The output V is reset when a word has been read out or when the register has been cleared.

When "contents" reaches zero, the buffer is empty. If an attempt is made to read out of an empty buffer, the output "buffer empty" (L) is set. The output ZA then has the contents "zero". Output L is cleared again when a word has been written in or when the buffer is cleared.

If the parameters ZE and ZA are supplied with data words, these data words must be in the data block specified by the parameter DB.

**Program structure**




The variables k and x are fixed by the user when the function block is called.

## Stack

### Functional description REG : LIFO FB 13

The function block REG : LIFO contains a one word (16-bit) stack of variable length (LIFO = last in/first out). The stack is located in a data block for which the parameters and the depth of the stack can be specified when the function block is called. Before the function block is colled a data block which is long enough must be opend. The data block area which is used must be erased in the cold restart branch.

The function block REG : LIFO has inputs for writing and reading 16-bit words and clearing the stack.

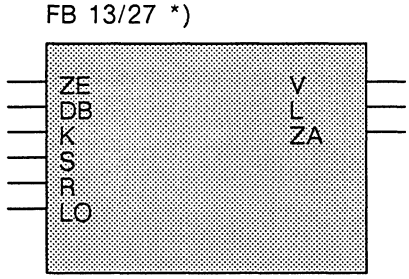
The function block REG : LIFO generates the messages "stack full" and "stack empty".

### Calling the function block

Representation in STL

```
      :  JU FB 13/27
NAME  :  REG: LIFO
ZE    :
DB    :
K     :  KF
S     :
R     :
LO    :
V     :
L     :
ZA    :
```

Graphic representation



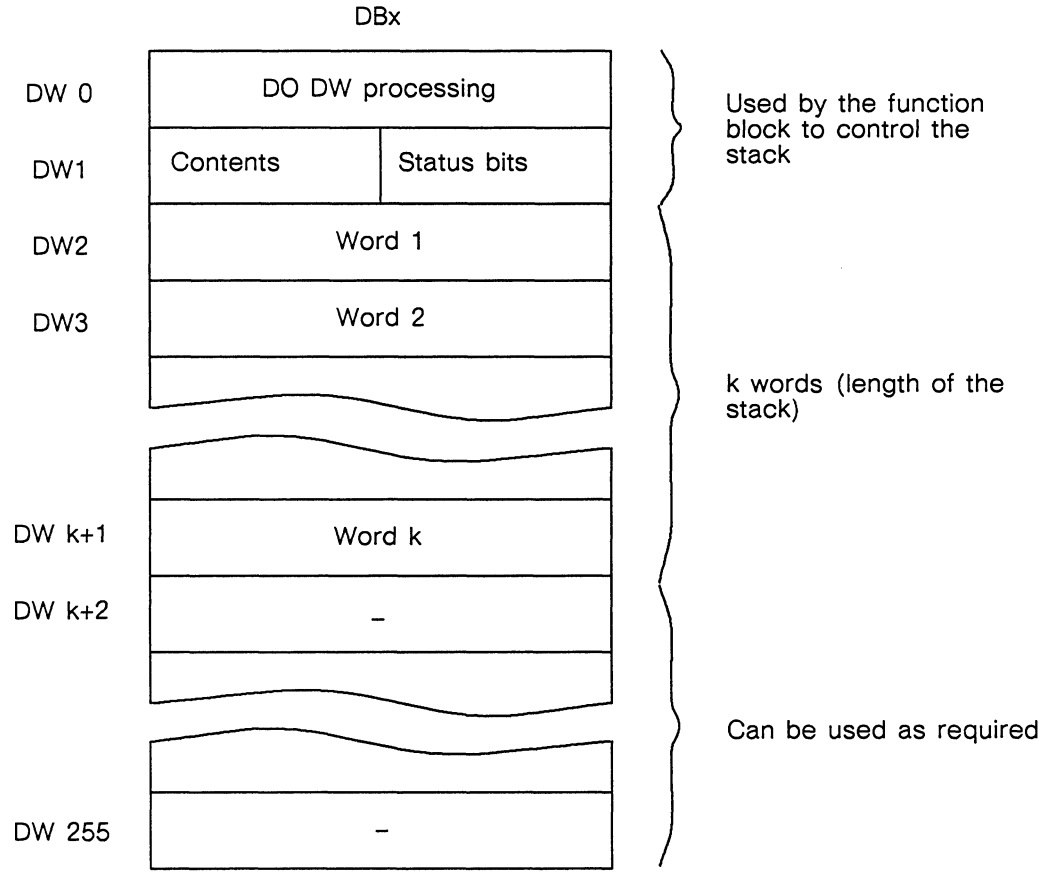

### Explanation of the inputs and outputs (parameters)

| Name | Para | Data | Designation | Notes |
|---|---|---|---|---|
| ZE | I | W | Input word | Input data |
| DB | B | – | Specifies the data block | – |
| K | D | KF | Specifies the stack length | (KF = 1 to 254) |
| S | I | BI | Command write to register | On a signal change from "0" to "1" |
| R | I | BI | Command read from register | On a signal change from "0" to "1" |
| LO | I | BI | Command clear register | On a signal change from "0" to "1" |
| V | Q | BI | Stack is full | If signal status "1" |
| L | Q | BI | Stack is empty | If signal status "1" |
| ZA | Q | W | Output word | Output data |

*) FB number dependent on the CPU used; see technical data

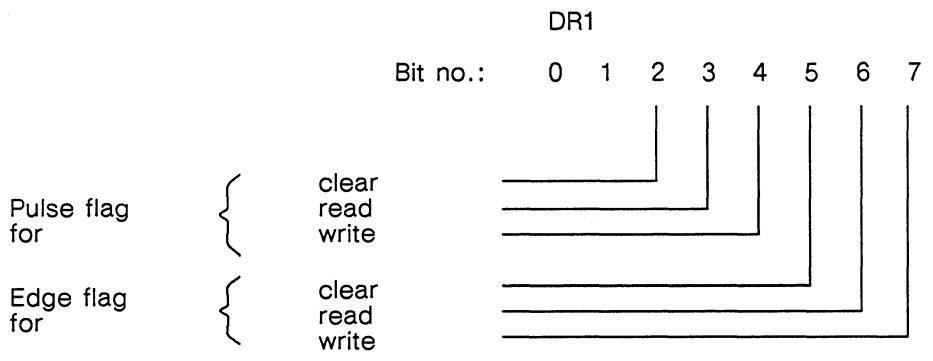
The binary inputs S, R and LO react to a positive-going edge, i.e. the stack is written to, read from or deleted only when the signal changes from "0" to "1".

**Assignment of the stack**




Numerical range x : 2 to 255
k : 1 to 250

**Assignment of the status bits**



**Function of the stack**

The binary inputs "write" (S), "read" (R) and "clear" (LO) only react to a positive-going edge. The status bits required are stored in the right byte of data word 1. They are loaded in flag byte FY 255 when the function block is processed.

The word at input ZE is written in on the positive-going edge of the signal at input S.

The word is written in starting from data word 2 onwards. The contents are incremented by 1. The current write pointer is, if necessary, calculated by the function block from the maximum depth of the stack (K) and the contents (DL 1). After the word has been written in, the contents are incremented by 1. The next word is written to data word 3 etc.

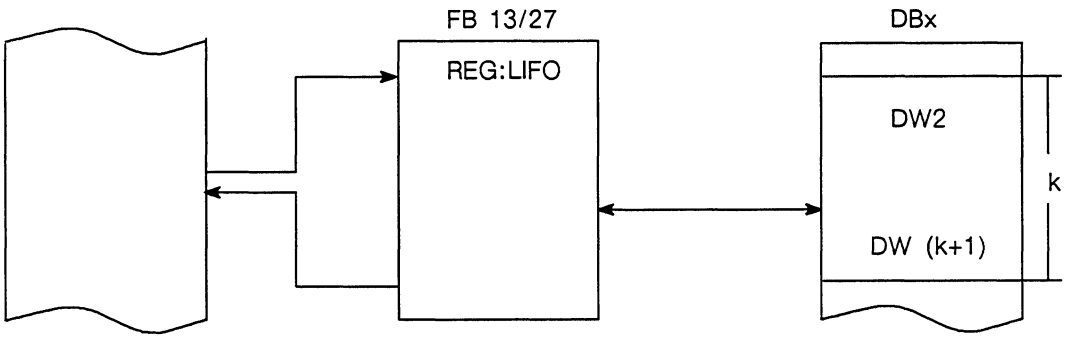
Reading begins with the last data word read in. The current read pointer is, if necessary, calculated by the function block from the maximum depth of the stack (K) and the contents (DL 1). After reading out, the contents are decremeted by 1.

When "contents" reaches the value K the stack is full. If an attempt is made to write to the full stack, the output "stack full" (V) is set. Output V is reset when a word has been read out or when the stack is cleared.

If the last entry is read out of the stack and an attempt is then made to read out from the empty stack, the output "stack empty" (L) is set. The output ZA then has the value 0. The output L is reset when a word is entered or when the stack is cleared.

If the parameters ZE and ZA are supplied with data words, these data words are located in the data block specified with the parameter DB.

**Program structure**




The variables k and x are fixed by the user when the function block is called.

# 4 Analog input/output

This Section describes the standard function blocks for reading in and outputting analog values. They can be used for the following modules:

| | | |
|---|---|---|
| Analog input | 6ES5 464-8MA11 | 6ES5 464-8MD11 |
| | 6ES5 464-8MA21 | 6ES5 464-8ME11 |
| | 6ES5 464-8MB11 | 6ES5 464-8MF11 |
| | 6ES5 464-8MC11 | 6ES5 464-8MF21 |
| | 6ES5 460-4UA1x | 6ES5 460-7LA1x |
| | 6ES5 465-4UA1x | 6ES5 465-7LA1x |
| | 6ES5 463-4UA1x | |
| | 6ES5 466-4LA1x | |
| Analog output | 6ES5 470-4UA11 | |
| | 6ES5 470-4UB11 | |
| | 6ES5 470-4UC11 | |

## 4.1 Overview

### Analog value input

Function block FB30 to FB33 read an analog value XE from an analog value input and supplys a proportional output value XA within the selectable limit values UGR ... OGR related to the rated value. The analog value can be scanned singly or cyclically.

### Analog value output

Function block RLG:AA converts an input value XE into an output value for an analog output module within the range specified by OGR and UGR. If this range is exceeded, an error message is output.

## 4.2 Description

## Reading in the analog value: AE:464 FB30

### Summary

The present programmer's guide describes the function block

| | | | |
|---|---|---|---|
| FB 30 | AE:464 | Read analog values from the modules | |
| | | 6ES5 464-8MA11 | 6ES5 464-8MD11 |
| | | 6ES5 464-8MA21 | 6ES5 464-8ME11 |
| | | 6ES5 464-8MB11 | 6ES5 464-8MF11 |
| | | 6ES5 464-8MC11 | 6ES5 464-8MF21 |

The usage of FB 30 for the analog input modules 6ES5 464-8M... is only useful if

- the analog input modules are directly on the AG S5-100U or
- an interface connection via the great central units (S5-115U, 135U, 155U) to the ET 100 station was realized using IM 308/318; in this case, the analog input modules type 464 are in the ET 100 station.

The present programmer's guide expects the knowledge of the instruction manual for the analog input module and the appropriate automation system.

When the peripheral areas Q, IM3 and IM4 in AG S5-115U are used in connection with an interface module, the PESP signal must not be evaluated on the central unit interface module (see instruction manual of the interface module).

#### Functional description

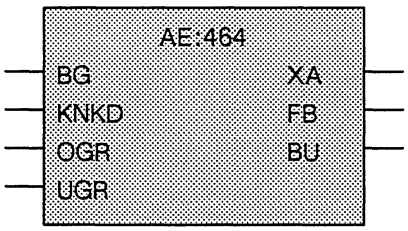
Function block FB 30 (AE:464) "Read analog value" reads an analog value from the analog input module 6ES5 464-8Mxxx. It outputs a standardized value at the output XA independent of the type of module. The user defines the range of values of the value delivered by the function block by the parameters OGR and UGR.

Function block FB 30 for AGs S5-095U, S5-100U and S5-115U supports the standardized value as 16 bit fixed-point number, the AGs S5-135U and S5-155U supports them as 32 bit floating-point number.

**Calling the function block**

```
      :  SPA FB 30
NAME  :  AE:464
BG    :
KNKD  :
OGR   :
UGR   :
XA    :
FB    :
BU    :
```

FB 30




**Explanation of the inputs and outputs (parameters)**

| Name | Para | Data | Designation |
|---|---|---|---|
| BG | D | KY | Details of the peripheral area and the module adress |
| KNKT | D | KY | Information about the channel number and the channel representation |
| OGR | D | KF (1)<br>KG (2) | Upper limit of the output value<br>Upper limit of the output value |
| UGR | D | KF (1)<br>KG (2) | Lower limit of the output value<br>Lower limit of the output value |
| XA | Q(3) | W (1)<br>D (2) | Output value as fixed/point number<br>Output value as flating/point number |
| FB | Q | BI | Error bit |
| BU | Q | BI | Message range exceeded |

(1) valid for AGs S5-095U, S5-100U and S5-115U

(2) valid for AGs S5-135U and S5-155U

(3) If a data (double) word is used as output word the associated data block must be opened before calling FB 30.

**Attention:** If flags are used for parameterization only flag areas from MB 0 to MB 199 may be assigned.

## Parameter assignment

**BG** Peripheral area, module address

KY = 0, 64 to 126(1)
(for AGs S5-095U and S5-100U)
KY = 0, 128 to 254(2)
(for AGs S5-115U, S5-135U and S5-155U in peripheral area P)
KY = 1, 0 to 254(2)
(for AGs S5-115U, S5-135U and S5-155U in peripheral area Q)
KY = 2, 0 to 254(2)
(for AGs S5-115U, S5-135U and S5-155U in peripheral area IM3)
KY = 3, 0 to 254(2)
(for AGs S5-115U, S5-135U and S5-155U in peripheral area IM4)
KY = 2, 0 to 254(2)
(for AGs S5-115U, S5-135U and S5-155U in peripheral area IM3)

**Attention:** Depending on the channel number the highest subaddresse must not exceed 127 for (1) and 255 for (2).

**KNKD** Channel number, channel representation

KY = x, y
x = 0 to 3 channel number
y = 7 to 13 channel representation

7 fixed-point representation bipolar (nominal range 2048)
8 fixed-point representation bipolar (nominal range -200 to +1369)
9 fixed-point representation bipolar (nominal range -200 to +1200)
10 fixed-point representation bipolar (nominal range -199 to +900)
11 fixed-point representation bipolar (nominal range -200 to +1700)
12 fixed-point representation unipolar (nominal range +512 to +2560)
13 fixed-point representation unipolar (nominal range 0 to +2048)

**OGR** Upper limit
KF = -32 768 to +32 767
(for AGs S5-095U, S5-100U and S5-115U)
KG = -1701412+39 to +1701412+39
(for AGs S5-135U and S5-155U)

**UGR** Lower limit
KF = -32 768 to +32 767
(for AGs S5-095U, S5-100U and S5-115U)
KG = -1701412+39 to +1701412+39
(for AGs S5-135U and S5-155U)

**XA** Output value
contains the standardized analog value within the limits of UGR and OGR as 16 bit fixed-point value (for AGs S5-095U, S5-100U and S5-115U) or as 32 bit floating-point value (for AGs S5-135U and S5-155U).

**FB** Error bit
Signal state is "1"
- if Bit $2^1$ in read analog value is signal state "1" (wire break)
- if less than 384 units are read in KD = 12

**BU** Over-range
Signal state is "1"
- if Bit $2^0$ in read analog value is signal state "1" (overflow)
- if the nominal range is exceeded (depending on parameter KD)

### Using the function block

Function block FB 30 (AE:464) reads the analog value from the analog input module 6ES5 464-8Mxxx and makes it available at parameter XA as standardized value (linearly converted within the limits OGR and UGR) in fixed-point representation or in floating-point representation (depending on the AG) for further processing. The function block takes into account the different nominal ranges using parameter KD (channel representation) for standardizing.

**Note:** For standardization, function block FB 30 for the AGs S5-095U, S5-100U and S5-115U requires the standard function block FB4 which the program package Basic Functions also contains.

**Channel representation**

The representation differs from the nominal range of the read analog value according to the analog input modul used. So that the function block can correctly carry through standardization within the limits OGR and UGR, the parameter must be supplied according to the following list.

KD = 7 fixed-point representation bipolar with nominal range -2048 to +2048 )
for modules 6ES5 464-8MA11 ± 50 mV
6ES5 464-8MA21 ± 50 mV without linearization and without temperature compensation)
6ES5 464-8MB11 ( ± 1 V)
6ES5 464-8MC11 ( ± 10 V)
6ES5 464-8MD11 ( ± 20 mA)
6ES5 464-8MF11 ( ±500 mV)
6ES5 464-8MF21 ( ±500 mV without linearization

KD = 8 fixed-point representation bipolar with nominal range -200 to +1369
for module 6ES5 464-8MA21 (thermocouple type K)

KD = 9 fixed-point representation bipolar with nominal range -200 to +1200
for module 6ES5 464-8MA21 (thermocouple type J)

KD = 10 fixed-point representation bipolar with nominal range -199 to +900
for module 6ES5 464-8MA21 (thermocouple type L)

KD = 11 fixed-point representation bipolar with nominal range -200 to +1700
for module 6ES5 464-8MF21 (Pt 100 with linearization)

KD = 12 fixed-point representation unipolar with nominal range +512 to +2560
for module 6ES5 464-8ME11 (4 to 20 mA)

KD = 13 fixed-point representation unipolar with nominal range 0 to +2048
for modules 6ES5 464-8MF11 (Pt 100)
6ES5 464-8MF21 (Pt 100 without linearization)

**Troubleshooting**

when the function block is left after correct processing the result of logic operation is set to "0" and accumulator 1 contains the value which is read from the module. The standardized value is available at output XA. Signal state "0" is assigned to the parameters FB and BU.

When an unvalid value is assigned, the function block signals this parameter assignment error by the result of logic operation "1" and by an error number in accumulator 1. In this case, the output XA is assigned zero. The parameters FB and BU are assigned signal state "0".

| | | | |
|---|---|---|---|
| Error number KF = | 1: | AG 095U/100U: | Parameter BG < 64 |
| | | AG 115U/135U/155U | Parameter BG < 128 at peripheral P |
| | 2: | parameter KN > 3 | |
| | 3: | sum from parameter BG and 2 times parameter KN | |
| | | AG 095U/100U: | > 127 |
| | | AG 115U/135U/155U | > 255 |
| | 4: | parameter KD < 7 or >13 | |
| | 5: | parameter OGR < = Parameter UGR | |
| | 6: | parameter BG: x, y: x > 3 | |

If a non-assigned module location is selected zero is read (AGs S5-095U and S5-100U). Depending on the setting, time-out is signalled at interface 308.

If error bit FB is set, zero is output at output XA. The result of logic operation is set to "1". Accumulator 1 contains the value which is read from the module.

If the overflow bit BU is set the nominal range limit value is output at output XA. The result of logic operation is set to "1". Accumulator 1 contains the value which is read from the module.

## Reading in an analog value: 460 FB 31

### Summary

The present programmer's guide describes the function block

| | | |
|---|---|---|
| FB 31 | AE:460 | Read analog values from the modules<br>6ES5 460-4UA1x 6ES5 460-7LA1x<br>6ES5 465-4UA1x 6ES5 465-7LA1x |

The present programmer's guide expects the knowledge of the instruction manual for the analog input module and the appropriate automation system.

When the peripheral areas Q, IM3 and IM4 in AG S5-115U are used in connection with an interface module, the PESP signal must not be evaluated on the central unit interface module (see instruction manual of the interface module).

### Functional description

Function block FB 31 (AE:460) "Read analog value" reads an analog value from the analog input module 6ES5 460-4UA/7LAxx or 6ES5 465-4UA/7LAxx. It outputs a standardized value at the output XA independent of the type of module. The user defines the range of values of the value delivered by the function block by the parameters OGR and UGR.

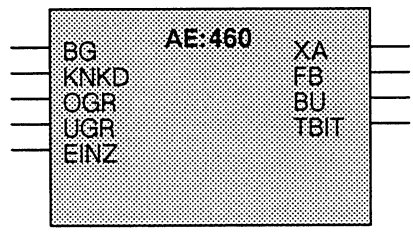
Function block FB 31 for AG S5-115U supports the standardized value as 16 bit fixed-point number, the function blocs FB 31 for AGs S5-135U and S5-155U support the standardized value as 32 bit floating-point number.

The function block FB 31 allows the analog value of a channel to be read by cyclic (asyncronous) coding as well as by individual scanning.

### Calling the function block

```
      :  SPA FP31
NAME  :  AE:460
BG    :
KNKD  :
OGR   :
UGR   :
EINZ  :
XA    :
BU    :
TBIT  :
```

FB 31




### Explanation of the parameters

| Name | Para | Data | Designation |
|---|---|---|---|
| BG | D | KY | details of the peripheral area and the module address |
| KNKD | D | KY | Information about the channel number and the channel representation |
| OGR | D | KF(1)<br>KG(2) | upper limit of the output value<br>upper limit of the output value |
| UGR | D | KF(1)<br>KG(2) | lower limit of the output value<br>lower limit of the output value |
| EINZ | I | BI | individual scan |
| XA | Q(3) | W(1)<br>D(2) | output value as fixed-point number<br>output value as floating-point number |
| FB | Q | BI | error bit |
| BU | Q | BI | message range exceeded |
| TBIT | Q | BI | job bit of the function block |

(1) valid for AG S5-115U

(2) valid for AGs S5-135U and S5-155U

(3) If a data (double) word is used as output word the associated data block must be opened before calling FB 31.

| **Attention:** | If flags are used for parameterization only flag areas from MB 0 to MB 199 may be assigned. |
|---|---|

**Parameter assignment**

**BG** Peripheral area, module address

KY = 0, 128 to 224/240 (16 channels/8 channels)[1]
(for AGs S5-115U, S5-135U and S5-155U in peripheral area P)
KY = 1, 0 to 224/240
(for AGs S5-115U, S5-135U and S5-155U in peripheral area Q)
KY = 2, 0 to 224/240
(for AGs S5-115U, S5-135U and S5-155U in peripheral area IM3)
KY = 3, 0 to 224/240
(for AGs S5-115U, S5-135U and S5-155U in peripheral area IM4)

**KNKD** Channel number, channel representation
KY = x, y
x = 0 to 15 channel number
y = 3 to 6 channel representation

3 fixed-point representation unipolar (nominal range +512 to +2560)
4 fixed-point representation unipolar (nominal range 0 to +2048)
5 absolute-value representation bipolar (nominal range -2048 to +2048)
6 fixed-point representation bipolar(nominal range -2048 to +2048)

**OGR** Upper limit
KF = -32 768 to +32 767
(for AGs S5-115U)

KG = -1701412+39 to +1701412+39
(for AGs S5-135U and S5-155U)

**UGR** Lower limit
KF = -32 768 to +32 767
(for AGs S5-115U)

KG = -1701412+39 to +1701412+39
(for AGs S5-135U and S5-155U)

**XA** Output value
contains the standardized analog value within the limits of UGR and OGR as 16 bit fixed-point value (for AGs S5-115U) or as 32 bit floating-point value (for AGs S5-135U and S5-155U).
Line break: If a line break is detected XA is set to zero (XA = 0)

**EINZ** Individual scan
initiates an individual scanning when signal state is "1".

**FB** Error bit
Signal state is "1"
- if Bit $2^1$ in read analog value is signal state "1" (wire break)
- if less than 384 units are read in KD = 3

**BU** Over-range
- if Bit $2^1$ in read analog value is signal state "1" (overflow)
- if the nominal range is exceeded (depending on parameter KD)

**TBIT** Activity bit
Signal state if "1" is an individual scanning is carried through (initiated by the function block)

### Using the function block

Function block FB 31 (AE:460) reads the analog value from one of the analog input module 6ES5 460-4UAxx, 6ES5 460-7LAxx, 6ES5 465-4UAxx or 6ES5 465-7LAxx and makes it available at parameter XA as standardized value (linearly converted within the limits OGR and UGR) in fixed-point representation or in floating-point representation (depending on the AG) for further processing. The function block takes into account the different nominal ranges and the numerical representation using parameter KD (channel representation) for standardizing.

### Channel representation

The numerical representation and the nominal range of the read analog value differs according to the setting of the analog input modul. So that the function block can correctly carry through standardization within the limits OGR and UGR, the parameter must be supplied according to the following list.

KD = 3 fixed-point representation unipolar with nominal range +512 to +2560

KD = 4 fixed-point representation unipolar with nominal range 0 to +2048

KD = 5 fixed-point representation bipolar with nominal range -2048 to +2048

KD = 6 fixed-point representation bipolar with nominal range -2048 to +2048

### Individual scanning

If the parameter EINZ is set to signal state "1" the function block initiates an individual scanning on the selected channel. Due to the long coding time, the analog value is read only during the next processing cycle (during the next program run; if the module had finished the coding). The module must not be accessed by other function block calls until the individually coded value is read. The parameter TBIT of the currently active function block call is set to signal state "1" only during the coding time; thus further calls can be locked.

| | |
|---|---|
| **Note:** | The analog input module hardware must be set to individual scanning. |

**Troubleshooting**

When the function block is left after correct processing the result of logic operation is set to "0" and accumulator 1 contains the value which is read from the module. The standardized value is available at output XA. Signal state "0" is assigned to the parameters FB and BU.

When a value is assigned to a parameter which is not within the defined value range, the function block signals this parameter assignment error by the result of logic operation "1" and by an error number in accumulator 1. In this case, the output XA is assigned zero. The parameters FB and BU are assigned signal state "0".

Error number KF =
1: parameter BG < 128 for peripheral P
2: parameter KN > 15
3: sum from parameter BG and 2 times parameter KN > 255
4: parameter KD < 3 or > 6
5: parameter OGR < = parameter UGR
6: parameter BG: x, y: x > 3

If a non-existing module or a non-existing channel is selected time-out is signalled.

If error bit FB is set, zero is output at XA. The result of logic operation is set to "1". Accumulator 1 contains the value which is read from the module.

If the overflow bit BU is set the nominal range limit value is output at output XA. The result of logic operation is set to "1". Accumulator 1 contains the value which is read from the module.

## Reading in an analog value: 463 FB 32

### Summary

The present programmer's guide describes the function block

FB 32 AE:463 Read analog values from the module
6ES5 463-4UA1x
6ES5 463-4UB1x

The present programmer's guide expects the knowledge of the instruction manual for the analog input module and the appropriate automation system.

When the peripheral areas Q, IM3 and IM4 in AG S5-115U are used in connection with an interface module, the PESP signal must not be evaluated on the central unit interface module (see instruction manual of the interface module).

### Functional description

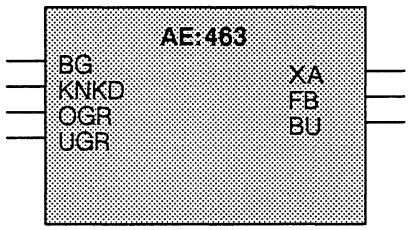
Function block FB 32 (AE:463) "Read analog value" reads an analog value from the analog input module 6ES5 463-4UAxx or 6ES5 463 - 4UBxx. It outputs a standardized value at the output XA independent of the type of module. The user defines the range of values of the value delivered by the function block by the parameters OGR and UGR.

Function block FB 32 for AG S5-115U supplies the standardized value as 16 bit fixed-point number, the function blocks for AGs S5-135U and S5-155U supply them as 32 bit floating-point number.

### Calling the function block

```
      :  SPA FB 32
NAME  :  AE:463
BG    :
KNKD  :
OGR   :
UGR   :
XA    :
FB    :
BU    :
```

FB 32




### Explanation of the parameters

| Name | Para | Data | Designation |
|---|---|---|---|
| BG | D | KY | details of the peripheral area and the module address |
| KNKD | D | KY | Information about the channel number and the channel representation |
| OGR | D | KF(1)<br>KG(2) | upper limit of the output value<br>upper limit of the output value |
| UGR | D | KF(1)<br>KG(2) | lower limit of the output value<br>lower limit of the output value |
| XA | Q(3) | W(1)<br>D(2) | output value as fixed-point number<br>output value as floating-point number |
| FB | Q | BI | error bit |
| BU | Q | BI | message range exceeded |

(1) valid for AG S5-115U

(2) valid for AGs S5-135U and S5-155U

(3) If a data (double) word is used as output word the associated data block must be opened before calling FB 32.

**Attention:** If flags are used for parameterization only flag areas from MB 0 to MB 199 may be assigned.

**Parameter assignment**

**BG** Peripheral area, module address

KY = 0, 128 to 248 (4 channels)
(for AGs S5-115U, S5-135U and S5-155U in peripheral area P)

KY = 1, 0 to 248
(for AGs S5-115U, S5-135U and S5-155U in peripheral area Q)

KY = 2, 0 to 248
(for AGs S5-115U, S5-135U and S5-155U in peripheral area IM3)

KY = 3, 0 to 248
(for AGs S5-115U, S5-135U and S5-155U in peripheral area IM4)

**KNKD** Channel number, channel representation

KY = x, y

x = 0 to 3 channel number

y = 20 to 21 channel representation

20 fixed-point representation unipolar (nominal range 0 to 1024)

21 fixed-point representation unipolar (nominal range 256 to 1280)

**OGR** Upper limit

KF = -32 768 to +32 767
(for AG S5-115U)

KG = -1701412+39 to +1701412+39
(for AGs S5-135U and S5-155U)

**UGR** Lower limit

KF = -32 768 to +32 767
(for AG S5-115U)

KG = -1701412+39 to +1701412+39
(for AGs S5-135U and S5-155U)

**XA** Output value

contains the standardized analog value within the limits of UGR and OGR as 16 bit fixed-point value (for AG S5-115U) or as 32 bit floating-point value (for AGs S5-135U and S5-155U).

**FB** Error bit

Signal state is "1" if less than 0 units (KD=20) or less than 192 units (KD=21) are read.

**BU** Over-range
Signal state is "1"
- if Bit $2^0$ in read analog value is signal state "1" (overflow)
- if the nominal range is exceeded (depending on parameter KD)

### Using the function block

Function block FB 32 (AE:463) reads the analog value from the analog input module 6ES5 463-4UAxx and makes it available at parameter XA as standardized value (linearly converted within the limits OGR and UGR) in fixed-point representation or in floating-point representation (depending on the AG) for further processing. The function block takes into account the different nominal ranges and numerical representations using parameter KD (channel representation) for standardizing.

### Channel representation

The nominal range of the read analog value differs according to the analog input modul used. So that the function block can correctly carry through standardization within the limits OGR and UGR, the parameter must be supplied according to the following list.

KD = 20 fixed-point representation unipolar with nominal range 0 to +1024

KD = 21 fixed-point representation unipolar with nominal range +256 to +1280

### Troubleshooting

When the function block is left after correct processing the result of logic operation is set to "0" and accumulator 1 contains the value which is read from the module. The standardized value is available at output XA. Signal state "0" is assigned to the parameters FB and BU.

When an unvalid value is assigned, the function block signals this parameter assignment error by the result of logic operation "1" and by an error number in accumulator 1. In this case, the output XA is assigned zero. The parameters FB and BU are assigned signal state "0".

Error number KF =
1: parameter BG < 128 for peripheral P
2: parameter KN > 3
3: sum from parameter BG and 2 times parameter KN > 255
4: parameter KD not 20 or 21
5: parameter OGR < = parameter UGR
6: parameter BG: x, y: x > 3

If a non-existing module or a non-existing channel is selected time-out is signalled.

If error bit FB is set, zero is output at XA. The result of logic operation is set to "1". Accumulator 1 contains the value which is read from the module.

If the overflow bit BU is set the nominal range limit value is output at XA. The result of logic operation is set to "1". Accumulator 1 contains the value which is read from the module.

## Reading in an analog value: 466 FB 33

### Summary

The present programmer's guide describes the function block

| | | |
|---|---|---|
| FB 33 | AE:466 | Read analog values from the module 6ES5 466-3LA1x |

The present programmer's guide expects the knowledge of the instruction manual for the analog input module and the appropriate automation system.

When the peripheral areas Q, IM3 and IM4 in AG S5-115U are used in connection with an interface module, the PESP signal must not be evaluated on the central unit interface module (see instruction manual of the interface module).

### Functional description

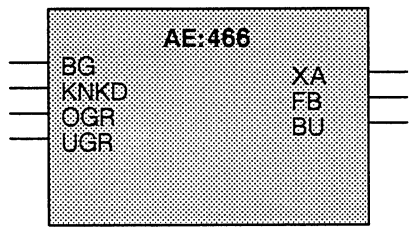
Function block FB 33 (AE:466) "Read analog value" reads an analog value from the analog input module 6ES5 466-3LAxx. It outputs a standardized value at the XA independent of the data format. The user defines the range of values of the value delivered by the function block by the parameters OGR and UGR.

Function block FB 33 for AG S5-115U supplies the standardized value as 16 bit fixed-point number, the function blocks FB 33 for AGs S5-135U and S5-155U supply them as 33 bit floating-point number.

### Calling the function block

```
      :  SPA FB 33
NAME  :  AE:466
BG    :
KNKD  :
OGR   :
UGR   :
XA    :
FB    :
BU    :
```

FB 33




### Explanation of the parameters

| Name | Para | Data | Designation |
|---|---|---|---|
| BG | D | KY | details of the peripheral area and the module address |
| KNKD | D | KY | Information about the channel number and the channel representation |
| OGR | D | KF(1)<br>KG(2) | upper limit of the output value<br>upper limit of the output value |
| UGR | D | KF(1)<br>KG(2) | lower limit of the output value<br>lower limit of the output value |
| XA | Q(3) | W(1)<br>D(2) | output value as fixed-point number<br>output value as floating-point number |
| FB | Q | BI | error bit |
| BU | Q | BI | message range exceeded |

(1) valid for AG S5-115U

(2) valid for AGs S5-135U and S5-155U

(3) If a data (double) word is used as output word the associated data block must be opened before calling FB 33.

**Attention:** If flags are used for parameterization only flag areas from MB 0 to MB 199 may be assigned.

**Parameter assignment**

**BG** Peripheral area, module address

KY = 0, 128 to 224 (16 channels)
(for AGs S5-115U, S5-135U and S5-155U in peripheral area P)

KY = 1, 0 to 224
(for AGs S5-115U, S5-135U and S5-155U in peripheral area Q)

KY = 2, 0 to 224
(for AGs S5-115U, S5-135U and S5-155U in peripheral area IM3)

KY = 3, 0 to 224
(for AGs S5-115U, S5-135U and S5-155U in peripheral area IM4)

**KNKD** Channel number, channel representation

KY = x, y
x = 0 to 15 channel number
y = 22 to 25 channel representation
22 fixed-point representation bipolar (nominal range -2048 to +2048)
23 absolut value representation bipolar (nominal range -2048 to +2048)
24 binary representation unipolar (nominal range 0 to 4095)
25 fixed-point representation unipolar (nominal range +512 to +2559)

**OGR** Upper limit

KF = -32 768 to +32 767
(for AG S5-115U)

KG = -1701412+39 to +1701412+39
(for AGs S5-135U and S5-155U)

**UGR** Lower limit

KF = -32 768 to +32 767
(for AG S5-115U)

KG = -1701412+39 to +1701412+39
(for AGs S5-135U and S5-155U)

**XA** Output value
contains the standardized analog value within the limits of UGR and OGR as 16 bit fixed-point value (for AG S5-115U) or as 32 bit floating-point value (for AGs S5-135U and S5-155U).

**FB** Error bit
Signal state is "1"
- if Bit $2^1$ in read analog value is signal state "1"
- if less than 384 units are read in KD = 25

**BU** Over-range
Signal state is "1" if Bit $2^0$ in read analog value is signal state "1" (overflow).

### Using the function block

Function block FB 33 (AE:466) reads the analog value from the analog input module 6ES5 466-3LAxx and makes it available at parameter XA as standardized value (linearly converted within the limits OGR and UGR) in fixed-point representation or in floating-point representation (depending on the AG) for further processing. The function block takes into account the different nominal ranges and numerical representations using parameter KD (channel representation) for standardizing.

### Channel representation

The nominal range of the read analog value differs according to the analog input modul used. So that the function block can correctly carry through standardization within the limits OGR and UGR, the parameter must be supplied according to the following list.

KD = 22 fixed-point representation bipolar with nominal range -2048 to +2047

KD = 23 absolut value representation bipolar with nominal range -2048 to +2047

KD = 24 fixed-point representation unipolar with nominal range 0 to +4095

KD = 25 fixed-point representation unipolar with nominal range +512 to +2559

### Troubleshooting

When the function block is left after correct processing the result of logic operation is set to "0" and accumulator 1 contains the value which is read from the module. The standardized value is available at output XA. Signal state "0" is assigned to the parameters FB and BU.

When an unvalid value is assigned, the function block signals this parameter assignment error by the result of logic operation "1" and by an error number in accumulator 1. In this case, the output XA is assigned zero. The parameters FB and BU are assigned signal state "0".

Error number KF =
1: parameter BG < 128 for peripheral P
2: parameter KN > 15
3: sum from parameter BG and 2 times parameter KN > 255
4: parameter KD < 22 or > 25
5: parameter OGR parameter UGR
6: parameter BG: x, y: x > 3

If a non-existing module or a non-existing channel is selected time-out is signalled.

If error bit FB is set, zero is output at XA. The result of logic operation is set to "1". Accumulator 1 contains the value which is read from the module.

If the overflow bit BU is set the nominal range limit value is output at XA. The result of logic operation is set to "1". Accumulator 1 contains the value which is read from the module.

## Outputting an analog value

### Functional description RLG:AA FB41

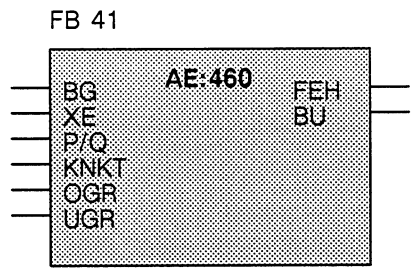
Function block RLG:AA converts an input value XE into an output value for an analog output module within the range specified by OGR and UGR. If this range is exceeded, an error message is output.

### Calling the function block

Representation in STL

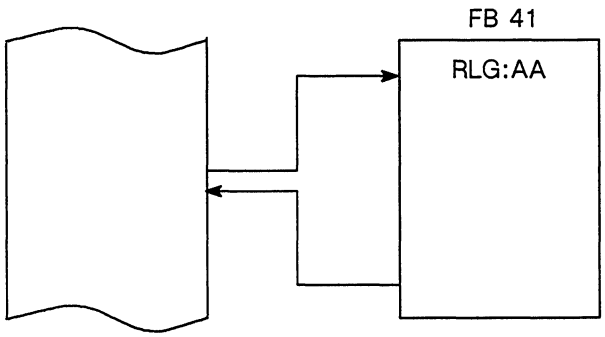
```
      :  SPA FB 41
NAME  :  RGL:AA
XE    :
BG    :
P/Q   :
KNKT  :
OGR   :
UGR   :
FEH   :
BU    :
```

Graphic representation




| Name | Para | Data | Designation | Notes |
|---|---|---|---|---|
| XE | I | D | address of input value XE | input value (floating-point number) within the limits of UGR...OGR |
| BG | D | KF | address of output value XE | P/Q=P: BG= +128...240<br>P/Q=Q: BG= + 0 ....240 |
| P/Q | D | KC | peripheral area normal/extended range | P/Q=P: normal range<br>P/Q=Q: extended range |
| KNKT | D | KY | channel number<br>channel type | KN= 0...7<br>KT= 0/1 (see Note) |
| OGR | D | KG | upper limit of input value XE | -1701412+39<=OGR<=+1701412 +39 |
| UGR | D | KG | lower limit of input value XE | -1701412+39<=OGR<=+1701412 +39 |
| FEH | Q | BI | address for message FEH | FEH=0: UGR not equal OGR<br>FEH=1: UGR equals OGR |
| BU | Q | BI | address for message range exceeded | BU=0: UGR<=XE<=OGR<br>BU=1: XE<=UGR or XE <=OGR |

**Program structure**




**Notes**

- Channel type KT
  The usable analog outputs convert the input value XE to an output value XA in two different forms of representation. The form of representation can be selected with the parameter KT, as follows:

  KT = 0 unipolar representation
  KT = 1 bipolar representation

  If a value other than zero is selected for the value KT, the function block selects the form of representation KT=1.

- Output value XA
  The input value XE is converted depending on the selected channel type KT according to the following formulas:

$$KT=0 \qquad XA = \frac{1024 * (XE-UGR)}{OGR-UGR}$$

$$KT=1 \qquad XA = \frac{1024 * (2 * XE- (OGR-UGR))}{OGR-UGR}$$

Key:

| | |
|---|---|
| OGR | upper limit |
| UGR | lower limit |
| XA | output value |
| XE | input value |

- Range limits UGR, OGR
  If suitable limit values are selected, the input value XE can be specified directly as a physical value.

**Example**

| Input value XE | Range limits UGR | OGR | Output value XA |
|---|---|---|---|
| -5000 ... +5000 | -5000000+04 | +5000000+04 | -1024 ... +1024 |

# 5 Special functions

## 5.1 Overview

### Timer block

The function block is used to read and set the hardware clock on CPU 946/947.

### Read/Write for expanded periphals

The function blocks are used to read and write from/to the expanded peripherals.

### Save and load

The function blocks save and load saved the data or transfers the saved data of the "flag scratchpad area" when cyclic program execution is interrupted.

### Code converter

The function block COD:B8 converts BCD code to binary code.
The function block COD:32 converts binary code to BCD code.

### Copying data areas

The function blocks copy an optional data area from one data block to another.

## 5.2 Description

## Date and time FB129

### Functional description

The function block SST:UHR is used to set or read the time of day. The new/current time of day is transferred to/taken from data block 5 before/after the function block is called. Before the function block is called, the function identifier 1/2 to set/read the time of day must be entered in data word 0. The time and date are maintained on the CPU (battery-buffered); the current time of day is therefore also available following a power failure. After the function block has been called, the time of day function identifier is cleared.

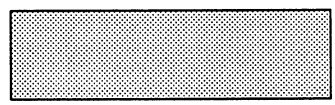
### Calling the function block

Representation in STL

```
      :  JU FB 129
NAME  :  SST: UHR
```

Graphic representation

FB 129



### Assignment of the data block

| | | | | |
|---|---|---|---|---|
| DW 0 | 0 | | Function identifier | |
| DW 1 | Tens Sec. | Ones Sec. | x | |
| DW 2 | Tens Hr. | Ones Hr. | Tens Min. | Ones Min. |
| DW 3 | Tens Day | Ones Day | Weekday | 0 |
| DW 4 | Tens Year | Ones Year | Tens Mon. | Ones Mon. |
| DW 5 | Tens Sec. | Ones Sec. | n * 10 ms | |
| DW 6 | Tens Hr. | Ones Hr. | Tens | Ones |
| DW 7 | Tens Day | Ones Day | Weekday | 0 |
| DW 8 | Tens Year | Ones Year | Tens Mon. | Ones Mon. |
| DW 9 | PAFE | | | |
| DW 10 | Reserved | | | |
| DW 11 | Reserved | | | |

### Note

The data block must be made available by the user and have the correct length!

## Permissible values:

| | | |
|---|---|---|
| ones | Seconds: | 0..9 |
| tens | Seconds: | 0..5 |
| ones | Minutes: | 0..9 |
| tens | Minutes: | 0..5 |
| ones | Hours: | 0..9 |
| tens | Hours: | 0..1 / 0..2 (*) |
| Weekday: | | Mon..Sun = 1..7 |
| ones | Days: | 0..9 |
| tens | Days: | 0..3 |
| ones | Months: | 0..9 |
| tens | Months: | 0..1 |
| ones | Years: | 0..9 |
| tens | Years: | 0..9 |
| n (*10ms): | | 0..99 |

(*): Bit 15:
1: 24 hour format
0: 12 hour format
Bit 14:
0: AM
1: PM

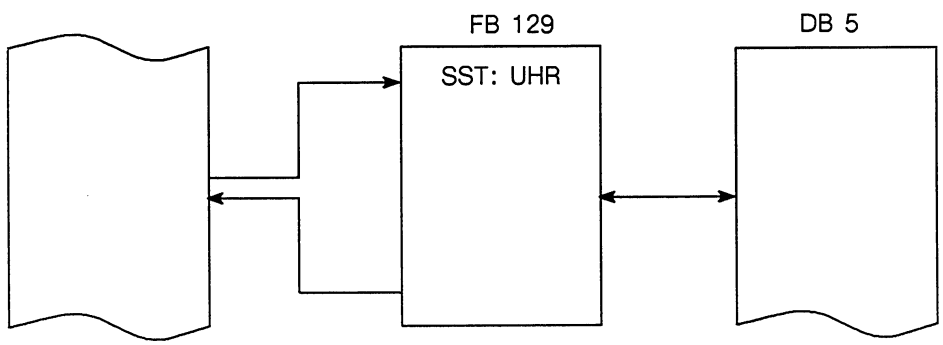
## PAFE:

Error messages when using the function "set time of day" (fct. no.:1):

Error number (transfer in accumulator 1):

| | |
|---|---|
| F101H | Specified year invalid |
| F102H | Specified month invalid |
| F103H | Specified day invalid |
| F104H | Specified weekday invalid |
| F105H | Specified hour invalid |
| F106H | Specified minute invalid |
| F107H | Specified second invalid |

### Program structure



# Read and write extended peripherals FB 196/FB 197 (PER:ET)

## Summary

The present programmer's guide describes the function block

**FB 196 (PER:ET)** Read and write expanded peripherals (direct parameterization)

**FB 197 (PER:ET)** Read and write expanded peripherals (indirect parameterization)

The function blocks are installed into the AGs

**S5-115U, S5-135U and S5-155U**

connected with the following interfaces:

IM 300-3, IM 300-5(-5CA11), IM 300-5(-5LB11), IM 301-3, IM 301-5, IM 304, IM 305, IM 306, IM 307, and IM 308.

The automation systems in which the individual interfaces can be installed are listed in the appropriate manual.

The present programmer's guide expects the knowledge of the instruction manual for the analog input module and the appropriate automation system.

**Functional description**

Function block "Read/write extended peripherals" transfers a peripheral area through an interface mentioned above to a CPU internal area or vice versa (as required, can be selected with parameters).

Thus input bytes can be read from the expanded peripherals and output bytes can be written to the expanded peripherals.

The following table lists all source and target areas which are addressable.

| Area | | S5-115U | S5-135U | | S5-155U |
|---|---|---|---|---|---|
| | | | CPU 922, 928-.UA. | CPU 928-.UBA. | |
| Peripheral area | P (byte no.) | 128 - 255 | 128 - 255 | 128 - 255 | 128 - 255 |
| | Q (byte no.) | 0 - 255 | 0 - 255 | 0 - 255 | 0 - 255 |
| | M3 (byte no.) | 0 - 255 | 0 - 255 | 0 - 255 | 0 - 255 |
| | IM4 (byte no.) | 0 - 255 | 0 - 255 | 0 - 255 | 0 - 255 |
| internal area | DB (DW no.) | 10 - 2042 | 10 - 2042 | 10 - 2042 | 10 - 2042 |
| | DX (DW no.) | - | 10 - 2042 | 10 - 2042 | 10 - 2042 |
| | Flag (byte no.) | 0 - 235 | 0 - 235 | 0 - 235 | 0 - 235 |
| | S flag (byte no.) | - | - | 0 - 1023 | 0 - 4095 |

If an DB or DX module is selected as internal area the user must establish the module in the required length before the function block call.

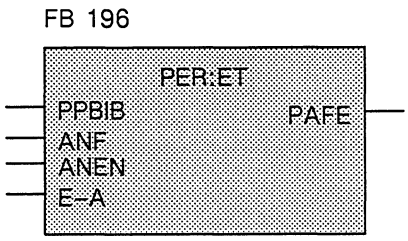
**Note:** The I/O modules addressed when one block is called, must have consecutive addresses. Following each gap in the addresses, the function block PER:ET must be called again. It is therefore advisable to assign the input bytes and output bytes consecutively for all the interfaces, which means that PER:ET needs only be called once for each direction of transfer.

### Calling the function block

Representation in STL

```
      :  JU FB 196
NAME  :  PER:ET
PBIB  :
ANF   :
ANEN  :
E-A   :
PAFE  :
```

Graphic representation




### Explanation of the inputs and outputs (parameters)

| Name | Para | Data | Designation |
|---|---|---|---|
| PBIB | D | KY | Specifies the area to be processed, |
| ANF | D | KF | specifies the first byte of the internal area |
| ANEN | D | KY | Specifies the first ans last peripheral byte |
| E-A | I | BI | Specifies the direction of transfer |
| PAFE | Q | BI | Parameter assignment error |

### Parameter assignment

PBIB The area to be processed is specfied

KY = x, y x = Selection of the area types between which the data are to be transferred

| | | | | |
|---|---|---|---|---|
| x = | 16 | area P | ↔ | DB block |
| | 17 | | | DX block(1) |
| | 18 | | | flag |
| | 19 | | | S flag (2) |
| x = | 32 | area Q(3) | ↔ | DB block |
| | 33 | | | DX block |
| | 34 | | | flag |
| | 35 | | | S flag (2) |
| x = | 48 | area IM3(3) | ↔ | DB block |
| | 49 | | | DX block |
| | 50 | | | flag |
| | 51 | | | S flag (2) |
| x = | 64 | area IM4(3) | ↔ | DB block |
| | 65 | | | DX block |
| | 66 | | | flag |
| | 67 | | | S flag (2) |

(1) not valid for AG S5-115U

(2) valid for AG S5-135U, CPU 928B and AG S5-155U, CPU 946/947

(3) If this area is used with AG S5-115U the PESP signal may not be evaluated on the interface in the CPU.

y = DB-/DX number (when transferred to/from a data block)
1 <= y<= 255

y= not applicable (for copy of flag or S flag area)

If the DB/DX area was selected on parameter PBIB of the function block PER:ET the appropriate area must be established (see "Data exchange through DB/DX block" for length of the block).

If the flag area was selected on parameter PBIB of the function block PER:ET the parameters ANF and ANEN define the flag assignment (see "Data exchange through flag/S-flag" for flag bytes).

**ANF:** Start byte of the internal area (number of DW or flag byte):
The parameter defines the start address of the block to be transferred in the data or flag area.

| | | |
|---|---|---|
| KF = +x | 0 <= x <= 2042 | DW number for data blocks |
| | 0 <= x <= 199 | byte number for flags |
| | 0 <= x <= 1023 | byte number for S-flags (AG 135U, CPU 928B) |
| | 0 <= x <= 4095 | byte number for S-flags (AG 155U) |

**ANEN:** KY = x, y x = number of start byte
0 <= x <= 255

y = number of end byte
x <= x <= 255

y - x <= 254 (only 255 bytes can be transferred at one time)

**E-A:** Direction of transfer: "0" signal: read inputs
"1" signal: write outputs

**PAFE:** Parameter assignment error

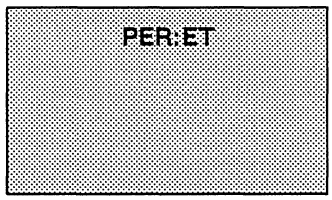
If the function block FB 196 is parameterized uncorrectly parameter PAFE signals "1". The error number can then be read from the accu 1:

| | | |
|---|---|---|
| Error number | 1: | area identification not permitted |
| | 2: | DB/DX number not permitted |
| | 3: | DB/DX not existing/too short |
| | 4: | parameter ANF not permitted |
| | 5: | parameter ANEN not permitted |
| | 6: | over range |

**Calling the function block**

```
      :  L KY x,y
      :  L KF +z
      :
      :  SPA FB 197
NAME  :
      :
      :  T MB ..
```

**FB 197**




Assignment of accumulators 1 and 2 before function block 197 call:

**Accu 2:** KY=x,y

| | | |
|---|---|---|
| x = | area that contains the parameter array | |
| x = | 00 parameter array contained in | DB block |
| | 01 | DX block(1) |
| | 02 | flag |
| | 03 | S flag(2) |

(1) not valid for AG 115U

(2) valid only for AG 135U, CPU 928B and AG S5-155U, CPU 946/947

| | |
|---|---|
| y = | DB number (x = 00), 1 <= y <= 255 |
| y = | DX number (x = 01), 1 <= y <= 255 |
| y = | not applicable (x = 02 or X = 03) |

If the parameter array is defined in a DB/DX block for the indirect parameterization this area must be established and preassigned before the FB 197 is called.

**Accu 1:** KF = +z address of the first parameter within the assigned area:
e.g. x = 5
Parameter array starts at
- DW 5 (x = 00 or x = 01)
- MB 5 (x = 02)
- SY 5 (x = 03)

Area for

| | | |
|---|---|---|
| - DW: | 0 <= z <= 2039 | |
| - MB: | 0 <= z <=193 | |
| - SY: | 0 <= z <= 1017 | (CPU 928B) |
| - SY: | 0 <=z <= 4089 | (CPU 946/947) |

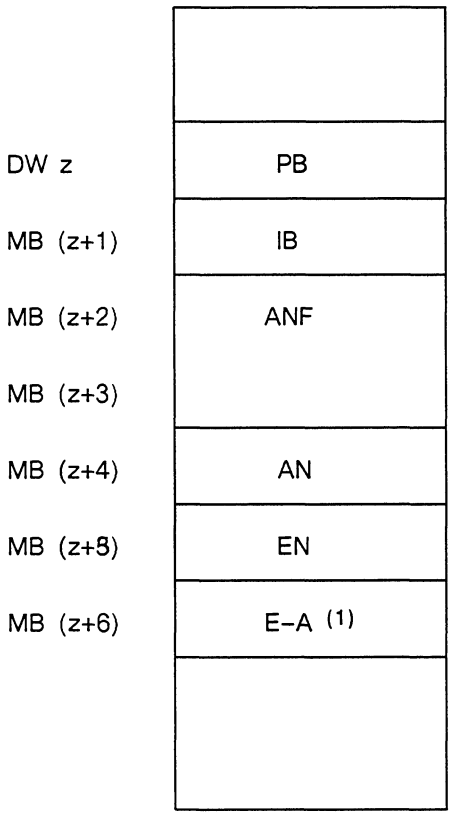
The parameter array must be preassigned as follows before the function block FB 197 is called:

The parameter array is in DB/DX block (x = 00 or x = 01)

DB/DX block

| | | |
|---|---|---|
| | | |
| DW z | PB | IB |
| DW (z+1) | ANF | |
| DW (z+2) | AN | EN |
| DW (z+3) | E-A (1) | -- |
| | | |

The parameter array is in flag/S-flag (x = 02 or x = 03)




(1) The least significant bit is assigned for parameter E-A.

A parameter assignment error is signalled by VKE = "1" after the function block FB 197 is left. The error number can then be read from the accu 1:

| | | |
|---|---|---|
| Error number | 1: | area identification not permitted |
| | 2: | DB/DX number not permitted |
| | 3: | DB/DX not existing/too short |
| | 4: | parameter ANF not permitted |
| | 5: | parameter ANEN not permitted |
| | 6: | over range |
| | 7: | area identification in accu 2 not permitted |
| | 8: | DB/DX number in accu 2 not permitted |
| | 9: | start of parameter array (accu 1) not permitted |
| | 10 | :DB/DX not existing/too short |

The error numbers 7 to 10 r efer to the contents of accu 1 and 2 before the function block FB 197 was called.

### Using the function blocks

The user may apply DB/DX blocks, flags or S-flags for the internal area (image of the inputs, image of the outputs).

Before the function blocks PER:ET are called a data block which is long enough must be established if a DB/DX block was selected as internal area. The block must be established at least to data word x where

$$X = ANF + \frac{EN - AN}{2}$$ even numbered for AN and EN or odd numbered for AN and EN

$$X = ANF + \frac{EN - AN - 1}{2}$$ even numbered for AN and odd numbered for EN

$$X = ANF + \frac{EN - AN + 1}{2}$$ odd numbered for AN and even numbered for EN

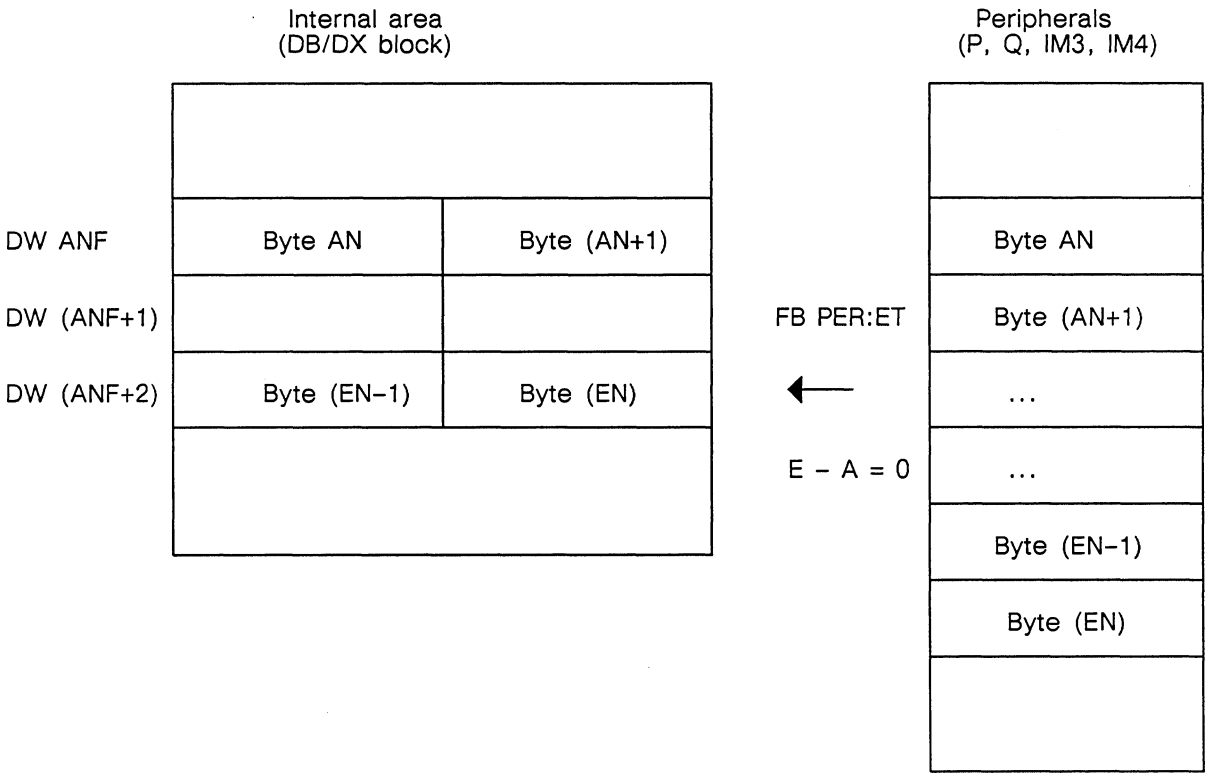
The same applies for DX blocks.

### Using the flag/S-flag areas

If peripheral areas are copied through flags or S-flags the following bytes are assigned:

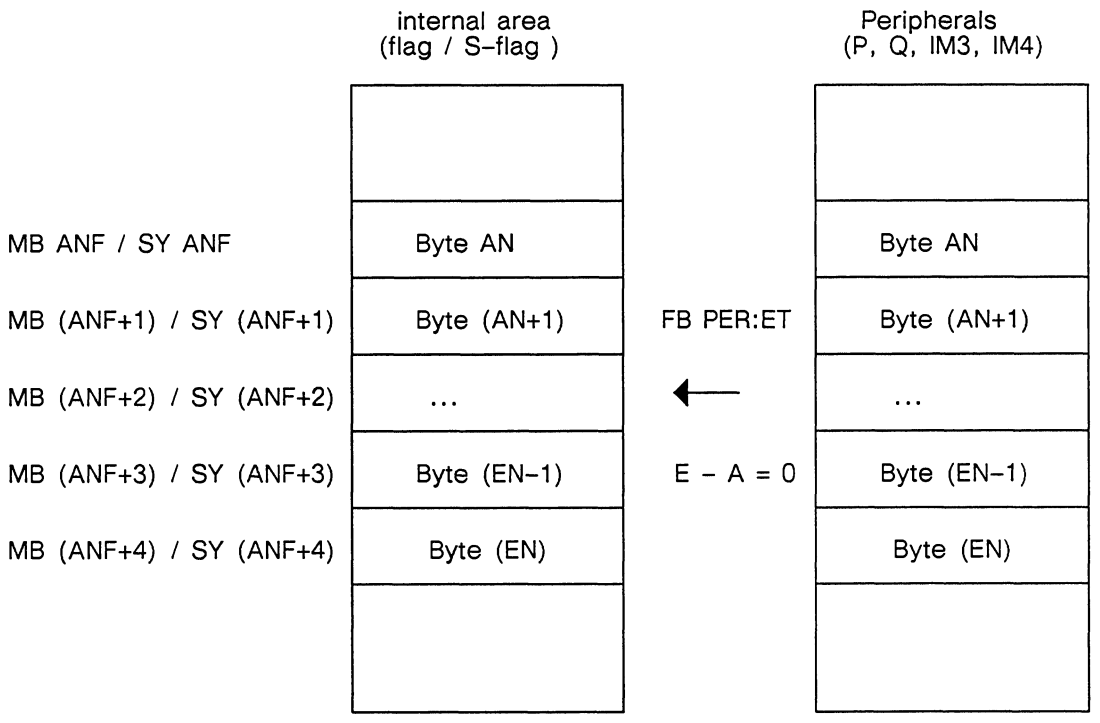
MB ANF to MB (ANF+EN-AN) in the flag area
SY ANF to SY (ANF+EN-AN) in S-flag area

**Reading the inputs in DB/DX block**

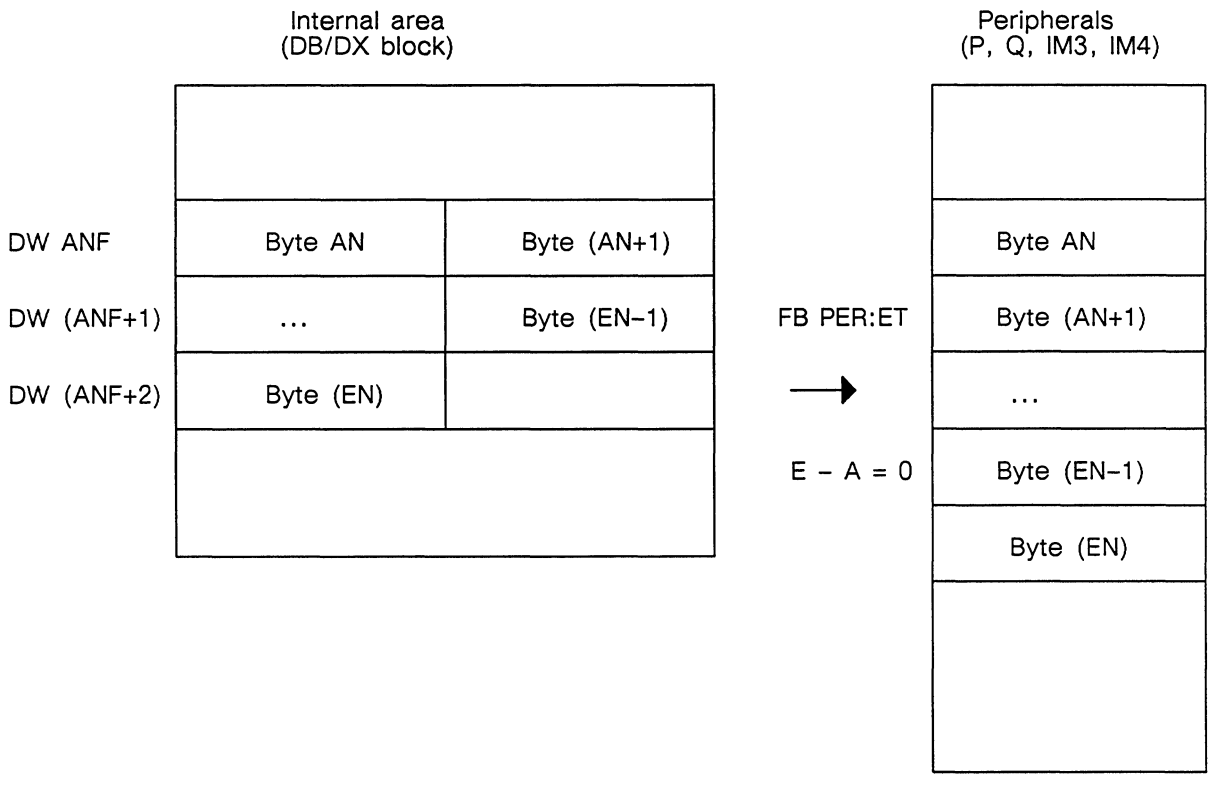



The left-hand data byte is always assigned an even peripheral byte, the right-hand data byte is always assigned an odd peripheral byte.

**Reading the inputs in flag/S-flag**

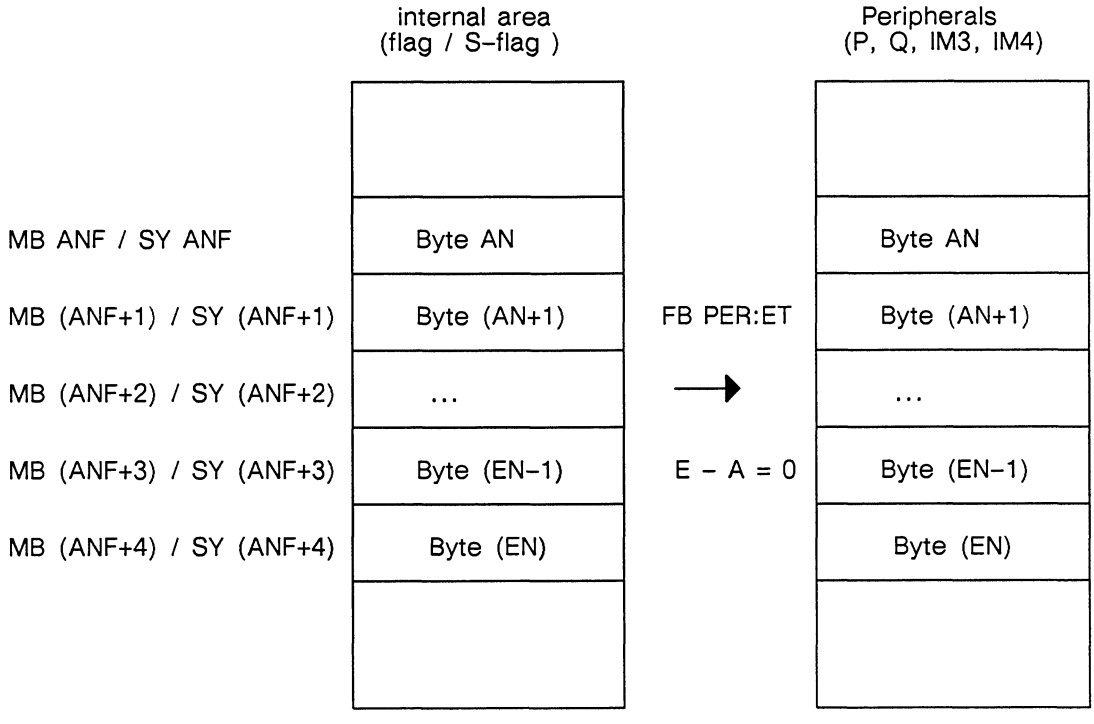


**Transferring the outputs from DB/DX block**




The left-hand data byte is always assigned an even peripheral byte, the right-hand data byte is always assigned an odd peripheral byte. In this case an even byte was supposed for AN and for EN.

**Transferring the outputs from flag/S-flags**



## Save / load

### Functional description

The function block "save" saves the data of the "scratchpad flag" when cyclic program execution is interrupted by time/interrupt-driven processing as well as by reset.
If a cyclic user program with open loop controls has time-driven closed loop controls and/or interrupt-driven programs added, there is a risk that the data of the function blocks in the scratchpad area (FW 200 to FW 254) of the cyclic program will be overwritten, since the function blocks of the time or interrupt-driven programs also use this scratchpad area.
FB 38 should be called at the beginning of each time or interrupt-driven program section. This saves the data of the scratchpad area in its data block.
The function block "load" transfers the saved data of the "scratchpad area" back to the "scratchpad area" following a time or interrupt-driven stoppage in the cyclic program.
The blocks FB 38 and FB 39 should always be used as pairs.
FB 39 is called at the end of a time or interrupt-driven stoppage after FB 38 has transferred the data of the scratchpad area to the data block at the beginning of the interruption. The function blocks can be called in a maximum of 16 different interrupt levels and save the "scratchpad area" in a "stack".

### Conditions

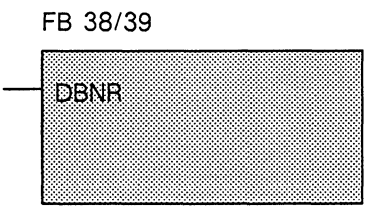
FB 38 and FB 39 must use the same data block.
The data block must be 817 data words long (DW 0 to DW 816).
The blocks FB 38 and FB 39 should be used with all time and interrupt-driven programs/as well as by reset.

### Calling the function block

Representation in STL

```
      :  JU FB 38/39
NAME  :  SAVE/LOAD
DBNR  :
```

Graphic representation




### Explanation of the inputs and outputs (parameters)

| Name | Para | Data | Designation |
|---|---|---|---|
| DB | B | - | Data block in which the data of the "scratchpad flag" are to be written |

| **Note** | In case of an error, the CPU goes to STOP. |
|---|---|

Possible errors (error number in accu. 1):

F001 ≙ DB save/load not available

F002 ≙ flag stack overflow

F003 ≙ DB save/load too short

**Remedy**

F001:
transfer DB save/load to PC.

F002:
always use FB 38/39 as a pair.
Call FB 38 at the beginning of the interrupting OB and FB 39 at the end.
The OB must not be terminated with BEC between these calls.

F003:
load DB save/load with a length of 817.

## Code converter

### Function

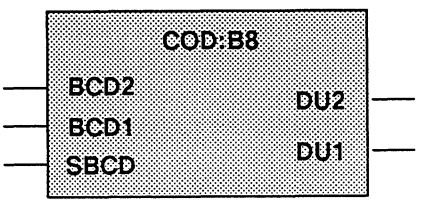
Converts BCD code to binary code.

**COD:B8 (FB 21)** Input BCD number, 8 decades with sign
Result is fixed-point binary number 31 + 1 bit.

### Calling the function block FB 21

```
      :  SPA FB21
NAME  :  COD:B8
BCD2  :
BCD1  :
SBCD  :
DU2   :
DU1   :
```

**FB 21**




### Explanation of the parameters

| Name | Para | Data | Designation |
|---|---|---|---|
| BCD2 | I | W | BCD number, dec. 4 to 7, -99999999 to +99999999 permitted |
| BCD1 | I | W | BCD number, dec. 0 to 3, -99999999 to +99999999 permitted |
| SBCD | I | BI | sign BCD number, "1" = negative |
| DU2 | Q | W | binary number, bit 0 to 15 |
| DU1 | Q | W | binary number, bit 16 to 31 |

Counter used: Z0

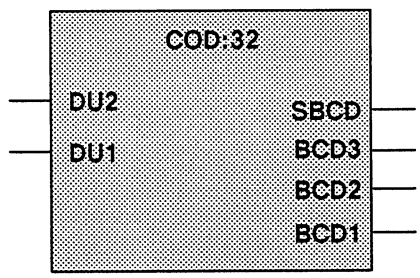
**Function**

Converts binary code to BCD code.

**COD:32 (FB 23)** Input fixed-point binary number 31 + 1 bit,
Result is BCD number, 10 decades, with sign

```
      :  SPA FB23
NAME  :  COD:32
DU2   :
DU1   :
SBCD  :
BCD3  :
BCD2  :
BCD1
```

**FB 23**




**Explanation of the parameters**

| Name | Para | Data | Designation | |
|---|---|---|---|---|
| DU2 | I | W | binary number, bit 16 to 31 | $-2^{31}$ to $+2^{31}-1$ permitted |
| DU1 | I | W | binary number, bit 0 to 15 | $-2^{31}$ to $+2^{31}-1$ permitted |
| SBCD | Q | BI | sign BCD number, "1" = negative | |
| BCD3 | Q | BY | BCD number, dec. 8 to 9 | |
| BCD2 | Q | W | BCD number, dec. 4 to 7 | |
| BCD1 | Q | W | BCD number, dec. 0 to 3 | |

## Copying data areas

### Summary

The present programmer's guide describes the function blocks

FB 44 (direct parameterization)

FB 45 (indirect parameterization)

### Functional description

The function blocks copy a data area defined by the user (0 – 255 data words) from one DB/DX data block to another.

Both function blocks (FB 44 and FB 45) have the same function. Only their parameterization is different. The parameters for FB 44 are given when calling them (direct parameterization) while the function block by passing a flag to a parameter array (indirect parameterization).

The parameter array consists of 5 data words which are assigned the parameters QTYP, QANF, LAEN, ZTYP and ZANF. The user defines the data area to be copied by the parameter QANF (start data word of the source) and LAEN (length of the block). The source block is selected by the parameter QTYP (DB/DX, block number) and the destination data block is selected by the parameter ZTYP. The parameter ZANF (start data word of the destination) defines the place of the first word copied.

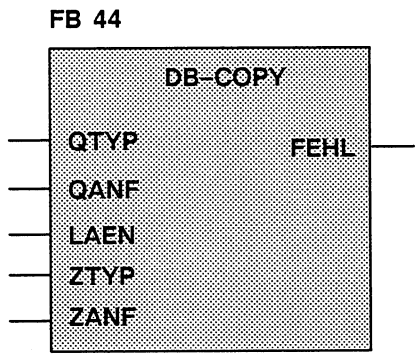
### Function

Copying of data blocks from one data block to another, which are in the data area (DB) as well as in the expanded data area (DX).
Please note that for AG S5-115U the simple data area (DB) is applicable. If a block of the DX data area is parameterized an error message will appear.

**Calling the function blocks**

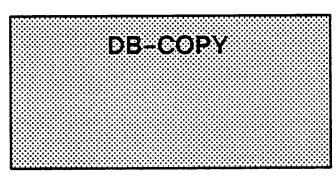
**FB 44**

```
      :  JU FB 44
NAME  :  DB-COPY
QTYP  :
QANF  :
LAEN  :
ZTYP  :
ZANF  :
FEHL  :
```




**FB 45**

```
      :  L KY x,y
      :  L KF + z
      :
      :  JU FB4 45
NAME  :  DB-COPY
      :
      :  T    MB ..
```




Assignment of the accumulators 2 and 1 before calling the function block FB 45:

**Accu 2:** KY=x,y

| | | |
|---|---|---|
| x= | block that contains the parameter array | |
| x = | 1 | DB block |
| | 2 | DX block(1) |

(1) not valid for AG 115U

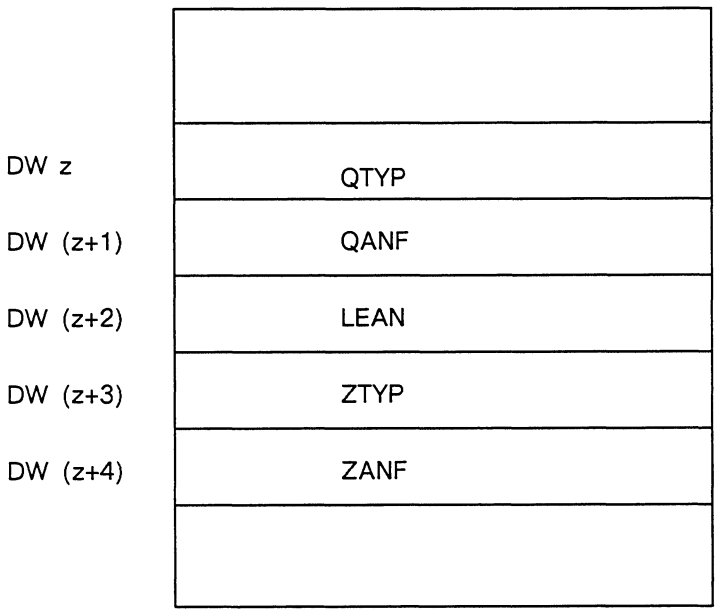
y = DB number (x = 1), 1 <= y <= 255
y = DX number (x = 2), 1 <= y <= 255

**Accu 1:** KF = +z address of the first parameter (number of the first data word) within the defined block.

0 <= z <= 251

The parameter array must be preassigned as follows before the function block is called:

DB/DX block




Explanation of the parameters

| Name | Para | Data | Designation |
|---|---|---|---|
| QTYP | D | KY | defines the source DB type and the DB number |
| QANF | D | KF | source start data word |
| LAEN | D | KF | lenght of the data area to be copied |
| ZTYP | D | KY | defines the destination DB type and the DB number |
| ZANF | D | KF | destination-start data word |
| FEHL | Q | BY | error byte (not for FB 45) |

**Parameter assignment**

QTYP: KY=x,y x = type of block
x = 1 → DB
x = 2 → DX

y = number of block
1 <= y <= 255

ZTYP: corresponding to QTYP

QANF: KF = 0 to 2042

ZANF: corresponding to QANF

LAEN: KF = 0 to 255

QANF + LAEN <= 2043

ZANF + LAEN <= 2043

FEHL: contains the error number if parameterization was not permitted

A parameter assignment error is signalled by VKE = "1" after the function block FB 45 is left. The error number can then be read from the accu 1:

**The following error numbers can be output:**

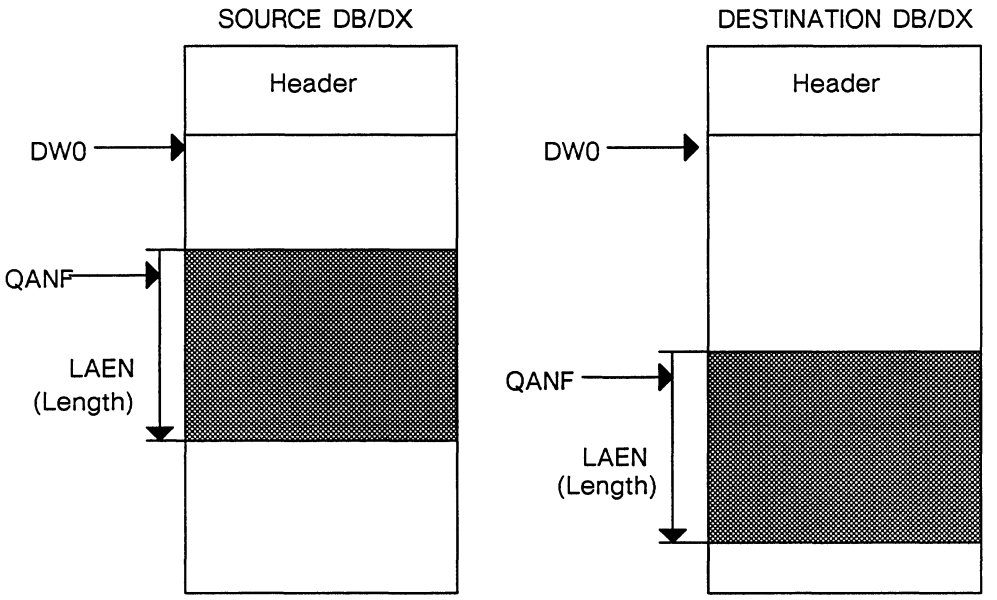
Err.no. 1 source DB QTYP; x not permitted
Err.no. 2 source DB QTYP; y not permitted
Err.no. 3 ZTYP, x not permitted
Err.no. 4 ZTYP, y not permitted
Err.no. 5 data word number (QANF) of the source > 2042
Err.no. 6 data word number (ZANF) of the destination> 2042
Err.no. 7 number of data words to be copied > 255
Err.no. 8 the sum of data word number of the source (QNAF) plus block length (LEAN) is greater than the data block length of 2042
Err.no. 9 the sum of data word number of the destination (ZNAF) plus block length (LEAN) is greater than the data block length of 2042
Err.no. 10 source data block does not exist
Err.no. 11 source data block too small
Err.no. 12 destination data block does not exist
Err.no. 13 destination data block too small
Err.no. 14 destination data block is an EPROM
Err.no. 15 parameter DB accu 2, x not permitted
Err.no. 16 accu 2, y: DB/DX not existing
Err.no. 17 accu 2, y: DB/DX number not permitted
Err.no. 18 address of the first parameter outside the valid data area (accu 1). 0 <= z <= 251
Err.no. 19 overlapping of source and destination data area

The error numbers 15 to 18 refer to the contents of accu 1 and 2 before calling the function block FB 45.

**Assignment of the data area**

No data blocks are opened in these function blocks.

**Using the function blocks**




The function blocks copy a defined number of data words (LAEN) from a source DB/DX (QTYP) to a destination DB/DX (ZTYP).

The parameters QTYP and ZTYP define the block type (DB or DX) of the source and the destination as well as the block number (1 – 255).

The first data word to be copied is defined by the source data word number (QANF). The place where the data to be copied are written, is defined by the destination data word number (ZANF).

A number of plausibility questions guarantees that false parameter inputs that are documented as error byte are intercepted.

# 6 Technical data

## S5-95U, S5-100U (CPU 103): Program file S5AA01ST.S5D

| Block designation | | Library number P71200- | Block length/ words | Call length/ words | Secondary blocks | Nesting depth | block | Execution time in ms | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | S5-95U | S5-100U (CPU 103 |
| FB 1 | ADD: 32 | S 1001-A-1 | 108 | 10 | -- | 0 | FB 1 | 1,8 | 2,1 |
| FB 2 | SUB: 32 | S 1002-A-1 | 108 | 10 | -- | 0 | FB 2 | 1,8 | 2,1 |
| FB 3 | MUL: 32 | S 1003-A-1 | 399 | 11 | -- | 0 | FB 3 | 8,3 - 31,4 | 3,2 - 5,4 |
| FB 4 | DIV: 32 | S 1004-A-2 | 395 | 14 | -- | 0 | FB 4 | 2,1 -25,8 | 2,4 - 4,2 |
| FB 5 | RAD: 16 | S 1005-A-1 | 126 | 6 | -- | 0 | FB 5 | 5,3 | 1.3 |
| FB 10 | REG: SCHB | S 1010-A-1 | 250 | 14 | -- | 0 | FB 10<br>8-Bit<br>128-Bit | 5,3<br>77,4 | 2,3<br>28,8 |
| FB 11 | REG: SCHW | S 1011-A-1 | 125 | 10 | -- | 0 | FB 11<br>K=64-254<br>bei SR<br>bei SL | <br><br>20,1 - 81,0<br>31,4 - 122,8 | <br><br>27,0 - 96,0<br>37,0 - 144,0 |
| FB 12 | REG:FIFO | S 1012-A-1 | 160 | 11 | -- | 0 | FB 12 | 1,4 | 1,6 |
| FB 13 | REG: LIFO | S 1013-A-1 | 117 | 11 | -- | 0 | FB 13 | 1,4 | 1,4 |
| FB 30 | AE:464 | S 1030-A-1 | 395 | 9 | FB 4,<br>FB 242 | 1 | FB 30 | 5,8 | 3,5 |

Notes:

- Block analog value output is integrated in the operating system of the AG.
- Assignment in the flag area: MB 200 - 255.
- Because the program can be interrupted at command boundaries, the OBs must be sa ved in a data block at the start and called at the end of all interrupting OBs in which the flags MB 200 - MB 255 are used.

## S5-115U (CPU 941 - 944 **):

Program file S5AA52ST.S5D

| Block designation | | Library number P71200- | Block length/ words | Call length/ words | Secondary blocks | Nesting depth | Execution time in ms |
|---|---|---|---|---|---|---|---|
| FB 1 | ADD: 32 | S 5001-A-1 | 108 | 10 | -- | 0 | see next Page |
| FB 2 | SUB: 32 | S5002-A-1 | 108 | 10 | -- | 0 | |
| FB 3 | MUL: 32 | S 5003-A-1 | 399 | 11 | -- | 0 | |
| FB 4 | DIV: 32 | S 5004-A-2 | 395 | 14 | -- | 0 | |
| FB 5 | RAD: 16 | S 5005-A-1 | 126 | 6 | -- | 0 | |
| FB 10 | REG: SCHB | S 5010-A-1 | 250 | 14 | -- | 0 | |
| FB 11 | REG: SCHW | S 5011-A-1 | 125 | 10 | -- | 0 | |
| FB 12 | REG:FIFO | S 5012-A-1 | 160 | 11 | -- | 0 | |
| FB 13 | REG: LIFO | S 5013-A-1 | 117 | 11 | -- | 0 | |
| FB 21 | COD:B8 | S 5021-A-1 | 183 | 7 | -- | 0 | |
| FB 23 | COD:32 | S 5023-A-1 | 302 | 8 | -- | 0 | |
| FB 30 | AE:464 | S 5030-A-1 | 418 | 9 | FB4, FB242 | 1 | |
| FB 31 | AE:460 | S 5031-A-1 | 315 | 11 | FB242 | 1 | |
| FB 32 | AE:463 | S 5032-A-1 | 235 | 9 | FB242 | 1 | |
| FB 33 | AE:466 | S 5033-B-1 | 238 | 9 | FB4, FB242 | 1 | |
| FB 44 | DB-COPY | S 5044-A-1 | 244 | 8 | -- | 0 | |
| FB 45 | DB-COPY | S 5045-A-1 | 302 | 2 | -- | 0 | |
| FB 196 | PER:ET | S 5196-A-1 | 213 | 7 | -- | 0 | |
| FB 197 | PER:ET | S 5197-B-1 | 272 | 2 | -- | 0 | |

Notes:

- Block analog value output is integrated in the operating system of the AG.
- Assignment in the flag area: MB 200 - 255.
- The counter Z0 ist set in FB 1.
- Because the program can be interrupted at command boundaries, the OBs must be sa ved in a data block at the start and called at the end of all interrupting OBs in which the flags MB 200 - MB 255 are used.

**) from order number 6ES5 94x-7UBxx on

## S5-115U (CPU 941 - 944 **): Program file S5AA52ST.S5D

| Block | Excution time (in ms) CPU | | | |
|---|---|---|---|---|
| | CPU 941 **) | CPU 942 **) | CPU 943 **) | CPU 944 **) |
| FB 1 | 2,8 | 2,8 | 2,6 | < 0,2 |
| FB 2 | 2,5 | 2,5 | 2,3 | < 0,2 |
| FB 3 | 4,0 - 8,2 | 4,0 - 8,2 | 3,5 - 5,6 | 0,6 - 2,7 |
| FB 4 | 5,9 - 6,8 | 5,9 - 6,8 | 4,2 - 4,7 | 1,7 - 2,2 |
| FB 5 | 1,6 | 1,6 | 1,4 | 0,3 |
| FB 10<br>by 8 bit<br>by 128 bit | <br>6,7<br>33,4 | <br>6,7<br>33,4 | <br>6,4<br>31,4 | <br>0,5<br>2,5 |
| | | | | |
| FB 11<br>k = 64-254<br>by SR<br>by SL | <br><br>31,0 - 117,0<br>44,1 - 170,4 | <br><br>31,0 - 117,0<br>44,1 - 170,4 | <br><br>29,9 - 113,1<br>43,0 - 166,0 | <br><br>1,5 - 5,2<br>2,3 - 8,7 |
| FB 12<br>by SR<br>by LO | <br>1,8<br>2,3 | <br>1,8<br>2,3 | <br>1,7<br>2,2 | <br>0,2<br>0,3 |
| FB 13<br>by SR<br>by LO | <br>1,8<br>2,4 | <br>1,8<br>2,4 | <br>1,7<br>2,3 | <br>< 0,2<br>0,3 |
| FB 21 | 3,3 | 3,3 | 2,5 | 0,9 |
| FB 23 | 3,5 | 3,5 | 2,5 | 1,2 |
| FB 30 | 4,5 | 4,5 | 3,9 | 2,7 |
| FB 31 | 4,9 | 4,9 | 4,6 | 2,8 |
| FB 32 | 4,2 | 4,2 | 4,0 | 2,7 |
| FB 33 | 10,6 | 10,6 | 8,4 | 4,8 |
| FB 44 | 2,4 - 24,3 *) | 2,4 - 24,3 *) | 2,0 - 22,2 *) | 1,0 - 1,2 *) |
| FB 45 | 2,6 - 24,7 *) | 2,6 - 24,7 *) | 2,4 - 22,5 *) | 0,9 - 1,5 *) |
| FB 196 | 2,3 - 10,9 *) | 2,3 - 10,9 *) | 2,1 - 10,7 *) | 0,4 - 0,6 *) |
| FB 197 | 2,1 - 10,7 *) | 2,1 - 10,7 *) | 1,9 - 10,5 *) | 0,7 - 0,9 *) |

*) The execution times depend on the bytes to be transferred.

**) from order number 6ES5 94x-7UBxx on

## S5-135U (CPU 922, 928, 928B):

Program file S5AA20ST.S5D

| Block designation | | Library number P71200- | Block length/ words | Call length/ words | Secondary blocks | Nesting depth | Execution time in ms |
|---|---|---|---|---|---|---|---|
| FB 1 | ADD: 32 | S 9001-A-0 | 100 | 10 | -- | 0 | see next Page |
| FB 2 | SUB: 32 | S9002-A-0 | 100 | 10 | -- | 0 | |
| FB 3 | MUL: 32 | S 9003-A-0 | 302 | 11 | -- | 0 | |
| FB 4 | DIV: 32 | S 9004-A-0 | 242 | 14 | -- | 0 | |
| FB 5 | RAD: 16 | S 9005-A-0 | 126 | 6 | -- | 0 | |
| FB 6 | RAD:GP | S 9006-A-0 | 133 | 8 | -- | 0 | |
| FB 10 | REG: SCHB | S 9010-A-0 | 250 | 14 | -- | 0 | |
| FB 11 | REG: SCHW | S 9011-A-0 | 125 | 10 | -- | 0 | |
| FB 12 | REG:FIFO | S 9012-A-0 | 160 | 11 | -- | 0 | |
| FB 13 | REG: LIFO | S 9013-A-0 | 117 | 11 | -- | 0 | |
| FB 30 | AE:464 | S 9030-A-1 | 384 | 11 | -- | 0 | |
| FB 31 | AE:460 | S 9031-A-1 | 300 | 13 | -- | 0 | |
| FB 32 | AE:463 | S 9032-A-1 | 219 | 11 | -- | 0 | |
| FB 33 | AE:466 | S 9033-B-1 | 252 | 11 | -- | 0 | |
| FB 41 | RLG:AA | S 9041-A-1 | 101 | 12 | -- | 0 | |
| FB 44 | DB-COPY | S 9044-A-1 | 196 | 8 | -- | 1 **) | |
| FB 45 | DB-COPY | S 9045-A-1 | 246 | 2 | -- | 1 **) | |
| FB 196 | PER:ET | S 9196-A-2 | 267 | 7 | -- | 1 **) | |
| FB 197 | PER:ET | S 9197-A-1 | 322 | 2 | -- | 1 **) | |

**) Special functions of the operation system are called which are considered to be "normal" block calls.

Notes:

- Assignment in the flag area: MB 200 - 255.
- Because the program can be interrupted at command boundaries, the OBs must be saved in a data block at the start and called at the end of all interrupting OBs in which the flags MB 200 - MB 255 are used.

## S5-135 U (CPU 922, 928, 928B)

Program file S5AA20ST.S5D

| Block | Excution time (in ms) CPU | | |
|---|---|---|---|
| | CPU 922 | CPU 928A | CPU 928B |
| FB 1 | 1,6 | 0,76 | 0,3 |
| FB 2 | 1,6 | 0,74 | 0,4 |
| FB 3 | 11,6 | 4,32 | 0,5 |
| FB 4 | 1,2 - 37 | 0,9 - 13,3 | 1,2 |
| FB 5 | 0,6 - 9,7 | 0,42 - 6,9 | 0,4 |
| FB 6 | 3,8 - 10,8 | 2,19 - 7,1 | 0,4 |
| FB 10 | 15 bei 32 Bit | 1,9 bei 32 Bit | 0,7 |
| FB 11 | 1,0 - 100<br>bei K = 1-254 | 0,8 - 67<br>bei K = 1-254 | 0,4 |
| FB 12 | 1,2 | 0,6 | 0,1 |
| FB 13 | 0,9 | 0,52 | 0,1 |
| FB 30 | 3,6 | 1,9 | 0,6 |
| FB 31 | 3,5 | 1,6 | 0,6 |
| FB 32 | 3,0 | 1,6 | 0,6 |
| FB 33 | 3,3 | 1,6 | 0,5 |
| FB 41 | 2,9 | 2,4 | 0,2 |
| FB 44 | 2,3 - 2,8 *) | 1,4 - 1,6 *) | 0,3 - 0,6 *) |
| FB 45 | 3,0 - 3,3 *) | 1,7 - 2,0 *) | 0,4 - 0,7 *) |
| FB 196 | 3,2 - 4,7 *) | 1,4 - 2,6 *) | 0,9 - 2,1 *) |
| FB 197 | 3,9 - 5,4 *) | 1,8 - 3,0 *) | 1,0 - 2,2 *) |

*) The execution times depend on the bytes to be transferred.

## S5-155U (CPU 946/947):

Program file S5AA60ST.S5D

| Block designation | | Library number P71200- | Block length/ words | Call length/ words | Secondary blocks | Nesting depth | Execution time in ms |
|---|---|---|---|---|---|---|---|
| FB 3 | ADD: 32 | S 6003-A-1 | 69 | 10 | -- | 0 | 0,13 |
| FB 7 | SUB: 32 | S6007-A-1 | 69 | 10 | -- | 0 | 0,13 |
| FB 11 | MUL: 32 | S 6011-A-1 | 197 | 11 | -- | 0 | 0,7 |
| FB 15 | DIV: 32 | S 6015-A-1 | 203 | 14 | -- | 0 | 0,2 - 2 |
| FB 18 | RAD: 16 | S 6018-A-1 | 128 | 6 | -- | 0 | 0,7 |
| FB 19 | RAD:GP | S 6019-A-1 | 129 | 5 | -- | 0 | 0,3 - 0,4 |
| FB 24 | REG: SCHB | S 6024-A-1 | 252 | 14 | -- | 0 | 0,2 - 12 |
| FB 25 | REG: SCHW | S 6025-A-2 | 117 | 10 | -- | 0 | 0,25 |
| FB 26 | REG:FIFO | S 6026-A-1 | 162 | 11 | -- | 0 | 0,2 |
| FB 27 | REG: LIFO | S 6027-A-1 | 119 | 11 | -- | 0 | 0,2 |
| FB 30 | AE:464 | S 6030-A-1 | 386 | 11 | -- | 0 | 0,4 |
| FB 31 | AE:460 | S 6031-A-1 | 305 | 13 | -- | 0 | 0,4 |
| FB 32 | AE:463 | S 6032-A-1 | 221 | 11 | -- | 0 | 0,5 |
| FB 33 | AE:466 | S 6033-A-1 | 254 | 11 | -- | 0 | 0,7 |
| FB 38 | RETTEN | S 6038-B-1 | 105 | 3 | -- | 1 **) | 0,24 |
| FB 39 | LADEN | S 6039-B-1 | 96 | 3 | -- | 1 **) | 0,24 |
| FB 41 | RLG:AA | S 6041-B-1 | 105 | 12 | -- | 0 | 0,27 |
| FB 44 | DB:COPY | S 6044-A-1 | 233 | 8 | -- | 0 | 0,3 - 0,7 *) |
| FB 45 | DB:COPY | S 6045-A-1 | 303 | 2 | -- | 0 | 0,3 - 0,8 *) |
| FB 129 | SST:UHR | S 6129-B-1 | 53 | 2 | -- | 1 **) | 0,1 - 0,4 |
| FB 196 | PER:ET | S 6196-B-3 | 301 | 7 | -- | 0 | 1,1 - 1,3 *) |
| FB 197 | PER:ET | S 6197-B-1 | 369 | 2 | -- | 0 | 1,3 - 1,5 *) |

*) The execution times depend on the bytes to be transferred.

**) Special functions of the operation system are called which are considered to be "normal" block calls.

Notes:

- Assignment in the flag area: MB 200 - 255.
- Because the program can be interrupted at command boundaries, the OBs must be sa ved in a data block at the start and called at the end of all interrupting OBs in which the flags MB 200 - MB 255 are used.