

## **S7-300 Instruction List**

### **CPU Specifications**

**CPU 312C, 313C, 313C-2 PtP, 313C-2 DP, 314C-2 PtP, 314C-2 DP**

**CPU 312, 314, 315-2 DP**

**This Instruction List has the order number:**

**6ES7398-8AA10-8BN0**

**Edition 09/2002**

**A5E00105517-02**

### **Copyright © Siemens AG 2002 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.

Siemens AG  
Bereich Automation and Drives  
Geschäftsgebiet Industrial Automatiuon Systems  
Postfach 4848, D- 90327 Nuernberg

Siemens Aktiengesellschaft

### **Disclaim 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.

© Siemens AG 2002  
Technical data subject to change.

6ES7398-8AA10-8BN0



## Contents

Validity Range of the Instructions List .....	5
Address Identifiers and Parameter Ranges .....	6
Abbreviations and Mnemonics .....	12
Registers .....	14
Examples of Addressing .....	17
Examples of how to calculate the pointer .....	20
Execution Times with Indirect Addressing .....	21
Calculating the Execution Time Using a CPU 314C–2 DP as an Example .....	23
List of Instructions .....	28
Bit Logic Instructions .....	29
Bit Logic Instructions with Parenthetical Expressions .....	35
ORing of AND Operations .....	37
Logic Instructions with Timers and Counters .....	38
Word Logic Instructions with the Contents of Accumulator 1 .....	42
Evaluating Conditions Using AND, OR and EXCLUSIVE OR .....	44
Edge-Triggered Instructions .....	46

---

Setting/Resetting Bit Addresses .....	48
Instructions Directly Affecting the RLO .....	51
Timer Instructions .....	53
Counter Instructions .....	55
Load Instructions .....	57
Load Instructions for Timers and Counters .....	62
Transfer Instructions .....	63
Load and Transfer Instructions for Address Registers .....	68
Load and Transfer Instructions for the Status Word .....	70
Load Instructions for DB Number and DB Length .....	71
Integer Math (16 Bits) .....	72
Integer Math (32 Bits) .....	73
Floating-Point Math (32 Bits) .....	74
Square Root and Square Instructions (32 Bits) .....	76
Logarithmic Function (32 Bits) .....	77
Trigonometrical Functions (32 Bits) .....	78
Adding Constants .....	79

---

Adding Using Address Registers .....	80
Comparison Instructions with Integers (16 Bits) .....	81
Comparison Instructions with Integers (32 Bits) .....	82
Comparison Instructions with Real Numbers (32 Bits) .....	83
Shift Instructions .....	84
Rotate Instructions .....	86
Accumulator Transfer Instructions, Incrementing and Decrementing .....	87
Program Display and Null Operation Instructions .....	88
Data Type Conversion Instructions .....	89
Forming the Ones and Twos Complements .....	91
Block Call Instructions .....	92
Block End Instructions .....	94
Exchanging Shared Data Block and Instance Data Block .....	95
Jump Instructions .....	96
Instructions for the Master Control Relay (MCR) .....	100

---

Organisation Blocks (OB) .....	101
Function Blocks (FB) .....	106
Functions (FC) .....	106
Data Blocks .....	107
Memory required by the SFBs for the integrated inputs and outputs .....	108
System Functions (SFC) .....	109
System Function Blocks (SFB) .....	114
IEC Functions .....	119
System Status Sublist .....	123
PROFIBUS-DP Sublists .....	129
Alphabetical Index of Instructions .....	131

## Validity Range of the Instructions List

CPU	As of order no.	As of Version		In the following referred to as
		Firmware	Hardware	
CPU 312	6ES7 312-1AD10-0AB0	V2.0.0	01	CPU 312
CPU 312C	6ES7 312-5BD01-0AB0			
CPU 313C	6ES7 313-5BE01-0AB0	V2.0.0	01	CPU 31x
CPU 313C-2 PtP	6ES7 313-6BE01-0AB0			
CPU 313C-2 DP	6ES7 313-6CE01-0AB0			
CPU 314	6ES7 314-1AF10-0AB0			
CPU 314C-2 PtP	6ES7 314-6BF01-0AB0			
CPU 314C-2 DP	6ES7 314-6CF01-0AB0			
CPU 315-2 DP	6ES7 315-2AG10-0AB0			

## Address Identifiers and Parameter Ranges

Addr. ID	Parameter Ranges	Description
Q	0.0 to 127.7	Output (in PIQ)
QB	0 to 127	Output byte (in PIQ)
QW	0 to 126	Output word (in PIQ)
QD	0 to 124	Output double word (in PIQ)

## Address Identifiers and Parameter Ranges

Addr. ID	Parameter Ranges		Description
DBX	0.0 to 16383.7		Data bit in data block
DB	1 to 511 (CPUs except 315-2 DP)	1 to 1023 (only CPU 315-2 DP)	Data block
DBB	0 to 16383		Data byte in DB
DBW	0 to 16382		Data word in DB
DBD	0 to 16380		Data double word in DB
DIX	0.0 to 16383.7		Data bit in instance DB
DI	1 to 511 (CPUs except 315-2 DP)	1 to 1023 (only CPU 315-2 DP)	Instance data block
DIB	0 to 16383		Data byte in instance DB
DIW	0 to 16382		Data word in instance DB
DID	0 to 16380		Data double word in instance DB

Addr. ID	Parameter Ranges			Description
	312	31x		
I	0.0 to 127.7	0.0 to 127.7		Inputs (in PII)
IB	0 to 127	0 to 127		Input byte (in PII)
IW	0 to 126	0 to 126		Input word (in PII)
ID	0 to 124	0 to 124		Input double word (in PII)
Addr. ID	Parameter Ranges			Description
	312	313C, 314, 314C	315-2 DP	
L	0.0 to 255.7	0.0 to 511.7	0.0 to 1023.7	Local data bit
LB	0 to 255	0 to 511	0 to 1023	Local data byte
LW	0 to 254	0 to 510	0 to 1022	Local data word
LD	0 to 252	0 to 508	0 to 1020	Local data double word

## Address Identifiers and Parameter Ranges

Addr. ID	Parameter Ranges			Description
	312	313C, 314, 314C	315-2 DP	
M	0.0 to 127.7	0.0 to 255.7	0.0 to 2047.7	Bit memory bit
MB	0 to 127	0 to 255	0 to 2047	Bit memory byte
MW	0 to 126	0 to 254	0 to 2046	Bit memory word
MD	0 to 124	0 to 252	0 to 2044	Bit memory double word

## Address Identifiers and Parameter Ranges

Addr. ID	Parameter Ranges		Description
	except for CPU 315-2 DP	only CPU 315-2 DP	
PQB	0 to 1023	0 to 2047	Peripheral output byte (direct I/O access)
PQW	0 to 1022	0 to 2046	Peripheral output word (direct I/O access)
PQD	0 to 1020	0 to 2044	Peripheral output double word (direct I/O access)
PIB	0 to 1023	0 to 2047	Peripheral input byte (direct I/O access)
PIW	0 to 1022	0 to 2046	Peripheral input word (direct I/O access)
PID	0 to 1020	0 to 2044	Peripheral input double word (direct I/O access)

Addr. ID	Parameter Ranges		Description
	312	31x	
T	0 – 127	0 – 256	Timer
Z	0 – 127	0 – 256	Counter
Parameter			Instruction addressed via parameter
B#16# W#16# DW#16#	–	–	Byte Word Double word hexadecimal
D#	–	–	IEC date constant
L#	–	–	32-bit integer constant
P#	–	–	Pointer constant
S5T#	–	–	S5 time constant <sup>1</sup> (16 bits), T#1D_5H-3M_1S_2MS
T#	–	–	IEC time constant, T#1D_5H-3M_1S_2MS
TOD#	–	–	time constant (16-/32-Bit), T#1D_5H-3M_1S_2MS
C#	–	–	Counter constant (BCD-codiert)
2#	–	–	Binary constant
B (b1,b2) B (b1,b2; b3,b4)	–	–	Constant, 2 or 4 Byte

<sup>1</sup> for loading of S5 timers

## Abbreviations and Mnemonics

The following abbreviations and mnemonics are used in the Instruction List:

Abbreviations	Description	Example
k8	8-bit constant	32
k16	16-bit constant	631
k32	32-bit constant	1272 5624
i8	8-bit integer	-155
i16	16-bit integer	+6523
i32	32-bit integer	-2 222 222
m	P#x.y (pointer)	P#240.3
n	Binary constant	1001 1100
p	Hexadecimal constant	EA12
q	Real number (32-bit floating-point number)	12.34567E+5
LABEL	Symbolic jump address (max. 4 characters)	DEST
a	Byte address	2
b	Bit address	x.1
c	Operand range	I, Q, M, L, DBX, DIX

<b>Abbreviations</b>	<b>Description</b>	<b>Example</b>
f	Timer/Counter No.	5
g	Operand range	IB, QB, PIB, MB, LB, DBB, DIB
h	Operand range	IW, QW, PIW, MW, LW, DBW, DIW
i	Operand range	ID, QD, PID, MD, LD, DBD, DID
r	Block No.	10

## Registers

### ACCU1 and ACCU2 (32 Bits)

The accumulators are registers for processing bytes, words or double words. The operands are loaded into the accumulators, where they are logically gated. The result of the logic operation (RLO) is in ACCU1.

Accumulator designations:

ACCU	Bits
ACCU <sub>x</sub> (x = 1 to 2)	Bits 0 to 31
ACCU <sub>x</sub> -L	Bits 0 to 15
ACCU <sub>x</sub> -H	Bits 16 to 31
ACCU <sub>x</sub> -LL	Bits 0 to 7
ACCU <sub>x</sub> -LH	Bits 8 to 15
ACCU <sub>x</sub> -HL	Bits 16 to 23
ACCU <sub>x</sub> -HH	Bits 24 to 31

## Address Registers AR1 and AR2 (32 Bits)

The address registers contain the area-internal or area-crossing addresses for instructions using indirect addressing. The address registers are 32 bits long.

The area-internal and/or area-crossing addresses have the following syntax:

- Area-internal address

00000000 00000bbb bbbbbbbb bbbbxxxx

- Area-crossing address

1000yyy 00000bbb bbbbbbbb bbbbxxxx

Legend:   b                   Byte address  
          x                   Bit number  
          y                   Area identifier (see section “Examples of Addressing”)

## Status Word (16 Bits)

The status word bits are evaluated or set by the instructions.

The status word is 16 bits long.

Bit	Assignment	Description
0	$\overline{FC}$	First check bit , Bit cannot be evaluated in the user program with the L STW instruction since it is not updated at program runtime
1	RLO	Result of (previous) logic operation
2	STA	Status, Bit cannot be evaluated in the user program with the L STW instruction since it is not updated at program runtime
3	OR	Or, Bit cannot be evaluated in the user program with the L STW instruction since it is not updated at program runtime
4	OS	Stored overflow
5	OV	Overflow
6	CC 0	Condition code
7	CC 1	Condition code
8	BR	Binary result
9 ... 15	Unassigned	–

## Examples of Addressing

Addressing Examples	Description
Immediate Addressing	
L +27	Load 16-bit integer constant "27" into ACCU1
L L#-1	Load 32-bit integer constant "-1" into ACCU1
L 2#1010101010101010	Load binary constant into ACCU1
L DW#16#A0F0_BCFD	Load hexadecimal constant into ACCU1
L 'END'	Load ASCII character into ACCU1
L T#500 ms	Load time value into ACCU1
L C#100	Load count value into ACCU1
L B#(100,12)	Load 2-byte constant
L B#(100,12,50,8)	Load 4-byte constant
L P#10.0	Load area-internal pointer into ACCU1
L P#E20.6	Load area-crossing pointer into ACCU1
L -2.5	Load real number into ACCU1
L D#1995-01-20	Load date
L TOD#13:20:33.125	Load time of day

Addressing Examples	Description
<b>Direct Addressing</b>	
A I 0.0	ANDing of input bit 0.0
L IB 1	Load input byte 1 into ACCU1
L IW 0	Load input word 0 into ACCU1
L ID 0	Load input double word 0 into ACCU1
<b>Indirect Addressing of Timers/Counters</b>	
SP T [LW 8]	Start timer; the timer number is in local word 8
CU C [LW 10]	Start counter; the counter number is in local data word 10
<b>Area-Internal Memory-Indirect Addressing</b>	
A I [LD 12] Example: L P#22.2 T LD 12 A I [LD 12]	AND operation: The address of the input is in local data double word 12 as pointer
A I [DBD 1]	AND operation: The address of the input is in data double word 1 of the DB as pointer
A Q [DID 12]	AND operation: The address of the output is in data double word 12 of the instance DB as pointer
A Q [MD 12]	AND operation: The address of the output is in memory marker double word 12 of the instance DB as pointer

Addressing Examples	Description																																				
<b>Area-Internal Register-Indirect Addressing</b>																																					
A I [AR1,P#12.2]	AND operation: The address of the input is calculated from the “pointer value in AR1+ P#12.2”																																				
<b>Area-Crossing Register-Indirect Addressing</b>																																					
For area-crossing register-indirect addressing, bits 24 to 26 of the address must also contain an area identifier. The address is in the address register.																																					
<table border="1"> <thead> <tr> <th data-bbox="161 415 344 477">Area identifier</th> <th data-bbox="344 415 592 477">Coding (binary)</th> <th data-bbox="592 415 839 477">Coding (hex.)</th> <th data-bbox="839 415 1444 477">Area</th> </tr> </thead> <tbody> <tr> <td data-bbox="161 477 344 505">P</td> <td data-bbox="344 477 592 505">1000 0000</td> <td data-bbox="592 477 839 505">80</td> <td data-bbox="839 477 1444 505">I/O area</td> </tr> <tr> <td data-bbox="161 505 344 533">I</td> <td data-bbox="344 505 592 533">1000 0001</td> <td data-bbox="592 505 839 533">81</td> <td data-bbox="839 505 1444 533">Input area</td> </tr> <tr> <td data-bbox="161 533 344 561">Q</td> <td data-bbox="344 533 592 561">1000 0010</td> <td data-bbox="592 533 839 561">82</td> <td data-bbox="839 533 1444 561">Output area</td> </tr> <tr> <td data-bbox="161 561 344 589">M</td> <td data-bbox="344 561 592 589">1000 0011</td> <td data-bbox="592 561 839 589">83</td> <td data-bbox="839 561 1444 589">Bit memory area</td> </tr> <tr> <td data-bbox="161 589 344 617">DB</td> <td data-bbox="344 589 592 617">1000 0100</td> <td data-bbox="592 589 839 617">84</td> <td data-bbox="839 589 1444 617">Data area</td> </tr> <tr> <td data-bbox="161 617 344 645">DI</td> <td data-bbox="344 617 592 645">1000 0101</td> <td data-bbox="592 617 839 645">85</td> <td data-bbox="839 617 1444 645">Instance data area</td> </tr> <tr> <td data-bbox="161 645 344 673">L</td> <td data-bbox="344 645 592 673">1000 0110</td> <td data-bbox="592 645 839 673">86</td> <td data-bbox="839 645 1444 673">Local data area</td> </tr> <tr> <td data-bbox="161 673 344 745">VL</td> <td data-bbox="344 673 592 745">1000 0111</td> <td data-bbox="592 673 839 745">87</td> <td data-bbox="839 673 1444 745">Predecessor local data (access to local data of invoking block see page 15)</td> </tr> </tbody> </table>	Area identifier	Coding (binary)	Coding (hex.)	Area	P	1000 0000	80	I/O area	I	1000 0001	81	Input area	Q	1000 0010	82	Output area	M	1000 0011	83	Bit memory area	DB	1000 0100	84	Data area	DI	1000 0101	85	Instance data area	L	1000 0110	86	Local data area	VL	1000 0111	87	Predecessor local data (access to local data of invoking block see page 15)	
Area identifier	Coding (binary)	Coding (hex.)	Area																																		
P	1000 0000	80	I/O area																																		
I	1000 0001	81	Input area																																		
Q	1000 0010	82	Output area																																		
M	1000 0011	83	Bit memory area																																		
DB	1000 0100	84	Data area																																		
DI	1000 0101	85	Instance data area																																		
L	1000 0110	86	Local data area																																		
VL	1000 0111	87	Predecessor local data (access to local data of invoking block see page 15)																																		
L B [AR1,P#8.0]	Load byte into ACCU1: The address is calculated from the “pointer value in AR1+ P#8.0”																																				
A [AR1,P#32.3]	AND operation: The address of the operand is calculated from the “pointer value in AR1+ P#32.3”																																				
<b>Addressing Via Parameters</b>																																					
A Parameter	Addressing via parameters																																				

## Examples of how to calculate the pointer

- **Example for sum of bit addresses  $\leq 7$ :**

LAR1 P#8.2

A I [AR1,P#10.2]

Result: Input 18.4 is addressed (by adding the byte and bit addresses)

- **Example for sum of bit addresses  $> 7$ :**

L MD 0 Random pointer, e.g. P#10.5

LAR1

A I [AR1,P#10.7]

Result: Input 21.4 is addressed (by adding the byte and bit addresses with carry)

## Execution Times with Indirect Addressing

You must calculate the execution times when using indirect addressing. This chapter shows you how.

### Two-Part Statement

A statement with indirectly addressed instructions consists of two parts:

**Part 1:** Load the address of the instruction

**Part 2:** Execute the instruction

In other words, you must calculate the execution time of a statement with indirectly addressed instructions from these two parts.

### Calculating the Execution Time

*The total execution time is calculated as follows:*

$$\begin{array}{l} \textit{Time required for loading the address} \\ + \textit{ execution time of the instruction} \\ \hline \underline{\underline{= \textit{Total execution time of the instruction}}} \end{array}$$

The execution times listed in the chapter entitled “List of Instructions” apply to the execution times of the second part of an instruction, i.e. for the actual execution of an instruction.

You must then add the time required for loading the address of the instruction to this execution time (see Table on following page).

The execution time for loading the address of the instruction from the various areas is shown in the following table.

Address is in ...	Execution Time in $\mu\text{s}$	
	312	31x
Bit memory area M Word Double word	0.7 1.6	0.4 0.9
Data block DB/DX Word Double word	1.5 3.7	0.8 2.0
Local data area L Word Double word	0.9 2.2	0.5 1.2
AR1/AR2 (area-internal)	1.0	0.5
AR1/AR2 (area-crossing)	3.0	1.6
Parameter (word) ... for: • Timers • Counters • Block calls	2.0	1.0
Parameter (double word) ... for Bits, bytes, words and double words	4.0	2.0

The pages that follow contain examples for calculating the instruction run time for the various indirectly addressed instructions.

## Calculating the Execution Time Using a CPU 314C–2 DP as an Example

You will find a few examples here for calculating the execution times for the various methods of indirect addressing. Execution times are calculated for the CPU 314C-2 DP.

### Calculating the Execution Times for Area-Internal Memory-Indirect Addressing

Example: A I [DBD 12]

Step 1: Load the contents of DBD 12 (time required is listed in the table on page 22)

Address is in ...	Execution Time in $\mu\text{s}$
Bit memory area M	
Word	0.4
Double word	0.9
Data block DB/DI	
Word	0.8
Double word	<b>2.0</b>

## Calculating the Execution Time Using a CPU 314C–2 DP as an Example

Step 2: AND the input addressed in this way (you will find the execution time in the tables in the chapter entitled “List of Instructions”)

Typical Execution Time in $\mu\text{s}$	
Direct Addressing	Indirect Addressing
0.1 :	Time for AI $\rightarrow$ 1.6+ :

Total execution time:

$$\begin{array}{r} 2.0 \mu\text{s} \\ + \quad 1.6 \mu\text{s} \\ \hline = \quad 3.6 \mu\text{s} \end{array}$$

## Calculating the Execution Time for Area-Internal Register-Indirect Addressing

Example: A I [AR1, P#34.3]

Step 1: Load the contents of AR1, and increment it by the offset 34.3 (the time required is listed in the table on page 22)

Address is in ...	Execution Time in $\mu\text{s}$
:	:
AR1/AR2 (area-internal)	<b>0.5</b>
:	:

Step 2: AND the input addressed in this way (you will find the execution time in the tables in the chapter entitled “List of Instructions”)

Typical Execution Time in $\mu\text{s}$	
Direct Addressing	Indirect Addressing
0.1	Time for <b>1.6+</b>
:	A I : /

Total execution time:

$$\begin{array}{r}
 0.5 \mu\text{s} \\
 + \quad 1.6 \mu\text{s} \\
 \hline
 = \quad \underline{\underline{2.1 \mu\text{s}}}
 \end{array}$$

## Calculating the Execution Time for Area-Crossing Memory-Indirect Addressing

Example: A [AR1, P#23.1] ... with I 1.0 in AR1

Step 1: Load the contents of AR1, and increment them by the offset 23.1 (the time required is in the table on page 22)

Address is in ...	Execution Time in $\mu\text{s}$
:	:
AR1/AR2 (area-crossing)	1.6
:	:

Step 2: AND the input addressed in this way (you will find the execution time in the tables in the chapter entitled “List of Instructions”)

Typical Execution Time in $\mu\text{s}$	
Direct Addressing	Indirect Addressing
0.1	Time for <b>1.6+</b>
:	AI : /

Total execution time:

$$\begin{array}{r}
 1.6 \mu\text{s} \\
 + \quad 1.6 \mu\text{s} \\
 \hline
 = \quad 3.2 \mu\text{s}
 \end{array}$$

## Execution Time for Addressing via Parameters

Example: A Parameter ... with I 0.5 in the block parameter list

Step 1: Load input I 0.5 addressed via the parameter (the time required is in the table on page 22).

Address is in ...	Execution Time in $\mu\text{s}$
:	:
:	:
Parameter (double word)	<b>2.0</b>

Step 2: AND the input addressed in this way (you will find the execution time in the tables in the chapter entitled "List of Instructions")

Typical Execution Time in $\mu\text{s}$	
Direct Addressing	Indirect Addressing
0.1	Time for <b>1.6+</b>
:	A I / :

Total execution time:

$$\begin{array}{r}
 2.0 \mu\text{s} \\
 + \quad 1.6 \mu\text{s} \\
 \hline
 = \quad 3.6 \mu\text{s}
 \end{array}$$

## List of Instructions

This chapter contains the complete list of S7-300 instructions. The descriptions have been kept as concise as possible. You will find a detailed functional description in the various STEP 7 reference manuals.

**Please note** that, in the case of indirect addressing (examples see page 18), you must add the time required for loading the address of the particular instruction to the execution times listed (see page 22).

## Bit Logic Instructions

Examining the signal state of the addressed instruction and gating the result with the RLO according to the appropriate logic function.

Instruction	Address Identifier		Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s						
					Direct Addressing		Indirect Addressing <sup>1</sup>				
					312	31x	312	31x			
A	I/Q	a.b	Input/output	1/2	0.2	0.1	3.0+	1.6+			
	M	a.b	Bit memory	1/2	0.4	0.2	3.2+	1.7+			
	L	a.b	Local data bit	2	0.7	0.3	3.7+	2.0+			
	DBX	a.b	Data bit	2	2.9	1.4	4.5+	2.4+			
	DIX	a.b	Instance data bit	2	2.9	1.4	4.5+	2.4+			
	c[AR1,m]		Register-ind., area-internal (AR1)	2	–	–	+	+			
	c[AR2,m]		Register-ind., area-internal (AR2)	2	–	–	+	+			
	[AR1,m]		Area-crossing via (AR1)	2	–	–	+	+			
	[AR2,m]		Area-crossing via (AR2)	2	–	–	+	+			
	Parameter		Via parameter	2	–	–	+	+			
Status word for: <b>A</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:			–	–	–	–	–	Yes	–	Yes	Yes
Instruction affects:			–	–	–	–	–	Yes	Yes	Yes	1

<sup>1</sup> Plus time required for loading the address of the instruction (see page 22)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

Instruction	Address Identifier		Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s						
					Direct Addressing		Indirect Addressing <sup>1</sup>				
					312	31x	312	31x			
AN	I/Q	a.b	Input/output	1/2	0.3	0.2	3.2+	1.7+			
	M	a.b	Bit memory	1/2	0.4	0.2	3.4+	1.8+			
	L	a.b	Local data bit	2	0.8	0.4	3.9+	2.1+			
	DBX	a.b	Data bit	2	3.0	1.5	4.7+	2.5+			
	DIX	a.b	Instance data bit	2	3.0	1.5	4.7+	2.5+			
	c[AR1,m]		Register-ind., area-internal (AR1)	2	–	–	+	+			
	c[AR2,m]		Register-ind., area-internal (AR2)	2	–	–	+	+			
	[AR1,m]		Area-crossing via (AR1)	2	–	–	+	+			
	[AR2,m]		Area-crossing via (AR2)	2	–	–	+	+			
	Parameter		Via parameter	2	–	–	+	+			
Status word for: <b>AN</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:			–	–	–	–	–	Yes	–	Yes	Yes
Instruction affects:			–	–	–	–	–	Yes	Yes	Yes	1

<sup>1</sup> Plus time required for loading the address of the instruction (see page 22)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

In- struc- tion	Address Identifier		Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s						
					Direct Addressing		Indirect Addressing <sup>1</sup>				
					312	31x	312	31x			
O	I/Q	a.b	Input/output	1/2	0.2	0.1	3.0+	1.6+			
	M	a.b	Bit memory	1/2	0.3	0.2	3.2+	1.7+			
	L	a.b	Local data bit	2	0.7	0.3	3.7+	2.0+			
	DBX	a.b	Data bit	2	2.9	1.4	4.6+	2.4+			
	DIX	a.b	Instance data bit	2	2.9	1.4	4.6+	2.4+			
	c[AR1,m]		Register-ind., area-internal (AR1)	2	–	–	+	+			
	c[AR2,m]		Register-ind., area-internal (AR2)	2	–	–	+	+			
	[AR1,m]		Area-crossing via (AR1)	2	–	–	+	+			
	[AR2,m]		Area-crossing via (AR2)	2	–	–	+	+			
	Parameter		Via parameter	2	–	–	+	+			
Status word for: <b>O</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO2	FC
Instruction depends on:			–	–	–	–	–	–	–	Yes	Yes
Instruction affects:			–	–	–	–	–	0	Yes	Yes	1

<sup>1</sup> Plus time required for loading the address of the instruction (see page 22)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

Instruction	Address Identifier		Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s						
					Direct Addressing		Indirect Addressing <sup>1</sup>				
					312	31x	312	31x			
ON	I/Q	a.b	Input/output	1/2	0.3	0.2	3.2+	1.7+			
	M	a.b	Bit memory	1/2	0.4	0.2	3.5+	1.8+			
	L	a.b	Local data bit	2	0.8	0.4	3.9+	2.1+			
	DBX	a.b	Data bit	2	3.0	1.5	4.7+	2.5+			
	DIX	a.b	Instance data bit	2	3.0	1.5	4.7+	2.5+			
	c[AR1,m]		Register-ind., area-internal (AR1)	2	–	–	+	+			
	c[AR2,m]		Register-ind., area-internal (AR2)	2	–	–	+	+			
	[AR1,m]		Area-crossing via (AR1)	2	–	–	+	+			
	[AR2,m]		Area-crossing via (AR2)	2	–	–	+	+			
	Parameter		Via parameter	2	–	–	+	+			
Status word for: <b>ON</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:			–	–	–	–	–	–	–	Yes	Yes
Instruction affects:			–	–	–	–	–	0	Yes	Yes	1

<sup>1</sup> Plus time required for loading the address of the instruction (see page 22)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

Instruction	Address Identifier		Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s						
					Direct Addressing		Indirect Addressing <sup>1</sup>				
					312	31x	312	31x			
X	I/Q	a.b	EXCLUSIVE OR								
				Input/output	1/2	0.2	0.1	2.9+	1.6+		
	M	a.b	Bit memory	1/2	0.3	0.2	3.2+	1.7+			
				Local data bit	2	0.7	0.3	3.7+	2.0+		
	DBX	a.b	Data bit	2	2.9	1.4	4.5+	2.4+			
				Instance data bit	2	2.9	1.4	4.5+	2.4+		
	c[AR1,m] c[AR2,m] [AR1,m] [AR2,m] Parameter		Register-ind., area-internal (AR1)	2	–	–	+	+			
			Register-ind., area-internal (AR2)	2	–	–	+	+			
			Area-crossing via (AR1)	2	–	–	+	+			
			Area-crossing via (AR2)	2	–	–	+	+			
		Via parameter	2	–	–	+	+				
Status word for: <b>X</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:			–	–	–	–	–	–	–	Yes	Yes
Instruction affects:			–	–	–	–	–	0	Yes	Yes	1

<sup>1</sup> Plus time required for loading the address of the instruction (see page 22)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

Instruction	Address Identifier		Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s						
					Direct Addressing		Indirect Addressing <sup>1</sup>				
					312	31x	312	31x			
XN	I/Q	a.b	EXCLUSIVE OR NOT								
		a.b	Input/output	1/2	0.3	0.2	3.2+	1.7+			
	M	a.b	Bit memory	1/2	0.4	0.2	3.5+	1.8+			
		a.b	Local data bit	2	0.8	0.4	3.9+	2.1+			
	DBX	a.b	Data bit	2	3.0	1.5	4.7+	2.5+			
		a.b	Instance data bit	2	3.0	1.5	4.7+	2.5+			
	c[AR1,m]		Register-ind., area-internal (AR1)	2	–	–	+	+			
			Register-ind., area-internal (AR2)	2	–	–	+	+			
	[AR1,m]		Area-crossing via (AR1)	2	–	–	+	+			
	[AR2,m]		Area-crossing via (AR2)	2	–	–	+	+			
Parameter		Via parameter	2	–	–	+	+				
Status word for: <b>XN</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:			–	–	–	–	–	–	–	Yes	Yes
Instruction affects:			–	–	–	–	–	0	Yes	Yes	1

<sup>1</sup> Plus time required for loading the address of the instruction (see page 22)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

## Bit Logic Instructions with Parenthetical Expressions

Saving the BR, RLO and OR bits and a function identifier (A, AN, ...) to the nesting stack. Seven nesting levels are possible per block.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu\text{s}$							
				312	31x						
A(		AND left parenthesis	1	3.2	1.6						
AN(		AND NOT left parenthesis	1	3.3	1.6						
O(		OR left parenthesis	1	3.0	1.5						
ON(		OR NOT left parenthesis	1	3.0	1.5						
X(		EXCLUSIVE OR left parenthesis	1	3.0	1.5						
XN(		EXCLUSIVE OR NOT left parenthesis	1	3.0	1.5						
Status word for:	<b>A(, AN(, O(, ON(, X(, XN(</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
Instruction depends on:			Yes	–	–	–	–	Yes	–	Yes	Yes
Instruction affects:			–	–	–	–	–	0	1	–	0

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s							
				312	31x						
)		Right parenthesis, popping an entry off the nesting stack, gating the RLO with the current RLO in the processor	1	1.0	1.0						
Status word for:	)		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:			-	-	-	-	-	-	-	Yes	-
Instruction affects:			Yes	-	-	-	-	Yes	1	Yes	1

## ORing of AND Operations

The ORing of AND operations is implemented according to the rule: AND before OR.

Instruction	Address Identifier	Description	in Length Words		Typical Execution Time in $\mu$ s								
			312	31x	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
O		ORing of AND operations according to the rule: AND before OR	1										
Status word for: <b>O</b>													
Instruction depends on:			–	–	–	–	–	–	Yes	–	Yes	Yes	
Instruction affects:			–	–	–	–	–	–	Yes	1	–	Yes	

## Logic Instructions with Timers and Counters

Examining the signal state of the addressed timer/counter and gating the result with the RLO according to the appropriate logic function.

Instruction	Address Identifier	Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu\text{s}$							
				Direct Addressing				Indirect Addressing <sup>1</sup>			
				312	31x	312	31x				
A	T	AND Timer	1/2+	0.6	0.3	2.1+	1.1+				
	C	Counter	1/2+	0.3	0.2	2.0+	1.1+				
	Timer para. Counter p.	Timer/counter (addressed via parameter)	2	–	–	+	+				
Status word for: <b>A</b>			CC 1	BR	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:			–	–	–	–	–	Yes	–	Yes	Yes
Instruction affects:			–	–	–	–	–	Yes	Yes	Yes	1

<sup>1</sup> Plus time required for loading the address of the instruction (see page 22)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

In- struction	Address Identifier	Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s						
				Direct Addressing		Indirect Addressing <sup>1</sup>				
				312	31x	312	31x			
AN	T	AND NOT Timer	1/2	0.8	0.4	2.3+	1.2+			
	C	Counter	1/2	0.5	0.3	2.2+	1.2+			
	Timer para. Counter p.	Timer/counter (addressed via parameter)	2	–	–	+	+			
Status word for: <b>AN</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:		–	–	–	–	–	Yes	–	Yes	Yes
Instruction affects:		–	–	–	–	–	Yes	Yes	Yes	1

<sup>1</sup> Plus time required for loading the address of the instruction (see page 22)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

In- struction	Address Identifier	Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s						
				Direct Addressing		Indirect Addressing <sup>1</sup>				
				312	31x	312	31x			
O	T	OR timer	1/2	0.6	0.3	2.1+	1.1+			
	C	OR counter	1/2	0.3	0.2	2.0+	1.0+			
	Timerpara. Counter p.	OR timer/counter (addressed via parameter)	2	–	–	+	+			
ON	T	OR NOT timer	1/2	0.8	0.4	2.3+	1.2+			
	C	OR NOT counter	1/2	0.5	0.3	2.2+	1.1+			
	Timerpara. Counter p.	OR NOT timer/counter (addressed via parameter)	2	–	–	+	+			
X	T	EXCLUSIVE OR timer	1/2	0.6	0.3	2.1+	1.1+			
	C	EXCLUSIVE OR counter	1/2	0.4	0.2	2.0+	1.1+			
	Timerpara. Counter p.	EXCLUSIVE OR timer/counter (addressed via parameter)	2	–	–	+	+			
Status word for: <b>O, ON, X</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:		–	–	–	–	–	–	–	Yes	Yes
Instruction affects:		–	–	–	–	–	0	Yes	Yes	1

<sup>1</sup> Plus time required for loading the address of the instruction (see page 22)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

In- struc- tion	Address Identifier	Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s							
				Direct Addressing		Indirect Addressing <sup>1</sup>					
				312	31x	312	31x				
XN	T	EXCLUSIVE OR NOT timer	1/2	0.8	0.4	2.3+	1.2+				
	C	EXCLUSIVE OR NOT counter	1/2	0.5	0.3	2.2+	1.2+				
	Timerpara. Counter p.	EXCLUSIVE OR NOT timer/counter (ad- dressed via parameter)	2	–	–	+	+				
Status word for: <b>XN</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:			–	–	–	–	–	–	–	Yes	Yes
Instruction affects:			–	–	–	–	–	0	Yes	Yes	1

<sup>1</sup> Plus time required for loading the address of the instruction (see page 22)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

## Word Logic Instructions with the Contents of Accumulator 1

Gating the contents of ACCU1 and/or ACCU1-L with a word or double word according to the appropriate function. The word or double word is either a constant in the instruction or in ACCU2. The result is in ACCU1 and/or ACCU1-L.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s							
				312	31x						
AW		AND ACCU2-L	1	0.6	0.3						
AW	k16	AND 16-bit constant	2	0.6	0.3						
OW		OR ACCU2-L	1	0.6	0.3						
OW	k16	OR 16-bit constant	2	0.6	0.3						
XOW		EXCLUSIVE OR ACCU2-L	1	0.6	0.3						
XOW	k16	EXCLUSIVE OR 16-bit constant	2	0.6	0.3						
AD		AND ACCU2	1	1.9	1.0						
AD	k32	AND 32-bit constant	3	2.1	1.0						
Status word for:		<b>AW, OW, XOW, AD</b>	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:			–	–	–	–	–	–	–	–	–
Instruction affects:			–	Yes	0	0	–	–	–	–	–

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s						
				312	31x					
OD		OR ACCU2	1	1.9	1.0					
OD	k32	OR 32-bit constant	3	2.1	1.0					
XOD		EXCLUSIVE OR ACCU2	1	1.9	1.0					
XOD	k32	EXCLUSIVE OR 32-bit constant	3	2.1	1.0					
Status word for: <b>OD, XOD</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:		–	–	–	–	–	–	–	–	–
Instruction affects:		–	Yes	0	0	–	–	–	–	–

## Evaluating Conditions Using AND, OR and EXCLUSIVE OR

Examining the specified conditions for their signal status, and gating the result with the RLO according to the appropriate function.

Instruc- tion	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s						
				312	31x					
A/ O/ X	==0	AND, OR, EXCLUSIVE OR Result=0 (CC 1=0)and (CC 0=0)	1	0.3	0.2					
	>0	Result>0 (CC 1=1) and (CC 0=0)	1	0.5	0.3					
	<0	Result<0 (CC 1=0)and (CC 0=1)	1	0.5	0.3					
	<>0	Result $\neq$ 0 ((CC1=0)and(CC 0=1)or (CC1=1)and(CC 0=0))	1	0.3	0.2					
	<=0	R<=0((CC 1=0) and (CC 0=1) or (CC1=0) and (CC 0=0))	1	0.3	0.2					
	>=0	R>=0((CC 1=1) and (CC 0=0) or (CC1=0) and (CC 0=0))	1	0.3	0.2					
	UO	unordered math instruction (CC 1=1) and (CC 0=1)	1	0.3	0.2					
	OS	OS=1	1	0.2	0.1					
	BR	BR=1	1	0.2	0.1					
OV	OV=1	1	0.2	0.1						
Status word for: <b>A/ O/ X</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:		Yes	Yes	Yes	Yes	Yes	Yes	–	Yes	Yes
Instruction affects:		–	–	–	–	–	Yes	Yes	Yes	1

## Evaluating Conditions Using AND, OR and EXCLUSIVE OR

Instruc- tion	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s						
				312	31x					
AN/ ON/ XN	==0	AND NOT, OR NOT, EXCLUSIVE OR NOT Result=0 (CC 1=0) and (CC 0=0)	1	0.3	0.2					
	>0	Result>0 (CC 1=1) and (CC 0=0)	1	0.5	0.3					
	<0	Result<0 (CC 1=0) and (CC 0=1)	1	0.5	0.3					
	<>0	Result $\neq$ 0 ((CC 1=0) and (CC 0=1) or (CC 1=1) and (CC 0=0))	1	0.5	0.3					
	<=0	Result<=0 ((CC 1=0) and (CC 0=1) or (CC 1=0) and (CC 0=0))	1	0.2	0.1					
	>=0	Result>=0 ((CC 1=1) and (CC 0=0) or (CC 1=0) and (CC 0=0))	1	0.2	0.1					
	UO	unordered math instruction (CC 1=1) and (CC 0=1)	1	0.5	0.3					
	OS	OS=1	1	0.3	0.2					
	BR	BR=1	1	0.3	0.2					
	OV	OV=1	1	0.3	0.2					
Status word for: <b>AN/ ON/ XN</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:		Yes	Yes	Yes	Yes	Yes	Yes	–	Yes	Yes
Instruction affects:		–	–	–	–	–	Yes	Yes	Yes	1

## Edge-Triggered Instructions

Detection of an edge change. The current signal state of the RLO is compared with the signal state of the instruction or "edge bit memory". FP detects a change in the RLO from "0" to "1"; FN detects a change in the RLO from "1" to "0".

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu\text{s}$							
				Direct Addressing		Indirect Addressing <sup>1</sup>		312	31x		
				312	31x	312	31x				
FP	I/Q a.b	Detecting the positive edge in the RLO. The bit addressed in the instruction is the auxiliary edge bit memory.	2	0.5	0.3	3.3+	1.8+				
	M a.b		2	1.0	0.5	3.6+	1.9+				
	L a.b		2	1.2	0.6	4.0+	2.1+				
	DBX a.b		2	3.6	1.8	5.2+	2.7+				
	DIX a.b		2	3.6	1.8	5.2+	2.7+				
	c[AR1,m]		2	–	–	+	+				
	c[AR2,m]		2	–	–	+	+				
	[AR1,m]		2	–	–	+	+				
	[AR2,m]		2	–	–	+	+				
	Parameter		2	–	–	+	+				
Status word for: <b>FP</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
Instruction depends on:			–	–	–	–	–	–	–	Yes	–
Instruction affects:			–	–	–	–	–	0	Yes	Yes	1

<sup>1</sup> Plus time required for loading the address of the instruction (see page 22)

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s							
				Direct Addressing		Indirect Addressing <sup>1</sup>					
				312	31x	312	31x				
FN	I/Q a.b	Detecting the negative edge in the RLO. The bit addressed in the instruction is the auxiliary edge bit memory.	2	0.7	0.3	3.5+	1.9+				
	M a.b		2	1.1	0.5	3.8+	2.0+				
	L a.b		2	1.3	0.7	4.2+	2.2+				
	DBX a.b		2	3.7	1.9	5.2+	2.8+				
	DIX a.b		2	3.7	1.9	5.2+	2.8+				
	c[AR1,m]		2	–	–	+	+				
	c[AR2,m]		2	–	–	+	+				
	[AR1,m]		2	–	–	+	+				
	[AR2,m]		2	–	–	+	+				
	Parameter		2	–	–	+	+				
Status word for: <b>FN</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:			–	–	–	–	–	–	–	Yes	–
Instruction affects:			–	–	–	–	–	0	Yes	Yes	1

<sup>1</sup> Plus time required for loading the address of the instruction (see page 22)

## Setting/Resetting Bit Addresses

Assigning the value “1” or “0” or the RLO o the addressed instruction. The instructions can be dependent on the MCR.

Instruction	Address Identifier	Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s							
				Direct Addressing		Indirect Addressing <sup>1</sup>					
				312	31x	312	31x				
S	I/Q a.b	Set input/output to “1” (MCR-dependent)	1/2	0.2 0.3	0.1 0.2	3.1+ 3.3+	1.7+ 1.8+				
	M a.b	Set bit memory to “1” (MCR-dependent)	1/2	0.4 1.8	0.2 0.9	3.4+ 3.7+	1.8+ 2.0+				
	L a.b	Set local data bit to “1” (MCR-dependent)	2	0.9 2.0	0.4 1.0	3.8+ 3.9+	2.0+ 2.1+				
	DBX a.b	Set data bit to “1” (MCR-dependent)	2	3.4 3.5	1.7 1.7	4.8+ 5.0+	2.6+ 2.7+				
	DIX a.b	Set instance data bit to “1” (MCR-dependent)	2	3.4 3.5	1.7 1.7	4.8+ 5.0+	2.6+ 2.7+				
	c[AR1,m]	Register-ind., area-internal (AR1)	2	–	–	+	+				
	c[AR2,m]	Register-ind., area-internal (AR2)	2	–	–	+	+				
	[AR1,m]	Area-crossing via (AR1)	2	–	–	+	+				
	[AR2,m]	Area-crossing via (AR2)	2	–	–	+	+				
	Parameter	Via parameter	2	–	–	+	+				
Status word for: <b>S</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:			–	–	–	–	–	–	–	Yes	–
Instruction affects:			–	–	–	–	–	0	Yes	–	0

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

Instruction	Address Identifier		Description	Length in Word <sup>2</sup>	Typical Execution Time in $\mu$ s						
					Direct Addressing		Indirect Addressing <sup>1</sup>				
					312	31x	312	31x			
R	I/Q	a.b	Reset input/output to "0" (MCR-dependent)	1/2	0.3	0.1	3.2+	1.7+			
	M	a.b	Set bit memory to "0" (MCR-dependent)	1/2	0.3	0.2	3.5+	1.8+			
	L	a.b	Set local data bit to "0" (MCR-dependent)	2	0.5	0.3	3.5+	1.8+			
	DBX	a.b	Set data bit to "0" (MCR-dependent)	2	1.8	0.9	3.6+	1.9+			
	DIX	a.b	Set instance data bit to "0" (MCR-dependent)	2	0.9	0.4	3.9+	2.1+			
					2.0	1.0	4.0+	2.1+			
					3.4	1.7	5.0+	2.6+			
					3.6	1.8	5.1+	2.7+			
					3.4	1.7	5.0+	2.6+			
					3.6	1.8	5.1+	2.7+			
	c[AR1,m]		Register-ind., area-internal (AR1)	2	–	–	+	+			
	c[AR2,m]		Register-ind., area-internal (AR2)	2	–	–	+	+			
	[AR1,m]		Area-crossing via (AR1)	2	–	–	+	+			
	[AR2,m]		Area-crossing via (AR2)	2	–	–	+	+			
	Parameter		Via parameter	2	–	–	+	+			
Status word for: <b>R</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:			–	–	–	–	–	–	–	Yes	–
Instruction affects:			–	–	–	–	–	0	Yes	–	0

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

Instruction	Address Identifier		Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s							
					Direct Addressing		Indirect Addressing <sup>1</sup>					
					312	31x	312	31x				
=	I/Q	a.b	Assign RLO to input/output (MCR-dependent)	1/2	0.2 0.3	0.1 0.2	3.2+ 3.4+	1.7+ 1.8+				
	M	a.b	Assign RLO to bit memory (MCR-dependent)	1/2	0.6 1.8	0.3 0.9	3.5+ 3.7+	1.8+ 2.0+				
	L	a.b	Assign RLO to local data bit (MCR-dependent)	2	0.8 2.1	0.4 1.0	3.9+ 4.1+	2.0+ 2.2+				
	DBX	a.b	Assign RLO to data bit (MCR-dependent)	2	3.4 3.6	1.7 1.8	5.0+ 5.1+	2.6+ 2.7+				
	DIX	a.b	Assign RLO to instance data bit (MCR-dependent)	2	3.4 3.6	1.7 1.8	5.0+ 5.1+	2.6+ 2.7+				
	c[AR1,m]		Register-ind., area-internal(AR1)	2	–	–	+	+				
	c[AR2,m]		Register-ind., area-internal(AR2)	2	–	–	+	+				
	[AR1,m]		Area-crossing via (AR1)	2	–	–	+	+				
	[AR2,m]		Area-crossing via (AR2)	2	–	–	+	+				
	Parameter		Via parameter	2	–	–	+	+				
<b>Status word for: =</b>				BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:				–	–	–	–	–	–	–	Yes	–
Instruction affects:				–	–	–	–	–	0	Yes	–	0

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

## Instructions Directly Affecting the RLO

The following instructions have a direct effect on the RLO.

Instru- tion	Address Identifier	Description	Length in Words				Typical Execution Time in $\mu$ s				
							312		31x		
CLR		Set RLO to "0"	2				0.2		0.1		
Status word for: <b>CLR</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:			-	-	-	-	-	-	-	-	-
Instruction affects:			-	-	-	-	0	0	0	0	0
SET	SET	SET	2				0.2		0.1		
Status word for: <b>SET</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:			-	-	-	-	-	-	-	-	-
Instruction affects:			-	-	-	-	0	1	1	0	0
NOT		NOT	2				0.2		0.1		
Status word for: <b>NOT</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:			-	-	-	-	-	Yes	-	Yes	-
Instruction affects:			-	-	-	-	-	-	1	Yes	-

Instruc- tion	Address Identifier	Description	Length in Words				Typical Execution Time in $\mu$ s			
							312		31x	
SAVE			1				0.2		0.1	
Status word for: <b>SAVE</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:		-	-	-	-	-	-	-	Yes	-
Instruction affects:		Yes	-	-	-	-	-	-	-	-

## Timer Instructions

Starting or resetting a timer (addressed directly or via a parameter). The time value must be in ACCU1-L.

Instruc- tion	Address Identifier	Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s							
				Direct Addressing		Indirect Addressing <sup>1</sup>					
				312	31x	312	31x				
SP	T f	Start timer as pulse on edge change from "0" to "1"	4/6	4.4	2.3	5.4+	2.9+				
	Timer para.		2	–	–	+	+				
SE	T f	Start timer as exded pulse on edge change from "0" to "1"	4/6	2.2	1.1	2.2+	1.2+				
	Timer para.		2	–	–	+	+				
SD	T f	Start timer as ON delay on edge change from "0" to "1"	4/6	4.6	2.4	5.5+	3.0+				
	Timer para.		2	–	–	+	+				
SS	T f	Start timer as retive ON delay on edge change from "0" to "1"	4/6	4.7	2.4	5.7+	3.0+				
	Timer para.		2	–	–	+	+				
Status word for: <b>SP, SE, SD, SS, SF</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:			–	–	–	–	–	–	–	Yes	–
Instruction affects:			–	–	–	–	–	0	–	–	0

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

Instruction	Address Identifier	Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s							
				Direct Addressing		Indirect Addressing <sup>1</sup>					
				312	31x	312	31x				
SA	T f	Start timer as off-delay timer when the edge changes from "1" to "0".	4/6	4.9	2.5	5.9+	3.2+				
	Timer para.		2	–	–	+	+				
FR	T f	Enable timer for restarting on edge change from "0" to "1" (reset edge bit memory for starting timer)	4/6	2.3	1.2	2.8+	1.5+				
	Timer para.		2	–	–	+	+				
R	T f	Reset timer	4/6	2.3	1.1	2.8+	1.5+				
	Timer para.		2	–	–	+	+				
Status word for: <b>FR, R</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:			–	–	–	–	–	–	–	Yes	–
Instruction affects:			–	–	–	–	–	0	–	–	0

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> With direct instruction addressing

## Counter Instructions

The count value is in ACCU1-L or in the address transferred as parameter.

Instruction	Address Identifier	Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s							
				Direct Addressing		Indirect Addressing <sup>1</sup>					
				312	31x	312	31x				
S	C f	Presetting of counter on edge change from "0" to "1"	4/6	3.3	1.7	4.5+	2.4+				
	Counter p.		2	–	–	+	+				
R	C f	Reset counter to "0"	4/6	1.3	0.6	2.1+	1.1+				
	Counter p.		2	–	–	+	+				
CU	C f	Increment counter by 1 on edge change from "0" to "1"	4/6	1.9	1.0	2.9+	1.6+				
	Counter p.		2	–	–	+	+				
CD	C f	Decrement counter by 1 on edge change from "0" to "1"	4/6	1.9	0.9	2.9+	1.5+				
	Counter p.		2	–	–	+	+				
Status word for: <b>S, R, CU, CD</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:			–	–	–	–	–	–	–	Yes	–
Instruction affects:			–	–	–	–	–	0	–	–	0

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

Instruction	Address Identifier	Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s						
				Direct Addressing		Indirect Addressing <sup>1</sup>				
				312	31x	312	31x			
FR	C f	Enable counter on edge change from "0" to "1" (reset edge bit memory for up and down counting)	2	1.6	0.8	2.6+	1.4			
	Counter p.		2	–	–	+	+			
Status word for: <b>FR</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:		–	–	–	–	–	–	–	Yes	–
Instruction affects:		–	–	–	–	–	0	–	–	0

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

## Load Instructions

Loading address identifiers into ACCU1. The contents of ACCU1 and ACCU2 are saved first. The status word is not affected.

Instruction	Address Identifier		Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s			
					Direct Addressing		Indirect Addressing <sup>1</sup>	
					312	31x	312	31x
L	IB	a	Load ... Input byte	1/2	0.4	0.2	2.7+	1.4+
	QB	a	Output byte	1/2	0.4	0.2	2.7+	1.4+
	PIB	a	Peripheral input byte	1/2	70.2	35.1	108.4+	57.8+
	PIB	a	Digital Onboard I/O <sup>3</sup>	1/2	51.5	48.3	65.2+	55.6+
	PIB	a	Analog Onboard I/O <sup>4</sup>	1/2		162.1		169.4+
	MB	a	Bit memory byte	1/2	0.5	0.2	2.6+	1.4+
	LB	a	Local data byte	2	0.9	0.5	3.3+	1.7+
	DBB	a	Data byte	2	3.0	1.5	4.7+	2.5+
	DIB	a	Instance data byte ... into ACCU1	2	3.0	1.5	4.7+	2.5+
	g[AR1,m]		Register-ind., area-internal (AR1)	2	–	–	+	+
	g[AR2,m]		Register-ind., area-internal (AR2)	2	–	–	+	+
	B[AR1,m]		Area-crossing (AR1)	2	–	–	+	+
	B[AR2,m]		Area-crossing (AR2)	2	–	–	+	+
	Parameter		Via parameter	2	–	–	+	+

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

<sup>3</sup> Access to digital onboard I/O

<sup>4</sup> Access to analog onboard I/O

Instruction	Address Identifier	Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s				
				Direct Addressing		Indirect Addressing <sup>1</sup>		
				312	31x	312	31x	
L	IW	a	Input word	1/2	0.6	0.3	2.9+	1.6+
	QW	a	Output word	1/2	0.6	0.3	2.9+	1.6+
	PIW	a	Peripheral input word	2	76.7	38.4	131.1+	69.9+
	PIW	a	Digital Onboard I/O <sup>3</sup>	2	61.4	57.6	77.6+	66.3
	PIW	a	Analog Onboard I/O <sup>4</sup>	2	–	170.5	–	179.2
	MW	a	Bit memory word	1/2	0.8	0.4	3.2+	1.7+
	LW	a	Local data word	2	1.1	0.6	3.8+	2.0+
	DBW	a	Data word	1/2	3.5	1.8	5.6+	3.0+
	DIW	a	Instance data word ... into ACCU1	1/2	3.5	1.8	5.6+	3.0+
	h[AR1,m]		Register-ind., area-internal (AR1)	2	–	–	+	+
	h[AR2,m]		Register-ind., area-internal (AR2)	2	–	–	+	+
	W[AR1,m]		Area-crossing via (AR1)	2	–	–	+	+
	W[AR2,m]		Area-crossing via (AR2)	2	–	–	+	+
	Parameter		Via parameter	2	–	–	+	+

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> With direct instruction addressing

<sup>3</sup> Access to digital onboard I/O

<sup>4</sup> Access to analog onboard I/O

Instruction	Address Identifier		Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s			
					Direct Addressing		Indirect Addressing <sup>1</sup>	
					312	31x	312x	31x
L	ID	a	Load ...	1/2	0.8	0.4	3.1+	1.6+
	QD	a	Input double word	1/2	0.8	0.4	3.1+	1.6+
	PID	a	Output double word	2	95.9	47.5	150.6+	80.3+
	PID	a	Peripheral input double word	2	–	303.0	–	323.0+
	PID	a	Analog Onboard I/O <sup>3</sup>	2	–	–	–	–
	MD	a	Bit memory double word	1/2	1.0	0.5	3.8+	2.0+
	LD	a	Local data double word	2	1.5	0.7	4.4+	2.3+
	DBD	a	Data double word	2	4.7	2.3	6.9+	3.7+
	DID	a	Instance data double word	2	4.7	2.3	6.9+	3.7+
			... into ACCU1					
	i[AR1,m]		Register-ind., area-internal (AR1)	2	–	–	+	+
	i[AR2,m]		Register-ind., area-internal (AR2)	2	–	–	+	+
	D[AR1,m]		Area-crossing via (AR1)	2	–	–	+	+
	D[AR2,m]		Area-crossing via (AR2)	2	–	–	+	+
Parameter		Via parameter	2	–	–	+	+	

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

<sup>3</sup> Access to analog onboard I/O

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu\text{s}$			
				Direct Addressing		Indirect Addressing <sup>1</sup>	
				312	31x	312	31x
L	k8	Load ... 8-bit constant into ACCU1-LL	1	0.4	0.2	–	–
	k16	16-bit constant into ACCU1-L	2	0.4	0.2	–	–
	k32	32-bit constant into ACCU1	3	0.5	0.3	–	–
	Parameter	Load constant into ACCU1 (addressed via parameter)	2	–	–	+	+
L	2#n	Load 16-bit binary constant into ACCU1-L	2	0.4	0.2	–	–
		Load 32-bit binary constant into ACCU1	3	0.5	0.3	–	–
L	B#8#p	Load 8-bit hexadecimal constant into ACCU1-L	1	0.4	0.2	–	–
	W#16#p	Load 16-bit hexadecimal constant into ACCU1-L	2	0.4	0.2	–	–
	DW#16#p	Load 32-bit hexadecimal constant into ACCU1-L	3	0.5	0.3	–	–

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s	
				312	31x
L	'x'	Load 1 characters		0.4	0.2
L	'xx'	Load 2 characters	2	0.4	0.2
L	'xxx'	Load 3 characters		0.5	0.3
L	'xxxx'	Load 4 characters	3	0.5	0.3
L	D# date	Load IEC date (BCD)	3	0.5	0.3
L	S5T# time value	Load S5 time constant (16 bits)	2	0.5	0.3
L	TOD# time value	Load 32-bit time constant IEC – daytime	3	0.5	0.3
L	T# time value	Load 16-bit timer constant	2	0.4	0.2
		Load 32-bit timer constant	3	0.5	0.3
L	C# count value	Load 16-bit counter constant	2	0.4	0.2
L	P# bit pointer	Load bit pointer	3	0.5	0.3
L	L# integer	Load 32 bit integer constant	3	0.5	0.3
L	Real number	Load real number	3	0.5	0.3

## Load Instructions for Timers and Counters

Loading a time value or count value into ACCU1. The contents of ACCU1 are first saved to ACCU2. The bits of the status word are not affected.

Instruction	Operand	Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s			
				Direct Addressing		Indirect Addressing <sup>1</sup>	
				312	31x	312	31x
L	T f	Load time value	1/2	1.7	0.8	2.0+	1.1+
	Timer para.	Load time value (addressed via parameter)	2	–	–	+	+
L	C f	Load count value	1/2	1.4	0.7	2.3+	1.2+
	Counter para.	Load count value (addressed via parameter)	2	–	–	+	+
LD	T f	Load time value in BCD	1/2	4.2	2.2	5.0+	2.5+
	Timer para.	Load time value in BCD (addressed via parameter)	2	–	–	+	+
LD	C f	Load count value in BCD	1/2	4.4	2.2	5.4+	2.9+
	Counter para.	Load count value (addressed via parameter)	2	–	–	+	+

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

## Transfer Instructions

Transferring the contents of ACCU1 to the addressed Inrand. The status word is not affected. Remember that some transfer instructions depend on the MCR.

Instruction	Address Identifier		Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s			
					Direct Addressing		Indirect Addressing <sup>1</sup>	
					312	31x	312	31x
T	IB	a	Transfer contents of ACCU1-LL to ... input byte (MCR-dependent)	1/2	0.2	0.1	2.4+	1.3+
					1.1	0.5	2.7+	1.5+
	QB	a	output byte (MCR-dependent)	1/2	0.2	0.1	2.4+	1.3+
					1.1	0.5	2.7+	1.5+
	PQB	a	peripheral output byte (MCR-dependent)	1/2	58.7	29.4	104.8+	55.9+
	PQB	a	Digital Onbaord I/O <sup>3</sup> (MCR-dependent)	1/2	58.8	29.4	105.2+	56.1+
PQB	a	Analoge Onboard I/O <sup>4</sup> (MCR-dependent)	1/2	57.3	53.9	70.6+	61.0+	
					58.2	54.4	71.2+	61.3+
					–	49.2		56.3+
					–	49.7		56.8+

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

<sup>3</sup> Access to digital onboard I/O

<sup>4</sup> Access to analog onboard I/O

Instruction	Address Identifier		Description	Length in Words	Typical Execution Time in $\mu$ s			
					Direct Addressing		Indirect Addressing <sup>1</sup>	
					312	31x	312	31x
T	MB	a	bit memory byte (MCR-dependent)	1/2	0.2	0.1	2.4+	1.3+
	LB	a	local data byte (MCR-dependent)	2	1.2	0.6	2.7+	1.5+
					0.4	0.2	3.3+	1.7+
	DBB	a	data byte (MCR-dependent)	2	1.5	0.8	2.9+	1.5+
2.7					1.3	4.1+	2.2+	
DIB	a	instance data byte (MCR-dependent)	2	2.7	1.3	4.5+	2.4+	
				2.4	1.3	4.1+	2.2+	
T	g[AR1,m]		Register-ind., area-internal (AR1)	2	-	-	+	+
			Register-ind., area-internal (AR2)	2	-	-	+	+
	g[AR2,m]		Area-crossing (AR1)	2	-	-	+	+
			Area-crossing (AR2)	2	-	-	+	+
	B[AR1,m]		Via parameter	2	-	-	+	+
B[AR2,m]			2	-	-	+	+	
	Parameter							

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

Instruction	Address Identifier	Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s			
				Direct Addressing		Indirect Addressing <sup>1</sup>	
				312	31x	312	31x
T	IW	Transfer contents of ACCU1-L to... input word (MCR-dependent)	1/2	0.4 1.1	0.2 0.6	2.6+ 2.9+	1.4+ 1.5+
	QW	output word (MCR-dependent)	1/2	0.4 1.1	0.2 0.6	2.6+ 2.9+	1.4+ 1.5+
	PQW	peripheral output word (MCR-dependent)	1/2	64.4 64.6	32.2 32.3	121.6+ 120.5+	64.8+ 64.3+
	PQW	Digital Onbaord I/O <sup>3</sup> (MCR-dependent)	1/2	70.5 71.1	66.1 66.4	85.8+ 86.4+	74.2+ 74.8+
	PQW	Analoge Onboard I/O <sup>4</sup> (MCR-dependent)	1/2	– –	66.1 66.4	– –	74.2+ 74.8+
T	MW	bit memory word (MCR-dependent)	1/2	0.4 1.5	0.2 0.7	3.2+ 3.5+	1.7+ 1.9+
	LW	local data word (MCR-dependent)	2	0.5 1.6	0.2 0.8	3.8+ 3.3+	2.0+ 1.8+
	DBW	data word (MCR-dependent)	2	3.2 3.2	1.6 1.6	4.8+ 5.2+	2.6+ 2.8+
	DIW	Instanz-data word (MCR-dependent)	2	3.2	1.5	4.8+	2.6+
				3.2	1.6	5.2+	2.8+

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> With direct instruction addressing/ with indirect instruction addressing

<sup>3</sup> Access to digital onboard I/O

<sup>4</sup> Access to analog onboard I/O

Instruction	Address Identifier	Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s			
				Direct Addressing		Indirect Addressing <sup>1</sup>	
				312	31x	312	31x
T	h[AR1,m]	Register-ind., area-internal(AR1)	2	–	+	+	+
	h[AR2,m]	Register-ind., area-internal(AR2)	2	–	+	+	+
	W[AR1,m]	Area-crossing (AR1)	2	–	+	+	+
	W[AR2,m]	Area-crossing (AR2)	2	–	+	+	+
	Parameter	Via parameter	2	–	+	+	+

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> Plus time required for loading the address of the instruction (see page 21)

Instruction	Address Identifier	Description	Length in Words <sup>2</sup>	Typical Execution Time in $\mu$ s			
				Direct Addressing		Indirect Addressing <sup>1</sup>	
				312	31x	312	31x
T	ID	Transfer contents of ACCU1 to ... input double word (MCR-dependent)	1/2	0.6	0.3	2.8+	1.5+
		output double word (MCR-dependent)	1/2	1.4	0.7	3.2+	1.7+
	QD	peripheral output double word (MCR-dependent)	1/2	0.6	0.3	2.8+	1.5+
		peripheral output double word (MCR-dependent)	1/2	1.4	0.7	3.2+	1.7+
	PQD	peripheral output double word (MCR-dependent)	1/2	73.1	36.6	130.1+	69.3+
		Analoge Onboard I/O <sup>3</sup> (MCR-dependent)	1/2	73.4	36.7	128.0+	68.2+
				–	91.3	–	100.4+
				–	91.9	–	101.3+
T	MD	bit memory double word (MCR-dependent)	1/2	0.6	0.3	3.8+	2.0+
		local data double word (MCR-dependent)	2	1.7	0.8	4.2+	2.3+
	LD	Data double word (MCR-dependent)	2	0.9	0.4	4.4+	2.4+
		Data double word (MCR-dependent)	2	2.0	1.0	4.0+	2.1+
	DID	Instanz data double word (MCR-dependent)	2	4.5	2.2	5.7+	3.0+
			2	4.4	2.2	5.7+	3.0+
T	i[AR1,m]	Register-ind., area-internal (AR1)	2	–	–	+	+
	i[AR2,m]	Register-ind., area -internal (AR2)	2	–	–	+	+
	D[AR1,m]	Area-crossing (AR1)	2	–	–	+	+
	D[AR2,m]	Area-crossing (AR2)	2	–	–	+	+
	Parameter	Via parameter	2	–	–	+	+

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> With direct instruction addressing

<sup>3</sup> Access to analog onboard I/O (MCR-dependent)

## Load and Transfer Instructions for Address Registers

Loading a double word from a memory area or register into AR1 or AR2.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s	
				312	31x
LAR1	–	Load contents from ... ACCU1	1	0.2	0.1
	AR2	Address register 2	1	0.2	0.1
	DBD    a	Data double word	2	4.6	2.3
	DID    a	Instance data double word	2	4.6	2.3
	m	32-bit constant as pointer	3	0.3	0.2
	LD     a	Local data double word	2	1.5	0.7
	MD     a	Bit memory double word ... into AR1	2	1.0	0.5
LAR2	–	Load contents from ... ACCU1	1	0.2	0.1
	DBD    a	Data double word	2	0.2	0.1
	DID    a	Instance data double word	2	4.6	2.3
	m	32-bit constant as pointer	3	4.6	2.3
	LD     a	Local data double word	2	0.3	0.2
	MD     a	Bit memory double word ... into AR2	2	1.5	0.7
				1.0	0.5

Instruction	Address Identifier		Description	Length in Words	Typical Execution Time in $\mu$ s	
					312	31x
TAR1	–		Transfer contents of AR1 to ... ACCU1	1	0.3	0.2
	AR2		Address register 2	1	0.2	0.1
	DBD	a	Data double word	2	4.4	2.2
	DID	a	Instance data double word	2	4.4	2.2
	m		32-bit constant as pointer	2	0.9	0.4
	LD	a	Local data double word	2	0.6	0.3
	MD	a	Bit memory double word			
TAR2	–		Transfer contents of AR2 to ... ACCU1	1	0.3	0.2
	DBD	a	Data double word	2	0.2	0.1
	DID	a	Instance data double word	2	4.4	2.2
	m		32-bit constant as pointer	2	4.4	2.2
	LD	a	Local data double word	2	0.9	0.4
	MD	a	Bit memory double word			
TAR	–		Exchange the contents of AR1 and AR2	1	0.6	0.3

## Load and Transfer Instructions for the Status Word

Instruc- tion	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s						
				312			31x			
L	STW	Load status word <sup>1</sup> into ACCU1		1.1	0.6					
Status word for: <b>L STW</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:		Yes	Yes	Yes	Yes	Yes	0	0	Yes	0
Instruction affects:		–	–	–	–	–	–	–	–	–
T	STW	Transfer ACCU1 (bits 0 to 8) to the status word <sup>1</sup>		1.1	0.6					
Status word for: <b>T STW</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:		–	–	–	–	–	–	–	–	–
Instruction affects:		Yes	Yes	Yes	Yes	Yes	–	–	Yes	–

<sup>1</sup> For the structure of the status word see page 15

## Load Instructions for DB Number and DB Length

Loading the number/length of a data block into ACCU1. The old contents of ACCU1 are saved to ACCU2. The condition code bits are not affected.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s	
				312	31x
L	DBNO	Load number of data block	1	2.4	1.3
L	DINO	Load number of instance data block	1	2.4	1.3
L	DBLG	Load length of data block into byte	1	0.5	0.3
L	DILG	Load length of instance data block into byte	1	0.5	0.3

## Integer Math (16 Bits)

Math instructions on two 16-bit words. The result is in ACCU1 and ACCU1-L, resp.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu\text{s}$						
				312	31x					
+l	–	Add 2 integers (16 bits) $(\text{ACCU1-L}) = (\text{ACCU1-L}) + (\text{ACCU2-L})$	1	1.3	0.6					
-l	–	Subtract 1 integer from another (16 bits) $(\text{ACCU1-L}) = (\text{ACCU2-L}) - (\text{ACCU1-L})$	1	1.5	0.7					
*l	–	Multiply 1 integer by another (16 bits) $(\text{ACCU1}) = (\text{ACCU2-L}) * (\text{ACCU1-L})$	1	2.2	1.1					
/l	–	Divide 1 integer by another (16 bits) $(\text{ACCU1-L}) = (\text{ACCU2-L}) : (\text{ACCU1-L})$ The remainder is in ACCU1-H	1	2.6	1.3					
Status word for: +l, -l,*l,/l		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:		–	–	–	–	–	–	–	–	–
Instruction affects:		–	Yes	Yes	Yes	Yes	–	–	–	–

## Integer Math (32 Bits)

Math instructions on two 32-bit words. The result is in ACCU1.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s							
				312	31x						
+D	–	Add 2 integers (32 bits) (ACCU1)=(ACCU2)+ (ACCU1)	1	1.6	0.8						
–D	–	Subtract 1 integer from another (32 bits) (ACCU1)=(ACCU2)– (ACCU1)	1	2.2	1.1						
*D	–	Multiply 1 integer by another (32 bits) (ACCU1)=(ACCU2)* (ACCU1)	1	7.1	3.5						
/D	–	Divide 1 integer by another (32 bits) (ACCU1)=(ACCU2):(ACCU1)	1	5.7	2.8						
MOD	–	Divide 1 integer by another (32 bits) and load the remainder into ACCU1: (ACCU1)=remainder of [(ACCU2):(ACCU1)]	1	3.8	1.9						
Status word for:		<b>+D, –D,*D, /D, MOD</b>	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:			–	–	–	–	–	–	–	–	–
Instruction affects:			–	Yes	Yes	Yes	Yes	–	–	–	–

## Floating-Point Math (32 Bits)

The result of the math instruction is in ACCU1. The execution time of the instruction depends on the value to be calculated.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s						
				312	31x					
+R	–	Add 2 real numbers (32 bits) (ACCU1)=(ACCU2)+(ACCU1)	1	5.5	2.7					
–R	–	Subtract 1 real number from another (32 bits) (ACCU1)=(ACCU2)–(ACCU1)	1	5.5	2.7					
*R	–	Multiply 1 real number by another (32 bits) (ACCU1)=(ACCU2)*(ACCU1)	1	6.4	3.2					
/R	–	Divide 1 real number by another (32 bits) (ACCU1)=(ACCU2):(ACCU1)	1	6.1	3.0					
Status word for: +R, –R, *R, /R		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:		–	–	–	–	–	–	–	–	–
Instruction affects:		–	Yes	Yes	Yes	Yes	–	–	–	–

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s						
				312	31x					
NEGR	–	Negate the real number in ACCU1	1	0.8	0.4					
ABS	–	Form the absolute value of the real number in ACCU1	1	0.8	0.4					
Status word for: <b>NEGR, ABS</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:		–	–	–	–	–	–	–	–	–
Instruction affects:		–	–	–	–	–	–	–	–	–

## Square Root and Square Instructions (32 Bits)

The result of the instruction is in ACCU1. The instructions can be interrupted.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s							
				312	31x						
SQRT	–	Calculate the square root of a real number in ACCU1	1	643	322						
SQR	–	Form the square of a real number in ACCU1	1	177	89						
Status word for:	<b>SQRT, SQR</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:			–	–	–	–	–	–	–	–	–
Instruction affects:			–	Yes	Yes	Yes	Yes	–	–	–	–

## Logarithmic Function (32 Bits)

The result of the logarithmic function is in ACCU1. The instructions can be interrupted.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s						
				312	31x					
LN	–	Form the natural logarithm of a real number in ACCU1	1	455	227					
EXP	–	Calculate the exponential value of a real number in ACCU1 to the base e (= 2.71828)	1	898	449					
Status word for: <b>LN, EXP</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:		–	–	–	–	–	–	–	–	–
Instruction affects:		–	Yes	Yes	Yes	Yes	–	–	–	–

## Trigonometrical Functions (32 Bits)

The result of the instruction is in ACCU1. The instructions can be interrupted.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu\text{s}$							
				312	31x						
SIN <sup>1</sup>	–	Calculate the sine of a real number	1	545	272						
ASIN <sup>2</sup>	–	Calculate the arcsine of a real number	1	1584	792						
COS <sup>1</sup>	–	Calculate the cosine of a real number	1	606	303						
ACOS <sup>2</sup>	–	Calculate the arccosine of a real number	1	1762	881						
TAN <sup>1</sup>	–	Calculate the tangent of a real number	1	549	274						
ATAN <sup>2</sup>	–	Calculate the arctangent of a real number	1	595	297						
Status word for: <b>SIN, ASIN, COS, ACOS, TAN, ATAN</b>		BIE	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:			–	–	–	–	–	–	–	–	–
Instruction affects:			–	Yes	Yes	Yes	Yes	–	–	–	–

<sup>1</sup> Specify the angle in radians; the angle must be given as a floating point value in ACCU 1.

<sup>2</sup> The result is an angle in radians.

## Adding Constants

Adding integer constants and storing the result in ACCU1. The condition code bits are not affected.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s	
				312	31x
+	i8	Add an 8-bit integer constant	1	0.2	0.1
+	i16	Add a 16-bit integer constant	2	0.2	0.1
+	i32	Add a 32-bit integer constant	3	0.3	0.2

## Adding Using Address Registers

Adding a 16-bit integer to the contents of the address register. The value is in the instruction or in ACCU1-L. The condition code bits are not affected.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s	
				312	31x
+AR1	–	Add the contents of ACCU1-L to those of AR1	1	0.2	0.1
+AR1	m	Add a pointer constant to the contents of AR1	2	0.4	0.2
+AR2	–	Add the contents of ACCU1-L to those of AR2	1	0.2	0.1
+AR2	m	Add pointer constant to the contents of AR2	2	0.4	0.2

## Comparison Instructions with Integers (16 Bits)

Comparing the 16-bit integers in ACCU1-L and ACCU2-L. RLO = 1 if the condition is satisfied.

Identifier	Address Instruction	Description	Length in Words	Typical Execution Time in $\mu\text{s}$						
				312	31x					
==I	–	ACCU2-L=ACCU1-L	1	1.4	0.7					
<>I	–	ACCU2-L $\neq$ ACCU1-L	1	1.6	0.8					
<I	–	ACCU2-L<ACCU1-L	1	1.6	0.7					
<=I	–	ACCU2-L<=ACCU1-L	1	1.4	0.7					
>I	–	ACCU2-L>ACCU1-L	1	1.3	0.7					
>=I	–	ACCU2-L>=ACCU1-L	1	1.4	0.7					
Status word for: ==I, <>I, <I, <=I, >I, >=I		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
Instruction depends on:		–	–	–	–	–	–	–	–	–
Instruction affects:		–	Yes	Yes	0	–	0	Yes	Yes	1

## Comparison Instructions with Integers (32 Bits)

Comparing the 32-bit integers in ACCU1 and ACCU2. RLO = 1 if the condition is satisfied.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu\text{s}$							
				312	31x						
==D	–	ACCU2=ACCU1	1	1.4	0.7						
<>D	–	ACCU2 $\neq$ ACCU1	1	1.4	0.7						
<D	–	ACCU2<ACCU1	1	1.4	0.7						
<=D	–	ACCU2<=ACCU1	1	1.4	0.7						
>D	–	ACCU2>ACCU1	1	1.3	0.7						
>=D	–	ACCU2>=ACCU1	1	1.3	0.7						
Status word for:		==D,<>D, <D, <=D, >D, >=D	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
Instruction depends on:			–	–	–	–	–	–	–	–	–
Instruction affects:			–	Yes	Yes	0	–	0	Yes	Yes	1

## Comparison Instructions with Real Numbers (32 Bits)

Comparing the 32-bit real numbers in ACCU1 and ACCU2. RLO = 1 if the condition is satisfied. The execution time of the instruction depends on the value to be compared.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu\text{s}$							
				312	31x						
==R	–	ACCU2=ACCU1	1	6.3	3.1						
<>R	–	ACCU2 $\neq$ ACCU1	1	6.3	3.1						
<R	–	ACCU2<ACCU1	1	6.4	3.2						
<=R	–	ACCU2<=ACCU1	1	6.3	3.1						
>R	–	ACCU2>ACCU1	1	6.3	3.1						
>=R	–	ACCU2>=ACCU1	1	6.4	3.2						
Status word for:		==R, <>R, <R, <=R, >R, >=R	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
Instruction depends on:			–	–	–	–	–	–	–	–	–
Instruction affects:			–	Yes	Yes	Yes	Yes	0	Yes	Yes	1

## Shift Instructions

Shifting the contents of ACCU1 and ACCU1-L to the left or right by the specified number of places. If no address identifier is specified, shift the number of places into ACCU2-LL. Any positions that become free are padded with zeros or the sign. The last bit shifted is in condition code bit CC 1.

Instruc- tion	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu\text{s}$							
				312	31x						
SLW	–	Shift the contents of ACCU1-L to the left. Positions that become free are provided with zeros.	1	1.9	1.0						
SLW	0 ... 15			0.6	0.3						
SLD	–	Shift the contents of ACCU1 to the left. Positions that become free are provided with zeros.	1	2.5	1.2						
SLD	0 ... 32			2.5	1.3						
SRW	–	Shift the contents of ACCU1-L to the right. Positions that become free are provided with zeros.	1	1.9	0.9						
SRW	0 ... 15			0.6	0.3						
SRD	–	Shift the contents of ACCU1 to the right. Positions that become free are provided with zeros.	1	2.5	1.2						
SRD	0 ... 32			2.5	1.3						
Status word for:		<b>SLW, SLD, SRW, SRD</b>	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:			–	–	–	–	–	–	–	–	–
Instruction affects:			–	Yes	Yes	Yes	–	–	–	–	–

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu\text{s}$						
				312	31x					
SSI	–	Shift the contents of ACCU1-L with sign to the right. Positions that become free are provided with the sign (bit 15).	1	1.8	0.9					
SSI	0 ... 15			0.6	0.3					
SSD	–	Shift the contents of ACCU1 with sign to the right	1	2.5	1.2					
SSD	0 ... 32			2.5	1.3					
Status word for: <b>SSI, SSD</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:		–	–	–	–	–	–	–	–	–
Instruction affects:		–	Yes	Yes	Yes	–	–	–	–	–

## Rotate Instructions

Rotate the contents of ACCU1 to the left or right by the specified number of places. If no address identifier is specified, rotate the number of places into ACCU2-LL.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu\text{s}$						
				312	31x					
RLD	–	Rotate the contents of ACCU1 to the left	1	2.2	1.1					
RLD	0 ... 32			3.2	1.6					
RRD	–	Rotate the contents of ACCU1 to the right	1	2.2	1.1					
RRD	0 ... 32			2.4	1.2					
Status word for: <b>RLD, RRD</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
Instruction depends on:		–	–	–	–	–	–	–	–	–
Instruction affects:		–	Yes	Yes	Yes	–	–	–	–	–
RLDA	–	Rotate the contents of ACCU1 one bit position to the left t						1.7	0.8	
RRDA	–									
Status word for: <b>RLDA, RRDA</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
Instruction depends on:		–	–	–	–	–	–	–	–	–
Instruction affects:		–	Yes	0	0	–	–	–	–	–

## Accumulator Transfer Instructions, Incrementing and Decrementing

The status word is not affected.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s	
				312	31x
CAW	–	Reverse the order of the bytes in ACCU1-L. LL, LH becomes LH, LL.	1	0.2	0.1
CAD	–	Reverse the order of the bytes in ACCU1. LL, LH, HL, AA becomes HH, HL, LH, LL.	1	0.4	0.2
TAK	–	Swap the contents of ACCU1 and ACCU2	1	0.5	0.3
PUSH	–	The contents of ACCU1 are transferred to ACCU2.	1	0.2	0.1
POP	–	The contents of ACCU2 are transferred to ACCU1:	1	0.2	0.1
INC	0 ... 255	Increment ACCU1-LL	1	0.2	0.1
DEC	0 ... 255	Decrement ACCU1-LL	1	0.2	0.1

## Program Display and Null Operation Instructions

The status word is not affected.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s	
				312	31x
BLD	0 ... 255	Program display instruction: Is treated by the CPU like a null operation instruction.	1	0.2	0.1
NOP	0 1	Null Operation instruction:	1	0.2 0.2	0.1 0.1

## Data Type Conversion Instructions

The results of the conversion are in ACCU1. When converting real numbers, the execution time depends on the value.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu\text{s}$							
				312	31x						
BTI	–	Conv. cont. of ACCU1 from BCD to integer (16 bits) ( <b>BCD To Int</b> )	1	3.9	1.9						
BTD	–	Conv. cont. of ACCU1 from BCD to double int. (32 bits) ( <b>BCD To Doubleint</b> )	1	8.6	4.3						
DTR	–	Convert contents of ACCU1 from double integer to real (32 bits) ( <b>Doubleint To Real</b> )	1	5.5	2.7						
ITD	–	Convert contents of ACCU1 from integer (16 bits) to double int. (32 bits) ( <b>Int To Doubleint</b> )	1	0.2	0.1						
Status word for:		<b>BTI, BTD, DTR, ITD</b>	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:			–	–	–	–	–	–	–	–	–
Instruction affects:			–	–	–	–	–	–	–	–	–
ITB	–	Conv. cont. of ACCU1 from int. (16 bits) to BCD from 0 to +/- 999 ( <b>Int To BCD</b> )	1	4.4	2.2						
DTB	–	Conv. cont. of ACCU1 f. double int. (32 bits) t. BCD f. 0 to +/-9 999 999 ( <b>Doubleint To BCD</b> )	1	10.0	5.0						

Instruction	Address Identifier	Description	Length in Words		Typical Execution Time in $\mu\text{s}$								
			312	31x	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
RND	–	Convert a real number into a 32-bit integer.	1	6.5	3.2								
RND–	–	Convert a real number into a 32-bit integer. The number is	1	6.5	3.3								
Status word for:		<b>ITB, DTB, RND, RND–, RND+,TRUNC</b>	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$		
Instruction depends on:			–	–	–	–	–	–	–	–	–	–	–
Instruction affects:			–	–	–	Yes	Yes	–	–	–	–	–	–
RND+	–	Convert a real number into a 32-bit integer. The number is rounded to the next whole number.	1	6.7	3.3								
TRUNC	–	Convert a real number into a 32-bit integer. The places after the decimal point are truncated.	1	6.3	3.1								
Status word for:		<b>ITB, DTB, RND, RND–, RND+,TRUNC</b>	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$		
Instruction depends on:			–	–	–	–	–	–	–	–	–	–	–
Instruction affects:			–	–	–	Yes	Yes	–	–	–	–	–	–

## Forming the Ones and Twos Complements

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu\text{s}$						
				312	31x					
INVI	–	Form the ones complement of ACCU1-L	1	0.2	0.1					
INVD	–	Form the ones complement of ACCU1	1	0.2	0.1					
Status word for: <b>INVI, INVD</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:		–	–	–	–	–	–	–	–	–
Instruction affects:		–	–	–	–	–	–	–	–	–
NEGI	–	Form the twos complement of ACCU1-L (integer)	1	1.4		0.7				
NEGD	–	Form the twos complement of ACCU1 (double integer)	1	1.6		0.8				
Status word for: <b>NEGI, NEGD</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:		–	–	–	–	–	–	–	–	–
Instruction affects:		–	Yes	Yes	Yes	Yes	–	–	–	–

## Block Call Instructions

In- struction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s						
				Direct Addressing		Indirect Addressing <sup>1</sup>				
				312	31x	312	31x			
CALL	FB q, DB q	Unconditional call of an FB, with parameter transfer	1	16.4	8.8	–	–			
CALL	SFB q, DB q	Unconditional call of an SFB, with parameter transfer	2	2	2	–	–			
CALL	FC q	Unconditional call of a function, with parameter transfer	1	15.6	7.5	–	–			
CALL	SFC q	Unconditional call of an SFC, with parameter transfer	2	2	2	–	–			
Status word for:	<b>CALL</b>	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:		–	–	–	–	–	–	–	–	–
Instruction affects:		–	–	–	–	0	0	1	–	0

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> See chapter System Functions (SFCs)/ see chapter System Function Blocks (SFBs)

In- struction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s							
				Direct Addressing		Indirect Addressing <sup>1</sup>					
				312	31x	312	31x				
UC	FB q FC q Parameter	Unconditional call of blocks without parameter transfer FB/FC call via parameter	1 <sup>3</sup>	9.1	6.0	9.8+	6.4+				
				9.1	6.0	9.8+	6.4+				
				9.1	6.0	9.8+	6.4+				
CC	FB q FC q Parameter	Conditional call of blocks without parameter transfer FB/FC call via parameter	1 <sup>3</sup>	9.4	6.2	9.9+	6.6+				
				9.4	6.2	9.9+	6.6+				
				9.4	6.2	9.9+	6.6+				
Status word for: <b>UC, CC</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:			–	–	–	–	–	–	–	–	–
Instruction affects:			–	–	–	–	0	0	1	–	0
OPN	DB q DI q Parameter	Open: Data block Instance data block Data block using parameters	1/2 <sup>2</sup> 2 2	0.7		1.2+		1.2+			
				0.7		1.2+		1.2+			
				0.7		1.2+		1.2+			
Status word for: <b>OPN</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:			–	–	–	–	–	–	–	–	–
Instruction affects:			–	–	–	–	–	–	–	–	–

<sup>1</sup> Plus time required for loading the address of the instruction (see page 21)

<sup>2</sup> Block No. > 255

<sup>3</sup> With direct instruction addressing

## Block End Instructions

In- struc- tion	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s						
				312	31x					
BE	–	End block	1	4.4	2.2					
BEU	–	End block unconditionally	1	4.4	2.2					
Status word for:	<b>BE, BEU</b>	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:		–	–	–	–	–	–	–	–	–
Instruction affects:		–	–	–	–	0	0	1	–	0
BEC	–	End block conditionally if RLO = "1"						1.2	0.6	
Status word for:	<b>BEC</b>	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:		–	–	–	–	–	–	–	Yes	–
Instruction affects:		–	–	–	–	Yes	0	1	1	0

## **Exchanging Shared Data Block and Instance Data Block**

Exchanging the two current data blocks. The current shared data block becomes the current instance data block, and vice versa. The condition code bits are not affected.

<b>Instruction</b>	<b>Address Identifier</b>	<b>Description</b>	<b>Length in Words</b>	<b>Typical Execution Time in <math>\mu</math>s</b>	
				<b>312</b>	<b>31x</b>
CDB		Exchange shared data block and instance data block	1	0.2	0.1

## Jump Instructions

Jumping as a function of conditions. With 8-bit operands the jump width is between –128 and +127. In the case of 16-bit operands, the jump width lies between –32768 and –129 (+128 and +32767).

**Note:**

Please note for S7-300 CPU programs that the jump destination always forms the **beginning** of a Boolean logic string in the case of jump instructions. The jump destination must not be included in the logic string.

Instruction	Address Identifier	Description	Length in Words		Typical Execution Time in $\mu$ s						
			312	31x	BR	CC 1	CC 0	OV	OS	OR	STA
JU	LABEL	Jump unconditionally	1 1/2		3.6	1.8					
Status word for: <b>JU</b>			BR	CC 1	CC 0	OV	OS	OR	STA		
Instruction depends on:			–	–	–	–	–	–	–	–	–
Instruction affects:			–	–	–	–	–	–	–	–	–
JC	LABEL	Jump if RLO = "1"	1 1/2		3.8	1.9					
JCN	LABEL	Jump if RLO = "0"	2		3.8	1.9					
Status word for: <b>JC, JCN</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:			–	–	–	–	–	–	–	Yes	–
Instruction affects:			–	–	–	–	–	0	1	1	0

<sup>1</sup> 1 word long for jump widths between –128 and +127

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu\text{s}$							
				312	31x			FC			
JCB	LABEL	Jump if RLO = "1". Save the RLO in the BR bit	2	3.8	1.9						
JNB	LABEL	Jump if RLO = "0". Save the RLO in the BR bit	2	3.8	1.9						
Status word for: <b>JCB, JNB</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:			-	-	-	-	-	-	-	Yes	-
Instruction affects:			Yes	-	-	-	-	0	1	1	0
JBI	LABEL	Jump if BR = "1"	2	3.8	1.9						
JNBI	LABEL	Jump if BR = "0"	2	3.8	1.9						
Status word for: <b>JBI, JNBI</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:			Yes	-	-	-	-	-	-	-	-
Instruction affects:			-	-	-	-	-	0	1	-	0
JO	LABEL	Jump on stored overflow (OV = "1")	1 <sup>1/2</sup>	3.8	1.9						
Status word for: <b>JO</b>			BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:			-	-	-	Yes	-	-	-	-	-
Instruction affects:			-	-	-	-	-	-	-	-	-

<sup>1</sup> 1 word long for jump widths between -128 and +127

Instruction	Address Identifier	Description	Length in Words				Typical Execution Time in $\mu\text{s}$				
			312		31x						
JOS	LABEL	Jump on stored overflow (OS = "1")	2				3.8		1.9		
Status word for: <b>JOS</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$	
Instruction depends on:		–	–	–	–	Yes	–	–	–	–	
Instruction affects:		–	–	–	–	0	–	–	–	–	
JUO	LABEL	Jump if "unordered instruction" (CC 1=1 and CC 0=1)	2				3.8		1.9		
JZ	LABEL	Jump if result=0 (CC 1=0 and CC 0=0)	1 1/2				3.8		1.9		
JP	LABEL	Jump if result>0 (CC 1=1 and CC 0=0)	1 1/2				3.8		1.9		
JM	LABEL	Jump if result<0 (CC 1=0 and CC 0=1)	1 1/2				3.8		1.9		
Status word for: <b>JUO, JZ, JP, JM</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$	
Instruction depends on:		–	Yes	Yes	–	–	–	–	–	–	
Instruction affects:		–	–	–	–	–	–	–	–	–	

<sup>1</sup> 1 word long for jump widths between –128 and +127

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s						
				312	31x					
JN	LABEL	Jump if result $\neq 0$ (CC 1=1 and CC 0=0) or (CC 1=0) and (CC 0=1)	1 1/2	3.8	1.9					
JMZ	LABEL	Jump if result $\leq 0$ (CC 1=0 and CC 0=1) or (CC 1=0 and CC 0=0)	2	3.8	1.9					
JPZ	LABEL	Jump if result $\geq 0$ (CC 1=1 and CC 0=0) or (CC 1=0) and (CC 0=0)	2	3.8	1.9					
Status word for: <b>JN, JMZ, JPZ</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:		–	Yes	Yes	–	–	–	–	–	–
Instruction affects:		–	–	–	–	–	–	–	–	–
JL	LABEL	Jump distributor This instruction is followed by a list of jump instructions. The operand is a jump label to subsequent instructions in this list. ACCU1-L contains the number of the jump instruction to be executed.	2	5.0	2.5					
LOOP	LABEL	Decrement ACCU1-L and jump if ACCU1-L $\neq 0$ (loop programming)	2	3.5	1.8					
Status word for: <b>JL, LOOP</b>		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:		–	–	–	–	–	–	–	–	–
Instruction affects:		–	–	–	–	–	–	–	–	–

<sup>1</sup> 1 word long for jump widths between –128 and +127

## Instructions for the Master Control Relay (MCR)

MCR=1→MCR is deactivated

MCR=0→MCR is activated; “T” and “=” instructions write “0” to the corresponding address identifiers; “S” and “R” instructions leave the memory contents unchanged.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in $\mu$ s						
				312	31x					
MCR(		Open an MCR zone. Save the RLO to the MCR stack.	1	1.3	0.8					
MCR(		Open an MCR zone. Save the RLO to the MCR stack.	1	1.3	0.8					
Status word for:	<b>MCR(</b>	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:		–	–	–	–	–	–	–	Yes	–
Instruction affects:		–	–	–	–	–	0	1	–	0
MCRA		Activate the MCR	1					0.2		0.1
MCRD		Deactivate the MCR	1					0.2		0.1
Status word for:	<b>)MCR</b>	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:		–	–	–	–	–	–	–	–	–
Instruction affects:		–	–	–	–	–	–	–	–	–

## Organisation Blocks (OB)

A user program for an S7-300 consists of blocks which contain the instructions, parameters, and data for the respective CPU. The individual CPUs of the S7-300 differ in the number of blocks which you can define for the respective CPU, and of those which are supplied by the operating system of the CPU. You can find a detailed description of the OBs and their use in the *STEP 7 online help system*.

Organisation Blocks	312	31x	Starting Events (Hexadecimal Values)
Cycle:			
OB 1	x	x	1101 <sub>H</sub> OB1 starting event 1103 <sub>H</sub> Running OB1 start event (conclusion of the free cycle)
Time-of-day interrupt:			
OB 10	x	x	1111 <sub>H</sub> Time-of-day interrupt event
Delay Interrupt:			
OB 20	x	x	1121 <sub>H</sub> Delay interrupt event
Watchdog interrupt:			
OB 35	x	x	1136 <sub>H</sub> Watchdog interrupt event

Organisation Blocks	312	31x	Starting Events (Hexadecimal Values)
Process interrupt:			
OB 40	x	x	1141 <sub>H</sub> Process interrupt
DPV1-Interrupt (only CPU 315-2 DP and 31xC-2 DP)			
OB 55	–	x	1155 <sub>H</sub> Status interrupt
OB 56	–	x	1156 <sub>H</sub> Update-interrupt
OB 57	–	x	1157 <sub>H</sub> Manufactur-specific interrupt
Error responses:			
OB 80	x	x	3501 <sub>H</sub> Cycle time violation 3502 <sub>H</sub> OB or FB request error 3505 <sub>H</sub> Time-of-day interrupt elapsed due to time jump 3507 <sub>H</sub> Multiple OB request error caused start info buffer overflow
Diagnostic interrupt:			
OB 82	x	x	3842 <sub>H</sub> Module o. k. 3942 <sub>H</sub> Module fault

Organisation Blocks	312	31x	Starting Events (Hexadecimal Values)
OB 85	x	x	35A1 <sub>H</sub> No OB or FB 35A3 <sub>H</sub> Error during access of a block by the operating system 39B1 <sub>H</sub> I/O access error during process image updating of the inputs (during each access) 39B2 <sub>H</sub> I/O access error during transfer of the process image to the output modules (during each access) 38B3 <sub>H</sub> I/O access error during process image updating of the inputs (outgoing event) 38B4 <sub>H</sub> I/O access error during transfer of the process image to the output modules (outgoing event) 39B4 <sub>H</sub> I/O access error during transfer of the process image to the output modules (incoming event)
OB 86	–	only DP	38C4 <sub>H</sub> Distributed I/O: station failed, outgoing 39C4 <sub>H</sub> Distributed I/O: station failed, incoming

Organisation Blocks	312	31x	Starting Events (Hexadecimal Values)
OB 87	x	x	35E1 <sub>H</sub> Incorrect frame identifier in GD 35E2 <sub>H</sub> GD packet status cannot be entered in DB 35E6 <sub>H</sub> GD whole status cannot be entered in DB
Restart:			
OB 100	x	x	1381 <sub>H</sub> Manual restart requests 1382 <sub>H</sub> Automatic restart requests

Organisation Blocks	312	31x	Starting Events (Hexadecimal Values)
Synchronous error responses:			
OB 121	x	x	2521 <sub>H</sub> BCD conversion error 2522 <sub>H</sub> Range length error during reading 2523 <sub>H</sub> Range length error during writing 2524 <sub>H</sub> Range error during reading 2525 <sub>H</sub> Range error during writing 2526 <sub>H</sub> Timer number error 2527 <sub>H</sub> Counter number error 2528 <sub>H</sub> Alignment error during reading 2529 <sub>H</sub> Alignment error during writing 253A <sub>H</sub> DB not loaded 253C <sub>H</sub> FC not loaded 253E <sub>H</sub> FB not loaded
OB 122	x	x	2944 <sub>H</sub> I/O access error at nth read access (n > 1) 2945 <sub>H</sub> I/O access error at nth write access (n > 1)

## Function Blocks (FB)

The following tables list the quantities, numbers, and maximal sizes of the function blocks, functions and data blocks that you can define in the individual CPUs of the S7-300.

<b>Blocks</b>	<b>31x (except for 315-2 DP)</b>	<b>315-2 DP</b>
Quantity <sup>1</sup>	512	2048
Admissible numbers	0 to 511	0 to 2047
Maximal size of an FB (process-relevant code)	16 kB	16 kB

## Functions (FC)

<b>Blocks</b>	<b>31x (except for 315-2 DP)</b>	<b>315-2 DP</b>
Quantity <sup>1</sup>	512	2048
Admissible numbers	0 to 511	0 to 2048
Maximal size of an FC (process-relevant code)	16 kB	16 kB

<sup>1</sup> Entire number FB, FC, DB: 1024

## Data Blocks

<b>Blocks</b>	<b>31x (except for 315-2 DP)</b>	<b>315-2 DP</b>
Quantity <sup>1</sup>	511	1023
Admissible numbers	1 to 511	1 to 1023
Maximal size of an FB (process-relevant code)	16 kB	16 kB

<sup>1</sup> Entire number: FB, FC, DB: 1024

## Memory required by the SFBs for the integrated inputs and outputs

SFB	Data	Load memory	Work memory (RAM)
41 CONT_C	126	330	162
42 CONT_S	90	266	126
43 PULSEGEN	34	168	70
44 ANALOG	98	316	134
46 DIGITAL	88	286	124
47 COUNT	34	178	70
48 FREQUENC	34	176	70
49 PULSE	24	138	60
60 SEND_PTP	40	290	76
61 RCV_PTP	44	298	80
62 RES_RCVB	28	272	64
63 SEND_RK	432	1074	468
64 FETCH_RK	432	1074	468
65 SERVE_RK	408	1032	444

## System Functions (SFC)

The following tables show the system functions offered by the operating systems of the S7-300 CPUs, and the execution times on the respective CPUs.

SFC No.	SFC Name	Description	Execution Time in $\mu\text{s}$	
			312	31x
0	SET_CLK	Sets the clock time	235	195
1	READ_CLK	Reads the clock time	70	60
2	SET_RTM	Sets the operating hours counter	75	65
3	CTRL_RTM	Starts/stops the operating hours counter	70	60
4	READ_RTM	Reads the operating hours counter	105	90
5	GADR_LGC	Determine logical channel address Rack-0 Internal DP	160	135
6	RD_SINFO	Reads start information of the current OB.	135	110
7	DP_PRAL <sup>1</sup>	Triggers a process interrupt from the user program of the CPU as DP slave through to DP master.	–	90
11	SYC_FR <sup>1</sup>	Synchronizes groups of DP slaves	–	300
12	D_ACT_DP <sup>1</sup>	Activates or deactivates DP slaves	–	410

<sup>1</sup> only CPU 31x-2 DP

SFC No.	SFC Name	Description	Execution Time in $\mu$ s	
			312	31x
13	DPNRM_DG <sup>1</sup>	Reads the DP-compliant slave diagnosis (CPU31)	–	150
14	DPRD_DAT	Reads/writes consistent data (n bytes)	–	150
15	DPWR_DAT		–	150
17	ALARM_SQ	Generates block-related messages that can be acknowledged	250	250
18	ALARM_S	Generates block-related messages that can not be acknowledged	250	250
19	ALARM_SC	Acknowledgment state of the last ALARM_SQ received message	110	110
20	BLKMOV	Copies variables within the working memory	90 + 2.6/ Byte	75 + 1.6/ Byte
21	FILL	Sets array default variables within the working memory	90 + 2.6/ Byte	75 + 2.2/ Byte
22	CREAT_DB	Generates a data block	110 + 3.5/ DB	110 + 3.5/ DB
23	DEL_DB	Deletes a data block	402	402
24	TEST_DB	Tests a data block	130	110
28	SET_TINT	Sets the times of a time-of-day interrupt	190	160

<sup>1</sup> only CPU 31x-2 DP

SFC No.	SFC Name	Description	Execution Time in $\mu$ s	
			312	31x
29	CAN_TINT	Cancels a time-of-day interrupt	85	70
30	ACT_TINT	Activates a time-of-day interrupt	140	120
31	QRY_TINT	Queries the status of a time-of-day interrupt	90	75
32	SRT_DINT	Starts a delay interrupt	90	75
33	CAN_DINT	Cancels a delay interrupt	60	50
34	QRY_DINT	Queries started delay interrupts	85	71
36	MSK_FLT	Masks sync faults	132	110
37	DMSK_FLT	Enables sync faults	143	120
38	READ_ERR	Reads event status register	140	120
39	DIS_IRT	Disables the handling of new interrupts	180	155
40	EN_IRT	Enables the handling of new interrupt events	125	105
41	DIS_AIRT	Delays the handling of interrupts	50	45
42	EN_AIRT	Enables the handling of interrupts	55	45
43	RE_TRIGR	Re-triggers the scan time monitor	50	40
44	REPL_VAL	Copies a substitute value into accumulator 1	60	50

46	STP	Forces the CPU into the STOP mode	–	
47	WAIT	Delays program execution in addition to waiting times	250	250
49	LGC_GADR	Converts a free address to the slot and rack for a module	250	210
50	RD_LGADR	Reads all the declared free addresses for a module	500	420
51	RDSYSST	Reads out the information from the system state list. SFC 51 is not interruptible through interrupts.	250 $\mu$ s + 10 $\mu$ s pro byte	224 $\mu$ s + 10 $\mu$ s pro byte
52	WR_USMSG	Writes specific diagnostic information in the diagnostic buffer	280	235
55	WR_PARM	Writes dynamic parameters to a module	2000	1700
56	WR_DPARM	Writes predefined dynamic parameters to a module	1750	1750
57	PARM_MOD	Assigns a module's parameters	<1650	<1400
58	WR_REC	Writes a module-specific data record	1400 $\mu$ s + 32 $\mu$ s pro byte	1400 $\mu$ s + 32 $\mu$ s pro byte
59	RD_REC	Reads a module-specific data record	500	500
64	TIME_TICK	Reads out the system time	55	50

SFC No.	SFC Name	Description	Execution Time in $\mu$ s	
			312	31x
65	X_SEND	Sends data to a communication partner external to your own S7 station	310	310
66	X_RCV	Receives data from a communication partner external to your own S7 station	120	120
67	X_GET	Reads data from a communication partner external to your own S7 station	190	190
68	X_PUT	Writes data to a communication partner external to your own S7 station	190	190
69	X_ABORT	Aborts connection to a communication partner external to your own S7 station	100	100
72	I_GET	Reads data from a communication partner within your own S7 station	190	190
73	I_PUT	Writes data to a communication partner within your own S7 station	190	190
74	I_ABORT	Aborts connection to a communication partner within your own S7 station	100	100
81	UBLKMOV	Copy the variable without interruption, length of the data to be copied up to 32 bytes	$90\mu\text{s} + 2\mu\text{s}$ pro Byte	$75\mu\text{s} + 2\mu\text{s}$ pro Byte
82	CREA_DBL	Create data block in load memory.	<1250	<1050
83	READ_DBL	Read from a data block in load memory	<1100	<950
84	WRIT_DBL	Write to a data block in load memory.	<1100	<900
101	RTM	Handling the Run-time meter	170	150
102	RD_DPARA	Read predefined parameter.	<1750	<1500

## System Function Blocks (SFB)

The following table lists the system function blocks supplied by the operating system of the S7-300's CPUs, and the execution times on the respective CPUs.

SFB No.	SFB Name	Description	Execution Time in $\mu$ s	
			312	31x
0	CTU	Counts up	101	90
1	CTD	Counts down	101	90
2	CTUD	Counts up and counts down	109	100
3	TP	Generates a pulse	135	115
4	TON	Delays a leading edge	120	101
5	TOF	Delays a falling edge	120	100
32	DRUM	Implements a sequence processor with a maximum of 16 s	90	80
<b>SFBs for the integrated inputs/outputs (only CPU 31xC)</b>				
41	CONT_C	Continuous control	3300	
42	CONT_S	Step control	2800	
43	PULSEGEN	Pulse generation	1500	

SFB No.	SFB Name	Description	Execution Time in $\mu\text{s}$	
			312	31x
44	ANALOG <sup>1</sup>	positioning with analog output idle run start positioning run request	–	880 2900 1300
46	DIGITAL <sup>1</sup>	positioning with digital outputs idle run start positioning run request	–	810 2200 1200
47	COUNT	counting	1222	1222
48	FREQUENC	frequency measurement	1240	1240
49	PULSE	pulse width modulation	1101	1101

<sup>1</sup> only CPU 314C

SFB No.	SFB Name	Description	Execution Time in $\mu\text{s}$	
			312	31x
52	RDREC	Read Data set from DP slave or central module	500	
53	WRREC	Write Data set to DP slave or central module	1400 $\mu\text{s}$ + 32 $\mu\text{s}$ pro byte	
54	RALRM	Read out interrupt status information from interrupts of a DP slave or of a central module in the respective OB	650	
60	SEND_PTP <sup>1</sup>	send data (n characters) idle run operationalmode	–	405 600+n*11 (1≤n≤1024)
61	RCV_PTP <sup>1</sup>	receive data (n characters) idle run operationalmode	–	430 600+n*7 (1≤n≤1024)

<sup>1</sup> only CPU 31xC-2 PtP

SFB No.	SFB Name	Description	Execution Time in $\mu\text{s}$	
			312	31x
62	RES_RCVB <sup>1</sup>	clear input buffer	–	
		idle run		390
		operational mode		700
63	SEND_RK <sup>2</sup>	send data (n characters, data exceeding a length of 128 characters are transferred in blocks with a maximum length of 128 characters)	–	450
		idle run		$1210+n*7$
		operational mode		$(1 \leq n \leq 1024)$
64	FETCH_RK <sup>2</sup>	send data (n characters, data exceeding a length of 128 characters are transferred in blocks with a maximum length of 128 characters)	–	620
		idle run		$1680+n*7$
		operational mode		$(1 \leq n \leq 1024)$

<sup>1</sup> only CPU 31xC–2 PtP

<sup>2</sup> only CPU 314C–2 PtP

SFB No.	SFB Name	Description	Execution Time in $\mu\text{s}$	
			312	31x
65	SERVE_RK <sup>1</sup>	receive/provide data (n characters, data exceeding a length of 128 characters are transferred in blocks with a maximum length of 128 characters) idle run operational mode	–	510 $1320+n*7$ ( $1 \leq n \leq 1024$ )
75	SALRM	Set desired interrupts of I-slaves	–	–

<sup>1</sup> only CPU 314C–2 PtP

## IEC Functions

You can use the following functions in STEP 7:

FC No.	FC Name	Description
<b>DATE_AND_TIME</b>		
3	D_TOD_DT	Concatenates the data formats DATE and TIME_OF_DAY (TOD) and converts to data format DATE_AND_TIME.
6	DT_DATE	Extracts the DATE data format from the DATE_AND_TIME data format.
7	DT_DAY	Extracts the day of the week from the data format DATE_AND_TIME.
8	DT_TOD	Extracts the TIME_OF_DAY data format from the DATE_AND_TIME data format.
<b>Time Formats</b>		
33	S5TI_TIM	Converts S5 TIME data format to TIME data format
40	TIM_S5TI	Converts TIME data format to S5 TIME data format
<b>Duration</b>		
1	AD_DT_TM	Adds a duration in the TIME format to a time in the DT format. The result is a new time in the DT format.
35	SB_DT_TM	Subtracts a duration in the TIME format from a time in the DT format. The result is a new time in the DT format.
34	SB_DT_DT	Subtracts two times in the DT format. The result is a duration in the TIME format.

FC No.	FC Name	Description
<b>Compare DATE_AND_TIME</b>		
9	EQ_DT	Compares the contents of two variables in the DATE_AND_TIME format for equal to.
12	GE_DT	Compares the contents of two variables in the DATE_AND_TIME format for greater than or equal to.
14	GT_DT	Compares the contents of two variables in the DATE_AND_TIME format for greater than.
18	LE_DT	Compares the contents of two variables in the DATE_AND_TIME format for less than or equal to.
23	LT_DT	Compares the contents of two variables in the DATE_AND_TIME format for less than.
28	NE_DT	Compares the contents of two variables in the DATE_AND_TIME format for not equal to.
<b>Compare STRING</b>		
10	EQ_STRNG	Compares the contents of two variables in the STRING format for equal to.
13	GE_STRNG	Compares the contents of two variables in the STRING format for greater than or equal to.
15	GT_STRNG	Compares the contents of two variables in the STRING format for greater than.
19	LE_STRNG	Compares the contents of two variables in the STRING format for less than or equal to.
24	LT_STRNG	Compares the contents of two variables in the STRING format for less than.
29	NE_STRNG	Compares the contents of two variables in the STRING format for not equal to.

FC-Nr.	FC-Name	Description
<b>STRING Variable Processing</b>		
21	LEN	Reads the length of a STRING variable.
20	LEFT	Reads the first L characters of a STRING variable.
32	RIGHT	Reads the last L characters of a STRING variable.
26	MID	Reads the middle L characters of a STRING variable (starting at the defined character).
2	CONCAT	Concatenates two STRING variables in one STRING variable.
17	INSERT	Inserts a STRING variable into another STRING variable at a defined point.
4	DELETE	Deletes L characters of a STRING variable.
31	REPLACE	Replaces L characters of a STRING variable with a second STRING variable.
11	FIND	Finds the position of the second STRING variable in the first STRING variable.

FC No.	FC Name	Description
<b>Format Conversions with STRING</b>		
16	I_STRNG	Converts a variable from INTEGER format to STRING format.
5	DI_STRNG	Converts a variable from INTEGER (32-bit) format to STRING format.
30	R_STRNG	Converts a variable from REAL format to STRING format.
38	STRNG_I	Converts a variable from STRING format to INTEGER format.
37	STRNG_DI	Converts a variable from STRING format to INTEGER (32-bit) format.
39	STRNG_R	Converts a variable from STRING format to REAL format.
<b>Number Processing</b>		
22	LIMIT	Limits a number to a defined limit value.
25	MAX	Selects the largest of three numeric variables.
27	MIN	Selects the smallest of three numeric variables.
36	SEL	Selects one of two variables.

## System Status Sublist

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
0111 <sub>H</sub>	<b>CPU identification</b> One record of the sublist	0001 <sub>H</sub> 0006 <sub>H</sub> 0007 <sub>H</sub>	CPU type and version number Identification of the basic hardware Identification of the basic firmware
0012 <sub>H</sub> 0112 <sub>H</sub> 0F12 <sub>H</sub>	<b>CPU features</b> All records of the sublist Only those records of a group of features Header information only	0000 <sub>H</sub> 0100 <sub>H</sub> 0300 <sub>H</sub>	STEP 7 processing Time system in the CPU STEP 7 operation set
0013 <sub>H</sub>	<b>User memory areas</b>	–	Work memory

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
0014 <sub>H</sub>	<b>Operating system areas</b>	–	Process image of the inputs (number in bytes) Process image of the outputs (number in bytes) Number of memory markers Number of timers Number of counters Size of the I/O address area Entire local data area of the CPU (in bytes)

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
0015 <sub>H</sub>	<b>Block types</b> All records of the sublist	–	OBs (number and size) DBs (number and size) SDBs (number and size) FCs (number and size) FBs (number and size)
0019 <sub>H</sub> 0074 <sub>H</sub> 0174 <sub>H</sub>  0F19 <sub>H</sub> 0F74 <sub>H</sub>	<b>State of module LEDs</b> Status of each LED    Header information only	–  0001 <sub>H</sub> 0004 <sub>H</sub> 0005 <sub>H</sub> 0006 <sub>H</sub> 0011 <sub>H</sub>	–  SF-LED RUN-LED STOP-LED FRCE-LED BF-LED
011C <sub>H</sub>	<b>MMC copy protection</b>	0008 <sub>H</sub>	serial number of MMC

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
0132 <sub>H</sub>	<b>Communications status</b> on the communications type specified	0001 <sub>H</sub> 0004 <sub>H</sub>  0005 <sub>H</sub> 0008 <sub>H</sub>  000B <sub>H</sub> 000C <sub>H</sub>	Number and type of connections CPU protection level, position of the key switch, version identification of the user program and configuration Diagnostic status data Target system, correction factor, Run-time meter, Date/Time Run-time meter (32 bits) 0 to 7 Run-time meter (32 bits) 8 to 15
0222 <sub>H</sub>	<b>Interrupt status</b> Record for the specified interrupt	OB number	–

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
0232 <sub>H</sub>	<b>CPU Protection Level</b>	0004 <sub>H</sub>	CPU protection level and position of the key switch, version identification of the user program and hardware configuration
0092 <sub>H</sub> 0292 <sub>H</sub> 0692 <sub>H</sub>	<b>Status information of module racks</b> Expected status of the module rack in the central configuration Actual status of module rack in the central configuration OK status of the expansion devices in the central configuration	0000 <sub>H</sub>	Information about the status of the module rack in the central configuration
0D91 <sub>H</sub>	<b>Module status information</b> of all modules in the specified rack (all CPUs)	0000 <sub>H</sub> 0001 <sub>H</sub> 0002 <sub>H</sub> 0003 <sub>H</sub>	Features/parameters of the module plugged in Rack 0 Rack 1 Rack 2 Rack 3

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
00A0 <sub>H</sub> 01A0 <sub>H</sub>	<b>Diagnostic buffer</b> All entered event information The x latest information entries	–	Event information The information in each case depends on the event
00B1 <sub>H</sub> 00B2 <sub>H</sub> 00B3 <sub>H</sub>	<b>Module diagnostics</b> Data record 0 of the module diagnostics information Complete module-dependent record of the module diagnostics information Complete module-dependent record of the module diagnostics information	Module starting address Module rack and slot number Module starting address	Module-dependent diagnostics information

## PROFIBUS-DP Sublists

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
0A91 <sub>H</sub>	<p><b>Module status information in the CPU</b></p> <p>Status information of all DP subsystems and DP masters</p>		
0C91 <sub>H</sub>	Module status information of a module	Module starting address	Features/parameters of the module plugged in
0D91 <sub>H</sub>	<p><b>Module status information</b></p> <p>In the station named (for CPU 315-2 DP)</p>	xxyy <sub>H</sub>	<p>All modules of station yy in the DP subnet xx</p> <p>As DP slave: Status data for transfer memory areas</p>

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
0092 <sub>H</sub>	<p><b>Status information of module racks or stations in the DP network</b></p> <p>Target status of racks in central configuration or of stations in a subnet</p>	0000 <sub>H</sub>	Information on the state of the mounting rack in the central configuration
0292 <sub>H</sub>	Actual status of racks in central configuration or of stations in a subnet	DP-Mastersystem-ID	Information of status of stations in subnet
0692 <sub>H</sub>	OK status of expansion racks in central configuration or of stations in a subnet		
00B4 <sub>H</sub>	<p><b>Module diagnostics</b></p> <p>All standard diagnostic data of a station (only with DP master)</p>	Module start address (Diagnostic address)	Module-dependent diagnostic information

## Alphabetical Index of Instructions

Instruction	Page	Instruction	Page
)	36	=	50
)MCR	100	==D	82
+	79	==I	81
+AR1	80	==R	83
+AR2	80	<=D	82
+D	73	<=I	81
+I	72	<=R	83
+R	74	<>D	82
-D	73	<>I	81
-I	72	<>R	83
-R	74	<D	82
*D	73	<I	81
*I	72	<R	83
*R	74	>=D	82
/D	73	>=I	81
/I	72	>=R	83
/R	74	>D	82

*Alphabetical Index of Instructions*

<b>Instruction</b>	<b>Page</b>	<b>Instruction</b>	<b>Page</b>
>I	81	CAD	87
>R	83	CALL	92
A	29, 38, 44	CAW	87
A(	35	CC	93
ABS	75	CD	55
ACOS	78	CDB	95
AD	42	CLR	51
AN	30, 39, 45	COS	78
AN(	35	CU	55
ASIN	78	DEC	87
ATAN	78	DTB	89
AW	42	DTR	89
BE	94	EXP	77
BEC	94	FN	47
BEU	94	FP	46
BLD	88	FR	54, 56
BTD	89	INC	87
BTI	89		

*Alphabetical Index of Instructions*

<b>Instruction</b>	<b>Page</b>	<b>Instruction</b>	<b>Page</b>
INVD	91	LD	62
INVI	91	LN	77
ITB	89	LOOP	99
ITD	89	MCR{	100
JBI	97	MCRA	100
JC	96	MCRD	100
JCB	97	MOD	73
JCN	96	NEGD	91
JL	99	NEGI	91
JNB	97	NEGR	75
JNBI	97	NOP	88
JO	97	NOT	51
JOS	98	O	31, 37, 40, 44
JU	96	O{	35
L	57 – 62, 70, 71	OD	43
LAR1	68	ON	32, 40, 45
LAR2	68	ON{	35
		OPN	93

<b>Instruction</b>	<b>Page</b>	<b>Instruction</b>	<b>Page</b>
OW	<b>42</b>	SLD	<b>84</b>
POP	<b>87</b>	SLW	<b>84</b>
PUSH	<b>87</b>	SP	<b>53</b>
R	<b>49, 54, 55</b>	SQR	<b>76</b>
RLD	<b>86</b>	SQRT	<b>76</b>
RLDA	<b>86</b>	SRD	<b>84</b>
RND	<b>90</b>	SRW	<b>84</b>
RND+	<b>90</b>	SS	<b>53</b>
RND-	<b>90</b>	SSD	<b>85</b>
RRD	<b>86</b>	SSI	<b>85</b>
RRDA	<b>86</b>	T	<b>63 –67, 70</b>
S	<b>48, 55</b>		
SAVE	<b>52</b>		
SD	<b>53</b>		
SE	<b>53</b>		
SET	<b>51</b>		
SA	<b>54</b>		
SIN	<b>78</b>		

<b>Instruction</b>	<b>Page</b>	<b>Instruction</b>	<b>Page</b>
TAK	87	X	33, 44, 40
TAN	78	X(	35
TAR	69	XN	34, 41, 45
TAR1	69	XN(	35
TAR2	69	XOD	43
TRUNC	90	XOW	42
UC	93		

