SIEMENS

SIMATIC

S7-300
Instruction list S7-300 CPUs and ET 200 CPUs

Parameter Manual

This manual is part of the documentation package with order number:
6ES7398-8FA10-8BA0

06/2011
A5E02354744-05

Legal information

Warning notice system

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

⚠ DANGER

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

⚠ WARNING

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

⚠ CAUTION

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

CAUTION

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

NOTICE

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

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

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

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

Disclaimer of Liability

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

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E02354744-05
℗ 05/2011

Table of contents

Validity Range of the Instructions List

1

Table 1- 1 This instruction list applies to the CPUs listed below:

	Order number	As of firmware version	In the following referred to as [1]
S7-300 CPUs			
CPU 312	6ES7312-1AE14-0AB0	V3.3	312
CPU 312C	6ES7312-5BF04-0AB0	V3.3	
CPU 313C	6ES7313-5BG04-0AB0	V3.3	313
CPU 313C-2 PtP	6ES7313-6BG04-0AB0	V3.3	
CPU 313C-2 DP	6ES7313-6CG04-0AB0	V3.3	
CPU 314	6ES7314-1AG14-0AB0	V3.3	314
CPU 314C-2 PtP	6ES7314-6BH04-0AB0	V3.3	
CPU 314C-2 DP	6ES7314-6CH04-0AB0	V3.3	
CPU 314C-2 PN/DP	6ES7314-6EH04-0AB0	V3.3	
CPU 315-2 DP	6ES7315-2AH14-0AB0	V3.3	315
CPU 315-2 PN/DP	6ES7315-2EH14-0AB0	V3.2	
CPU 317-2 DP	6ES7317-2AK14-0AB0	V3.3	317
CPU 317-2 PN/DP	6ES7317-2EK14-0AB0	V3.2	
CPU 319-3 PN/DP	6ES7318-3EL01-0AB0	V3.2	319
ET 200 CPUs			
IM151-7 CPU	6ES7151-7AA21-0AB0	V3.3	151
IM151-8 PN/DP CPU	6ES7151-8AB01-0AB0	V3.2	
IM154-8 PN/DP CPU	6ES7154-8AB01-0AB0	V3.2	154

[1] except in those tables requiring a detailed differentiation

Address Identifiers and Parameter Ranges

2

The following address identifiers and address areas are used.

Because the values of CPU 313C-2 DP, 314C-2 DP and 314C-2 PN/DP deviate from the generally applicable table, there follows a separate table with the values of the listed CPUs.

Address	Parameter Ranges								Description
	312	313	314	315	317	319	151	154	
Q	0.0 to 127.7 (can be set up 1023.7)			0.0 to 127.7 (can be set up to 2047.7)	0.0 to 255.7 (can be set up to 8191.7)		0.0 to 127.7 (can be set up to 2047.7)		Output (in PIQ)
QB	0 to 127 (can be set up 1023)			0 to 127 (can be set up to 2047)	0 to 255 (can be set up to 8191)		0 to 127 (can be set up 2047)		Output byte (in PIQ)
QW	0 to 126 (can be set up 1022)			0 to 126 (can be set up to 2046)	0 to 254 (can be set up to 8190)		0 to 126 (can be set up 2046)		Output word (in PIQ)
QD	0 to 124 (can be set up 1020)			0 to 124 (can be set up to 2044)	0 to 252 (can be set up to 8188)		0 to 124 (can be set up 2044)		Output double word (in PIQ)
DB	1 to 16000								Data block
DBX	0.0 to 32731.7 [1]	0.0 to 65533.7							Data bit in data block
DBB	0.0 to 32731 [1]	0 to 65533							Data byte in DB
DBW	0.0 to 32730 [1]	0 to 65532							Data word in DB
DBD	0.0 to 32728 [1]	0 to 65530							Data double word in DB
DI	1 to 16000								Instance data block
DIX	0.0 to 32731.7 [1]	0.0 to 65533.7							Data bit in instance DB
DIB	0.0 to 32731 [1]	0 to 65533							Data byte in instance DB
DIW	0.0 to 32730 [1]	0 to 65532							Data word in instance DB
DID	0.0 to 32728 [1]	0 to 65530							Data double word in instance DB

[1] The same parameter ranges apply to the CPU 312C as for the other CPUs.

Address	Parameter Ranges								Description
	312	313	314	315	317	319	151	154	
I	0.0 to 127.7 (can be set up to 1023.7)			0.0 to 127.7 (can be set up to 2047.7)	0.0 to 255.7 (can be set up to 8191.7)		0.0 to 127.7 (can be set up to 2047.7)		Input (in PII)
IB	0 to 127 (can be set up to 1023)			0 to 127 (can be set up to 2047)	0 to 255 (can be set up to 8191)		0 to 127 (can be set up to 2047)		Input byte (in PII)
IW	0 to 126 (can be set up to 1022)			0 to 126 (can be set up to 2046)	0 to 254 (can be set up to 8190)		0 to 126 (can be set up to 2046)		Input word (in PII)
ID	0 to 124 (can be set up to 1020)			0 to 124 (can be set up to 2044)	0 to 252 (can be set up to 8188)		0 to 124 (can be set up to 2044)		Input double word (in PII)
M	0.0 to 255.7			0.0 to 2047.7	0.0 to 4095.7	0.0 to 8191	0.0 to 255.7	0.0 to 2047.7	Bit memory
MB	0 to 255			0 to 2047	0 to 4095	0 to 8191	0 to 255	0 to 2047	Bit memory byte
MW	0 to 254			0 to 2046	0 to 4094	0 to 8190	0 to 254	0 to 2046	Bit memory word
MD	0 to 252			0 to 2044	0 to 4092	0 to 8188	0 to 252	0 to 2044	Bit memory double word
L [2)]	0.0 to 2047.7								Local data
LB [2)]	0 to 2047								Local data byte
LW [2)]	0 to 2046								Local data word
LD [2)]	0 to 2044								Local data double word

[2)] When using temporary variables, please note that these are only valid within the particular block or are available as parent local data of other blocks called in this block.
After exit and renewed block call, it is not certain that the temporary variables will still include the same values that were present when the block call was closed previously.
Temporary variables are initially undefined during the block call and must always be re-initialized each time they are used for the first time in the block.

Address	Parameter ranges								Description
	312	313	314	315	317	319	151	154	
PQB	0 to 1023			0 to 2047	0 to 8191		0 to 2047		I/O output byte
PQW	0 to 1022			0 to 2046	0 to 8190		0 to 2046		I/O output word
PQD	0 to 1020			0 to 2044	0 to 8188		0 to 2044		I/O output double word
PIB	0 to 1023			0 to 2047	0 to 8191		0 to 2047		I/O input byte
PIW	0 to 1022			0 to 2046	0 to 8190		0 to 2046		I/O input word
PID	0 to 1020			0 to 2044	0 to 8188		0 to 2044		I/O input double word
T	0 to 255				0 to 511	0 to 2047	0 to 255		Timer
C	0 to 255				0 to 511	0 to 2047	0 to 255		Counter

The following addresses and address ranges apply to CPU 313C-2 DP, 314C-2 DP and 314C-2 PN/DP:

Address	Parameter ranges			Description
	313C-2 DP	314C-2 DP	314C-2 PN/DP	
Q	0.0 to 127.7 (can be set up 2047.7)		0.0 to 255.7 (can be set up to 2047.7)	Output (in PIQ)
QB	0 to 127 (can be set up to 2047)		0 to 255 (can be set up to 2047)	Output byte (in PIQ)
QW	0 to 126 (can be set up to 2046)		0 to 254 (can be set up to 2046)	Output word (in PIQ)
QD	0 to 124 (can be set up to 2044)		0 to 252 (can be set up to 2044)	Output double word (in PIQ)
DB	1 to 16000			Data block
DBX	0.0 to 65533.7			Data bit in data block
DBB	0 to 65533			Data byte in DB
DBW	0 to 65532			Data word in DB
DBD	0 to 65530			Data double word in DB
DI	1 to 16000			Instance data block
DIX	0.0 to 65533.7			Data bit in instance DB
DIB	0 to 65533			Data byte in instance DB
DIW	0 to 65532			Data word in instance DB
DID	0 to 65530			Data double word in instance DB

Address	Parameter ranges			Description
	313C-2 DP	314C-2 DP	314C-2 PN/DP	
I	0.0 to 127.7 (can be set up to 2047.7)		0.0 to 255.7 (can be set up to 2047.7)	Inputs (in PII)
IB	0 to 127 (can be set up to 2047)		0 to 255 (can be set up to 2047)	Input byte (in PII)
IW	0 to 126 (can be set up to 2046)		0 to 254 (can be set up to 2046)	Input word (in PII)
ID	0 to 124 (can be set up to 2044)		0 to 252 (can be set up to 2044)	Input double word (in PII)
M	0.0 to 255.7			Bit memory
MB	0 to 255			Bit memory byte
MW	0 to 254			Bit memory word
MD	0 to 252			Bit memory double word
L [1)]	0.0 to 2047.7			Local data
LB [1)]	0 to 2047			Local data byte
LW [1)]	0 to 2046			Local data word
LD [1)]	0 to 2044			Local data double word
PQB	0 to 2047			I/O output byte
PQW	0 to 2046			I/O output word
PQD	0 to 2044			Peripheral output double word
PIB	0 to 2047			I/O input byte
PIW	0 to 2046			I/O input word
PID	0 to 2044			Peripheral input double word
T	0 to 255			Timer
C	0 to 255			Counter

[1)] When using temporary variables, please note that these are only valid within the particular block or are available as parent local data of other blocks called in this block.
After exit and renewed block call, it is not certain that the temporary variables will still include the same values that were present when the block call was closed previously.
Temporary variables are initially undefined during the block call and must always be re-initialized each time they are used for the first time in the block.

Constants

3

Table 3- 1 The following constants are used:

Constant	Description
Parameter	Address, addressed via parameter
B#16#	Byte hexadecimal
W#16#	Word hexadecimal
DW#16#	Double word hexadecimal
D#Date	IEC date constant
L#Integer	32-bit integer constant
P#Bitpointer	Pointer constant
S5T#Time	S5 time constant [1] (16 bit), T#1D_5H_3M_1S_2MS
T#Time	Time constant (16/32 bit), T#1D_5H_3M_1S_2MS
TOD#Time	IEC time constant, T#1D_5H_3M_1S_2MS
C#Count value	Counter constant (BCD coded)
2#n	Binary constant
B (b1, b2) or B (b1, b2, b3, b4)	Constant, 2 or 4 byte

[1] Serves for loading the S5-Timer

Abbreviations

4

Table 4- 1 The abbreviations are used:

Abbreviation	... 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	Pointer constant	P#240.3
n	Binary constant	1001 1100
p	Hexadecimal constant	EA12
q	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	Address range (bit)	I, Q, M, L, DBX, DIX
f	Timer/counter number	5
g	Address range (byte)	IB, QB, PIB, PQB MB, LB, DBB, DIB
h	Address range (word)	IW, QW, PIW, PQW, MW, LW, DBW, DIW
i	Address range (double word)	ID, QD, PID, PAD MD, LD, DBD, DID
r	Block number	10
AC	Range of address memory cell	
RE	Range error (invalid range)	

Registers

5

ACCU1 and ACCU2 (32 bit)

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

The accumulators are 32 bit long.

Table 5- 1 Designations:

Accumulator	Bit
ACCUx (x = 1 to 2)	Bit 0 to 31
ACCUx-L	Bit 0 to 15
ACCUx-H	Bit 16 to 31
ACCUx-LL	Bit 0 to 7
ACCUx-LH	Bit 8 to 15
ACCUx-HL	Bit 16 to 23
ACCUx-HH	Bit 24 to 31

Address Registers AR1 and AR2 (32 bit)

The address registers contain the areainternal or areacrossing addresses for instructions using indirect addressing. The address registers are 32 bit long.

The areainternal and/or areacrossing addresses have the following syntax:

- Areainternal address:

  00000000 00000bbb bbbbbbbb bbbbbxxx

- Areacrossing address:

  10000yyy 00000bbb bbbbbbbb bbbbbxxx

Legend for structure of addresses:

- b: Byte address
- x: Bit number
- y: Area identifier (see section: Examples of Addressing (Page 21))

Status Word

6

Status word (16 bit)

The status word bits are evaluated or set by the instructions.
The status word is 16 bit long.

Bit	Assignment	Description
0	/FC [1) 2)]	First check, bit cannot be written and evaluated in the user program because it is not updated at program runtime.
1	RLO	Result of logic operation
2	STA [1) 2)]	Status, bit cannot be written and evaluated in the user program because it is not updated at program runtime.
3	OR [1) 2)]	Or, bit cannot be written and evaluated in the user program because it is not updated at program runtime.
4	OS	Stored overflow
5	OV	Overflow
6	CC0	Result display
7	CC1	Result display
8	BR	Binary result
9 to 15	Unassigned	-

1) In the U-Stack display, the value "0" is always output.

2) In the display to STATUS block and breakpoint, the bit is displayed/refreshed correctly.

Addressing

7

7.1 Address types

Table 7- 1 The following address types are used:

		1. access								2. access							
	Commands	I	Q	M	P	L	DB	DI	V	I	Q	M	P	L	DB	DI	V
	A, AN, O, ON, X, XN, =, R, S, FP, FN -																
Direct	c 0.0	–	–	–	–	–	–	–	–	c	c	c	–	c	c	c	–
Memory indirect	c [AC D 0]	–	–	AC	–	AC	AC	AC	–	c	c	c	–	c	c	c	–
Memory indirect via block parameter	[#par]	–	–	–	–	–	–	–	–	c	c	c	RE	RE	c	c	c
Register indirect, area-internal	c[AR1, P#..] c[AR2, P#..]	–	–	–	–	–	–	–	–	c	c	c	–	c	c	c	–
Register indirect, area-crossing	[AR1, P#..] [AR2, P#..]	–	–	–	–	–	–	–	–	c	c	c	RE	c	c	c	c
	L, T -																
Direct	cB 0. cW 0. cD 0	–	–	–	–	–	–	–	–	c	c	c	c	c	c	c	–
Memory indirect	cB[AC D 0] cW[AC D 0] cD]AC D 0]	–	–	AC	–	AC	AC	AC	–	c	c	c	c	c	c	c	–
Memory indirect via block parameter	Bpar, Wpar, Dpar	–	–	–	–	–	–	–	–	c	c	c	c	RE	c	c	c
Register indirect, area-internal	cB[AR1, P#..] cW[AR1, P#..] cD[AR1, P#..] cB[AR2, P#..] cW[AR2, P#..] cD[AR2, P#..]	–	–	–	–	–	–	–	–	c	c	c	c	c	c	c	–
Register indirect, area-crossing	B[AR1, P#..] W[AR1, P#..] D[AR1, P#..] B[AR2, P#..] W[AR2, P#..] D[AR2, P#..]	–	–	–	–	–	–	–	–	c	c	c	c	c	c	c	c

		1. access								2. access							
	Commands	I	Q	M	P	L	DB	DI	V	I	Q	M	P	L	DB	DI	V
	SP, SE, SD, SS, SF, R, FR, L, LC, A, AN, O, ON, X, XN -																
Direct	T 0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Memory indirect	T[AC W 0]	–	–	AC	–	AC	AC	AC	–	–	–	–	–	–	–	–	–
Memory indirect via block parameter	#Tpar	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	S, CU, CD, R, FR, L, LC, A, AN, O, ON, X, XN -																
Direct	C 0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Memory indirect	C[AC W 0]	–	–	AC	–	AC	AC	AC	–	–	–	–	–	–	–	–	–
Memory indirect via block parameter	#Zpar	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	UC, CC -																
Direct	FB 0. FC 0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Memory indirect	FB[AC W 0], FC[AC W 0]	–	–	AC	–	AC	AC	AC	–	–	–	–	–	–	–	–	–
Memory indirect via block parameter	FBpar, #FCpar	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	OPN -																
Direct	DB 0, DI 0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Memory indirect	DB[AC W 0], DI[AC W 0]	–	–	AC	–	AC	AC	AC	–	–	–	–	–	–	–	–	–
Memory indirect via block parameter	DBpar, #FCpar [1)]	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

[1)] The STL syntax prohibits opening the 2nd data block as block parameter.

Definition of abbreviations

- c= address range (bit)
- AC= range of address memory cell;
- RE= Range error (invalid range)

See also

Abbreviations (Page 13)

Examples of addressing (Page 21)

7.2 Examples of addressing

Examples of addressing	Description
Immediate addressing	
L +27	Load 16bit integer constant "27" into ACCU1
L L#–1	Load 32bit integer constant "-1" into ACCU1
L 2#1010101010101010	Load binary constant into ACCU1
L DW#16#A0F0BCFD	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 constant as 2 byte
L B#(100,12,50,8)	Load constant as 4 byte
L P#10.0	Load areainternal pointer into ACCU1
L P#E20.6	Load areacrossing 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
Direct Addressing	
A I 0.0	ANDing of input bit 0.0
L IB1	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
Indirect Addressing of Timers/Counters	
SP T [LW 8]	Start timer; the timer number is in local data word 8
CU C [LW 10]	Start counter; the counter number is in local data word 10
AreaInternal MemoryIndirect Addressing	
A I [LD 12]	AND operation: The address of the input is in local data double word 12 as pointer **Example:** L P#22.2 T LD 12 A I [LD 12]
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 double word 12 as pointer

Examples of Addressing	Description			
Area-Internal Register-Indirect Addressing				
A I [AR1,P#12.2]	AND operation: The address of the input is calculated from the “pointer value in address register 1+ pointer P#12.2"			
Area-Crossing Register-Indirect Addressing [1]				
	For areacrossing registerindirect addressing, bit 24 to 26 of the address must also contain an area identifier. The address is in the address register.			
	Area ID	**Coding binary**	**Coding hexadecimal**	**Area**
	P	1000 0**000**	80	I/O area
	I	1000 0**001**	81	Input area
	Q	1000 0**010**	82	Output area
	M	1000 0**011**	83	Bit memory area
	DB	1000 0**100**	84	Data area
	DI	1000 0**101**	85	Instance data area
	L	1000 0**110**	86	Local data area
	VL	1000 0**111**	87	Predecessor local data area (access to local data of invoking block)
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 is calculated from the "pointer value in address register 1+ pointer P#32.3"			
Addressing Via Parameters				
A Parameter	Addressing via parameters			

[1] Logic Instructions with timers and counters (Page 28)

Examples of how to calculate the pointer

8

Example for sum of bit addresses ≤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)

Instruction list

9

This chapter contains the complete list of S7-300 instructions. The descriptions have been kept as concise as possible.

Note

Execution times

For indirect addressing and special addresses, you have to also add to the execution times a time for loading of the address or the respective address.

See also:

- Examples of addressing (Page 21)
- Address types (Page 19)
- Execution Time (Page 59)

Additional information

Detailed descriptions of the function are included in the STEP 7 reference manuals.

See also

Load Instructions for Timers and Counters (Page 35)

9.1 Logic instructions

9.1.1 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	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
A	1)	AND	1/2	0,10	0,07	0,06	0,05	0,03	0,004	0,06	0,05
AN	1)	AND NOT									

Status word for: A, AN	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	Yes	-	Yes	Yes
Instruction affects:	-	-	-	-	-	Yes	Yes	Yes	1

O	1)	OR	1/2	0,10	0,07	0,06	0,05	0,03	0,004	0,06	0,05
ON	1)	OR NOT									
X	1)	EXCLUSIVE OR									
XN	1)	EXCLUSIVE OR NOT									

Status word for: O, ON, X, XN	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	Yes	Yes
Instruction affects:	-	-	-	-	-	0	Yes	Yes	1

1) for valid addresses and parameter ranges see Address types (Page 19), Logic Instructions with Timers and Counters (Page 28)

9.1.2 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.

The listed parentheses also apply to the "right parenthesis" instructions.

Instruction	Description	Length in words	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
A(	AND left parenthesis	1	0.28	0.18	0.15	0.12	0.05	0.013	0.15	0.12
AN(	AND NOT left parenthesis									
O(	OR left parenthesis									
ON(	OR NOT left parenthesis									
X(	EXCLUSIVE OR left parenthesis									
XN(	EXCLUSIVE OR NOT left parenthesis									

Status word for: A(, AN(, O(, ON(, X(, XN(	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	Yes	-	-	-	-	Yes	-	Yes	Yes
Instruction affects:	-	-	-	-	-	0	1	-	0

)	Right parenthesis, popping an entry off the nesting stack, gating the RLO with the current RLO in the processor	1	0.28	0.18	0.15	0.12	0.05	0.013	0.15	0.12

Status word for:)	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	Yes	-	-	-	-	Yes	1	Yes	1

O	ORing of AND operations according to the rule: **AND before OR**	1	0.08	0.06	0.05	0.04	0.02	0.008	0.05	0.04

Status word for: O	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	Yes	-	Yes	Yes
Instruction affects:	-	-	-	-	-	Yes	1	-	Yes

9.1.3 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	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
A	T f [1]	AND Timer	1/2	0,60	0,30	0,26	0,23	0,13	0,02	0,26	0,23
	C f [1]	AND Counter		0,30	0,15	0,12	0,10	0,05	0,01	0,12	0,10
AN	T f [1]	AND NOT Timer		0,60	0,30	0,26	0,23	0,13	0,02	0,26	0,23
	C f [1]	AND NOT Counter		0,30	0,15	0,12	0,10	0,05	0,01	0,12	0,10

Status word for: A, AN	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	Yes	-	Yes	Yes
Instruction affects:	-	-	-	-	-	Yes	Yes	Yes	1

O	T f [1]	OR Timer	1/2	0,60	0,30	0,26	0,23	0,13	0,02	0,26	0,23
	C f [1]	OR Counter		0,30	0,15	0,12	0,10	0,05	0,01	0,12	0,10
ON	T f [1]	OR NOT Timer		0,60	0,30	0,26	0,23	0,13	0,02	0,26	0,23
	C f [1]	OR NOT Counter		0,30	0,15	0,12	0,10	0,05	0,01	0,12	0,10
X	T f [1]	EXCLUSIVE OR Timer		0,60	0,30	0,26	0,23	0,13	0,02	0,26	0,23
	C f [1]	EXCLUSIVE OR Counter		0,30	0,15	0,12	0,10	0,05	0,01	0,12	0,10
XN	T f [1]	EXCLUSIVE OR NOT Timer		0,60	0,30	0,26	0,23	0,13	0,02	0,26	0,23
	C f [1]	EXCLUSIVE OR NOT Counter		0,30	0,15	0,12	0,10	0,05	0,01	0,12	0,10

Status word for: O, ON, X, XN	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	Yes	Yes
Instruction affects:	-	-	-	-	-	0	Yes	Yes	1

[1] for valid parameter ranges, see Address types (Page 19)

9.1.4 Logic Instructions 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.

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
A		AND									
O		OR									
X		EXCLUSIVE OR									
	== 0	Result = 0 (CC1 = 0) and (CC0 = 0)	1	0,30	0,11	0,09	0,08	0,03	0,01	0,09	0,08
	> 0	Result > 0 (CC1 = 1) and (CC0 = 0)									
	< 0	Result < 0 (CC1 = 0) and (CC0 = 1)									
	<> 0	Result ≠ 0 ((CC1 = 0) and (CC0 = 1) or (CC1 = 1) and (CC0 = 0))									
	<= 0	Result ≤ 0 ((CC1 = 0) and (CC0 = 1) or (CC1 = 0) and (CC0 = 0))									
	>= 0	Result ≥ 0 ((CC1 = 1) and (CC0 = 0) or (CC1 = 0) and (CC0 = 0))									
	AO	AND unordered/ invalid (CC1 = 1) and (CC0 = 1)									
	OS	AND OS = 1									
	BR	AND BR = 1									
	OV	AND OV = 1									

Status word for: A, O, X	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	Yes	-	Yes	Yes
Instruction affects:	-	-	-	-	-	Yes	Yes	Yes	1

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
AN		AND NOT									
ON		OR NOT									
XN		EXCLUSIVE OR NOT									
	== 0	Result = 0 (CC1 = 0) and (CC0 = 0)	1	0,30	0,11	0,09	0,08	0,03	0,01	0,09	0,08
	> 0	Result > 0 (CC1 = 1) and (CC0 = 0)									
	< 0	Result < 0 (CC1 = 0) and (CC0 = 1)									
	<> 0	Result00 ((CC1 = 0) and (CC0 = 1) or (CC1 = 1) and (CC0 = 0))									
	<= 0	Result ≤ 0 ((CC1 = 0) and (CC0 = 1) or (CC1 = 0) and (CC0 = 0))									
	>= 0	Result ≥ 0 ((CC1 = 1) and (CC0 = 0) or (CC1 = 0) and (CC0 = 0))									
	AO	AND unordered/ invalid (CC1 = 1) and (CC0 = 1)									
	OS	AND OS = 1									
	BR	AND BR = 1									
	OV	AND OV = 1									

Status word for: AN, ON, XN	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	Yes	-	Yes	Yes
Instruction affects:	-	-	-	-	-	Yes	Yes	Yes	1

9.2 EdgeTriggered Instructions

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

Auxiliary edge bit memory is the bit addressed in the instruction.

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
FP	1)	Detecting the positive edge in the RLO.	2	0,26	0,19	0,17	0,15	0,08	0,015	0,17	0,15
FN	1)	Detecting the negative edge in the RLO.									

Status word for: FP, FN	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	-	-	-	-	-	0	Yes	Yes	1

1) for all valid addresses and parameter ranges see Address types (Page 19)

9.3 Setting/Resetting Bit Addresses

Assigning the value "1" or "0" or the RLO of the addressed instruction.

The instructions can be MCR-dependent.

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
S	1)	Set input/output/bit memory/local data bit/data bit/instance data bit to "1"	2	0,14	0,10	0,09	0,08	0,04	0,01	0,09	0,08
R	1)	Set input/ output/bit memory/local data bit/data bit/instance data bit to "0"									
=	1)	Assign RLO to input/output/bit memory/local data bit/data bit/instance data bit									

Status word for: S, R, =	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	-	-	-	-	-	0	Yes	-	0

1) for all valid addresses and parameter ranges see Address types (Page 19)

9.4 Instructions Directly Affecting the RLO

The following instructions have a direct effect on the RLO.

Instruction	Description	Length in words	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
CLR	Set RLO to "0"	2	0,07	0,06	0,05	0,04	0,02	0,004	0,05	0,04

Status word for: CLR	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	0	0	0	0

SET	Set RLO to "1"	2	0,07	0,06	0,05	0,04	0,02	0,004	0,05	0,04

Status word for: SET	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	0	1	1	0

NOT	Negate RLO	2	0,07	0,06	0,05	0,04	0,02	0,004	0,05	0,04

Status word for: NOT	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	Yes	-	Yes	-
Instruction affects:	-	-	-	-	-	-	1	Yes	-

SAVE	Retain the RLO in the Bit BR	2	0,08	0,06	0,05	0,04	0,02	0,004	0,05	0,04

Status word for: SAVE	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	Yes	-	-	-	-	-	-	-	-

9.5 Timer Instructions

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

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
SP	T f [1)]	Start timer as pulse on edge change from "0" to "1"	4/6	1.20	0.79	0.63	0.48	0.19	0.075	0.63	0.48
SE	T f [1)]	Start timer as extended pulse on edge change from "0" to "1"		1.11	0.73	0.57	0.46	0.18	0.065	0.57	0.46
SD	T f [1)]	Start timer as ON delay on edge change from "0" to "1"		1.31	0.90	0.69	0.53	0.21	0.080	0.69	0.53
SS	T f [1)]	Start timer as retentive ON delay on edge change from "0" to "1"		1.25	0.84	0.66	0.51	0.20	0.070	0.66	0.51
SF	T f [1)]	Start timer as OFF delay on edge change from "1" to "0"		1.37	0.84	0.72	0.55	0.21	0.080	0.72	0.55
FR	T f [1)]	Enable timer for restarting on edge change from "0" to "1" (reset edge bit memory for starting timer)		1.28	0.83	0.67	0.52	0.20	0.060	0.67	0.52
R	T f [1)]	Reset timer		1.51	0.98	0.79	0.61	0.24	0.115	0.79	0.61

Status word for: SP, SE, SD, SS, SF, FR, R	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	-	-	-	-	-	0	-	-	0

[1)] for valid parameter ranges, see Address types (Page 19)

9.6 Counter Instructions

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

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
S	C f [1)]	Presetting of counter on edge change from "0" to "1"	4/6	1.76	1.20	0.92	0.71	0.28	0.090	0.92	0.71
R	C f [1)]	Reset counter to “0" on edge change from "0" to "1"		1.15	0.73	0.60	0.46	0.17	0.050	0.60	0.46
CU	C f [1)]	Increment counter by 1 on edge change from "0" to "1"		1.22	0.79	0.64	0.49	0.20	0.055	0.64	0.49
CD	C f [1)]	Decrement counter by 1 on edge change from "0" to "1"		1.31	0.84	0.69	0.53	0.20	0.060	0.69	0.53
FR	C f [1)]	Enable counter on edge change from "0" to "1" (reset edge bit memory for up and down counting)	2	1.19	0.76	0.62	0.48	0.19	0.055	0.62	0.48

Status word for: S, R, CU, CD, FR	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	-	-	-	-	-	0	-	-	0

[1)] for valid parameter ranges, see Address types (Page 19)

9.7 Load Instructions

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

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
L		Load ...									
	B [1]	Byte	1/2	0,24	0,15	0,12	0,09	0,03	0,007	0,12	0,09
	W [1]	Word		0,28	0,18	0,14	0,11	0,04	0,010	0,14	0,11
	DW [1]	Double word		0,32	0,20	0,16	0,12	0,04	0,015	0,16	0,12
	k8 [2]	8bit constant in ACCU1-LL	1	0,24	0,15	0,12	0,09	0,03	0,007	0,12	0,09
	k16 [2]	16bit constant in ACCU1-L	2								
	k32 [2]	32bit constant in ACCU1	3								

[1] for all valid addresses and parameter ranges, see Address types (Page 19)

[2] valid for all Constants (Page 11)

9.8 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 condition code bits are not affected.

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
L	T f [1]	Load time value	1/2	1,70	1,30	0,80	0,80	0,34	0,175	0,80	0,80
LC	T f [1]	Load time value BCD coded		2,71	1,73	1,41	1,09	0,43	0,280	1,41	1,09
L	C f [1]	Load count value		1,11	0,70	0,58	0,45	0,14	0,050	0,58	0,45
LC	C f [1]	Load count value BCD coded		1,71	1,10	0,89	0,69	0,27	0,155	0,89	0,69

[1] for valid parameter ranges, see Address types (Page 19)

9.9 Transfer Instructions

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

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
T		Transfer contents of...									
	B [1]	ACCU1-LL input byte	1/2	0.20	0.13	0.10	0.08	0.03	0.007	0.10	0.08
	W [1]	ACCU1-L input word		0.24	0.15	0.12	0.09	0.03	0.008	0.12	0.09
	DW [1]	ACCU1 input double word		0.28	0.18	0.14	0.11	0.04	0.010	0.14	0.11

[1] for all valid addresses and parameter ranges, see Address types (Page 19)

9.10 Load and Transfer Instructions for Address Registers

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

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
LAR1		Load contents from ... ... into AR1									
	-	ACCU1 ...	1	0,20	0,15	0,10	0,10	0,03	0,01	0,10	0,10
	AR2	Address register 2...	1	0,20	0,15	0,10	0,10	0,03	0,01	0,10	0,10
	DBD a	Data double word ...	2	0,51	0,34	0,27	0,21	0,08	0,02	0,27	0,21
	DID a	Instance data double word ...	2	0,98	0,61	0,51	0,40	0,15	0,05	0,51	0,40
	m	32bit constant as pointer ...	3	0,30	0,18	0,15	0,12	0,04	0,01	0,15	0,12
	LD a	Local data double word ...	2	0,51	0,34	0,27	0,21	0,08	0,02	0,27	0,21
	MD a	Bit memory double word ...	2	0,51	0,34	0,27	0,21	0,08	0,02	0,27	0,21

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
LAR2		Load contents from ... ... into AR2									
	-	ACCU1 ...	1	0,20	0,15	0,10	0,10	0,03	0,01	0,10	0,10
	DBD a	Data double word ...	2	0,51	0,34	0,27	0,21	0,08	0,02	0,27	0,21
	DID a	Instance data double word ...	2	0,98	0,61	0,51	0,40	0,15	0,05	0,51	0,40
	m	32bit constant as pointer ...	3	0,30	0,18	0,15	0,12	0,04	0,01	0,15	0,12
	LD a	Local data double word ...	2	0,51	0,34	0,27	0,21	0,08	0,02	0,27	0,21
	MD a	Bit memory double word ...	2	0,51	0,34	0,27	0,21	0,08	0,02	0,27	0,21
TAR1		Transfer contents of AR1 to									
	-	ACCU1	1	0,30	0,19	0,16	0,13	0,04	0,02	0,16	0,13
	AR2	Address register 2	1	0,20	0,15	0,10	0,10	0,03	0,01	0,10	0,10
	DBD a	Data double word	2	0,39	0,26	0,21	0,17	0,06	0,02	0,21	0,17
	DID a	Instance data double word	2	0,93	0,59	0,49	0,38	0,14	0,045	0,49	0,38
	LD a	Local data double word	2	0,39	0,26	0,21	0,17	0,06	0,02	0,21	0,17
	MD a	Bit memory double word ...	2	0,39	0,26	0,21	0,17	0,06	0,02	0,21	0,17
TAR2		Transfer contents of AR2 to									
	-	ACCU1	1	0,30	0,19	0,16	0,13	0,04	0,02	0,16	0,13
	DBD a	Data double word	2	0,39	0,26	0,21	0,17	0,06	0,02	0,21	0,17
	DID a	Instance data double word	2	0,93	0,59	0,49	0,38	0,14	0,045	0,49	0,38
	LD a	Local data double word	2	0,39	0,26	0,21	0,17	0,06	0,02	0,21	0,17
	MD a	Bit memory double word	2	0,39	0,26	0,21	0,17	0,06	0,02	0,21	0,17
TAR		Exchange the contents of AR1 and AR2	1	0,28	0,19	0,16	0,13	0,04	0,01	0,16	0,13

9.11 Load and Transfer Instructions for the Status Word

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
L	STW	Load status word [1] into ACCU1	1	0,63	0,43	0,33	0,26	0,09	0,025	0,33	0,26

Status word for: L STW	BR	CC1	CC0	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 [1]	1	0,58	0,38	0,31	0,24	0,09	0,020	0,31	0,24

Status word for: T STW	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	Yes	Yes	Yes	Yes	Yes	-	-	Yes	-

[1] Structure of the status word, see: Status word (Page 17)

9.12 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	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
L	DBNO	Load number of data block	1	0,27	0,18	0,15	0,12	0,04	0,01	0,15	0,12
L	DINO	Load number of instance data block									
L	DBLG	Load length of data block into byte	1	0,34	0,22	0,19	0,14	0,04	0,01	0,19	0,14
L	DILG	Load length of instance data block into byte									

9.13 Word Logic Instructions with the Contents of Accumulator 1

Linking the contents of ACCU1 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 or ACCU1-L.

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
AW		AND ACCU2-L	1	0.33	0.22	0.18	0.14	0.05	0.014	0.18	0.14
OW		OR ACCU2-L									
XOW		EXCLUSIVE OR ACCU2-L									
AW	k16	AND 16-bit constant	2	0.33	0.22	0.18	0.14	0.05	0.014	0.18	0.14
OW	k16	OR 16-bit constant									
XOW	k16	EXCLUSIVE OR 16-bit constant									

Status word for: AW, OW, XOW	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	0	0	-	-	-	-	-

AD		AND ACCU2	1	0.28	0.19	0.16	0.13	0.05	0.014	0.16	0.13
OD		OR ACCU2									
XOD		EXCLUSIVE OR ACCU2									
AD	k32	AND 32-bit constant	3	0.28	0.19	0.16	0.13	0.05	0.014	0.16	0.13
OD	k32	OR 32-bit constant									
XOD	k32	EXCLUSIVE OR 32-bit constant									

Status word for: AD, OD, XOD	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	0	0	-	-	-	-	-

9.14 Fixedpoint arithmetic (16/32-bit) / Floatingpoint arithmetic (32-bit)

Mathematical functions of two 16-/32-bit numbers. The result is in ACCU1 or ACCU1-L.

I = Integer → 16 bit,

D = Integer → 32 bit,

R = Real number → 32 bit

Instruction	Description	Length in words	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
Add 2 integers or real numbers		1								
+I	(ACCU1-L) = (ACCU1-L) + (ACCU2-L)		0.25	0.17	0.13	0.10	0.04	0.010	0.13	0.10
+D	(ACCU1) = (ACCU2) + (ACCU1)		0.22	0.15	0.12	0.09	0.03	0.010	0.12	0.09
+R	(ACCU1) = (ACCU2) + (ACCU1)		1.10	0.72	0.58	0.44	0.16	0.040	0.58	0.44
Subtract 2 integers or real numbers										
–I	(ACCU1-L) = (ACCU2-L) - (ACCU1-L)		0.25	0.17	0.13	0.10	0.04	0.010	0.13	0.10
–D	(ACCU1) = (ACCU2) - (ACCU1)		0.22	0.15	0.12	0.09	0.03	0.010	0.12	0.09
–R	(ACCU1) = (ACCU2) - (ACCU1)		1.10	0.72	0.58	0.44	0.16	0.040	0.58	0.44

Status word for: +I, +D, +R, -I, -D, -R	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	Yes	Yes	Yes	-	-	-	-

Instruction	Description	Length in words	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
Multiply 2 integers or real numbers		1								
*I	(ACCU1) = (ACCU2-L) * (ACCU1-L)		0.28	0.18	0.15	0.12	0.04	0.010	0.15	0.12
*D	(ACCU1) = (ACCU2) * (ACCU1)		0.21	0.15	0.12	0.09	0.03	0.008	0.12	0.09
*R	(ACCU1) = (ACCU2) * (ACCU1)		1.11	0.71	0.58	0.44	0.16	0.040	0.58	0.44
Divide 2 integers or real numbers										
/I	(ACCU1-L) = (ACCU2-L): (ACCU1-L) → The remainder of the division is in ACCU1-H		0.52	0.34	0.27	0.22	0.08	0.060	0.27	0.22
/D	(ACCU1) = (ACCU2): (ACCU1)		0.51	0.33	0.27	0.21	0.08	0.050	0.27	0.21
/R	(ACCU1) = (ACCU2): (ACCU1)		4.85	3.00	2.52	1.89	0.25	0.060	2.52	1.89
MOD	Divide 2 integers (32 bit) and load the remainder of the division in ACCU1: (ACCU1) = Remainder of [(ACCU2): (ACCU1)]		0.43	0.29	0.23	0.18	0.07	0.060	0.23	0.18

Status word for: *I, *D, *R, /I, /D, /R, MOD	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	Yes	Yes	Yes	-	-	-	-

NEGR	Negate the real number in ACCU1	1	0.20	0.14	0.12	0.09	0.03	0.005	0.12	0.09
ABS	Form the absolute value of the real number in ACCU1		0.20	0.14	0.12	0.09	0.03	0.005	0.12	0.09

Status word for: NEGR, ABS	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

9.15 Square root, square (32 bit)/logarithm function (32 bit)

The result of the instruction / logarithm function is in ACCU1. The instructions can be interrupted by interrupts.

Instruction	Description	Length in words	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
SQRT	Calculate the square root of a real number in ACCU1	1	8,14	5,16	4,22	3,24	1,26	0,475	4,22	3,24
SQR	Form the square of a real number in ACCU1		1,15	0,73	0,59	0,46	0,18	0,040	0,59	0,46
LN	Form the natural logarithm of a real number in ACCU1	1	7,34	4,65	3,80	2,92	1,20	0,455	3,80	2,92
EXP	Calculate the exponential value of a real number in ACCU1 to the base e (= 2.71828)		9,13	5,80	4,73	3,63	1,50	0,525	4,73	3,63

Status word for: SQRT, SQR, LN, EXP	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	Yes	Yes	Yes	-	-	-	-

9.16 Trigonometrical functions (32 bits)

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

Instruction	Description	Length in words	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
SIN[1)]	Calculate the sine of a real number	1	7,52	4,77	3,90	3,00	1,20	0,530	3,90	3,00
ASIN[2)]	Calculate the arcsine of a real number		15,80	10,23	8,40	6,44	1,30	0,480	8,40	6,44
COS[1)]	Calculate the cosine of a real number		9,19	5,78	4,75	3,65	1,50	0,530	4,75	3,65
ACOS[2)]	Calculate the arccosine of a real number		7,21	4,56	3,73	2,87	1,20	0,450	3,73	2,87
TAN[1)]	Calculate the tangent of a real number		10,92	6,93	5,67	4,35	1,80	0,620	5,67	4,35
ATAN[2)]	Calculate the arctangent of a real number		7,91	5,10	4,10	3,14	1,30	0,485	4,10	3,14

Status word for: SIN, ASIN, COS, ACOS, TAN, ATAN	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	Yes	Yes	Yes	-	-	-	-

[1)] Specify the angle in radians; the angle must be given as a floating point value in ACCU1.

[2)] The result is an angle in radians.

9.17 Adding Constants

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

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
+	i8	Add an 8bit integer constant	1	0,20	0,14	0,10	0,10	0,05	0,01	0,10	0,10
+	i16	Add an 16bit integer constant	2	0,20	0,14	0,10	0,10	0,05	0,01	0,10	0,10
+	i32	Add an 32bit integer constant	3	0,20	0,14	0,10	0,10	0,05	0,01	0,10	0,10

9.18 Adding Using Address Registers

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

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
+AR1	-	Add the contents of ACCU1-L to AR1	1	0.20	0.16	0.10	0.10	0.07	0.01	0.10	0.10
+AR1	m	Add a pointer constant to the contents of AR1	2	0.40	0.20	0.15	0.12	0.07	0.01	0.15	0.12
+AR2	-	Add the contents of ACCU1-L to AR2	1	0.20	0.16	0.10	0.10	0.07	0.01	0.10	0.10
+AR2	m	Add pointer constant to the contents of AR2	2	0.40	0.20	0.15	0.12	0.07	0.01	0.15	0.12

9.19 Comparison instructions with integers (16/32 bit) or with 32-bit real numbers

Comparison of integers (16 bit) in ACCU1-L and ACCU2-L. RLO = 1, if the condition is satisfied.

Comparison of integers (32 bit) in ACCU1 and ACCU2. RLO = 1, if the condition is satisfied.

Comparison of 32bit real numbers in ACCU1 and ACCU2. RLO = 1, if the condition is satisfied.

Instruction	Description	Length in words	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
==I	ACCU2-L = ACCU1-L	1	0,48	0,31	0,26	0,20	0,07	0,028	0,26	0,20
==D	ACCU2 = ACCU1		0,43	0,28	0,23	0,18	0,06	0,023	0,23	0,18
==R	ACCU2 = ACCU1		1,67	1,07	0,87	0,67	0,27	0,046	0,87	0,67
<>I	ACCU2-L ≠ ACCU1-L		0,48	0,31	0,26	0,20	0,07	0,028	0,26	0,20
<>D	ACCU ≠ ACCU1		0,43	0,28	0,23	0,18	0,06	0,023	0,23	0,18
<>R	ACCU ≠ ACCU1		1,67	1,07	0,87	0,67	0,27	0,046	0,87	0,67
<I	ACCU2-L < ACCU1-L		0,48	0,31	0,26	0,20	0,07	0,028	0,26	0,20
<D	ACCU2 < ACCU1		0,43	0,28	0,23	0,18	0,06	0,023	0,23	0,18
<R	ACCU2 < ACCU1		1,67	1,07	0,87	0,67	0,27	0,046	0,87	0,67
<=I	ACCU2-L ≤ ACCU1-L		0,48	0,31	0,26	0,20	0,07	0,028	0,26	0,20
<=D	ACCU2 ≤ ACCU1		0,43	0,28	0,23	0,18	0,06	0,023	0,23	0,18
<=R	ACCU2 ≤ ACCU1		1,67	1,07	0,87	0,67	0,27	0,046	0,87	0,67
>I	ACCU2-L > ACCU1-L		0,48	0,31	0,26	0,20	0,07	0,028	0,26	0,20
>D	ACCU2 > ACCU1		0,43	0,28	0,23	0,18	0,06	0,023	0,23	0,18
>R	ACCU2 > ACCU1		1,67	1,07	0,87	0,67	0,27	0,046	0,87	0,67
>=I	ACCU2-L ≥ ACCU1-L		0,48	0,31	0,26	0,20	0,07	0,028	0,26	0,20
>=D	ACCU2 ≥ ACCU1		0,43	0,28	0,23	0,18	0,06	0,023	0,23	0,18
>=R	ACCU2 ≥ ACCU1		1,67	1,07	0,87	0,67	0,27	0,046	0,87	0,67

Status word for: == I, ==D, <>I, <>D, <I, <D, <=I, <=D, >I, >D, >=I, >=D	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	Yes	0	-	0	Yes	Yes	1
Status word for: ==R, <>R, <R, <=R, >R, >=R	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	Yes	Yes	Yes	0	Yes	Yes	1

9.20 Shift Instructions

Shift the contents of ACCU1 or ACCU1-L to the left or right by the number of specified digits. If no address is specified, shift by the number of digits to ACCU2-LL. Any positions that become free are filled with zeros or with the sign. The last bit shifted is in condition code bit CC1.

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
SLW	–	Shift the contents of ACCU1-L to the left.	1	0.51	0.34	0.27	0.21	0.08	0.019	0.27	0.21
	0 ... 15	Positions that become free are filled with zeros.									
SLD	–	Shift the contents of ACCU1 to the left.		0.46	0.30	0.24	0.19	0.07	0.019	0.24	0.19
	0 ... 32	Positions that become free are filled with zeros.									
SRW	–	Shift the contents of ACCU1-L to the right.		0.51	0.24	0.27	0.21	0.08	0.019	0.27	0.21
	0 ... 15	Positions that become free are filled with zeros.									
SRD	–	Shift the contents of ACCU1 to the right.		0.46	0.30	0.24	0.19	0.07	0.019	0.24	0.19
	0 ... 32	Positions that become free are filled with zeros.									
SSI	–	Shift the contents of ACCU1–L with sign to the right.		0.60	0.36	0.30	0.23	0.09	0.019	0.30	0.23
	0 ... 15	Positions that become free are filled with the sign (bit 15).									
SSD	–	Shift the contents of ACCU1 with sign to the right.		0.46	0.31	0.27	0.19	0.08	0.019	0.27	0.19
	0 ... 32	Positions that become free are filled with the sign (bit 31).									

Status word for: SLW, SLD, SRW, SRD, SSI, SSD	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	Yes	Yes	-	-	-	-	-

9.21 Rotate Instructions

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

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
RLD	- 0 ... 32	Rotate the contents of ACCU1 to the left	1	0.45	0.29	0.24	0.19	0.07	0.019	0.24	0.19
RRD	- 0 ... 32	Rotate the contents of ACCU1 to the right		0.45	0.29	0.24	0.19	0.07	0.019	0.24	0.19

Status word for: RLD, RRD	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	Yes	Yes	-	-	-	-	-

RLDA	-	Rotate the contents of ACCU1 one bit position to the left via condition code bit CC1	1	0.30	0.20	0.16	0.13	0.05	0.012	0.16	0.13
RRDA	-	Rotate the contents of ACCU1 one bit position to the left via condition code bit CC1		0.30	0.20	0.16	0.13	0.05	0.015	0.16	0.13

Status word for: RLDA, RRDA	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	0	0	-	-	-	-	-

9.22 ACCU-transfer instructions, incrementing and decrementing

The status word is not affected.

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
TAW	-	Reverse the order of the bytes in ACCU1-L; LL, LH becomes LH, LL.	1	0.20	0.13	0.10	0.10	0.05	0.01	0.10	0.10
CAD	-	Reverse the order of the bytes in ACCU1. LL, LH, HL, HH becomes HH, HL, LH, LL.		0.40	0.24	0.20	0.16	0.06	0.01	0.20	0.16
TAK	-	Swap the contents of ACCU1 and ACCU2		0.25	0.17	0.14	0.11	0.04	0.01	0.14	0.11
PUSH	-	The contents of ACCU1 are transferred to ACCU2.		0.20	0.13	0.10	0.08	0.03	0.01	0.10	0.08
POP	-	The contents of ACCU2 are transferred to ACCU1.		0.20	0.14	0.10	0.08	0.03	0.01	0.10	0.08
INC	0 ... 255	Increment ACCU1-LL		0.20	0.14	0.10	0.10	0.05	0.01	0.10	0.10
DEC	0 ... 255	Decrement ACCU1-LL		0.20	0.14	0.10	0.10	0.05	0.01	0.10	0.10

9.23 Program Display and Null Operation Instructions

The status word is not affected.

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
BLD[1)]	0 ... 255	Program display instruction: Is treated by the CPU like a null operation instruction.	1	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
NOP[2)]	0 1	Null operation		0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00

[1)]The BLD instructions are generated and used by the programming device and should not be deleted, changed or added to.

[2)]The NOP1 instruction should not be used. If you require a NOP instruction, use NOP0.

9.24 Data Type Conversion Instructions

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

Instruction	Description	Length in words	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
BTI	Convert ACCU1 from BCD to integer (16 bit) (**B**CD **T**o **I**nteger)	1	0,73	0,46	0,39	0,30	0,11	0,040	0,39	0,30
BTD	Convert ACCU1 from BCD to integer (32 bit) (**B**CD **T**o **D**oubleinteger)		1,08	0,67	0,57	0,44	0,16	0,090	0,57	0,44
DTR	Convert ACCU1 from integer (32 bit) to real (32 bit) (**D**oubleint. **T**o **R**eal)		0,70	0,45	0,37	0,29	0,11	0,020	0,37	0,29
ITD	Convert ACCU1 from integer (16 bit) to integer (32 bit) (**I**nteger **T**o **D**oubleinteger)		0,21	0,14	0,10	0,09	0,03	0,008	0,10	0,09

Status word for: BTI, BTD, DTR, ITD	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

ITB	Convert ACCU1 from integer (16 bit) to BCD 0 to ±999 (**I**nteger **T**o **B**CD)	1	1,09	0,70	0,57	0,44	0,17	0,117	0,57	0,44
DTB	Convert ACCU1 from integer (32 bit) to BCD 0 to ±9 999 999 (**D**oubleinteger **T**o **B**CD)		2,98	1,90	1,54	1,19	0,47	0,315	1,54	1,19
RND	Convert a real number into a 32-bit integer.		4,82	3,06	2,49	1,92	0,15	0,025	2,49	1,92
RND-	Convert a real number into a 32-bit integer. The number is rounded to the next integer.		4,82	3,06	2,49	1,92	0,15	0,025	2,49	1,92
RND+	Convert a real number into a 32-bit integer. The number is rounded to the next integer.		4,82	3,06	2,49	1,92	0,15	0,025	2,49	1,92
TRUNC	Convert a real number into a 32-bit integer. The places after the decimal point are truncated.		4,82	3,06	2,49	1,92	0,15	0,025	2,49	1,92

Status word for: ITB, DTB, RND, RND-, RND+, TRUNC	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	Yes	Yes	-	-	-	-

9.25 Forming the Ones and Twos Complements

Instruction	Description	Length in words	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
INVI	Form the ones complement of ACCU1-L	1	0.13	0.10	0.08	0.07	0.04	0.010	0.08	0.07
INVD	Form the ones complement of ACCU1		0.11	0.09	0.07	0.06	0.03	0.005	0.07	0.06

Status word for: INVI, INVD	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

NEGI	Form the twos complement of ACCU1-L (integer)	1	0.16	0.12	0.10	0.08	0.05	0.010	0.10	0.08
NEGD	Form the twos complement of ACCU1 (double integer)		0.12	0.09	0.07	0.06	0.03	0.005	0.07	0.06

Status word for: NEGI, NEGD	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	Yes	Yes	Yes	-	-	-	-

9.26 Block Call Instructions

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
CALL	FB p, DB r	Unconditional call of an FB, with parameter transfer	1	5,10	3,25	2,65	2,05	0,78	0,35	2,65	2,05
CALL	SFB p, DB r	Unconditional call of an SFB, with parameter transfer	2	[1]							
CALL	FC p	Unconditional call of a function, with parameter transfer	1	4,87	3,15	2,59	2,03	0,83	0,35	2,59	2,03
CALL	SFC p	Unconditional call of an SFC, with parameter transfer	2	[1]							

Status word for: CALL	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	0	0	1	-	0

[1] in chapter:

- System Functions (SFC) (Page 71)
- System Function Blocks (SFB) (Page 79)

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
UC	FBq	Unconditional call of blocks without parameter transfer	1	3,97	2,53	2,06	1,59	0,62	0,30	2,06	1,59
	FCq			4,26	2,76	2,27	1,77	0,72	0,30	2,27	1,77
	Parameter	FB/FC call via parameter		4,26	2,76	2,27	1,77	0,72	0,30	2,27	1,77
CC	FBq	Conditional call of blocks without parameter transfer	1	3,97	2,53	2,06	1,59	0,62	0,30	2,06	1,59
	FCq			4,26	2,76	2,27	1,77	0,72	0,30	2,27	1,77
	Parameter	FB/FC call via parameter		4,26	2,76	2,27	1,77	0,72	0,30	2,27	1,77

Status word for: UC, CC	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	0	0	1	-	0

OPN [3]											
	DBp	Open data block	1/2 [2]	0,40	0,28	0,21	0,17	0,08	0,02	0,21	0,17
	DIp	Open instance data block	2	0,40	0,28	0,21	0,17	0,08	0,02	0,21	0,17
	Parameter	Open instance data block	2	0,40	0,28	0,21	0,17	0,08	0,02	0,21	0,17

Status word for: OPN	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

[2] For long block numbers (> 255)

[3] The CPUs offer high-performance support of symbolic programming. The fully qualified DB accesses (e.g. DB100.DBX 1.2) used here generally cause no additional runtimes. This applies also for the OPN DB command contained in the access.

9.27 Block End Instructions

Instruction	Description	Length in words	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
BE	End block	1	1,20	1,09	0,88	0,68	0,26	0,07	0,88	0,68
BEA	End block absolute		1,20	1,09	0,88	0,68	0,26	0,07	0,88	0,68

Status word for: BE, BEA	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	0	0	1	-	0

BEC	End block conditionally if RLO = "1"	1	1,20	1,09	0,88	0,68	0,26	0,07	0,88	0,68

Status word for: BEC	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	-	-	-	-	Yes	0	1	1	0

9.28 Exchange Data Blocks

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

Instruction	Description	Length in words	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
CDB	Exchange data blocks	1	0,20	0,15	0,10	0,10	0,10	0,05	0,10	0,10

9.29 Jump instructions

Jump depends on the condition.

- With 8bit addresses, the jump width is between (-128 and +127).
- In the case of 16bit addresses, the jump width lies between (-32768 and -129) or (+128 and +32767).

Note

Please note for S7-300 CPU programs that the jump instructions starting from a logic string or into a logic string are invalid.

Instructions that set /FC = 0 indicate the end of a logic string.
The beginning is the first logic instruction after the end of a logic string. Here the linear program sequence is relevant without taking into consideration the jump instructions.
Note that AND before OR also indicates the beginning of a new logic string.

Jump instructions into a different nesting level are also invalid.

Examples of jump instructions (Page 56)

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
JC	LABEL	Jump conditionally if RLO = "1"	1 [1] /2	0.39	0.26	0.21	0.16	0.10	0.01	0.21	0.16
JCN	LABEL	Jump conditionally if RLO = "0"	2	0.39	0.26	0.21	0.16	0.10	0.01	0.21	0.16

Status word for: JC, JCN	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	-	-	-	-	-	0	1	1	0

JCB	LABEL	Jump conditionally if RLO ="1"; Save the RLO in the BR bit	2	0.39	0.26	0.21	0.16	0.10	0.01	0.21	0.16
JNB	LABEL	Jump conditionally if RLO ="0"; Save the RLO in the BR bit	2	0.39	0.26	0.21	0.16	0.10	0.01	0.21	0.16

Status word for: JCB, JNB	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	Yes	-	-	-	-	0	1	1	0

[1] 1 word long for jump width of -128 to +127

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
JBI	LABEL	Jump conditionally if BR = "1"	2	0.39	0.26	0.21	0.16	0.10	0.01	0.21	0.16
JNBI	LABEL	Jump conditionally if BR = "0"	2	0.39	0.26	0.21	0.16	0.10	0.01	0.21	0.16

Status word for: JBI, JNBI	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	Yes	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	0	1	-	0

JO	LABEL	Jump conditionally on stored overflow (OS = "1")	1 [1] /2	0.39	0.26	0.21	0.16	0.10	0.01	0.21	0.16

Status word for: JO	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	Yes	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

JOS	LABEL	Jump conditionally on stored overflow (OS = "1")	2	0.39	0.26	0.21	0.16	0.10	0.01	0.21	0.16

Status word for: JOS	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	Yes	-	-	-	-
Instruction affects:	-	-	-	-	0	-	-	-	-

JII	LABEL	Jump conditionally if "invalid instruction" (CC1 = 1 and CC0 = 1)	2	0.39	0.26	0.21	0.16	0.10	0.01	0.21	0.16
JZ	LABEL	Jump conditionally result = 0 (CC1 = 0 and CC0 = 0)	1 [1] /2	0.39	0.26	0.21	0.16	0.10	0.01	0.21	0.16
JP	LABEL	Jump conditionally if result > 0 (CC1 = 1 and CC0 = 0)	1 [1] /2	0.39	0.26	0.21	0.16	0.10	0.01	0.21	0.16
JM	LABEL	Jump conditionally if result < 0 (CC1 = 0 and CC0 = 1)	1 [1] /2	0.39	0.26	0.21	0.16	0.10	0.01	0.21	0.16

Status word for: JII, JZ, JP, JM	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	Yes	Yes	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

[1] 1 word long for jump width of -128 to +127

Instruction	Address	Description	Length in words	Typ. execution time in µs							
				312	313	314	315	317	319	151	154
JN	LABEL	Jump conditionally if result ≠ 00; (CC1 = 1 and CC0 = 0) or (CC1 = 0) and (CC0 = 1)	1 [1] /2	0.39	0.26	0.21	0.16	0.10	0.01	0.21	0.16
JMZ	LABEL	Jump conditionally if result ≤ 0; (CC1 = 0 and CC0 = 1) or (CC1 = 0 and CC0 = 0)	2	0.39	0.26	0.21	0.16	0.10	0.01	0.21	0.16
JPZ	LABEL	Jump conditionally if result ≥ 0; (CC1 = 1 and CC0 = 0) or (CC1 = 0) and (CC0 = 0)	2	0.39	0.26	0.21	0.16	0.10	0.01	0.21	0.16

Status word for: JN, JMZ, JPZ	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	Yes	Yes	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

JU	LABEL	Jump unconditionally	1 [1] /2	0.39	0.26	0.21	0.16	0.10	0.010	0.21	0.16
JL	LABEL	Jump distributor The instruction is followed by a list of jump instructions. The address is a jump label to the following instruction in the list. ACCU1-L includes the number of the jump instruction to be executed	2	0.39	0.26	0.21	0.16	0.10	0.032	0.21	0.16
LOOP	LABEL	Decrement ACCU1-L and jump if ACCU1-L ≠ 00 (loop programming)	2	0.35	0.24	0.19	0.15	0.06	0.010	0.19	0.15

Status word for: JU, JL, LOOP	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

[1] 1 word long for jump width of -128 to +127

9.29.1 Examples of jump instructions

// Example 1: Invalid jump over the end of a logic string

```
      =    M   10.0   // End of logic string 1

      A    M    0.0   // Start of logic string 2
      JO   L01        // Jump is not permitted because it exceeds the end of the logic string
      A    M    0.1
      =    M   10.1   // End of logic string 2

L01:  A    M    2.0   // Start of logic string 3
      =    M   20.0   // End of logic string 3
```

// Example 2: Invalid jump at the end of a logic string

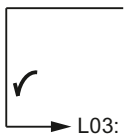
```
      =    M   10.0   // End of logic string 1

      A    M    0.0   // Start of logic string 2
      JO   L02        // End of logic string 2, because JC sets status bit /FC=0. Jump is permitted
                      because it is at the end of the logic string.

      A    M    0.1   // Start of logic string 3
      =    M   10.1   // End of logic string 3

L02:  A    M    2.0   // Start of logic string 4
      =    M   20.0   // End of logic string 4
```

// Example 3: Valid jump within a logic string



```
      =    M    10.0   // End of logic string 1

      A    M    0.0   // Start of logic string 2
      JO   L03         Jump within logic string is permitted.
                       JO does not end the logic string.

      A    M    0.1   // Logic instructions
      A    M    0.2
L03:  A    M    0.3   // Jump to label within logic string is permitted
      A    M    0.4
      =    M   10.1   // End of logic string 2

      A    M    2.0   // Start of logic string 3
      =    M   20.0   // End of logic string 3
```

// Example 4: Valid jump over and past a logic string

```
      =     M     10.0   // End of logic string 1

      JO    L04          // Jump over logic string is permitted
      A     M     0.0    // Start of logic string 2
      A     M     0.1    // Logic instructions
      A     M     0.2
      A     M     0.3
      A     M     0.4
      =     M     10.1   // End of logic string 2

L04:  A     M     2.0    // Start of logic string 3. Jump to label is permitted,
                         because jump is not within a logic string
      =     M     20.0   // End of logic string 3
```

// Example 5: Invalid jumps between nesting levels

```
      =     M     10.0   // End of logic string 1
      A(
      JZ    L05b         // Jump from nesting level is not permitted
      JO    L05a         // Jump to another nesting level is not permitted
      A     M     0.0
      A     M     0.1
      A     M     0.2
      )
      A(
      A     M     0.3
      A     M     0.4
L05a: A     M     2.0    // Label is located in different nesting level than jump
      )
L05b: =     M     20.0   // Label is located in different nesting level than jump
```

// Example 6: Invalid jumps in AND before OR gatings

```
      =     M     10.0   // End of logic string 1

      A     M     0.0    // Start of first AND or OR logic string
      JZ    L06b         // Jump from AND before OR logic string is not permitted
      JO    L06a         // Jump to second AND or OR logic string is not permitted
      A     M     0.1
      A     M     0.2
      O                  // AND before OR operation
      A     M     0.3    // Start of second AND before OR logic string
L06a: A     M     0.4    // Label is located in different logic string than jump
      A     M     0.5
      A     M     0.6
      =     M     20.0

L06b: =     M     20.0   // Label is outside of link
```

9.30 Instructions for the Master Control Relay (MCR)

MCR=1 → MCR is deactivated
MCR=0 → MCR is activated; "T" and "=" instructions write zeros to the corresponding addresses; "S" and "R" instructions leave the memory contents unchanged.

Instruction	Description	Length in words	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
MCR(	Open an MCR zone. Save the RLO to the MCR stack.	1	0,21	0,17	0,15	0,13	0,08	0,03	0,15	0,13
)MCR	Close an MCR zone. Save the RLO to the MCR stack.		0,21	0,17	0,15	0,13	0,08	0,03	0,15	0,13

Status word for: MCR (,)MCR	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	Yes	-
Instruction affects:	-	-	-	-	-	0	1	-	0

MCRA	Activate the MCR	1	0,20	0,15	0,10	0,10	0,07	0,03	0,10	0,10
MCRD	Deactivate the MCR		0,20	0,15	0,10	0,10	0,07	0,03	0,10	0,10

Status word for: MCRA, MCRD	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	-	-	-	-	-	-	-	-

9.31 Execution Times

9.31.1 Execution Time

You have to calculate the basic execution times for direct/indirect addressing. In this chapter we will explain the calculation to you.

Two-Part Statement

A statement consists of two parts:

Part 1: Executing the instruction (see starting with chapter: Logic Instructions (Page 26))

Part 2: Loading the address (as of next table)

This means that you also have to calculate the basic execution time of a statement with address from these two parts.

Calculating the Execution Time

The following applies to the basic execution time:

	Execution time of the instruction
+	Execution time for loading of address
=	Basic execution time of the instruction

The execution times listed in the chapter "List of Instructions" apply to the second part of an instruction, which means the actual execution of an instruction.

You must then add the time required for loading the address to this execution time (see following table).

9.31.2 Loading the Addresses and Operands

Address area	Example	Typ. execution time in µs							
		312	313	314	315	317	319	151	154
Direct addressing	L 1.234567e–36	0	0	0	0	0	0	0	0
I/O	A I a.b	0	0	0	0	0	0	0	0
M	A M a.b	0	0	0	0	0	0	0	0
L	A L a.b	0	0	0	0	0	0	0	0
DB/DI fully qualified [1]	DB100.DBX10.3	0	0	0	0	0	0	0	0
DB/DI partly qualified	DBX10.3 with unknown DB number (e.g. after ON DB[MW20])	0,12	0,09	0,06	0,04	0,02	0,01	0,06	0,04
Timer		0	0	0	0	0	0	0	0
Counter		0	0	0	0	0	0	0	0
I/O access		[2]							

[1] The CPUs offer high-performance support of symbolic programming. The fully qualified DB accesses (e.g. DB100.DBX 1.2) supported here generally cause no additional runtimes. This applies also for the OPN DB command contained in the access.

[2] See table: Execution times for address access to I/O - direct and indirect addressing (PI/PO) (Page 61)

9.31.3 Execution times for address access - indirect addressing

Address area	Example	Typ. execution time in µs							
		312	313	314	315	317	319	151	154
Areainternal, register indirect addressing (AR1/AR2)	= Q [AR1, P#1.1]	0,28	0,16	0,14	0,10	0,03	0,015	0,14	0,10
Areacrossing, register indirect addressing (AR1/AR2)	= [AR1, P#1.0]	0,88	0,55	0,44	0,33	0,11	0,05	0,44	0,33
Memory-indirect addressing	= Q [MD2]	0,64	0,40	0,32	0,24	0,08	0,04	0,32	0,24
Addressing via parameters	A FC_Parameter	0,12	0,08	0,06	0,04	0,02	0,01	0,06	0,04
Access to FB instance data	A FC_Parameter, L Var_Stat	0,12	0,08	0,06	0,04	0,02	0,01	0,06	0,04
Timer	L T [MW2]	0,96	0,60	0,48	0,36	0,12	0,10	0,48	0,36
Counter	L C [MW2]	0,96	0,60	0,48	0,36	0,12	0,10	0,48	0,36
I/O access		[1]							

[1] See table:
Execution times for address access to I/O - direct and indirect addressing (PI/PO) (Page 61)

9.31.4 Execution times for address access to I/O - direct and indirect addressing (PI/PO)

Address	I/O areas	Example	Additional runtimes for address access in µs (typ.)							
			312	313	314	315-2 DP 317-2 DP	315-2 PN/DP 317-2 PN/DP	319	151	154
Load byte	Central	L PIB 0	14.3						67.8	
Load Word		L PIW 0	18.1						71.8	
Load DWord		L PID 0	35.6						80.2	
Transfer byte		T PQB 0	11.2						63.4	
Transfer Word		T PQW 0	12.7						67.4	
Transfer DWord		T PQD 0	25.0						75.2	
Load byte	Digital on-board I/O [1]	L PIB 124	4.4			-			-	
Load Word		L PIW 124	4.5			-			-	
Transfer byte		T PQB 124	4.5			-			-	
Transfer Word		T PQW 124	-	4.2		-			-	
Load byte	Analog on-board I/O [2]	L PIB 752	-	4.7		-			-	
Load Word		L PIW 752	-	4.9		-			-	
Load DWord		L PID 752	-	6.1		-			-	
Transfer byte		T PQB 752	-	4.0		-			-	
Transfer Word		T PQW 752	-	4.1		-			-	
Transfer DWord		T PQD 752	-	4.4		-			-	
Load byte	Distributed (PROFIBUS)	L PIB 0	-	3.9 [3]		3.9		1.7	3.9	
Load Word		L PIW 0	-	4.1 [3]		4.1		1.8	4.1	
Load DWord		L PID 0	-	4.2 [3]		4.2		1.8	4.2	
Transfer byte		T PQB 0	-	3.9 [3]		3.9		0.7	3.9	
Transfer Word		T PQW 0	-	4.1 [3]		4.1		0.7	4.1	
Transfer DWord		T PQD 0	-	4.3 [3]		4.3		0.8	4.3	

[1] C-CPUs only

[2] CPU 313C, CPU 314C-2 DP, CPU 314C-2 PtP, and CPU 314C-2 PN/DP only

[3] CPU 313C-2 DP, 314C-2 DP and 314C-2 PN/DP only

Address	I/O areas	Example	Additional runtimes for address access in µs (typ.)							
			312	313	314	315-2 DP 317-2 DP	315-2 PN/DP 317-2 PN/DP	319	151	154
Load byte	Distributed (PROFINET)	L PIB 0	-		6.6 [4]	-	6.6	2.2	6.6 [5]	
Load Word		L PIW 0	-		6.7 [4]	-	6.7	2.2	6.7 [5]	
Load DWord		L PID 0	-		8.0 [4]	-	8.0	5.9	8.0 [5]	
Transfer byte		T PQB 0	-		7.8 [4]	-	7.8	2.2	7.8 [5]	
Transfer Word		T PQW 0	-		7.9 [4]	-	7.9	2.2	7.9 [5]	
Transfer DWord		T PQD 0	-		7.9 [4]	-	7.9	2.3	7.9 [5]	

[4] Only CPU 314C-2 PN/DP

[5] These values do not apply to IM151-7 CPU

9.32 Master Control Relay - active (MCR)

For the execution times in the active MCR area, an addition must be calculated for each command.

In the active MCR area, the runtime additions per command in µs are as follows:

312	313	314	315	317	319	151	154
0,40	0,35	0,30	0,20	0,07	0,04	0,30	0,20

9.33 Calculating the execution time using CPU 315-2 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 315-2 DP.

Calculating the Execution Time for AreaInternal MemoryDirect Addressing

Example: A M 0.0

1. Step: Execution time of instruction (times: Bit logic instructions (Page 26))

Instruction	Description	Typ. execution time in µs
A	AND	**0,05**

2. Step: Execution times of address access (times: Loading the Addresses and Operands (Page 60))

Address area	Typ. execution time in µs
M	**0**

Total execution time:

0.05 µs + 0.00 µs = 0.05 µs

Calculating the Execution Time for AreaInternal MemoryIndirect Addressing

Example: A I [DBD 12]

1. Step: Execution time of instruction (times: Bit logic instructions (Page 26))

Instruction	Description	Typ. execution time in µs
A	AND	**0,05**

2. Step: Execution times of address access (times: Execution times for address access - indirect addressing (Page 60))

Address area	Typ. execution time in µs
Memory-indirect addressing	**0,24**

Total execution time:

0.05 µs + 0.24 µs = 0.29 µs

Calculating the Execution Time for AreaInternal Register-Indirect Addressing

Example: A I [AR1, P#34.3]

1. Step: Execution time of instruction (times: Bit logic instructions (Page 26))

Instruction	Description	Typ. execution time in µs
A	AND	**0,05**

2. Step: Execution times of address access (times: Execution times for address access - indirect addressing (Page 60))

Address area	Typ. execution time in µs
Area-internal, register-indirect addressing	**0,10**

Total execution time:

0.05 µs + 0.10 µs = 0.15 µs

Calculating the Execution Time for AreaCrossing RegisterIndirect Addressing

Example: A [AR1, P#23.1] ... with P#E1.0 in AR1

1. Step: Execution time of instruction (times: Bit logic instructions (Page 26))

Instruction	Description	Typ. execution time in µs
A	AND	**0,05**

2. Step: Execution times of address access (times Execution times for address access - indirect addressing (Page 60))

Address area	Typ. execution time in µs
Areacrossing, registerindirect addressing	**0,33**

Total execution time:

0.05 µs + 0.33 µs = 0.38 µs

Execution Time for Addressing via Parameters

Example: A "Start"... The parameter "Start" is linked with I 0.5 at the block call.

1. Step: Execution time of instruction (times: Bit logic instructions (Page 26))

Instruction	Description	Typ. execution time in µs
A	AND	**0,05**

2. Step: Execution times of address access (times: Execution times for address access - indirect addressing (Page 60))

Address area	Typ. execution time in µs
Addressing via parameters	**0,04**

Total execution time:

0.05 µs + 0.04 µs = 0.09 µs

See also

Execution Time (Page 59)

9.34 Example of I/O Access

Example: L PIB 0 (centralized I/O)

1. Step: Time of the loading instructions - direct and indirect addressing (times: Load Instructions (Page 35))

Instruction	Address	Typ. execution time in µs
L	B	**0,09**

2. Step: Execution times of address access (times: Execution times for address access to I/O - direct and indirect addressing (PI/PO) (Page 61))

Address	Additional Runtimes for address access in µs
Load Byte	**14,3**

Total execution time:

0.09 µs + 14.30 µs = 14.39 µs

9.35 Organization Blocks (OB)

A user program for an S7-300 consists of blocks which contain the instructions, parameters and data for the respective CPU. The various S7-300 CPUs differ in the number of blocks that you can define for the respective CPU or the blocks provided by the CPU operating system. For a detailed description of the OBs and their applications, refer to the *STEP 7 Online Help*.

Organization blocks	312	313	314	315	317	319	151	154	Start events (hexadecimal value)	
Free cycle:										
OB 1	x	x	x	x	x	x	x	x	1101_H	OB1 start event
									1103_H	Running OB1 start event (conclusion of the free cycle)
Time-of-day interrupts:										
OB 10	x	x	x	x	x	x	x	x	1111_H	Timeofday interrupt event
Time-delay interrupts:										
OB 20	x	x	x	x	x	x	x	x	1121_H	Delay interrupt event
OB 21	x	x	x	x	x	x	x	x	1122_H	Delay interrupt event
Cyclic interrupts:										
OB 32	x	x	x	x	x	x	x	x	1133_H	Cyclic interrupt event
OB 33	x	x	x	x	x	x	x	x	1134_H	Cyclic interrupt event
OB 34	x	x	x	x	x	x	x	x	1135_H	Cyclic interrupt event
OB 35	x	x	x	x	x	x [1]	x	x	1136_H	Cyclic interrupt event
Hardware interrupts:										
OB 40	x	x	x	x	x	x	x	x	1141_H	Hardware interrupt
DPV1 interrupts (DP-CPUs only):										
OB 55	-	x	x	x	x	x	x	x	1155_H	Status interrupt
OB 56	-	x	x	x	x	x	x	x	1156_H	Update interrupt
OB 57	-	x	x	x	x	x	x	x	1157_H	Manufacturer-specific interrupt
Synchronous cycle interrupts:										
OB 61 [2]	-	-	x [3]	x	x	x	x [4]	x	1164_H	Synchronous program execution

[1] In addition to the setting in milliseconds for the OB 35 call interval, you can also select a setting in microseconds in STEP 7 for OB 35 in order to be able to set the parameters for even the shortest interrupt cycle of 500 µs and multiples thereof (the range of values can be set from 500 µs to 60000 ms).

[2] IM151-8 PN/DP CPU and CPU 314C-2 PN/DP: Synchronous cycle to PROFINET IO (not to PROFIBUS DP)
CPU 315, 154, 317 and 319: Synchronous cycle either to PROFIBUS DP or to PROFINET IO (because only one synchronous cycle interrupt OB is available)
CPU 313C-2 DP and CPU 314C-2 DP: No synchronous cycle

[3] Only applies to CPU 314C-2 PN/DP

[4] Does not apply to IM151-7 CPU

Organization blocks	312	313	314	315	317	319	151	154	Start events (hexadecimal value)	
Asynchronous error interrupts:										
OB 80	x	x	on	x	x	x	x	x	3501_H	Cycle time exceeded
									3502_H	OB or FB request error
									3505_H	Timeofday interrupt elapsed due to time jump
									3507_H	Multiple OB request error caused start info buffer overflow
OB 82 (Diagnostics interrupt)	x	x	x	x	x	x	x	x	3842_H	Module working properly
									3942_H	Module defective
OB 83	-	-	x [5]	x [5]	x [5]	x [5]	x [6] [7]	x [6]	3854_H	PROFINET IO submodule inserted and matches configured submodule
									3855_H	PROFINET IO submodule inserted but does not match configured submodule
									3861_H	Module inserted
									3951_H	PROFINET IO module removed
									3961_H	Module removed
OB 85	x	x	x	x	x	x	x	x	$35A1_H$	No OB or FB
									$35A3_H$	Error when operating system accesses a block
									$39B1_H$	I/O access error during process image update of the inputs (during each access)
									$39B2_H$	I/O access error during transfer of the process image to the output modules (during each access)
									$38B3_H$	I/O access error during process image update of the inputs (outgoing event)
									$39B3_H$	I/O access error during process image update of the inputs (incoming event)
									$38B4_H$	I/O access error during transfer of the process image to the output modules (outgoing event)
									$39B4_H$	I/O access error during transfer of the process image to the output modules (incoming event)

[5] For PROFINET IO only

[6] For central I/O devices and PROFINET IO

[7] On IM151-7 CPU, only applies to central I/O devices

Organization blocks	312	313	314	315	317	319	151	154	Start events (hexadecimal value)	
OB 86 [8]	-	x	x	x	x	x	x	x	$32C9_H$	PROFIBUS DP: Station activated by SFC 12 (mode 3)
									$33C9_H$	PROFIBUS DP: Station deactivated by SFC 12 (mode 4)
									$38C4_H$	Distributed I/O: Station failure, outgoing
									$39C4_H$	Distributed I/O: Station failure, incoming
									$32CF_H$	PROFINET IO: Station activated by SFC 12 (mode 3)
									$33CF_H$	PROFINET IO: Station deactivated by SFC 12 (mode 4)
									$38CB_H$	PROFINET IO: Station return
									$39CB_H$	PROFINET IO: Station failure
									$38F8_H$	PROFINET IO: Partial station return
									$39F8_H$	PROFINET IO: Partial station failure
OB 87	x	x	x	x	x	x	x [9]	x	$35E1_H$	Incorrect message frame ID in GD
									$35E2_H$	GD package status cannot be entered in DB
									$35E6_H$	GD total status cannot be entered in DB
Restart (warm restart):										
OB 100	x	x	x	x	x	x	x	x	1381_H	Manual restart (warm restart) request
									1382_H	Automatic restart (warm restart) request

[8] Only applies to CPUs with DP and/or PN interface

[9] Does not apply to IM151-8 PN/DP CPU

Organization blocks	312	313	314	315	317	319	151	154	Start events (hexadecimal value)	
Synchronous error interrupts:										
OB 121	x	x	x	x	x	x	x	x	2521_H	BCD conversion error
									2522_H	Area length error when reading
									2523_H	Area length error when writing
									2524_H	Range error when reading
									2525_H	Range error when writing
									2526_H	Timer number error
									2527_H	Counter number error
									2528_H	Alignment error when reading
									2529_H	Alignment error when writing
									2530_H	Write error during access to DB
									2531_H	Write error during access to DI
									2532_H	Block number error when opening a DB
									2533_H	Block number error when opening a DI
									2534_H	Block number error when calling an FC
									2535_H	Block number error when calling an FB
									$253A_H$	DB not loaded
									$253C_H$	FC not loaded
									$253E_H$	FB not loaded
OB 122	x	x	x	x	x	x	x	x	2944_H	I/O access error at nth read access (n > 1)
									2945_H	I/O access error at nth write access (n > 1)

9.36 Function Blocks (FBs)

The following table lists the quantity, number and maximum size of the function blocks that you can define in the individual CPUs of the S7-300.

Function blocks	312	312C	313	314	315	317	319	151	154
Quantity	1024					2048	4096	1024	
Permitted number	0 to 7999							0 to 7999	
Maximum size of an FB (processrelevant code)	32 KB	64 KB						64 KB	

9.37 Functions (FC)

The following table lists the quantity, number and maximum size of the functions that you can define in the individual CPUs of the S7-300.

Functions	312	312C	313	314	315	317	319	151	154
Quantity	1024					2048	4096	1024	
Permitted number	0 to 7999							0 to 7999	
Maximum size of an FC (processrelevant code)	32 KB	64 KB						64 KB	

9.38 Data blocks (DB)

The following table lists the quantity, number and maximum size of the data blocks that you can define in the individual CPUs of the S7-300.

Data blocks	312	312C	313	314	315	317	319	151	154
Quantity	1024					2048	4096	1024	
Permitted number	1 to 16000							1 to 16000	
Maximal size of a DB (data byte quantity)	32 KB	64 KB						64 KB	

9.39 System Functions (SFC)

The following tables show the system functions provided by the operating system of the S7-300 CPUs, including the execution times on the relevant CPU.

SFC no.	SFC name	Description	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
0	SET_CLK	Set time of day	21				21	7	21	
1	READ_CLK	Read time of day	7				6	3	7	
2	SET_RTM	Set operating hours counter	6				5	3	6	
3	CTRL_RTM	Starts/stops operating hours counter	6				5	2	6	
4	READ_RTM	Read operating hours counter	8				7	3	8	
5	GADR_LGC	Determine logical base address of a module	26				18	12	26	
6	RD_SINFO	Read start information of the current OB	11				5	3	11	
7	DP_PRAL	Trigger hardware interrupt from the user program of the CPU as DP slave to DP master	-	87 (for DP CPU only)		87		26	87 [1]	87
		Concurrently running jobs to different modules, max.	-	34 jobs together with SFB 75 jobs						
11	DPSYC_FR	Synchronize groups of DP slaves	-	65 (for DP CPU only)		65	54	23	65 [2]	65
		Concurrently running jobs, max.	-	2 jobs						
12	D_ACT_DP	Activate or deactivate DP slaves/PN IO devices	-	64 (for DP CPU only)		64	48	30	64 [2]	64
		Concurrently running jobs, max.	-	8 jobs						
13	DPNRM_DG	Read slave diagnostics	-	33 (for DP CPU only)		33	23	10	33 [2]	33
		Concurrently running jobs, max.	-	4 jobs						
14	DPRD_DAT	Read consistent user data (n byte)	-	27 (for DP CPU only)		27	20	15	27 [2]	27
15	DPWR_DAT	Write consistent user data (n byte)	-	26 (for DP CPU only)		26	24	15	26 [2]	26

[1] IM151-8 PN/DP CPU does not support this SFC

[2] With inserted DP master module

SFC no.	SFC name	Description	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
17	ALARM_SQ	Generate blockrelated alarms that can be acknowledged	126				99	67	126	
18	ALARM_S	Generate blockrelated alarms that cannot be acknowledged	126				101	68	126	
19	ALARM_SC	Acknowledgment state of the last ALARM_SQ incoming alarms	27				20	5	27	
20	BLKMOV	Copy variables within the working memory	10 + 0.01 per byte				7 + 0.01 per byte	2 + 0.003 per byte	10 + 0.01 per byte	
21	FILL	Preset field within the working memory	10 + 0.035 per byte				6 + 0.035 per byte	3 + 0.01 per byte	10 + 0.035 per byte	
22	CREAT_DB	Generate data block in the working memory	86				63	50	86	
23 [3)]	DEL_DB	Delete data block	94				87	52	94	
		Concurrently running jobs, max.	21 jobs							
24	TEST_DB	Test data block	13				7	5	13	
28	SET_TINT	Set time-of-day interrupt	17				11	5	17	
29	CAN_TINT	Cancel time-of-day interrupt	8				4	2	8	
30	ACT_TINT	Activate time-of-day Interrupt	10				5	2	10	
31	QRY_TINT	Query time-of-day interrupt	11				6	2	11	
32	SRT_DINT	Start time-delay interrupt	10				7		10	
33	CAN_DINT	Cancel time-delay interrupt	10				5		10	
34	QRY_DINT	Query time-delay interrupt	8				3		8	

[3)]The SFC 23 deletes data blocks in the RUN operating mode. If a SFC 23 call is present in the loaded project, additional tests are carried out when data blocks are accessed. These tests can increase the command run time on the DB operand area. If a data block is accessed that was deleted in RUN by SFC 23, the OB programming error (OB 121) is called. DBs are deleted in the background and the process may take as long as the OB1 cycle. Freeing up memory resources may take up many OB1 cycles.

SFC no.	SFC name	Description	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
36	MSK_FLT	Mask synchronous error events	8				5	3	8	
37	DMSK_FLT	Unmask synchronous error events	8				5	3	8	
38	READ_ERR	Read event status register	7				5	2	7	
39	DIS_IRT	Disable new events	24				15	9	24	
40	EN_IRT	Cancel disable of events	23				20	13	23	
41	DIS_AIRT	Delay interrupt events	24				24	10	24	
42	EN_AIRT	Cancel delay of interrupt events	13				13	7	13	
43	RE_TRIGR	Re-trigger cycle time monitoring	21				13	12	21	
44	REPL_VAL	Transfer substitute value to ACCU1	5				4	3	5	
46	STP	Put CPU in STOP	No time information							
47	WAIT	Delays program execution in addition to waiting times	of which + 0.1 % waiting time							
49	LGC_GADR	Determine slot that belongs to a logical address	20				10	8	20	
50	RD_LGADR	Determine all logical addresses of a module	38				22	18	38	
51	RDSYSST	Read out information from the system state list. SFC 51 is not interruptible through interrupts.	9 + 0.1 per byte				7 + 0.1 per byte	3 + 0.1 per byte	9 + 0.1 per byte	
		Concurrently running jobs, max.	4 jobs							
52	WR_USMSG	Write user alarm in diagnostic buffer	290					60	290	

SFC no.	SFC name	Description	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
55	WR_PARM	Write dynamic parameters	190							
		Concurrently running jobs, max.	1 job							
56	WR_DPARM	Write predefined dynamic parameters	95							
		Concurrently running jobs, max.	1 job							
57	PARM_MOD	Assign module parameters	95							
		Concurrently running jobs, max.	1 job							
58	WR_REC	Write data record	388 + 10 per byte					350 + 10 per byte	388 + 10 per byte	
		Concurrently running jobs to different modules, max.	4 jobs together with SFB 53 jobs				8 jobs together with SFB 53 jobs		4 jobs together with SFB 53 jobs	
59	RD_REC	Read data record	461 + 12 per byte					432 + 12 per byte	461 + 12 per byte	
		Concurrently running jobs to different modules, max.	4 jobs together with SFB 52 jobs				8 jobs together with SFB 52 jobs		4 jobs together with SFB 52 jobs	
64	TIME_TICK	Read out millisecond timer	6				4	2	6	

SFC no.	SFC name	Description	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
65	X_SEND	Send data to external partner	15				13	8	15 [1]	15
		Max. possible simultaneous SFC 65, SFC 66, SFC 67, SFC 68, SFC 72, or SFC 73 jobs to different remote communication partners [4]	4 jobs	6 jobs	10 jobs	14 jobs	30 jobs		10 jobs [1]	14 jobs
66	X_RCV	Receive data from external partner	19				9	8	19 [1]	19
		Max. possible simultaneous SFC 65, SFC 66, SFC 67, SFC 68, SFC 72, or SFC 73 jobs to different remote communication partners [4]	4 jobs	6 jobs	10 jobs	14 jobs	30 jobs		10 jobs [1]	14 jobs
67	X_GET	Read data from external partner	18				12	5	18 [1]	18
		Max. possible simultaneous SFC 65, SFC 66, SFC 67, SFC 68, SFC 72, or SFC 73 jobs to different remote communication partners [4]	4 jobs	6 jobs	10 jobs	14 jobs	30 jobs		10 jobs [1]	14 jobs
68	X_PUT	Write data to external partner	18				12	5	18 [1]	18
		Max. possible simultaneous SFC 65, SFC 66, SFC 67, SFC 68, SFC 72, or SFC 73 jobs to different remote communication partners [4]	4 jobs	6 jobs	10 jobs	14 jobs	30 jobs		10 jobs [1]	14 jobs
69	X_ABORT	Abort connection to external partner	7				5		7 [1]	7

[1] IM151-8 PN/DP CPU does not support this SFC

[4] Note: Only one SFC 65, SFC 66, SFC 67, SFC 68, SFC 72 or SFC 73 job is possible at the same time to different remote communication partners.

SFC no.	SFC name	Description	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
70	GEO_LOG	Determine start address of a module	23				9	8	23	
71	LOG_GEO	Determine slot that belongs to a logical address	21				11	8	21	
72	I_GET	Read data from internal partner	36				28	15	36	
		Max. possible simultaneous SFC 65, SFC 66, SFC 67, SFC 68, SFC 72, or SFC 73 jobs to different remote communication partners [4]	4 jobs	6 jobs	10 jobs	14 jobs	30 jobs		10 jobs	14 jobs
73	I_PUT	Write data to internal partner	28					15	28	
		Max. possible simultaneous SFC 65, SFC 66, SFC 67, SFC 68, SFC 72, or SFC 73 jobs to different remote communication partners [4]	4 jobs	6 jobs	10 jobs	14 jobs	30 jobs		10 jobs	14 jobs
74	I_ABORT	Abort connection to internal partner	8				6	2	8	
81	UBLKMOV	Copy the variable uninterruptible, length of the data to be copied up to 512 byte	11 + 0.01 per byte				8 + 0.01 per byte	3	11 + 0.01 per byte	
82	CREA_DBL	Create data block in load memory	46				39	20	46	
		Concurrently running jobs, max.	3 jobs							
83	READ_DBL	Read from a data block in load memory	47				36	20	47	
		Concurrently running jobs, max.	3 jobs							
84	WRIT_DBL	Write to a data block in load memory	50				36	20	50	
		Concurrently running jobs, max.	3 jobs							

[4] Note: Only one SFC 65, SFC 66, SFC 67, SFC 68, SFC 72 or SFC 73 job is possible at the same time to different remote communication partners.

SFC no.	SFC name	Description	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
99 [5]	WWW	Connection between user program and web server	-		17		15	4	17	
101	RTM	Handle the runtime meter	8				7	3	8	
102	RD_DPARA	Read predefined parameters	62				53	30	62	
		Concurrently running jobs, max.	1 job							
103	DP_TOPOL	Determine the bus topology in a DP master system	-	25 (for DP CPUs only)		25	7		25 [2]	25
105	READ_SI	Status dynamically assigned ALARM_Dx system resources	47 + 0.61 per alarm				45 + 0.26 per alarm	15 + 0.1 per alarm	47 + 0.61 per alarm	
106	DEL_SI	Enable dynamically assigned system resources	146 + 3.8 per alarm				140 + 3.6 per alarm	107 + 3.6 per alarm	146 + 3.8 per alarm	
107	ALARM_DQ	Create acknowledgeable alarm with associated value	127				98	69	127	
108	ALARM_D	Create non-acknowledgeable alarm with associated value	129				99	69	129	
109 [6]	PROTECT	Activate write protection	4				3	2	4	

[2] With inserted DP master module

[5] For PROFINET-CPUs only (CPU 31x PN/DP, IM15x-8 PN/DP CPU, and CPU 314C-2 PN/DP).
The SFC runtime can temporarily increase up to 800 µs when you initialize a WEB page.

[6] It is advisable to protect the CPU with a password to prevent unauthorized access. Please note the characteristic features of fail-safe systems.

SFC no.	SFC name	Description	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
112 [7]	PN_IN	Update inputs in the user program interface of PROFINET components	-		778	760	612	197	778	760
113 [7]	PN_OUT	Update outputs in the user program interface of PROFINET components	-		604	604	464	158	604	604
114 [7]	PN_DP	Update DP interconnection	-		153	150	132	105	153	150
126	SYNC_PI	Update partial process image of inputs in synchronous cycle	-		30 + 0,2 per byte [8]	30 + 0.2 per byte	29 + 0.2 per byte	22 + 0.15 per byte	30 + 0.2 per byte [9]	30 + 0.2 per byte
		Concurrently running jobs, max.	-		1 job					
127	SYNC_PO	Update partial process image output table in synchronous cycle	-		29 + 0.2 per byte [8]	29 + 0.2 per byte	28 + 0.2 per byte	25 + 0.15 per byte	29 + 0.2 per byte [9]	29 + 0.2 per byte
		Concurrently running jobs, max.	-		1 job					

[7] For CPU 31x PN/DP, IM15x-8 PN/DP, and CPU 314C-2 PN/DP: The runtimes of these blocks depend on their respective interconnection configuration. See also the CPU 31xC and CPU 31x Manual, Technical Specifications chapter: "Cycle time, extending the OB1 cycle for cyclical interconnections".

[8] For CPU 314C-2 PN/DP only with synchronous cycle on PROFINET IO

[9] Does not apply to IM151-7 CPU; applies only to the synchronous cycle of an IM151-8 PN/DP CPU on PROFINET IO

9.40 System Function Blocks (SFB)

The following table lists the system function blocks provided by the S7-300 CPU operating system, including their execution times on the relevant CPU.

SFB No.	SFB name	Description	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
0	CTU	Count up	13				9	4	13	
1	CTD	Count down	11				8	3	11	
2	CTUD	Count up and down	11				9	3	11	
3	TP	Generate pulse	13				11	5	13	
4	TON	Generate ON delay	13				9	5	13	
5	TOF	Generate OFF delay	12				8	3	12	
32	DRUM	Implement a sequence processor with a maximum of 16 steps	40				20	10	40	
41 [1]	CONT_C	Controller (PID) for continuous I/O sizes, integrated controller	-	58			-			
42 [1]	CONT_S	Step controller (PI), integrated controller	-	50			-			
43 [1]	PULSEGEN	Pulse shaper	-	39			-			
44 [1]	ANALOG	Positioning with analog output, integrated technology function:	-				-			
		• Idle			35					
		• Starting travel			65					
		• Job			65					
46 [1]	DIGITAL	Positioning with digital outputs, integrated technology function:	-				-			
		• Idle			35					
		• Starting travel			65					
		• Job			65					

[1] Supported by CPU 31xC only

SFB No.	SFB name	Description	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
47 [1]	COUNT	Counting, integrated technology function	75			-				
48 [1]	FREQUENC	Frequency measurement, integrated technology function	65			-				
49 [1]	PULSE	Pulse width modulation, integrated technology function	65			-				
52	RDREC	Read data record from DP slave, PROFINET I/O device or central module	483 + 12 per byte				469 + 12 per byte	432 + 12 per byte	483 + 12 per byte	
		Concurrently running jobs to different modules, max.	4 jobs together with SFB 59 jobs				8 jobs together with SFB 59 jobs		4 jobs together with SFB 59 jobs	
53	WRREC	Write data record to DP slave, PROFINET I/O device or central module	429 + 10 per byte					350 + 10 per byte	429 + 10 per byte	
		Concurrently running jobs to different modules, max.	4 jobs together with SFB 58 jobs				8 jobs together with SFB 58 jobs		4 jobs together with SFB 58 jobs	
54	RALRM	Read out interrupt status information from interrupts of a DP slave, PROFINET I/O device or of a central module in the respective OB	31				27	7	31	
		Concurrently running jobs, max.	1 job							

[1] Supported by CPU 31xC only

SFB no.	SFB name	Description	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
60 [2]	SEND_PTP	Send data in idle mode	-	70		-				
		Send data in productive mode								
		• 1 to 206 bytes	-	120		-				
		• 207 to 412 bytes	-	140		-				
		• 413 to 618 bytes	-	160		-				
		• 619 to 824 bytes	-	180		-				
		• 825 to 1024 bytes	-	200		-				
61 [2]	RCV_PTP	Receive data in idle mode	-	70		-				
		Send data in productive mode								
		• 1 to 206 bytes	-	110		-				
		• 207 to 412 bytes	-	125		-				
		• 413 to 618 bytes	-	140		-				
		• 619 to 824 bytes	-	155		-				
		• 825 to 1024 bytes	-	170		-				
62 [2]	RES_RCVB	Delete receive buffer in idle mode	-	70		-				
		Delete receive buffer in productive mode	-	70		-				

[2] Only for CPU 31xC-2 PtP

SFB no.	SFB name	Description	Typ. execution time in µs							
			312	313	314	315	317	319	151	154
63 [3]	SEND_RK	Send data in idle mode	-	-	145	-				
		Send data in productive mode	-	-	550	-				
64 [3]	FETCH_RK	Fetch data in idle mode	-	-	145	-				
		Fetch data productive mode	-	-	1250	-				
65 [3]	SERVE_RK	Receive/provide data in idle mode	-	-	145	-				
		Receive/provide data in productive mode	-	-	1250	-				
73 [4]	RCVREC	Receive data records in an I-device from a higher-level IO controller	-	-	90 + 0.015 per byte		60 + 0.01 per byte	35 + 0.005 per byte	90 + 0.015 per byte	
74 [4]	PRVREC	Provide data records in an I-device for a higher-level IO controller	-	-	90 + 0.015 per byte		60 + 0.01 per byte	35 + 0.005 per byte	90 + 0.015 per byte	
75 [5]	SALRM	Set any interrupt from I-slaves	-	41			32	30	41 [6]	41
		Concurrently running jobs to different modules, max.	-	34 jobs together with SFB 7 jobs						
81	RD_DPAR	Reading predefined parameters	50				30	20	50	
		Concurrently running jobs, max.	4 jobs							
104 [4]	IP_CONF	Distribution of the IP Suite and the device name from the user program	-		84	41	26	15	84	41

[3] Only for CPU 314C-2 PtP
Notice: if the length exceeds 128 characters, the data are transferred in blocks of up to 128 characters each.

[4] For PROFINET CPUs only

[5] For DP CPUs with slave functionality only

[6] IM151-8 PN/DP CPU does not support this SFB

9.41 Standard blocks for S7 communication

Predefined blocks (FBs/FCs) are available as interface in your STEP 7 user program for use in certain communication services. These blocks are saved in the standard library, communication blocks.

FB no.	FB name	Description	may be used with CPUs	
			without PN interface	with PN interface
8	USEND	Uncoordinated send of data	Communication via CP	Communication via CP or integrated PROFINET interface [1]
9	URCV	Uncoordinated receipt of data		
12	BSEND	Block-oriented send of data		
13	BRCV	Block-oriented receipt of data		
14	GET	Read data from a remote CPU		
15	PUT	Write data to a remote CPU		
28 [2]	USEND_E	Uncoordinated send of data with extended send ranges SD_1 to SD_4	-	Communication via integrated PROFINET interface
29 [2]	URCV_E	Uncoordinated reception of data with extended reception ranges RD_1 to RD_4	-	
34 [2]	GET_E	Read data from remote CPU with extended reception ranges RD_1 to RD_4	-	
35 [2]	PUT_E	Write data to a remote CPU with extended writing ranges SD_1 to SD_4)	-	

[1] Communication using these blocks is only possible on an IM151-8 PN/DP CPU and IM154-8 PN/DP CPU via the integrated PROFINET interface. These function blocks cannot be used on an IM151-7 CPU.

[2] As of V3.2

FC no.	FC name	Description	may be used with CPUs	
			without PN interface	with PN interface
62	C_CNTRL	Query connection status which belongs to a local connection ID.	Communication via CP	Communication via CP or integrated PROFINET interface [1]

[1] Communication using these blocks is only possible on an IM151-8 PN/DP CPU and IM154-8 PN/DP CPU via the integrated PROFINET interface. This FC cannot be used on an IM151-7 CPU.

9.42 Function Blocks for Open Communication via Industrial Ethernet

STEP 7 provides certain FBs and UDTs that can be used to exchange data with communication partners by means of the user program. These blocks are saved in the standard library, communication blocks.

FB no.	FB name	Description	CPU 315-2 PN/DP CPU 317-2 PN/DP	CPU 319-3 PN/DP IM151-8 PN/DP CPU IM154-8 PN/DP CPU	CPU 314-2 PN/DP	Communication protocols
63	TSEND	Sending data	as of V3.1	as of V3.2	as of V3.3	TCP, ISO-on-TCP
64	TRCV	Receiving data				TCP, ISO-on-TCP
65	TCON	Establishing a communication connection				TCP, ISO-on-TCP, UDP
66	TDISCON	Terminating a communication connection				TCP, ISO-on-TCP, UDP
67	TUSEND	Sending data				UDP
68	TURCV	Receiving data				UDP

9.43 IEC Functions

You can use the following IEC functions in STEP 7.

These blocks are saved in the standard library, IEC Function Blocks of STEP 7.

FC no.	FC name	Description
DATE_AND_TIME		
3	D_TOD_DT	Combines 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 DATE_AND_TIME data format.
8	DT_TOD	Extracts the TIME_OF_DAY data format from the DATE_AND_TIME data format.
Time formats		
33	S5TI_TIM	Converts S5 TIME data format to TIME data format.
40	TIM_S5TI	Converts TIME data format to S5 TIME data format.
Duration		
1	AD_DT_TM	Add a duration in TIME format to a point in time in DT format; the result is a new point in time in DT format.
35	SB_DT_TM	Subtract a duration in TIME format from a point in time in DT format; the result is a new point in time in DT format.
34	SB_DT_DT	Subtract two points in time in DT format; the result is a duration in TIME format.
Compare DATE_AND_TIME		
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 unequal to.
Compare STRING		
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 unequal to.

FC no.	FC name	Description
Processing STRING variables		
21	LEN	Reads out the actual length of a STRING variable.
20	LEFT	Reads the first L character of a STRING variable.
32	RIGHT	Reads the last L character of a STRING variable.
26	MID	Reads the center L character of a STRING variable. (starting from the defined character).
2	CONCAT	Combines 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.
Format conversions with STRING		
16	I_STRNG	Converts a variable from INTEGER format to STRING format.
5	DI_STRNG	Converts a variable from INTEGER (32bit) 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 (32bit) format.
39	STRNG_R	Converts a variable from STRING format to REAL format.
Processing of numbers		
22	LIMIT	Limits a number to a defined limit value.
25	MAX	Selects the largest of three numerical variables.
27	MIN	Selects the smallest of three numerical variables.
36	SEL	Selects one of two variables.

SSL partial list

10

SSL ID	Index	Message function
		Module identification
0111_H		Identification data record corresponding to the specified index
	0001_H	CPU type and version number
	0006_H	Identification of basic hardware
	0007_H	Identification of basic firmware
		CPU characteristics
0012_H	–	All characteristics
0112_H		Characteristics of a group
	0000_H	STEP 7 processing
	0100_H	Time system in the CPU
	0200_H	System behavior of the CPU
	0300_H	STEP 7 instruction supply
$0F12_H$	–	Header information only
		User memory areas
0013_H	–	All data records of available user memory areas
0113_H		One data record for the specified memory area
	0001_H	Working memory
		System areas
0014_H	–	Data records of all system areas
$0F14_H$	–	Header information only
		Block types
0015_H	–	Data records of all block types
		Status of the module LEDs
0019_H	–	Read the status of all LEDs
$0F19_H$	–	Header information only

SSL ID	Index	Message function
		Component identification
$001C_H$	–	Read all data records
$011C_H$		Data record for specified index
	0001_H	Station name
	0002_H	Name of the module
	0003_H	Plant ID of the module
	0004_H	Copyright entry
	0005_H	Serial number of the module
	0007_H	Module type name
	0008_H	Serial number of the micro memory card
	0009_H	Manufacturer and profile of a CPU module
	$000A_H$	OEM identifier
	$000B_H$	Location ID
$01FC_H$	–	Header information only
		Alarm status
0222_H		Data record for specified interrupt
	OB no.	Number of the OB (OB1 only)
		Assignment between process image partitions and CPUs (only for CPUs that support synchronous cycle)
0025_H	–	Assignment of all partial process images and OBs
0125_H	TPA no. (number of the partial process image)	Assignment of a partial process image to the corresponding OB
0225_H	OB no.	Assignment of an OB to the corresponding partial process images
$0F25_H$	–	Only SSL partial list header information
		Communication status data
0132_H		Communication status information on the specified communication unit (only one data record)
	0004_H	OMS/contactor
	0005_H	Diagnostics
	0008_H	Time system (TIME)
	$000B_H$	Runtime meter (32 bit) 0 to 7
	$000C_H$	Runtime meter (32 bit) 8 to 15
0232_H		Communication status information on specified communication unit
	0004_H	OMS/contactor

SSL ID	Index	Message function
		Status of the module LEDs
0074_H	–	Read the status of all LEDs
0174_H		Read the status of individual LEDs
	0001_H	GE, group error
	0004_H	RUN, RUN LED
	0005_H	STOP, STOP LED
	0006_H	FRCE, Force LED
	$000B_H$	BF1 LED
	$000C_H$	BF2 LED
	0014_H	BF3 LED
	0015_H	MAINT LED
		Module status information
0591_H	–	Module status information for all submodules that a host recognizes
$0A91_H$	–	Module status information of all DP master systems known to CPU (only CPUs with DP interface)
$0C91_H$		Module status information of a module
	Any logical address of a module/submodule	Module status information of a module using logical address
$0D91_H$		Module status information of a rack or station
	Centralized configuration: 0000_H: Rack 0 $0001h_H$: Rack 1 0002_H: Rack 2 0003_H: Rack 3 **PROFIBUS DP:** $xxyy_H$: DP subnet ID/station no. **PROFINET IO:** Slot address of PROFINET IO device: Bit 15: is always = 1 Bit 11-14: PN IO subsystem ID (value range 100-115; in which only 0 to 15 must be specified) Bit 0-10: Station number of the PROFINET IO device	Module status information of all modules in specified rack/station

SSL ID	Index	Message function
		Rack/station status information
0092H		Expected state of the rack in the central configuration or stations of a subnet
	0000H	Information about the state of the rack in the central configuration
	DP master system ID	Information about the state of the stations in the subnet
0292H		Actual state of the rack in the central configuration or stations of a subnet
	0000H	Information about the state of the rack in the central configuration
	DP master system ID	Information about the state of the stations in the subnet
0692H		Diagnosed state of the rack in the central configuration or stations of a subnet
	0000H	Information about the state of the rack in the central configuration
	DP master system ID	Information about the state of the stations in the subnet
		Rack/station status information
0094H		Expected state of the rack in the central configuration or stations of a subnet
	0000H	Information about the state of the rack in the central configuration
	DP master system ID or PN IO subsystem no.	Information about the state of the stations in the subnet
0194H		Activation status of stations of a subnet (only CPU with DP and/or PROFINET interface)
	DP master system ID or PN IO subsystem no.	Information about the state of the stations in the subnet
0294H		Actual state of the rack in the central configuration or stations of a subnet
	0000H	Information about the state of the rack in the central configuration
	DP master system ID or PN IO subsystem no.	Information about the state of the stations in the subnet
0694H		Diagnosed state of the rack in the central configuration or stations of a subnet
	0000H	Information about the state of the rack in the central configuration
	DP master system ID or PN IO subsystem no.	Information about the state of the stations in the subnet
0794H		Faulty- and/or maintenance status of station
	0000H	Information about the state of the rack in the central configuration
	DP master system ID or PN IO subsystem no.	Information about the state of the stations in the subnet
0F94H	–	Header information only

SSL ID	Index	Message function
		Extended DP master system information
0195_H	$xxyy_H$: DP master system ID/00_H	Extended DP master system information of a DP master system (only CPUs with DP interface)
$0F95_H$	–	Header information only (only CPUs with DP interface)
		Submodule status information
0696_H	Any logical address of a module/submodule	Status data of all submodules of a module
$0C96_H$	Any logical address of a module/submodule	Status data of a submodule
		ToolChanger information (only for CPUs with PN interface)
$009C_H$		Information about all tool changers and their tools in a PN IO subsystem
$019C_H$		Information about all tool changers
$029C_H$		Information about a tool changer and its tools
$039C_H$		Information about a tool and its IO device
$0F9C_H$		only header information
		Diagnostic buffer
$00A0_H$		All input event information (in the RUN of CPU default mode outputs only 10 entries; the number of event information output in RUN can be parameterized from 10 - 499)
$01A0_H$	x	The "x" most recent input event information
$0FA0_H$	–	Header info SSL only
		Diagnostic data on modules
$00B1_H$	Any logical address of a module/submodule	The first four diagnostic bytes of a module (diagnostics data record DS0)
$00B2_H$	Rack and slot number	All diagnostics data of a module (diagnostics data record DS1 - only for centrally mounted modules)
$00B3_H$	Any logical address of a module/submodule	All diagnostics data of a module (diagnostics data record DS1)
$00B4_H$	Logical basic address (diagnosis address of the slave)	Standard diagnostics data of a DP slave (only CPUs with DP interface)

Index

E

F

I

J

L

M

N

O

P

Q

R

S

T

U

W

X