S7-300 Instruction List

CPU Specifications
CPUs 312 IFM to 318-2 DP

This Instruction List is part of the documentation package with the order number:

6ES7398-8FA10-8BA0

This documentation can **no longer** be ordered under the given number!

Edition 10/2001

A5E00096292-03

Siemens AG
Bereich Automatisierungs- und Antriebstechnik
Geschaeftsgebiet Industrie-Automatisierungssysteme
Postfach 4848, D- 90327 Nuernberg

Disclaimer of Liability

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

Siemens Aktiengesellschaft

A5E00096292-03

Contents

Validity Range of the Instructions List

CPU	as of order no.	in the following referred to as
CPU 312 IFM	6ES7 312-5AC02-0AB0	CPU 312*
CPU 313	6ES7 313-1AD03-0AB0	CPU 313
CPU 314	6ES7 314-1AE04-0AB0	CPU 314
CPU 314 IFM	6ES7 314-5AE03-0AB0	CPU 314*
CPU 314 IFM	6ES7 314-5AE10-0AB0	
CPU 315	6ES7 315-1AF03-0AB0	CPU 315
CPU 315-2 DP	6ES7 315-2AF03-0AB0	CPU 315-2
CPU 316-2 DP	6ES7 316-2AG00-0AB0	CPU 316-2
CPU 318-2	6ES7 318-2AJ00-0AB0	CPU 318-2

Address Identifiers and Parameter Ranges

Addr. ID	Parameter Ranges						Description
	312*	313	314	314*	315/315-2/ 316-2	318-2	
Q	0.0 to 31.7	0.0 to 127.7		0.0 to 123.7	0.0 to 127.7	0.0 to 2047.7[1]	Output bit (in PIQ)
	124.7 to 127.7	–		124.0 to 127.7	–	–	... integrated outputs
QB	0 to 31	0 to 127		0 to 123	0 to 127	0 to 2047[1]	Output byte (in PIQ)
	124 to 127	–		124 to 127	–	–	... integrated outputs
QW	0 to 30	0 to 126		0 to 122	0 to 126	0 to 2046[1]	Output word in (PIQ)
	124 to 126	–		124 to 126	–	–	... integrated outputs
QD	0 to 28	0 to 124		0 to 120	0 to 124	0 to 2044[1]	Output double word (in PIQ)
	124	–		124	–	–	... integrated outputs
B	–				–	–	Byte with general register-indirect addressing
W	–				–	–	Word with general register-indirect addressing
D	–				–	–	Double word with general register-indirect addressing

[1] PIQ is preset to 256 byte

Addr. ID	Parameter Ranges						Description
	312*	313	314	314*	315/315-2/ 316-2	318-2	
DBX	0.0 to 8191.7				0.0 to 8191.7	0.0 to 65533.7	Data bit in data block
DB	1 to 63	1 to 127			1 to 127	0 to 2047	Data block
DBB	0 to 6143	0 to 8191			0 to 8191	0 to 65533	Data byte in DB
DBW	0 to 6142	0 to 8190			0 to 8190	0 to 65532	Data word in DB
DBD	0 to 6140	0 to 8188			0 to 8188	0 to 65530	Data double word in DB
DIX	0.0 to 8191.7				0.0 to 8191.7	0.0 to 65533.7	Data bit in instance DB
DI	1 to 63	1 to 127			1 to 127	1 to 2047	Instance data block
DIB	0 to 6143	0 to 8191			0 to 8191	0 to 65533	Data byte in instance DB
DIW	0 to 6142	0 to 8190			0 to 8190	0 to 65532	Data word in instance DB
DID	0 to 6140	0 to 8188			0 to 8188	0 to 65530	Data double word in instance DB

Addr. ID	Parameter Ranges						Description
	312*	313	314	314*	315/315-2/ 316-2	318-2	
I	0.0 to 31.7	0.0 to 127.7		0.0 to 123.7	0.0 to 127.7	0.0 to 2047.7 [1]	Input bit (in PII)
	124.0 to 127.7	–		124.0 to 127.7	–	–	... integrated inputs
IB	0 to 31	0 to 127		0 to 123	0 to 127	0 to 2047 [1]	Input byte (in PII)
	124 to 127	–		124 to 127	–	–	... integrated inputs
IW	0 to 30	0 to 127		0 to 122	0 to 126	0 to 2046 [1]	Input word (in PII)
	124 to 126	–		124 to 126	–	–	... integrated inputs
ID	0 to 28	0 to 124		0 to 120	0 to 124	0 to 2044 [1]	Input double word (in PII)
	124	–		124	–	–	... integrated inputs
L	0.0 to 255.7				0.0 to 255.7	0.0 to 8191.7 [2]	Local data bit
LB	0 to 255				0 to 255	0 to 8191 [2]	Local data byte
LW	0 to 254				0 to 254	0 to 8190 [2]	Local data word
LD	0 to 252				0 to 252	0 to 8188 [2]	Local data double word

[1] PII is preset to 256 byte
[2] Local data area is preset to 4096 byte

Addr. ID	Parameter Ranges							Description
	312*	**313**	**314**	**314***	**315**	**315-2 316-2**	**318-2**	
M	0.0 to 127.7	0.0 to 255.7			0.0 to 255.7		0.0 to 1023.0	Bit memory bit
MB	0 to 127	0 to 255			0 to 255		0 to 1023	Bit memory byte
MW	0 to 126	0 to 254			0 to 254		0 to 1022	Bit memory word
MD	0 to 124	0 to 252			0 to 252		0 to 1020	Bit memory double word
PQB	0 to 31	0 to 31	0 to 767	0 to 751	0 to 767	0 to 1023	0 to 8191	Peripheral output byte (direct I/O access)
	124							
	256 to 383	256 to 383						
PQW	0 to 30	0 to 30	0 to 766	0 to 750	0 to 766	0 to 1022	0 to 8190	Peripheral output word (direct I/O access)
	256 to 382	256 to 382						
PQD	0 to 28	0 to 28	0 to 764	0 to 748	0 to 764	0 to 1020	0 to 8188	Peripheral output double word (direct I/O access)
	256 to 380	256 to 380						

Addr. ID	Parameter Ranges							Description
	312*	**313**	**314**	**314***	**315**	**315-2 316-2**	**318-2**	
PIB	0 to 31	0 to 31	0 to 767	0 to 751	0 to 767	0 to 1023	0 to 8191	Peripheral input byte (direct I/O access)
	124 to 125							
	256 to 383	256 to 383						
PIW	0 to 30	0 to 30	0 to 766	0 to 750	0 to 766	0 to 1022	0 to 8190	Peripheral input word (direct I/O access)
	124							
	256 to 382	256 to 382						
PID	0 to 28	0 to 28	0 to 764	0 to 748	0 to 764	0 to 1020	0 to 8188	Peripheral input double word (direct I/O access)
	256 to 380	256 to 380						
T	0 to 63	0 to 127			0 to 127		0 to 511	Timer
Z	0 to 31	0 to 35	0 to 63		0 to 63		0 to 511	Counter
Parame-ter	–				–			Instruction addressed via parameter
B#16# W#16# DW#16#	–				–			Byte Word Double word hexadecimal

Addr. ID	Parameter Ranges							Description
	312*	313	314	314*	315	315-2 316-2	318-2	
D#	–				–			IEC data constant
L#	–				–			32-bit integer constant
P#	–				–			Pointer constant
S5T#	–				–			S5 time constant (16 bits) for loading of S5 timers
T#	–	1			1			Time constant (16/32 bits)
TOD#	–				–			IEC time constant
C#	–				–			Counter constant (BCD-codiert)
2#	–				–			Binary constant
B (b1,b2) B (b1,b2, b3,b4)	–				–			Constant, 2 or 4 Byte

1 T#1D_5H_3M_1S_2MS

Abbreviations and Mnemonics

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

Abbre-viations	Description	Example
k8	8-bit constant	32
k16	16-bit constant	62 531
k32	32-bit constant	127 624
i8	8-bit integer	–155
i16	16-bit integer	+6523
i32	32-bit integer	–2 222 222
m	P#x.y (pointer)	P#240.3
n	Binary constant	1001 1100
p	Hexadecimal constant	EA12
q	Real number (32-bit floating-point number)	12.34567E+5
LABEL	Symbolic jump address (max. 4 characters)	DEST
a	Byte address	2
b	Bit address	x.1
c	Operand range	I, Q, M, L, DBX, DIX

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

Registers

ACCU1 and ACCU2 (32 Bits)

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

CPU 318-2: also ACCU3 and ACCU4.

Accumulator designations:

ACCU	Bits
ACCUx (x = 1 to 4)	Bits 0 to 31
ACCUx-L	Bits 0 to 15
ACCUx-H	Bits 16 to 31
ACCUx-LL	Bits 0 to 7
ACCUx-LH	Bits 8 to 15
ACCUx-HL	Bits 16 to 23
ACCUx-HH	Bits 24 to 31

Address Registers AR1 and AR2 (32 Bits)

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

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

- Area-internal address

  `00000000 00000bbb bbbbbbbb bbbbbxxx`

- Area-crossing address

  `10000yyy` `00000bbb bbbbbbbb bbbbbxxx`

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

Status Word (16 Bits)

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

The status word is 16 bits long.

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

Examples of Addressing

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

Addressing Examples	Description
Direct Addressing	
A I 0.0	ANDing of input bit 0.0
L IB 1	Load input byte 1 into ACCU1
L IW 0	Load input word 0 into ACCU1
L ID 0	Load input double word 0 into ACCU1
Indirect Addressing of Timers/Counters	
SP T [LW 8]	Start timer; the timer number is in local word 8
CU C [LW 10]	Start counter; the counter number is in local data word 10
Area-Internal Memory-Indirect Addressing	
A I [LD 12] Example: L P#22.2 T LD 12 A I [LD 12]	AND operation: The address of the input is in local data double word 12 as pointer
A I [DBD 1]	AND operation: The address of the input is in data double word 1 of the DB as pointer
A Q [DID 12]	AND operation: The address of the output is in data double word 12 of the instance DB as pointer
A Q [MD 12]	AND operation: The address of the output is in memory marker double word 12 of the instance DB as pointer
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 AR1+ P#12.2”

<table>
<tr><th>Addressing Examples</th><th>Description</th></tr>
<tr><td colspan="2">Area-Crossing Register-Indirect Addressing</td></tr>
<tr><td colspan="2">For area-crossing register-indirect addressing, bits 24 to 26 of the address must also contain an area identifier. The address is in the address register.
<table>
<tr><th>Area identifier</th><th>Coding (binary)</th><th>Coding (hex.)</th><th>Area</th></tr>
<tr><td>P</td><td>1000 0000</td><td>80</td><td>I/O area</td></tr>
<tr><td>I</td><td>1000 0001</td><td>81</td><td>Input area</td></tr>
<tr><td>Q</td><td>1000 0010</td><td>82</td><td>Output area</td></tr>
<tr><td>M</td><td>1000 0011</td><td>83</td><td>Bit memory area</td></tr>
<tr><td>DB</td><td>1000 0100</td><td>84</td><td>Data area</td></tr>
<tr><td>DI</td><td>1000 0101</td><td>85</td><td>Instance data area</td></tr>
<tr><td>L</td><td>1000 0110</td><td>86</td><td>Local data area</td></tr>
<tr><td>VL</td><td>1000 0111</td><td>87</td><td>Predecessor local data (access to local data of invoking block see page 15)</td></tr>
</table></td></tr>
<tr><td>L B [AR1,P#8.0]</td><td>Load byte into ACCU1: The address is calculated from the “pointer value in AR1+ P#8.0”</td></tr>
<tr><td>A [AR1,P#32.3]</td><td>AND operation: The address of the operand is calculated from the “pointer value in AR1+ P#32.3”</td></tr>
<tr><td colspan="2">Addressing Via Parameters</td></tr>
<tr><td>A Parameter</td><td>Addressing via parameters</td></tr>
</table>

Examples of how to calculate the pointer

- **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)

Execution Times with Indirect Addressing

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

Two-Part Statement

A statement with indirectly addressed instructions consists of two parts:

Part 1: Load the address of the instruction

Part 2: Execute the instruction

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

Calculating the Execution Time

The total execution time is calculated as follows:

	Time required for loading the address
+	*execution time of the instruction*
=	*Total execution time of the instruction*

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

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

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

Address is in ...	Execution Time in µs			
	312*/313	**314/314***	**315/315-2/ 316-2**	**318-2**
Bit memory area M				
Word	1.7	0.7	0.8	0.2
Double word	3.5	2.3	2.1	0.3
Data block DB/DX				
Word	5.2	2.8	3.0	0.2
Double word	6.7	3.9	4.1	0.3
Local data area L				
Word	2.0	0.8	0.9	0.2
Double word	3.7	2.6	2.2	0.3
AR1/AR2 (area-internal)	3.0	1.9	1.7	0.0
AR1/AR2 (area-crossing)	4.9	3.9	3.2	0 .0
Parameter (word) ... for: • Timers • Counters • Block calls	4.0	2.5	2.1	0.2
Parameter (double word) ... for Bits, bytes, words and double words	7.3	5.3	4.3	0.3

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

Examples of Calculations (for the CPU 314)

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 314.

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

Example: A I [DBD 12]

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

Address is in ...	Execution Time in µs
Bit memory area M	
Word	0.7
Double word	2.3
Data block DB/DI	
Word	2.8
Double word	3.9

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

Typical Execution Time in µs	
Direct Addressing	**Indirect Addressing**
0.2	Time for A I 2.0+
:	:

Total execution time:

	3.9 ms
+	2.0 ms
=	5.9 ms

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

Example: A I [AR1, P#34.3]

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

Address is in ...	Execution Time in µs
:	:
AR1/AR2 (area-internal)	1.9
:	:

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

Typical Execution Time in µs	
Direct Addressing	**Indirect Addressing**
0.2	Time for A I 2.0+
:	:

Total execution time:

	1.9 ms
+	2.0 ms
=	3.9 ms

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

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

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

Address is in ...	Execution Time in µs
:	:
AR1/AR2 (area-crossing)	3.9
:	:

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

Typical Execution Time in µs	
Direct Addressing	**Indirect Addressing**
0.2	Time for A I 2.0+
:	:

Total execution time:

	3.9 ms
+	2.0 ms
=	5.9 ms

Execution Time for Addressing Via Parameters

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

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

Address is in ...	Execution Time in µs
:	:
:	:
Parameter (double word)	5.3

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

Typical Execution Time in µs	
Direct Addressing	**Indirect Addressing**
0.2	Time for A I 2.0+
:	:

Total execution time:

```
      5.3 ms
+     2.0 ms
=     7.3 ms
```

List of Instructions

This chapter contains the complete list of S7-300 instructions. The descriptions have been kept as concise as possible. You will find a detailed functional description in the various STEP 7 reference manuals.
Please note that, in the case of indirect addressing (examples see page 18), you must add the time required for loading the address of the particular instruction to the execution times listed (see page 22).

Bit Logic Instructions

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
A		AND									
	I/Q a.b	Input/output	1 [2]/2	0.7	0.2	0.3	0.1+	2.5+	2.0+	1.6+	0.1+
	M a.b	Bit memory	1 [2]/2	1.5	0.6	0.6	0.1+	2.7+	2.2+	1.7+	0.1+
	L a.b	Local data bit	2	2.2	0.8	0.9	0.1+	3.0+	2.2+	1.8+	0.1+
	DBX a.b	Data bit	2	5.2	2.7	2.8	0.1+	4.2+	2.8+	2.5+	0.1+
	DIX a.b	Instance data bit	2	5.2	2.7	2.8	0.1+	4.2+	2.8+	2.5+	0.1+
	c[AR1,m]	Register-ind., area-internal (AR1)	2	–	–	–	–	+	+	+	+
	c[AR2,m]	Register-ind., area-internal (AR2)	2	–	–	–	–	+	+	+	+
	[AR1,m]	Area-crossing via (AR1)	2	–	–	–	–	+	+	+	+
	[AR2,m]	Area-crossing via (AR2)	2	–	–	–	–	+	+	+	+
	Parameter	Via parameter	2	–	–	–	–	+	+	+	+

Status word for: **A**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	Yes	–	Yes	Yes
Instruction affects:	–	–	–	–	–	Yes	Yes	Yes	1

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing, Address area 0 to 127

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
AN		AND NOT									
	I/Q a.b	Input/output	1 [2]/2	1.4	0.5	0.5	0.1	2.9+	2.2+	1.9+	0.1+
	M a.b	Bit memory	1 [2]/2	1.9	0.7	0.8	0.1	3.1+	2.4+	2.1+	0.1+
	L a.b	Local data bit	2	2.5	0.9	1.0	0.1	3.4+	2.4+	2.2+	0.1+
	DBX a.b	Data bit	2	5.5	3.0	3.1	0.1	4.6+	2.9+	2.8+	0.1+
	DIX a.b	Instance data bit	2	5.5	3.0	3.1	0.1	4.6+	2.9+	2.8+	0.1+
	c[AR1,m]	Register-ind., area-internal (AR1)	2	–	–	–	–	+	+	+	+
	c[AR2,m]	Register-ind., area-internal (AR2)	2	–	–	–	–	+	+	+	+
	[AR1,m]	Area-crossing via (AR1)	2	–	–	–	–	+	+	+	+
	[AR2,m]	Area-crossing via (AR2)	2	–	–	–	–	+	+	+	+
	Parameter	Via parameter	2	–	–	–	–	+	+	+	+

Status word for: **AN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	Yes	–	Yes	Yes
Instruction affects:	–	–	–	–	–	Yes	Yes	Yes	1

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing, Address area 0 to 127

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
O		OR									
	I/Q a.b	Input/output	1 [2]/2	0.7	0.2	0.3	0.1	2.5+	2.0+	1.6+	0.1+
	M a.b	Bit memory	1 [2]/2	1.5	0.6	0.7	0.1	2.7+	2.2+	1.7+	0.1+
	L a.b	Local data bit	2	2.2	0.8	0.9	0.1	3.0+	2.2+	1.8+	0.1+
	DBX a.b	Data bit	2	5.2	2.7	2.9	0.1	4.2+	2.8+	2.5+	0.1+
	DIX a.b	Instance data bit	2	5.2	2.7	2.9	0.1	4.2+	2.8+	2.5+	0.1+
	c[AR1,m]	Register-ind., area-internal (AR1)	2	–	–	–	–	+	+	+	+
	c[AR2,m]	Register-ind., area-internal (AR2)	2	–	–	–	–	+	+	+	+
	[AR1,m]	Area-crossing via (AR1)	2	–	–	–	–	+	+	+	+
	[AR2,m]	Area-crossing via (AR2)	2	–	–	–	–	+	+	+	+
	Parameter	Via parameter	2	–	–	–	–	+	+	+	+

Status word for: **O**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	Yes
Instruction affects:	–	–	–	–	–	0	Yes	Yes	1

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
ON		OR NOT									
	I/Q a.b	Input/output	1 [2]/2	1.4	0.5	0.5	0.1	2.9+	2.2+	1.6+	0.1+
	M a.b	Bit memory	1[2]/2	1.9	0.7	0.8	0.1	3.1+	2.4+	2.0+	0.1+
	L a.b	Local data bit	2	2.5	0.9	1.0	0.1	3.4+	2.4+	2.2+	0.1+
	DBX a.b	Data bit	2	5.5	3.0	3.1	0.1	4.6+	2.9+	2.8+	0.1+
	DIX a.b	Instance data bit	2	5.5	3.0	3.1	0.1	4.6+	2.9+	2.8+	0.1+
	c[AR1,m]	Register-ind., area-internal (AR1)	2	–	–	–	–	+	+	+	+
	c[AR2,m]	Register-ind., area-internal (AR2)	2	–	–	–	–	+	+	+	+
	[AR1,m]	Area-crossing via (AR1)	2	–	–	–	–	+	+	+	+
	[AR2,m]	Area-crossing via (AR2)	2	–	–	–	–	+	+	+	+
	Parameter	Via parameter	2	–	–	–	–	+	+	+	+

Status word for: **ON**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	Yes
Instruction affects:	–	–	–	–	–	0	Yes	Yes	1

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
X		EXCLUSIVE OR									
	I/Q a.b	Input/output	2	0.7	0.2	0.3	0.1	2.5+	1.9+	1.6+	0.1+
	M a.b	Bit memory	2	1.5	0.6	0.7	0.1	2.7+	2.1+	1.7+	0.1+
	L a.b	Local data bit	2	2.2	0.8	0.9	0.1	3.0+	2.1+	1.9+	0.1+
	DBX a.b	Data bit	2	5.2	2.8	2.9	0.1	4.2+	2.6+	2.5+	0.1+
	DIX a.b	Instance data bit	2	5.2	2.8	2.9	0.1	4.2+	2.6+	2.5+	0.1+
	c[AR1,m]	Register-ind., area-internal (AR1)	2	–	–	–	–	+	+	+	+
	c[AR2,m]	Register-ind., area-internal (AR2)	2	–	–	–	–	+	+	+	+
	[AR1,m]	Area-crossing via (AR1)	2	–	–	–	–	+	+	+	+
	[AR2,m]	Area-crossing via (AR2)	2	–	–	–	–	+	+	+	+
	Parameter	Via parameter	2	–	–	–	–	+	+	+	+

Status word for: **X**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	Yes
Instruction affects:	–	–	–	–	–	0	Yes	Yes	1

[1] Plus time required for loading the address of the instruction (see page 22)

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
XN		EXCLUSIVE OR NOT									
	I/Q a.b	Input/output	2	1.4	0.5	0.5	0.1	2.9+	2.2+	1.9+	0.1+
	M a.b	Bit memory	2	1.9	0.7	0.8	0.1	3.1+	2.4+	2.0+	0.1+
	L a.b	Local data bit	2	2.5	0.9	1.0	0.1	3.4+	2.4+	2.2+	0.1+
	DBX a.b	Data bit	2	5.5	3.0	3.1	0.1	4.6+	2.9+	2.8+	0.1+
	DIX a.b	Instance data bit	2	5.5	3.0	3.1	0.1	4.6+	2.9+	2.8+	0.1+
	c[AR1,m]	Register-ind., area-internal (AR1)	2	–	–	–	–	+	+	+	+
	c[AR2,m]	Register-ind., area-internal (AR2)	2	–	–	–	–	+	+	+	+
	[AR1,m]	Area-crossing via (AR1)	2	–	–	–	–	+	+	+	+
	[AR2,m]	Area-crossing via (AR2)	2	–	–	–	–	+	+	+	+
	Parameter	Via parameter	2	–	–	–	–	+	+	+	+

Status word for: **XN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	Yes
Instruction affects:	–	–	–	–	–	0	Yes	Yes	1

1 Plus time required for loading the address of the instruction (see page 22)

Bit Logic Instructions with Parenthetical Expressions

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
A(		AND left parenthesis	1	2.9	1.7	1.7	0.1
AN(		AND NOT left parenthesis	1	2.9	1.7	1.7	0.1
O(		OR left parenthesis	1	2.9	1.4	1.7	0.1
ON(		OR NOT left parenthesis	1	2.9	1.4	1.7	0.1
X(		EXCLUSIVE OR left parenthesis	1	2.9	1.4	1.7	0.1
XN(		EXCLUSIVE OR NOT left parenthesis	1	2.9	1.4	1.7	0.1

Status word for: **A(, AN(, O(, ON(, X(, XN(**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	Yes	–	–	–	–	Yes	–	Yes	Yes
Instruction affects:	–	–	–	–	–	0	1	–	0

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
)		Right parenthesis, popping an entry off the nesting stack, gating the RLO with the current RLO in the processor	1	3.3	1.7	1.9	0.1

Status word for:)	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	Yes	–	–	–	–	Yes	1	Yes	1

ORing of AND Operations

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
O		ORing of AND operations according to the rule: AND before OR	1	1.4	0.3	0.5	0.1

Status word for: **O**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	Yes	–	Yes	Yes
Instruction affects:	–	–	–	–	–	Yes	1	–	Yes

Logic Instructions with Timers and Counters

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
A		AND									
	T	Timer	1 2/2	2.4	0.8	0.9	0.1	3.3+	2.2+	2.1+	0.1+
	C	Counter	1 2/2	1.7	0.6	0.6	0.1	3.0+	1.9+	1.8+	0.1+
	Timer para.	Timer/counter	2	–	–	–	–	+	+	+	+
	Counter p.	(addressed via parameter)		–	–	–	–	+	+	+	+

Status word for: **A**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	Yes	–	Yes	Yes
Instruction affects:	–	–	–	–	–	Yes	Yes	Yes	1

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
AN		AND NOT									
	T	Timer	1 [2]/2	3.0	1.0	1.1	0.1	3.7+	2.4+	2.3+	0.1+
	C	Counter	1 [2]/2	2.4	0.8	0.9	0.1	3.3+	2.2+	2.1+	0.1+
	Timer para.	Timer/counter	2	–	–	–	–	+	+	+	+
	Counter p.	(addressed via parameter)		–	–	–	–	+	+	+	+

Status word for: **AN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	Yes	–	Yes	Yes
Instruction affects:	–	–	–	–	–	Yes	Yes	Yes	1

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
O	T C	OR timer OR counter	1 [2]/2 1 [2]/2	2.4 1.7	0.8 0.6	0.9 0.6	0.1 0.1	3.3+ 3.0+	2.2+ 1.9+	2.1+ 1.8+	0.1+ 0.1+
	Timerpara. Counter p.	OR timer/counter (addressed via parameter)	2	– –	– –	– –	– –	+ +	+ +	+ +	+ +
ON	T C	OR NOT timer OR NOT counter	1 [2]/2 1 [2]/2	3.0 2.4	1.0 0.8	1.1 0.9	0.1 0.1	3.7+ 3.3+	2.4+ 2.2+	2.3+ 2.1+	0.1+ 0.1+
	Timerpara. Counter p.	OR NOT timer/counter (addressed via parameter)	2	– –	– –	– –	– –	+ +	+ +	+ +	+ +
X	T C	EXCLUSIVE OR timer EXCLUSIVE OR counter	2 2	2.4 1.7	0.8 0.6	0.9 0.6	0.1 0.1	3.3+ 3.0+	2.2+ 1.9+	2.1+ 1.8+	0.1+ 0.1+
	Timerpara. Counter p.	EXCLUSIVE OR timer/counter (addressed via parameter)	2 2	– –	– –	– –	– –	+ +	+ +	+ +	+ +

Status word for: **O, ON, X**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	Yes
Instruction affects:	–	–	–	–	–	0	Yes	Yes	1

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
XN	T C	EXCLUSIVE OR NOT timer EXCLUSIVE OR NOT counter	2 2	3.0 2.4	1.0 1.0	1.1 0.9	0.1 0.1	3.7+ 3.3+	2.4+ 1.2+	2.3+ 2.1+	0.1+ 0.1+
	Timerpara. Counter p.	EXCLUSIVE OR NOT timer/co-unter (addressed via parameter)	2	– –	– –	– –	– –	+ +	+ +	+ +	+ +

Status word for: **XN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	Yes
Instruction affects:	–	–	–	–	–	0	Yes	Yes	1

1 Plus time required for loading the address of the instruction (see page 22)

Word Logic Instructions with the Contents of Accumulator 1

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
AW		AND ACCU2-L	1	1.7	0.5	0.6	0.1
AW	k16	AND 16-bit constant	2	2.3	0.7	0.9	0.1
OW		OR ACCU2-L	1	1.7	0.5	0.6	0.1
OW	k16	OR 16-bit constant	2	2.3	0.7	0.9	0.1
XOW		EXCLUSIVE OR ACCU2-L	1	1.7	0.5	0.6	0.1
XOW	k16	EXCLUSIVE OR 16-bit constant	2	2.3	0.7	0.9	0.1
AD		AND ACCU2	1	3.4	1.9	2.0	0.1
AD	k32	AND 32-bit constant	3	4.1	2.1	2.3	0.15

Status word for: **AW, OW, XOW, AD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	Yes	0	0	–	–	–	–	–

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
OD		OR ACCU2	1	3.4	1.9	2.0	0.1
OD	k32	OR 32-bit constant	3	4.1	2.1	2.3	0.15
XOD		EXCLUSIVE OR ACCU2	1	3.4	1.9	2.0	0.1
XOD	k32	EXCLUSIVE OR 32-bit constant	3	4.1	2.1	2.3	0.15

Status word for: **OD, XOD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	Yes	0	0	–	–	–	–	–

Evaluating Conditions Using AND, OR and EXCLUSIVE OR

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312* 313	314 314*	315/315-2/ 316-2	318-2
A	==0	AND Result=0 (CC 1=0)and (CC 0=0)	1	1.5	0.5	0.6	0.1
	>0	Result>0 (CC 1=1) and (CC 0=0)	1	2.3	0.7	0.9	0.1
	<0	Result<0 (CC 1=0)and (CC 0=1)	1	2.3	0.7	0.9	0.1
	<>0	Result≠0 ((CC 1=0)and(CC 0=1)or (CC1=1)and(CC 0=0))	1	1.5	0.5	0.6	0.1
	<=0	Result<=0 ((CC 1=0) and (CC 0=1) or (CC1=0) and (CC 0=0))	1	1.5	0.5	0.6	0.1
	>=0	Result>=0 ((CC 1=1) and (CC 0=0) or (CC1=0) and (CC 0=0))	1	1.5	0.5	0.6	0.1

Status word for: **A**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	Yes	–	Yes	Yes
Instruction affects:	–	–	–	–	–	Yes	Yes	Yes	1

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs 312* 313	314 314*	315/315-2/ 316-2	318-2
A		AND	1	1.5	0.5	0.6	0.1
	UO	unordered math instruction (CC 1=1) and (CC 0=1)	1	0.7	0.2	0.3	0.1
	OS	AND OS=1	1	0.7	0.2	0.3	0.1
	BR	AND BR=1	1	0.7	0.2	0.3	0.1
	OV	AND OV=1	1	0.7	0.2	0.3	0.1

Status word for: **A**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	Yes	–	Yes	Yes
Instruction affects:	–	–	–	–	–	Yes	Yes	Yes	1

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312* 313	314 314*	315/315-2/ 316-2	318-2
AN	==0	AND NOT Result=0 (CC 1=0) and (CC 0=0)	1	1.5	0.5	0.6	0.1
	>0	Result>0 (CC 1=1) and (CC 0=0)	1	2.3	0.7	0.9	0.1
	<0	Result<0 (CC 1=0) and (CC 0=1)	1	2.3	0.7	0.9	0.1
	<>0	Result≠0 ((CC 1=0) and (CC 0=1) or (CC 1=1) and (CC 0=0))	1	2.3	0.7	0.9	0.1
	<=0	Result<=0 ((CC 1=0) and (CC 0=1) or (CC 1=0) and (CC 0=0))	1	0.7	0.2	0.3	0.1
	>=0	Result>=0 ((CC 1=1) and (CC 0=0) or (CC 1=0) and (CC 0=0))	1	0.7	0.7	0.3	0.1
	UO	unordered math instruction (CC 1=1) and (CC 0=1)	1	2.3	0.7	0.9	0.1
	OS	OS=1	1	1.5	0.5	0.6	0.1
	BR	BR=1	1	1.5	0.5	0.6	0.1
	OV	OV=1	1	1.5	0.5	0.6	0.1

Status word for: **AN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	Yes	–	Yes	Yes
Instruction affects:	–	–	–	–	–	Yes	Yes	Yes	1

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312* 313	314 314*	315/315-2/ 316-2	318-2
O	==0	OR Result=0 (CC 1=0) and (CC 0=0)	1	1.5	0.5	0.6	0.1
	>0	Result>0 (CC 1=1) and (CC 0=0)	1	2.3	0.7	0.9	0.1
	<0	Result<0 (CC 1=0) and (CC 0=1)	1	2.3	0.7	0.9	0.1
	<>0	Result≠0 ((CC 1=0) and (CC 0=1) or (CC 1=1) and (CC 0=0))	1	1.5	0.5	0.6	0.1
	<=0	Result<=0 ((CC 1=0) and (CC 0=1) or (CC 1=0) and (CC 0=0))	1	1.5	0.5	0.6	0.1
	>=0	Result>=0 ((CC 1=1) and (CC 0=0) or (CC 1=0) and (CC 0=0))	1	1.5	0.5	0.6	0.1
	UO	unordered math instruction (CC 1=1) and (CC 0=1)	1	1.5	0.5	0.6	0.1
	OS	OS=1	1	0.7	0.2	0.3	0.1
	BR	BR=1	1	0.7	0.2	0.3	0.1
	OV	OV=1	1	0.7	0.2	0.3	0.1

Status word for: **O**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	–	–	Yes	Yes
Instruction affects:	–	–	–	–	–	0	Yes	Yes	1

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312* 313	314 314*	315/315-2/ 316-2	318-2
ON	==0	OR NOT Result=0 (CC 1=0) and (CC 0=0)	1	1.5	0.5	0.6	0.1
	>0	Result>0 (CC 1=1) and (CC 0=0)	1	2.3	0.7	0.9	0.1
	<0	Result<0 (CC 1=0) and (CC 0=1)	1	2.3	0.7	0.9	0.1
	<>0	Result≠0 ((CC 1=0) and (CC 0=1) or (CC 1=1) and (CC 0=0))	1	2.3	0.7	0.9	0.1
	<=0	Result<=0 ((CC 1=0) and (CC 0=1) or (CC 1=0) and (CC 0=0))	1	0.7	0.2	0.3	0.1
	>=0	Result>=0 ((CC 1=1) and (CC 0=0) or (CC 1=0) and (CC 0=0))	1	0.7	0.7	0.3	0.1
	UO	unordered math instruction (CC 1=1) and (CC 0=1)	1	2.3	0.7	0.9	0.1
	OS	OS=1	1	1.5	0.5	0.6	0.1
	BR	BR=1	1	1.5	0.5	0.6	0.1
	OV	OV=1	1	1.5	0.5	0.6	0.1

Status word for: **ON**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	–	–	Yes	Yes
Instruction affects:	–	–	–	–	–	0	Yes	Yes	1

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312* 313	314 314*	315/315-2/ 316-2	318-2
X	==0	EXCLUSIVE OR Result=0 (CC 1=0) and (CC 0=0)	1	1.5	0.5	0.6	0.1
	>0	Result>0 (CC 1=1) and (CC 0=0)	1	2.3	0.7	0.9	0.1
	<0	Result<0 (CC 1=0) and (CC 0=1)	1	2.3	0.7	0.9	0.1
	<>0	Result≠0 ((CC 1=0) and (CC 0=1) or (CC 1=1) and (CC 0=0))	1	1.5	0.5	0.6	0.1
	<=0	Result<=0 ((CC 1=0) and (CC 0=1) or (CC 1=0) and (CC 0=0))	1	1.5	0.5	0.6	0.1
	>=0	Result>=0 ((CC 1=1) and (CC 0=0) or (CC 1=0) and (CC 0=0))	1	1.5	0.5	0.6	0.1
	UO	unordered math instruction (CC 1=1) and (CC 0=1)	1	1.5	0.5	0.6	0.1
	OS	OS=1	1	0.7	0.2	0.3	0.1
	BR	BR=1	1	0.7	0.2	0.3	0.1
	OV	OV=1	1	0.7	0.2	0.3	0.1

Status word for: **X**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	–	–	Yes	Yes
Instruction affects:	–	–	–	–	–	0	Yes	Yes	1

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312* 313	314 314*	315/315-2/ 316-2	318-2
XN	==0	EXCLUSIVE OR NOT Result=0 (CC 1=0) and (CC 0=0)	1	1.5	0.5	0.6	0.1
	>0	Result>0 (CC 1=1) and (CC 0=0)	1	2.3	0.7	0.9	0.1
	<0	Result<0 (CC 1=0) and (CC 0=1)	1	2.3	0.7	0.9	0.1
	<>0	Result≠0 ((CC 1=0) and (CC 0=1) or (CC 1=1) and (CC 0=0))	1	2.3	0.7	0.9	0.1
	<=0	Result<=0 ((CC 1=0) and (CC 0=1) or (CC 1=0) and (CC 0=0))	1	0.7	0.2	0.3	0.1
	>=0	Result>=0 ((CC 1=1) and (CC 0=0) or (CC 1=0) and (CC 0=0))	1	0.7	0.7	0.3	0.1
	UO	unordered math instruction (CC 1=1) and (CC 0=1)	1	2.3	0.7	0.9	0.1
	OS	OS=1	1	1.5	0.5	0.6	0.1
	BR	BR=1	1	1.5	0.5	0.6	0.1
	OV	OV=1	1	1.5	0.5	0.6	0.1

Status word for: **XN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	–	–	Yes	Yes
Instruction affects:	–	–	–	–	–	0	Yes	Yes	1

Edge-Triggered Instructions

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
FP	I/Q a.b	Detecting the positive edge in the RLO. The bit addressed in the instruction is the auxiliary edge bit memory.	2	2.0	0.7	0.8	0.2	3.6+	2.7+	2.4+	0.2+
	M a.b		2	3.5	1.4	1.5	0.2	3.9+	2.9+	2.7+	0.2+
	L a.b		2	3.8	1.5	1.6	0.2	4.1+	2.9+	2.7+	0.2+
	DBX a.b		2	6.7	2.0	4.0	0.2	5.7+	3.7+	3.6+	0.2+
	DIX a.b		2	6.7	2.0	4.0	0.2	5.7+	3.7+	3.6+	0.2+
	c[AR1,m]		2	–	–	–	–	+	+	+	+
	c[AR2,m]		2	–	–	–	–	+	+	+	+
	[AR1,m]		2	–	–	–	–	+	+	+	+
	[AR2,m]		2	–	–	–	–	+	+	+	+
	Parameter		2	–	–	–	–	+	+	+	+

Status word for: **FP**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	–	0	Yes	Yes	1

[1] Plus time required for loading the address of the instruction (see page 22)

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
FN	I/Q a.b	Detecting the negtive edge in the RLO. The bit addressed in the intruction is the auxiliary edge bit memory.	2	2.6	0.9	1.0	0.2	3.8+	2.9+	2.6+	0.2+
	M a.b		2	3.8	1.6	1.6	0.2	4.1+	3.1+	2.8+	0.2+
	L a.b		2	4.2	1.7	1.7	0.2	4.3+	3.1+	2.8+	0.2+
	DBX a.b		2	6.8	2.2	4.1	0.2	5.8+	4.0+	3.7+	0.2+
	DIX a.b		2	6.8	2.2	4.1	0.2	5.8+	4.0+	3.7+	0.2+
	c[AR1,m]		2	–	–	–	–	+	+	+	+
	c[AR2,m]		2	–	–	–	–	+	+	+	+
	[AR1,m]		2	–	–	–	–	+	+	+	+
	[AR2,m]		2	–	–	–	–	+	+	+	+
	Parameter		2	–	–	–	–	+	+	+	+

Status word for: **FN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	–	0	Yes	Yes	1

[1] Plus time required for loading the address of the instruction (see page 22)

Setting/Resetting Bit Addresses

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

In-struc-tion	Address Identifier		Description	Length in Words	Typical Execution Time in µs							
					Direct Addressing				Indirect Addressing [1]			
					312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
S	I/Q	a.b	Set input/output to “1”	1 [2]/2	0.7	0.3	0.3	0.2	3.3+	2.2+	2.2+	0.2+
			(MCR-dependent)		1.4	1.4	0.5	0.2	4.4+	3.4+	2.9+	0.2+
	M	a.b	Set bit memory to “1”	1 [2]/2	1.9	0.8	0.8	0.2	3.7+	2.4+	2.5+	0.2+
			(MCR-dependent)		3.9	3.0	2.3	0.2	4.4+	3.6+	3.0+	0.2+
	L	a.b	Set local data bit to “1”	2	3.0	1.2	1.3	0.2	3.8+	2.4+	2.5+	0.2+
			(MCR-dependent)		4.9	3.1	2.9	0.2	3.9+	3.6+	2.5+	0.2+
	DBX	a.b	Set data bit to “1”	2	6.2	3.3	3.7	0.2	5.5+	3.3+	3.5+	0.2+
			(MCR-dependent)		7.3	4.5	4.3	0.2	6.6+	4.4+	4.1+	0.2+
	DIX	a.b	Set instance data bit to “1”	2	6.2	3.3	3.7	0.2	5.5+	3.3+	3.5+	0.2+
			(MCR-dependent)		7.3	4.5	4.3	0.2	6.6+	4.4+	4.1+	0.2+
	c[AR1,m]		Register–ind., area–internal (AR1)	2	–	–	–	–	+	+	+	+
	c[AR2,m]		Register–ind., area–internal (AR2)	2	–	–	–	–	+	+	+	+
	[AR1,m]		Area-crossing via (AR1)	2	–	–	–	–	+	+	+	+
	[AR2,m]		Area-crossing via (AR2)	2	–	–	–	–	+	+	+	+
	Parameter		Via parameter	2	–	–	–	–	+	+	+	+

Status word for: **S**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	–	0	Yes	–	0

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing

In-struc-tion	Address Identifier		Description	Length in Word	Typical Execution Time in µs							
					Direct Addressing				Indirect Addressing [1]			
					312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
R	I/Q	a.b	Reset input/output to “0”	1 [2]/2	1.0	0.4	0.4	0.2	3.5+	2.4+	2.3+	0.2+
			(MCR-dependent)		1.4	1.4	0.5	0.2	4.6+	3.5+	3.0+	0.2+
	M	a.b	Set bit memory to “0”	1 [2]/2	2.2	0.9	0.9	0.2	3.8+	2.5+	2.6+	0.2+
			(MCR-dependent)		4.1	3.1	2.4	0.2	4.6+	3.7+	3.2+	0.2+
	L	a.b	Set local data bit to“ 0”	2	3.0	1.2	1.3	0.2	4.0+	2.5+	2.6+	0.2+
			(MCR-dependent)		5.1	3.2	3.0	0.2	4.1+	3.7+	2.7+	0.2+
	DBX	a.b	Set data bit to “0”	2	6.4	3.5	3.8	0.2	5.7+	3.4+	3.6+	0.2+
			(MCR-dependent)		7.3	4.6	4.3	0.2	6.7+	4.5+	4.3+	0.2+
	DIX	a.b	Set instance data bit to “0”	2	6.4	3.5	3.8	0.2	5.7+	3.4+	3.6+	0.2+
			(MCR-dependent)		7.3	4.6	4.3	0.2	6.7+	4.5+	4.3+	0.2+
	c[AR1,m]		Register–ind., area–internal (AR1)	2	–	–	–	–	+	+	+	+
	c[AR2,m]		Register–ind., area–internal (AR2)	2	–	–	–	–	+	+	+	+
	[AR1,m]		Area-crossing via (AR1)	2	–	–	–	–	+	+	+	+
	[AR2,m]		Area-crossing via (AR2)	2	–	–	–	–	+	+	+	+
	Parameter		Via parameter	2	–	–	–	–	+	+	+	+

Status word for: **R**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	–	0	Yes	–	0

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing

In-struc-tion	Address Identifier		Description	Length in Words	Typical Execution Time in µs							
					Direct Addressing				Indirect Addressing [1]			
					312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
=	I/Q	a.b	Assign RLO to input/output	1 [2]/2	0.7	0.2	0.3	0.2	3.3+	2.2+	2.2+	0.2+
			(MCR-dependent)		1.4	1.4	0.5	0.2	4.4+	3.4+	2.9+	0.2+
	M	a.b	Assign RLO to bit memory	1 [2]/2	2.2	0.9	0.9	0.2	3.7+	2.4+	2.5+	0.2+
			(MCR-dependent)		3.9	3.0	2.3	0.2	4.4+	3.6+	3.0+	0.2+
	L	a.b	Assign RLO to local data bit	2	2.7	1.0	1.1	0.2	3.8+	2.4+	2.5+	0.2+
			(MCR-dependent)		4.6	3.1	2.6	0.2	3.6+	3.6+	2.3+	0.2+
	DBX	a.b	Assign RLO to data bit	2	6.4	3.3	3.8	0.2	5.7+	3.3+	3.6+	0.2+
			(MCR-dependent)		7.5	5.3	4.4	0.2	6.7+	5.3+	4.3+	0.2+
	DIX	a.b	Assign RLO to instance data bit	2	6.4	3.3	3.8	0.2	5.7+	3.3+	3.6+	0.2+
			(MCR-dependent)		7.5	5.3	4.4	0.2	6.7+	5.3+	4.3+	0.2+
	c[AR1,m]		Register–ind., area–internal(AR1)	2	–	–	–	–	+	+	+	+
	c[AR2,m]		Register–ind., area–internal(AR2)	2	–	–	–	–	+	+	+	+
	[AR1,m]		Area-crossing via (AR1)	2	–	–	–	–	+	+	+	+
	[AR2,m]		Area-crossing via (AR2)	2	–	–	–	–	+	+	+	+
	Parameter		Via parameter	2	–	–	–	–	+	+	+	+

Status word for: =	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	–	0	Yes	–	0

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing

Instructions Directly Affecting the RLO

The following instructions have a direct effect on the RLO.

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315–2/ 316–2	318-2
CLR		Set RLO to ”0”	1	0.7	0.2	0.3	0.1

Status word for: **CLR**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	–	–	–	–	0	0	0	0

SET		Set RLO to ”1”	1	0.7	0.2	0.3	0.1

Status word for: **SET**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	–	–	–	–	0	1	1	0

NOT		Negate RLO	1	0.7	0.2	0.3	0.1

Status word for: **NOT**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	Yes	–	Yes	–
Instruction affects:	–	–	–	–	–	–	1	Yes	–

SAVE		Save RLO to the BR bit	1	0.7	0.2	0.3	0.1

Status word for: **SAVE**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	Yes	–	–	–	–	–	–	–	–

Timer Instructions

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
SP	T f	Start timer as pulse on edge change from “0” to “1”	1 [2]/2	14.0	8.4	9.2	0.2	14.3+	8.8+	9.7+	0.2+
	Timer para.		2	–	–	–	–	+	+	+	+
SE	T f	Start timer as exded pulse on edge change from “0” to “1”	1 [2]/2	14.0	8.4	9.2	0.2	14.3+	8.8+	9.7+	0.2+
	Timer para.		2	–	–	–	–	+	+	+	+
SD	T f	Start timer as ON delay on edge change from “0” to “1”	1 [2]/2	14.7	9.0	9.7	0.2	15.0+	9.4+	10.2+	0.2+
	Timer para		2	–	–	–	–	+	+	+	+
SS	T f	Start timer as retive ON delay on edge change from “0” to “1”	1 [2]/2	14.7	9.0	9.7	0.2	15.0+	9.4+	10.2+	0.2+
	Timer para.		2	–	–	–	–	+	+	+	+
SF	T f	Start timer as OFF delay on edge change from “1” to “0”	1 [2]/2	15.0	9.2	10.0	0.2	15.4+	9.6+	10.5+	0.2+
	Timer para.		2	–	–	–	–	+	+	+	+

Status word for: **SP, SE, SD, SS, SF**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	–	0	–	–	0

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
FR	T f	Enable timer for restarting on edge change from “0” to “1” (reset edge bit memory for starting timer)	1 [2]/2	3.9	2.0	2.1	0.2	4.3+	2.5+	2.7+	0.2+
	Timer para.		2	–	–	–	–	+	+	+	+
R	T f	Reset timer	1 [2]/2	3.5	1.8	1.8	0.2	3.8+	2.2+	2.4+	0.2+
	Timer para.		2	–	–	–	–	+	+	+	+

Status word for: **FR, R**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	–	0	–	–	0

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing

Counter Instructions

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
S	C f	Presetting of counter on edge change from “0” to “1”	1 [2]/2	9.8	6.0	6.6	0.2	10.2+	6.4+	7.1+	0.2+
	Counter p.		2	–	–	–	–	+	+	+	+
R	C f	Reset counter to “0”	1 [2]/2	3.4	1.7	1.8	0.2	3.8+	2.2+	2.3+	0.2+
	Counter p.		2	–	–	–	–	+	+	+	+
CU	C f	Increment counter by 1 on edge change from “0” to “1”	1 [2]/2	4.8	2.6	2.8	0.2	5.2+	3.1+	3.4+	0.2+
	Counter p.		2	–	–	–	–	+	+	+	+
CD	C f	Decrement counter by 1 on edge change from “0” to “1”	1 [2]/2	5.1	2.8	3.0	0.2	5.3+	3.2+	3.5+	0.2+
	Counter p.		2	–	–	–	–	+	+	+	+

Status word for: **S, R, CU, CD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	–	0	–	–	0

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
FR	C f	Enable counter on edge change from "0" to "1" (reset edge bit memory for up and down counting)	1 [2]/2	4.0	2.1	2.2	0.2	4.3+	2.5+	2.7	0.2+
	Counter p.		2	–	–	–	–	+	+	+	+

Status word for: **FR**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	–	0	–	–	0

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing

Load Instructions

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
L		Load ...									
	IB a	Input byte	1 [2]/2	1.7	0.6	0.6	0.1	2.7+	2.2+	1.7+	0.1+
	QB a	Output byte	1 [2]/2	1.7	0.6	0.6	0.1	2.7+	2.2+	1.7+	0.1+
	PIB a	Peripheral input byte	2	< 30/ < 21 [3]	< 24/ < 53[3]/ < 88[4]	<24	0.1	< 40/ < 30 [3]	26+/ <58 [3]/ <92 [4]	< 27	0.1+
	MB a	Bit memory byte	1 [2]/2	1.9	0.7	0.8	0.1	2.8+	2.2+	1.8+	0.1+
	LB a	Local data byte	2	2.9	1.0	1.1	0.1	3.2+	2.2+	2.0+	0.1+
	DBB a	Data byte	2	5.0	2.8	2.8	0.1	4.3+	2.8+	2.6+	0.1+
	DIB a	Instance data byte	2	6.8	2.8	2.8	0.1	4.3+	2.8+	2.6+	0.1+
		... into ACCU1	2								
	g[AR1,m]	Register–ind., area–internal (AR1)	2	–	–	–	–	+	+	+	+
	g[AR2,m]	Register–ind., area–internal (AR2)	2	–	–	–	–	+	+	+	+
	B[AR1,m]	Area-crossing (AR1)	2	–	–	–	–	+	+	+	+
	B[AR2,m]	Area-crossing (AR2)	2	–	–	–	–	+	+	+	+
	Parameter	Via parameter	2	–	–	–	–	+	+	+	+

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing
3 Integrated digital I/O for CPU 312* and 314*
4 Integrated analog I/O for CPU 314*

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
L		Load ...									
	IW a	Input word	1 [2]/2	2.4	0.8	0.9	0.1	2.9+	2.1+	1.9+	0.1+
	QW a	Output word	1 [2]/2	2.4	0.8	0.9	0.1	2.9+	2.1+	1.9+	0.1+
	PIW a	Peripheral input word		< 40	< 29/ <53[3]/ <88[4]	< 30	0.1	< 46/< 30 [3]	30+/ <58[3]/ <92[4]	< 32	0.1+
	MW a	Bit memory word	2	2.7	1.0	1.1	0.1	3.2+	2.4+	2.1+	0.1+
	LW a	Local data word	2	3.0	1.1	1.3	0.1	3.7+	2.8+	2.3+	0.1+
	DBW a	Data word	1 [2]/2	5.7	3.3	3.3	0.1	5.2+	3.7+	3.2+	0.1+
	DIW a	Instance data word	1 [2]/2	5.7	3.3	3.3	0.1	5.2+	3.7+	3.2+	0.1+
		... into ACCU1									
	h[AR1,m]	Register–ind., area–internal (AR1)	2	–	–	–	–	+	+	+	+
	h[AR2,m]	Register–ind., area–internal (AR2)	2	–	–	–	–	+	+	+	+
	W[AR1,m]	Area-crossing via (AR1)	2	–	–	–	–	+	+	+	+
	W[AR2,m]	Area-crossing via (AR2)	2	–	–	–	–	+	+	+	+
	Parameter	Via parameter	2	–	–	–	–	+	+	+	+

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing
3 Integrated digital I/O for CPU 312* and 314*
4 Integrated analog I/O for CPU 314*

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
L		Load ...									
	ID a	Input double word	1 [2]/2	2.9	0.9	1.1	0.2	3.2+	2.4+	2.1+	0.2+
	QD a	Output double word	1 [2]/2	2.9	0.9	1.1	0.2	3.2+	2.4+	2.1+	0.2+
	PID a	Peripheral input double word	2	< 45	37/ <190[3]	< 40	0.2	< 65	39+/ <200[3]	< 42	0.2+
	MD a	Bit memory double word	1 [2]/2	3.4	1.4	1.5	0.2	3.7+	2.7+	2.5+	0.2+
	LD a	Local data double word	2	3.7	1.5	1.6	0.2	4.2+	3.1+	2.7+	0.2+
	DBD a	Data double word	2	7.0	4.4	4.3	0.2	6.5+	4.7+	4.2+	0.2+
	DID a	Instance data double word ... into ACCU1	2	7.0	4.4	4.3	0.2	6.5+	4.7+	4.2+	0.2+
	i[AR1.m]	Register–ind., area–internal (AR1)	2	–	–	–	–	+	+	+	+
	i[AR2,m]	Register–ind., area–internal (AR2)	2	–	–	–	–	+	+	+	+
	D[AR1.m]	Area-crossing via (AR1)	2	–	–	–	–	+	+	+	+
	D[AR2,m]	Area-crossing via (AR2)	2	–	–	–	–	+	+	+	+
	Parameter	Via parameter	2	–	–	–	–	+	+	+	+

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing
3 Integrated analog I/O for CPU 314*

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
L		Load ...									
	k8	8-bit constant into ACCU1-LL	1	1.7	0.6	0.6	0.1	–	–	–	–
	k16	16-bit constant into ACCU1-L	2	1.7	0.6	0.6	0.1	–	–	–	–
	k32	32-bit constant into ACCU1	3	2.0	0.7	0.8	0.15	–	–	–	–
	Parameter	Load constant into ACCU1 (addressed via parameter)	2	–	–	–	–	+	+	+	+
L	2#n	Load 16-bit binary constant into ACCU1-L	2	1.7	0.6	0.6	0.1	–	–	–	–
		Load 32-bit binary constant into ACCU1	3	2.0	0.7	0.7	0.15	–	–	–	–
L	B#8#p	Load 8-bit hexadecimal constant into ACCU1-L	1	1.7	0.6	0.6	0.1	–	–	–	–
	W#16#p	Load 16-bit hexadecimal constant into ACCU1-L	2	1.7	0.6	0.6	0.1	–	–	–	–
	DW#16#p	Load 32-bit hexadecimal constant into ACCU1-L	3	2.0	0.7	0.7	0.15	–	–	–	–

[1] Plus time required for loading the address of the instruction (see page 22)

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
L	'x'	Load 1 characters		1.2	0.6	0.7	0.1
L	'xx'	Load 2 characters	2	1.2	0.6	0.7	0.1
L	'xxx'	Load 3 characters		1.4	0.7	0.88	0.15
L	'xxxx'	Load 4 characters	3	1.4	0.7	0.88	0.15
L	D# date	Load IEC date (BCD)	3	1.2	0.6	0.8	0.15
L	S5T# time value	Load S5 time constant (16 bits)	2	1.2	0.6	0.8	0.1
L	TOD# time value	Load 32-bit time constant IEC time of day	3	1.4	0.93	0.88	0.15
L	T# time value	Load 16-bit timer constant	2	1.2	0.7	0.88	0.1
		Load 32-bit timer constant	3	1.4	0.6	0.88	0.15
L	C# count value	Load 16-bit counter constant	2	1.2	0.6	0.88	0.1
L	P# bit pointer	Load bit pointer	3	1.4	0.7	0.88	0.15
L	L# integer	Load 32 bit integer constant	3	1.4	0.7	0.88	0.15
L	Real number	Load real number	3	1.4	0.93	0.88	0.15

Load Instructions for Timers and Counters

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
L	T f	Load time value	1 [2]/2	3.1	1.6	1.7	0.1	5.2+	0.8+	2.1+	0.1+
	Timer para.	Load time value (addressed via parameter)	2	–	–	–	–	+	+	+	+
L	C f	Load count value	1 [2]/2	2.9	1.6	1.5	0.1	5.2+	0.8+	2.1+	0.1+
	Counter para.	Load count value (addressed via parameter)	2	–	–	–	–	+	+	+	+
LD	T f	Load time value in BCD	1 [2]/2	8.1	5.4	5.4	0.3	15.6+	4.6+	5.9+	0.3+
	Timer para.	Load time value in BCD (addressed via parameter)	2	–	–	–	–	+	+	+	+
LD	C f	Load count value in BCD	1 [2]/2	7.4	5.0	4.9	0.3	14.2+	4.2+	5.4+	0.3+
	Counter para.	Load count value (addressed via parameter)	2	–	–	–	–	+	+	+	+

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing

Transfer Instructions

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

Instruction	Address Identifier		Description	Length in Words	Typical Execution Time in µs							
					Direct Addressing				Indirect Addressing [1]			
					312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
T			Transfer contents of ACCU1-LL to ...									
	IB	a	input byte	1 [2]/2	0.7	0.2	0.3	0.1	2.0+	1.6+	1.2+	0.1+
			(MCR-dependent)		2.6	1.3	1.4	0.1	2.5+	2.0+	1.6+	0.1+
	QB	a	output byte	1 [2]/2	0.7	0.2	0.3	0.1	2.0+	1.6+	1.2+	0.1+
			(MCR-dependent)		2.6	1.3	1.4	0.1	2.5+	2.0+	1.6+	0.1+
	PQB	a	peripheral output byte	1 [3]/2	< 30	24/ <40[4]/ <47[5]	< 24	0.1	<35.5/ < 19 [4]	25+/ <45 [4]/ < 50 [5]	< 27	0.1+
			(MCR-dependent)		< 32	25/ <41[4]/ <48[5]	< 25	0.1	<36.5/ < 20 [4]	26+/ < 46 [4] < 51 [5]	< 28	0.1+

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing
3 Direct addressing with PQB 0 to 255
4 Integrated digital I/O for CPU 312* and 314*
5 Integrated analog I/O for CPU 314*

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
T	MB a	bit memory byte	1 [2]/2	0.9	0.4	0.4	0.1	2.2+	1.7+	1.3+	0.1+
		(MCR-dependent)		2.7	1.8	1.5	0.1	2.7+	2.2+	1.7+	0.1+
	LB a	local data byte	2	1.5	0.6	0.6	0.1	2.5+	1.8+	1.5+	0.1+
		(MCR-dependent)		3.1	2.2	1.8	0.1	3.3+	2.4+	2.0	0.1+
	DBB a	data byte	2	4.6	2.9	2.5	0.1	3.9+	2.7+	2.3+	0.1+
		(MCR-dependent)		5.4	3.5	3.0	0.1	4.7+	3.3+	2.8+	0.1+
	DIB a	instance data byte	2	4.6	2.9	2.5	0.1	3.9+	2.7+	2.3+	0.1+
		(MCR-dependent)		5.4	3.5	3.0	0.1	4.7+	3.3+	2.8+	0.1+
T	g[AR1,m]	Register–ind., area–internal (AR1)	2	–	–	–	–	+	+	+	+
	g[AR2,m]	Register–ind., area–internal (AR2)	2	–	–	–	–	+	+	+	+
	B[AR1,m]	Area–crossing (AR1)	2	–	–	–	–	+	+	+	+
	B[AR2,m]	Area–crossing (AR2)	2	–	–	–	–	+	+	+	+
	Parameter	Via parameter	2	–	–	–	–	+	+	+	+

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
T		Transfer contents of ACCU1-L to...									
	IW	input word	1 [2]/2	1.4	0.5	0.5	0.1	2.3+	1.8+	1.5+	0.1+
		(MCR-dependent)		2.7	1.4	1.5	0.1	2.8+	2.2+	1.8+	0.1+
	QW	output word	1 [2]/2	1.4	0.5	0.5	0.1	2.3+	1.8+	1.5+	0.1+
		(MCR-dependent)		2.7	1.4	1.5	0.1	2.8+	2.2+	1.8+	0.1+
	PQW	peripheral output word	1 [3]/2	< 34	<27/ <42 [4]/ < 50 [5]	< 27	0.1	<40	29+/ <46 [4]/ <53 [5]	< 31	0.1+
		(MCR-dependent)		< 36	<28/ <44 [4]/ < 52 [5]	< 28	0.1	< 42	<30+/ <48 [4]/ <55 [5]	< 32	0.1+
	MW	bit memory word	1 [2]/2	1.7	0.7	0.8	0.1	2.7+	2.1+	1.7+	0.1+
		(MCR-dependent)		3.0	2.0	1.8	0.1	3.2+	2.6+	2.1+	0.1+
	LW	local data word	2	2.0	0.8	0.9	0.1	3.0+	2.2+	1.8+	0.1+
		(MCR-dependent)		3.4	2.4	2.0	0.1	3.8+	2.8+	2.3+	0.1+
	DBW	data word	2	5.2	3.6	3.0	0.1	4.7+	3.5+	2.9+	0.1+
		(MCR-dependent)		6.1	4.2	3.5	0.1	5.6+	4.1+	3.4+	0.1+
	DIW	instance data word	2	5.2	3.6	3.0	0.1	4.7+	3.5+	2.9+	0.1+
		(MCR-dependent)		6.1	4.2	3.5	0.1	5.6+	4.1+	3.4+	0.1+

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing
3 Direct addressing with PQW 0 to 254
4 Integrated digital I/O for CPU 314*
5 Integrated analog I/O for CPU 314*

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
T	h[AR1,m]	Register–ind., area–internal(AR1)	2	–	–	–	–	+	+	+	+
	h[AR2,m]	Register–ind., area–internal(AR2)	2	–	–	–	–	+	+	+	+
	W[AR1,m]	Area–crossing (AR1)	2	–	–	–	–	+	+	+	+
	W[AR2,m]	Area–crossing (AR2)	2	–	–	–	–	+	+	+	+
	Parameter	Via parameter	2	–	–	–	–	+	+	+	+
T		Transfer contents of ACCU1 to ...									
	ID	input double word	1 [2]/2	2.0	0.7	0.8	0.2	2.7+	2.0+	1.7+	0.2+
		(MCR-dependent)		3.0	1.7	1.8	0.2	3.2+	2.4+	2.1+	0.2+
	QD	output double word	1 [2]/2	2.0	0.7	0.8	0.2	2.7+	2.0+	1.7+	0.2+
		(MCR-dependent)		3.0	1.7	1.8	0.2	3.2+	2.4+	2.1+	0.2+
	PQD	peripheral output double word	1 [2]/2	< 38	< 31	< 31	0.2	< 42	< 32+	< 34	0.2+
		(MCR-dependent)		< 39	< 32	< 32	0.2	< 43	< 35+	< 35	0.2+

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
T	MD	bit memory double word	1 [2]/2	2.7	1.2	1.3	0.2	3.3+	2.7+	2.2+	0.2+
		(MCR-dependent)		3.7	2.0	2.3	0.2	3.8+	3.2+	2.6+	0.2+
	LD	local data double word	2	3.0	1.2	1.4	0.2	5.3+	3.3+	3.0+	0.2+
		(MCR-dependent)		4.1	2.2	2.6	0.2	6.2+	3.9+	3.5+	0.2+
	DBD	data double word	2	6.7	4.9	4.1	0.2	6.2+	4.9+	4.0+	0.2+
		(MCR-dependent)		7.6	5.5	4.6	0.2	7.1+	5.5+	4.5+	0.2+
	DID	instance data double word	2	6.7	4.9	4.1	0.2	6.2+	4.9+	4.0+	0.2+
		(MCR-dependent)		7.6	5.5	4.6	0.2	7.2+	5.5+	4.5+	0.2+
T	i[AR1,m]	Register–ind., area–internal (AR1)	2	–	–	–	–	+	+	+	+
	i[AR2,m]	Register–ind., area–internal (AR2)	2	–	–	–	–	+	+	+	+
	D[AR1,m]	Area–crossing (AR1)	2	–	–	–	–	+	+	+	+
	D[AR2,m]	Area–crossing (AR2)	2	–	–	–	–	+	+	+	+
	Parameter	Via parameter	2	–	–	–	–	+	+	+	+

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing

Load and Transfer Instructions for Address Registers

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
LAR1		Load contents from ...					
	–	ACCU1	1	0.7	0.2	0.3	0.2
	AR2	Address register 2	1	0.7	0.2	0.3	0.2
	DBD a	Data double word	2	6.3	4.0	3.8	0.3
	DID a	Instance data double word	2	6.3	4.0	3.8	0.3
	m	32-bit constant as pointer	3	1.4	0.4	0.5	0.2
	LD a	Local data double word	2	3.4	1.4	1.5	0.3
	MD a	Bit memory double word	2	3.0	1.2	1.4	0.3
		... into AR1					
LAR2		Load contents from ...					
	–	ACCU1	1	0.7	0.2	0.3	0.2
	DBD a	Data double word	2	6.3	4.0	3.8	0.3
	DID a	Instance data double word	2	6.3	4.0	3.8	0.3
	m	32-bit constant as pointer	3	1.4	0.4	0.5	0.2
	LD a	Local data double word	2	3.4	1.4	1.5	0.3
	MD a	Bit memory double word	2	3.0	1.2	1.4	0.3
		... into AR2					

Transferring a double word from AR1 or AR2 to a memory area or register. The status word is not affected.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
CAR1		Transfer contents of AR1 to ...					
	–	ACCU1	1	1.7	0.4	0.7	0.1
	AR2	Address register 2	1	0.7	0.2	0.3	0.2
	DBD a	Data double word	2	6.9	3.9	4.3	0.2
	DID a	Instance data double word	2	6.9	3.9	4.3	0.2
	m	32-bit constant as pointer	3	3.7	1.4	1.6	0.2
	LD a	Local data double word	2	3.4	1.2	1.5	0.2
	MD a	Bit memory double word	2				
CAR2		Transfer contents of AR2 to ...					
	–	ACCU1	1	1.7	0.4	0.7	0.1
	DBD a	Data double word	2	6.9	3.9	4.3	0.2
	DID a	Instance data double word	2	6.9	3.9	4.3	0.2
	m	32-bit constant as pointer	3	3.7	1.4	1.6	0.2
	LD a	Local data double word	2	3.4	1.2	1.5	0.2
	MD a	Bit memory double word	2				
CAR	–	Exchange the contents of AR1 and AR2	1	1.4	0.7	0.5	0.4

Load and Transfer Instructions for the Status Word

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
L	STW	Load status word [1] into ACCU1		2.4	1.4	1.5	0.1

Status word for: **L STW**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	Yes	Yes	Yes	Yes	Yes	0	0	Yes	0
Instruction affects:	–	–	–	–	–	–	–	–	–

T	STW	Transfer ACCU1 (bits 0 to 8) to the status word[1]		2.2	1.5	1.4	0.1

Status word for: **T STW**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	Yes	Yes	Yes	Yes	Yes	–	–	Yes	–

1 For the structure of the status word see page 16

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.

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
L	DBNO	Load number of data block	1	5.1	3.1	3.3	0.1
L	DINO	Load number of instance data block	1	5.1	3.1	3.3	0.1
L	DBLG	Load length of data block into byte	1	1.7	0.6	0.6	0.1
L	DILG	Load length of instance data block into byte	1	1.7	0.6	0.6	0.1

Integer Math (16 Bits)

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

CPU 318–2: ACCU3 and ACCU4 are then transferred to ACCU2 and ACCU3.

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315 315-2 316-2	318-2
+I	–	Add 2 integers (16 bits) (ACCU1-L)=(ACCU1-L)+ (ACCU2-L)	1	2.4	1.5	1.5	0.1
–I	–	Subtract 1 integer from another (16 bits) (ACCU1-L)=(ACCU2-L)– (ACCU1-L)	1	2.6	1.8	1.6	0.1
*I	–	Multiply 1 integer by another (16 bits) (ACCU1)=(ACCU2-L)* (ACCU1-L)	1	3.6	2.1	2.4	0.8
/I	–	Divide 1 integer by another (16 bits) (ACCU1-L)= (ACCU2-L):(ACCU1-L) The remainder is in ACCU1-H	1	5.0	3.2	3.4	0.8

Status word for: **+I, –I,*I, /I**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	Yes	Yes	Yes	Yes	–	–	–	–

Integer Math (32 Bits)

Math instructions on two 32-bit words. The result is in ACCU1.
CPU 318–2: ACCU3 and ACCU4 are then transferred to ACCU2 and ACCU3.

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315 315-2 316-2	318-2
+D	–	Add 2 integers (32 bits) (ACCU1)=(ACCU2)+ (ACCU1)	1	3.1	1.8	2.0	0.1
–D	–	Subtract 1 integer from another (32 bits) (ACCU1)=(ACCU2)– (ACCU1)	1	4.0	2.3	2.7	0.1
*D	–	Multiply 1 integer by another (32 bits) (ACCU1)=(ACCU2)* (ACCU1)	1	13.5	8.2	9.9	1.3
/D	–	Divide 1 integer by another (32 bits) (ACCU1)=(ACCU2): (ACCU1)	1	14.8	6.5	10.8	1.3
MOD	–	Divide 1 integer by another (32 bits) and load the remainder into ACCU1: (ACCU1)=remainder of [(ACCU2):(ACCU1)]	1	15.5	6.4	11.3	1.3

Status word for: **+D, –D,*D, /D, MOD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	Yes	Yes	Yes	Yes	–	–	–	–

Floating-Point Math (32 Bits)

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

CPU 318–2: ACCU3 and ACCU4 are then transferred to ACCU2 and ACCU3.

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
+R	–	Add 2 real numbers (32 bits) (ACCU1)= (ACCU2)+(ACCU1)	1	< 60	< 50	< 35	0.6
–R	–	Subtract 1 real number from another (32 bits) (ACCU1)= (ACCU2)–(ACCU1)	1	< 60	< 50	< 35	0.6
*R	–	Multiply 1 real number by another (32 bits) (ACCU1)=(ACCU2)*(ACCU1)	1	< 60	< 50	< 35	1.4
/R	–	Divide 1 real number by another (32 bits) (ACCU1)=(ACCU2):(ACCU1)	1	< 60	< 50	< 40	2.1

Status word for: **+R, –R, *R, /R**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	Yes	Yes	Yes	Yes	–	–	–	–

In-struc-tion	Address Identi-fier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
NEGR	–	Negate the real number in ACCU1	1	0.7	1.0	0.3	0.1
ABS	–	Form the absolute value of the real number in ACCU1	1	0.7	0.4	0.3	0.1

Status word for: **NEGR, ABS**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	–	–	–	–	–	–	–	–

Square Root and Square Instructions (32 Bits)

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*	313/314/314*	315/315-2/ 316-2	318-2
SQRT	–	Calculate the square root of a real number in ACCCU1	1	–	< 1000	< 1000	40
SQR	–	Form the square of a real number in ACCU1	1	–	< 300	< 300	1,4

Status word for: **SQRT, SQR**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	Yes	Yes	Yes	Yes	–	–	–	–

Logarithmic Function (32 Bits)

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*	313/314/314*	315/315-2/ 316-2	318-2
LN	–	Form the natural logarithm of a real number in ACCU1	1	–	< 650	< 650	< 35
EXP	–	Calculate the exponential value of a real number in ACCU1 to the base e (= 2.71828)	1	–	< 1500	< 1500	< 35

Status word for: **LN, EXP**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	Yes	Yes	Yes	Yes	–	–	–	–

Trigonometrical Functions (32 Bits)

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*	313/314/ 314*	315/315-2/ 316-2	318-2
SIN	–	Calculate the sine of a real number	1	–	< 900	< 900	31
ASIN	–	Calculate the arcsine of a real number	1	–	< 2500	< 2500	74
COS	–	Calculate the cosine of a real number	1	–	< 900	< 900	32
ACOS	–	Calculate the arccosine of a real number	1	–	< 2500	< 2500	77
TAN	–	Calculate the tangent of a real number	1	–	< 900	< 900	35
ATAN	–	Calculate the arctangent of a real number	1	–	< 900	< 900	32

Status word for: **SIN, ASIN, COS, ACOS, TAN, ATAN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	FC
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	Yes	Yes	Yes	Yes	–	–	–	–

Adding Constants

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
+	i8	Add an 8-bit integer constant	1	0.7	0.2	0.3	0.1
+	i16	Add a 16-bit integer constant	2	0.7	0.2	0.3	0.1
+	i32	Add a 32-bit integer constant	3	1.5	0.4	0.6	0.15

Adding Using Address Registers

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
+AR1	–	Add the contents of ACCU1-L to those of AR1	1	0.7	0.3	0.3	0.2
+AR1	m	Add a pointer constant to the contents of AR1	2	0.7	0.6	0.3	0.2
+AR2	–	Add the contents of ACCU1-L to those of AR2	1	0.7	0.3	0.3	0.2
+AR2	m	Add pointer constant to the contents of AR2	2	0.7	0.6	0.3	0.2

Comparison Instructions with Integers (16 Bits)

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
==I	–	ACCU2-L=ACCU1-L	1	2.3	1.4	1.4	0.1
<>I	–	ACCU2-L≠ACCU1-L	1	2.4	1.6	1.5	0.1
<I	–	ACCU2-L<ACCU1-L	1	2.4	1.6	1.5	0.1
<=I	–	ACCU2-L<=ACCU1-L	1	2.3	1.4	1.4	0.1
>I	–	ACCU2-L>ACCU1-L	1	2.4	1.3	1.5	0.1
>=I	–	ACCU2-L>=ACCU1-L	1	2.3	1.4	1.4	0.1

Status word for: **==I, <>I, <I, <=I, >I, >=I**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	Yes	Yes	0	–	0	Yes	Yes	1

Comparison Instructions with Integers (32 Bits)

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
==D	–	ACCU2=ACCU1	1	3.1	1.9	2.0	0.1
<>D	–	ACCU2≠ACCU1	1	3.1	1.9	2.0	0.1
<D	–	ACCU2<ACCU1	1	3.1	1.9	2.0	0.1
<=D	–	ACCU2<=ACCU1	1	3.1	1.9	2.0	0.1
>D	–	ACCU2>ACCU1	1	3.1	1.9	2.0	0.1
>=D	–	ACCU2>=ACCU1	1	3.1	1.9	2.0	0.1

Status word for: **==D,< >D, <D, <=D, >D, >=D**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	Yes	Yes	0	–	0	Yes	Yes	1

Comparison Instructions with Real Numbers (32 Bits)

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
==R	–	ACCU2=ACCU1	1	< 70	< 50	< 45	0.4
<>R	–	ACCU2≠ACCU1	1	< 70	< 50	< 45	0.4
<R	–	ACCU2<ACCU1	1	< 70	< 50	< 45	0.4
<=R	–	ACCU2<=ACCU1	1	< 70	< 50	< 45	0.4
>R	–	ACCU2>ACCU1	1	< 70	< 50	< 45	0.4
>=R	–	ACCU2>=ACCU1	1	< 70	< 50	< 45	0.4

Status word for: **==R, <>R, <R, <=R, >R, >=R**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	Yes	Yes	Yes	Yes	0	Yes	Yes	1

Shift Instructions

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
SLW	–	Shift the contents of ACCU1-L to the left. Positions that become free are provided with zeros.	1	3.0	1.5	2.0	0.1
SLW	0 ... 15			1.8	0.6	0.7	0.1
SLD	–	Shift the contents of ACCU1 to the left. Positions that become free are provided with zeros.	1	4.5	1.7	3.1	0.1
SLD	0 ... 32			4.9	2.9	3.1	0.1
SRW	–	Shift the contents of ACCU1-L to the right. Positions that become free are provided with zeros.	1	3.0	1.5	2.0	0.1
SRW	0 ... 15			1.8	0.6	0.7	0.1
SRD	–	Shift the contents of ACCU1 to the right. Positions that become free are provided with zeros.	1	4.5	1.7	3.1	0.1
SRD	0 ... 32			4.9	2.9	3.2	0.1

Status word for: **SLW, SLD, SRW, SRD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	Yes	Yes	Yes	–	–	–	–	–

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs 312*/313	314/314*	315/315-2/ 316-2	318-2
SSI	–	Shift the contents of ACCU1-L with sign to the right. Positions that become free are provided with the sign (bit 15).	1	2.9	1.6	1.8	0.1
SSI	0 ... 15			1.8	0.6	0.7	0.1
SSD	–	Shift the contents of ACCU1 with sign to the right	1	4.5	1.7	3.1	0.1
SSD	0 ... 32			4.9	2.9	3.2	0.1

Status word for: **SSI, SSD**	BR	CC 1		CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–		–	–	–	–	–	–	–
Instruction affects:	–	Yes		Yes	Yes	–	–	–	–	–

Rotate Instructions

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
RLD	–	Rotate the contents of ACCU1 to the left	1	4.8 5.3	3.3 3.4	3.3 3.4	0.1 0.1
RLD	0 ... 32						
RRD	–	Rotate the contents of ACCU1 to the right	1	5.0 5.4	3.3 3.4	3.5 3.5	0.1 0.1
RRD	0 ... 32						

Status word for: **RLD, RRD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	Yes	Yes	Yes	–	–	–	–	–

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
RLDA	–	Rotate the contents of ACCU1 one bit position to the left through condition code bit CC 1		2.9	1.9	1.9	0.1
RRDA	–	Rotate the contents of ACCU1 one bit position to the right through condition code bit CC 1		2.9	1.9	1.9	0.1

Status word for: **RLDA, RRDA**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	Yes	0	0	–	–	–	–	–

Accumulator Transfer Instructions, Incrementing and Decrementing

The status word is not affected.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
CAW	–	Reverse the order of the bytes in ACCU1-L. LL, LH becomes LH, LL.	1	0.7	0.2	0.3	0.1
CAD	–	Reverse the order of the bytes in ACCU1. LL, LH, HL, AA becomes HH, HL, LH, LL.	1	1.7	0.6	0.6	0.1
TAK	–	Swap the contents of ACCU1 and ACCU2	1	2.0	0.7	0.8	0.1
ENT	–	The contents of ACCU2 and ACCU3 are transferred to ACCU3 and ACCU4.	1	–	–	–	0.1
LEAVE	–	The contents of ACCU3 and ACCU4 are transferred to ACCU2 and ACCU3.	1	–	–	–	0.1

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
PUSH	–	The contents of ACCU1 are transferred to ACCU2 **CPU 318–2:** The contents of ACCU1, ACCU2 and ACCU3 are transferred to ACCU2, ACCU3 and ACCU4.	1	0.7	0.2	0.3	0.1
POP	–	The contents of ACCU2 are transferred to ACCU1 **CPU 318–2:** The contents of ACCU2, ACCU3 and ACCU4 are transferred to ACCU1, ACCU2 and ACCU3.	1	0.7	0.2	0.3	0.1
INC	0 ... 255	Increment ACCU1-LL	1	0.7	0.2	0.3	0.1
DEC	0 ... 255	Decrement ACCU1-LL	1	0.7	0.2	0.3	0.1

Program Display and Null Operation Instructions

The status word is not affected.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
BLD	0 ... 255	Program display instruction: Is treated by the CPU like a null operation instruction.	1	0.7	0.2	0.3	0.1
NOP	0 1	Null Operation instruction:	1	0.7 0.7	0.2 0.2	0.3 0.3	0.1 0.1

Data Type Conversion Instructions

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
BTI	–	Conv. cont. of ACCU1 from BCD to integer (16 bits) (**B**CD **T**o **I**nt)	1	6.6	4.5	4.7	0.2
BTD	–	Conv. cont. of ACCU1 from BCD to double int. (32 bits) (**B**CD **T**o **D**oubleint)	1	15.7	10.4	11.5	0.2
DTR	–	Convert contents of ACCU1 from double integer to real (32 bits) (**D**oubleint **T**o **R**eal)	1	< 26	< 20	< 15	0.3
ITD	–	Convert contents of ACCU1 from integer (16 bits) to double int. (32 bits) (**I**nt **T**o **D**oubleint)	1	0.7	0.2	0.1	0.1

Status word for: **BTI, BTD, DTR, ITD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	–	–	–	–	–	–	–	–

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
ITB	–	Conv. cont. of ACCU1 from int. (16 bits) to BCD from 0 to +/– 999 (**I**nt **T**o **B**CD)	1	7.2	5.2	5.1	0.2
DTB	–	Conv. cont. of ACCU1 f. double int. (32 bits) t. BCD f. 0 to +/–9 999 999 (**D**oubleint **T**o **B**CD)	1	16.1	4.1	11.8	0.2
RND	–	Convert a real number into a 32-bit integer.	1	< 35	< 28	< 20	0.4
RND–	–	Convert a real number into a 32-bit integer. The number is rounded to the next whole number.	1	< 35	< 28	< 20	0.4
RND+	–	Convert a real number into a 32-bit integer. The number is rounded to the next whole number.	1	< 35	< 28	< 20	0.4
TRUNC	–	Convert a real number into a 32-bit integer. The places after the decimal point are truncated.	1	< 35	< 28	< 20	0.4

Status word for: **ITB, DTB, RND, RND–, RND+,TRUNC**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	–	–	Yes	Yes	–	–	–	–

Forming the Ones and Twos Complements

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
INVI	–	Form the ones complement of ACCU1-L	1	0.7	0.2	0.3	0.1
INVD	–	Form the ones complement of ACCU1	1	0.7	0.2	0.3	0.1

Status word for: **INVI, INVD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	–	–	–	–	–	–	–	–

NEGI	–	Form the twos complement of ACCU1-L (integer)	1	2.3	1.6	1.5	0.1
NEGD	–	Form the twos complement of ACCU1 (double integer)	1	3.1	1.8	2.0	0.1

Status word for: **NEGI, NEGD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	Yes	Yes	Yes	Yes	–	–	–	–

Block Call Instructions

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
CALL	FB q, DB q	Unconditional call of an FB, with parameter transfer	1	9.2	7.7	5.3		–	–	–	–
CALL	SFB q, DB q	Unconditional call of an SFB, with parameter transfer	2	See execution time for SFB [2]	7.7	–	–	–	–	–	–
CALL	FC q	Unconditional call of a function, with parameter transfer	1	9.2	7.7	5.3		–	–	–	–
CALL	SFC q	Unconditional call of an SFC, with parameter transfer	2	See execution time for SFCs[2]				–			

Status word for: **CALL**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	–	–	–	0	0	1	–	0

1 Plus time required for loading the address of the instruction (see page 22)
2 In the *S7-300 Hardware and Installation Manual*

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312* 313	314 314*	315 315-2 316-2	318-2	312* 313	314 314*	315 315-2 316-2	318-2
UC	FB q FC q Parameter	Unconditional call of blocks without parameter transfer FB/FC call via parameter	1 [2]	9.2	7.7	5.3	1.4 1.4 1.4	9.8+	8.5+	6.1+	1.4+ 1.4+ 1.4+
CC	FB q FC q Parameter	Conditional call of blocks without parameter transfer FB/FC call via parameter	1 [2]	9.2	7.7	5.3	1.4 1.4 1.4	9.8+	8.5+	6.1+	1.4+ 1.4+ 1.4+

Status word for: **UC, CC**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	–	–	–	0	0	1	–	0

OPN	DB q DI q Parameter	Open: Data block Instance data block Data block using parameters	1/2[3] 2 2	2.9	1.6	1.5	0.3	4.0+	1.4+	2.6+	0.3+

Status word for: **OPN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	–	–	–	–	–	–	–	–

1 Plus time required for loading the address of the instruction (see page 22)
2 With direct instruction addressing
3 With direct instruction addressing Block No. > 255

Block End Instructions

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
BE	–	End block	1	4.9	4.1	2.8	2.0
BEU	–	End block unconditionally	1	–	–	–	–

Status word for: **BE, BEU**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	–	–	–	0	0	1	–	0

BEC	–	End block conditionally if RLO = “1”		5.9	4.4	3.2	2.2

Status word for: **BEC**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	Yes	0	1	1	0

Exchanging Shared Data Block and Instance Data Block

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

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
CDB	–	Exchange shared data block and instance data block	1	1.0	0.3	0.4	

Jump Instructions

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

Note:
Please note for S7–300 CPU programs that the jump destination always (not for 318–2) forms the **beginning** of a Boolean logic string in the case of jump instructions. The jump destination must not be included in the logic string.

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
JU	LABEL	Jump unconditionally	1 1/2	1.8	1.7	1.8	0.5

Status word for: **JU**	BR	CC 1	CC 0	OV	OS	OR	STA		
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	–	–	–	–	–	–	–	–

JC	LABEL	Jump if RLO = “1”	1 1/2	2.3	2.0	1.5	0.5
JCN	LABEL	Jump if RLO = “0”	2	2.6	2.3	1.6	0.5

Status word for: **JC, JCN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	–	0	1	1	0

1 1 word long for jump widths between –128 and +127

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
JCB	LABEL	Jump if RLO = “1”. Save the RLO in the BR bit	2	2.9	2.2	1.8	0.5
JNB	LABEL	Jump if RLO = “0”. Save the RLO in the BR bit	2	2.9	2.4	1.8	0.5

Status word for: **JCB, JNB**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	Yes	–	–	–	–	0	1	1	0

JBI	LABEL	Jump if BR = “1”	2	2.3	2.1	1.5	0.5
JNBI	LABEL	Jump if BR = “0”	2	2.3	2.1	1.5	0.5

Status word for: **JBI, JNBI**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	Yes	–	–	–	–	–	–	–	–
Instruction affects:	–	–	–	–	–	0	1	–	0

JO	LABEL	Jump on stored overflow (OV = “1”)	1 1/2	2.3	2.1	1.5	0.5

Status word for: **JO**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	Yes	–	–	–	–	–
Instruction affects:	–	–	–	–	–	–	–	–	–

[1] 1 word long for jump widths between –128 and +127

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs 312*/313	314/314*	315/315-2/ 316-2	318-2
JOS	LABEL	Jump on stored overflow (OS = "1")	2	2.6	2.2	1.6	0.5

Status word for: **JOS**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	Yes	–	–	–	–
Instruction affects:	–	–	–	–	0	–	–	–	–

JUO	LABEL	Jump if "unordered instruction" (CC 1=1 and CC 0=1)	2	2.8	2.3	1.8	0.5
JZ	LABEL	Jump if result=0 (CC 1=0 and CC 0=0)	1 1/2	2.7	2.2	1.7	0.5
JP	LABEL	Jump if result>0 (CC 1=1 and CC 0=0)	1 1/2	2.7	2.4	1.8	0.5
JM	LABEL	Jump if result<0 (CC 1=0 and CC 0=1)	1 1/2	3.0	2.4	1.8	0.5

Status word for: **JUO, JZ, JP, JM**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	Yes	Yes	–	–	–	–	–	–
Instruction affects:	–	–	–	–	–	–	–	–	–

[1] 1 word long for jump widths between –128 and +127

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
JN	LABEL	Jump if result ≠ 0 (CC 1=1 and CC 0=0) or (CC 1=0) and (CC 0=1)	1 1/2	2.8	2.3	1.8	0.5
JMZ	LABEL	Jump if result ≤ 0 (CC 1=0 and CC 0=1) or (CC 1=0 and CC 0=0)	2	2.4	2.1	1.5	0.5
JPZ	LABEL	Jump if result ≥ 0 (CC 1=1 and CC 0=0) or (CC 1=0) and (CC 0=0)	2	2.4	2.2	1.6	0.5

Status word for: **JN, JMZ, JPZ**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	Yes	Yes	–	–	–	–	–	–
Instruction affects:	–	–	–	–	–	–	–	–	–

1 1 word long for jump widths between –128 and +127

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
JL	LABEL	Jump distributor This instruction is followed by a list of jump instructions. The operand is a jump label to subsequent instructions in this list. ACCU1-L contains the number of the jump instruction to be executed.	2	3.2	3.9	2.7	0.7
LOOP	LABEL	Decrement ACCU1-L and jump if ACCU1-L ≠ 0 (loop programming)	2	2.4	1.7	1.6	0.5

Status word for: **JL, LOOP**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	–	–	–	–	–	–	–	–

Instructions for the Master Control Relay (MCR)

MCR=1→MCR is deactivated
MCR=0→MCR is activated; “T” and “=” instructions write “0” to the corresponding address identifiers; “S” and “R” instructions leave the memory contents unchanged.

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
MCR(		Open an MCR zone. Save the RLO to the MCR stack.	1	3.0	1.6	1.7	0.1

Status word for: **MCR(**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	–	0	1	–	0

)MCR		Close an MCR zone. Pop an entry off the MCR stack.	1	2.8	1.5	1.6	0.1

Status word for: **)MCR**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	–	–	–	–	0	1	–	0

In-struc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312*/313	314/314*	315/315-2/ 316-2	318-2
MCRA		Activate the MCR	1	0.7	0.2	0.3	0.1
MCRD		Deactivate the MCR	1	0.7	0.2	0.3	0.1

Status word for: **MCRA, MCRD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	–	–
Instruction affects:	–	–	–	–	–	–	–	–	–

Organisation Blocks (OB)

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

Organisation Blocks	312*	313	314 314*	315	315-2 316-2	318-2	Starting Events (Hexadecimal Values)
Cycle:							
OB 1	x	x	x	x	x	x	1101_H OB1 starting event 1103_H Running OB1 start event (conclusion of the free cycle)
Time-of-day interrupt:							
OB 10	–	x	x	x	x	x	1111_H Time-of-day interrupt event
OB 11	–	–	–	–	–	x	1112_H Time-of-day interrupt event
Delay Interrupt:							
OB 20	–	x	x	x	x	x	1121_H Delay interrupt event
OB 21	–	–	–	–	–	x	1122_H Delay interrupt event
Watchdog interrupt:							
OB 32	–	–	–	–	–	x	1133_H Watchdog interrupt event
OB 35	–	x	x	x	x	x	1136_H Watchdog interrupt event

Organisation Blocks	312*	313	314 314*	315	315-2 316-2	318-2	Starting Events (Hexadecimal Values)
Process interrupt:							
OB 40	x	x	x	x	x	x	1141_H Process interrupt
OB 41	x	–	–	–	–	x	1114_H Process interrupt
Diagnostic interrupt:							
OB 82	–	x	x	x	x	x	3842_H Module o. k. 3942_H Module fault
Error responses:							
OB 80	–	x	x	x	x	x	3501_H Cycle time violation 3502_H OB or FB request error 3505_H Time-of-day interrupt elapsed due to time jump 3507_H Multiple OB request error caused start info buffer overflow
OB 81	–	x	x	x	x	x	3822_H BAF: Backup voltage returns to CPU 3922_H BAF: No backup voltage in CPU

Organisation Blocks	312*	313	314 314*	315	315-2 316-2	318-2	Starting Events (Hexadecimal Values)
OB 85	–	x	x	x	x	x	$35A1_H$ No OB or FB $35A3_H$ Error during access of a block by the operating system $39B1_H$ I/O access error during process image updating 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 updating of the inputs (outgoing event) $39B3_H$ I/O access error during process image updating 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)

Organisation Blocks	312*	313	314 314*	315	315-2 316-2	318-2	Starting Events (Hexadecimal Values)
OB 86	–	–	–	–	–	x	$38C4_H$ Distributed I/O: station failed, outgoing $39C4_H$ Distributed I/O: station failed, incoming
OB 87	–	x	x	x	x	x	$35E1_H$ Incorrect frame identifier in GD $35E2_H$ $35E2_H$ GD packet status cannot be entered in DB $35E6_H$ GD whole status cannot be entered in DB
Background:							
OB 90	–	–	–	–	–	x	OB 90 initiated by... 1191_H Restart 1192_H Deletion of a block 1193_H Transfer of a block in RUN mode 1195_H 1195_H OB 90 start event
Restart:							
OB 100	x	x	x	x	x	x	1381_H Manual restart requests 1382_H Automatic restart requests
Cold start:							
OB 102	–	–	–	–	–	x	1385_H Manual cold-start requests 1386_H Automatic cold-start requests

Organisation Blocks	312*	313	314 314*	315	315-2 316-2	318-2	Starting Events (Hexadecimal Values)
Synchronous error responses:							
OB 121	–	x	x	x	x	x	2521_H BCD conversion error 2522_H Range length error during reading 2523_H Range length error during writing 2524_H Range error during reading 2525_H Range error during writing 2526_H Timer number error 2527_H Counter number error 2528_H Alignment error during reading 2529_H Alignment error during writing 2530_H Write error during access to DB 2531_H Write error during access to DI 2532_H Block number error opening a DB 2533_H Block number error opening a DI 2534_H Block number error at FC call 2535_H Block number error at FB call $253A_H$ DB not loaded $253C_H$ FC not loaded $253E_H$ FB not loaded

Organisation Blocks	312*	313	314 314*	315	315-2 316-2	318-2	Starting Events (Hexadecimal Values)
OB 122	–	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)

Function Blocks (FB)

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

Blocks	312*	313	314	314*	315	315-2	316-2	318-2
Function blocks								
Quantity	32	128	128	128	192	192	256	1024
Admissible numbers	0 to 31	0 to 127	0 to 127	0 to 127	0 to 191	0 to 191	0 to 255	0 to 1023
Maximal size of an FB (process–relevant code)	8 kB	8 kB	8 kB	8 kB	16 kB	16 kB	16 kB	64 kB

Functions (FC) and Data Blocks

Blocks	312*	313	314	314*	315	315-2	316-2	318-2
Functions								
Quantity	32	128	128	128	192	192	512	1024
Admissible numbers	0 to 31	0 to 127	0 to 127	0 to 127	0 to 191	0 to 191	0 to 511	0 to 1023
Maximal size of an FC (process–relevant code)	8 kB	8 kB	8 kB	8 kB	16 kB	16 kB	16 kB	64 kB
Data blocks								
Quantity	127	127	127	127	254	254	511	2047
Admissible numbers	1 to 127	1 to 127	1 to 127	1 to 127	1 to 254	1 to 254	1 to 511	1 to 2047
Maximal size of a data block (data byte quantity)	8 kB	8 kB	8 kB	8 kB	16 kB	16 kB	16 kB	64 kB

System Functions (SFC)

The following tables show the system functions offered by the

operating systems of the S7-300 CPUs, and the execution times on the respective CPUs.

SFC No.	SFC Name	Description	Execution Time in µs							
			312*	313	314	314*	315	315-2	316-2	318-2
0	SET_CLK	Sets the clock time	290	240			240			137
1	READ_CLK	Reads the clock time	205	190			185			28
2	SET_RTM	Sets the operating hours counter	–	65			60			21
3	CTRL_RTM	Starts/stops the operating hours counter	–	55			55			21
4	READ_RTM	Reads the operating hours counter	–	90			80			24
5	GADR_LGC	Determine logical channel address Rack–0 Internal DP	–	–			–	170		38
6	RD_SINFO	Reads start information of the current OB.	180	150			120			34
7	DP_PRAL	Triggers a process interrupt from the user program of the CPU as DP slave through to DP master.	–	–			–	100		29

SFC No.	SFC Name	Description	Execution Time in µs							
			312*	313	314	314*	315	315-2	316-2	318-2
11	SYC_FR	Synchronizes groups of DP slaves	–	–			–	328		124 +2,1*
12	D_ACT_DP	Activates or deactivates DP slaves	–	–			–	442		–
13	DPNRM_DG	Reads the DP-compliant slave diagnosis	–	–			–	180		97
14	DPRD_DAT	Reads/writes consistent data (n bytes)	–	–			–	180		47
15	DPWR_DAT		–	–			–	180		47
17	ALARM_SQ	Generates block-related messages that can be acknowledged	–	–	310		250			74
18	ALARM_S	Generates block-related messages that can not be acknowledged	–	–	310		250			74
19	ALARM_SC	Acknowledgment state of the last ALARM_SQ received message	–	–	130		110			56
20	BLKMOV	Copies variables within the working memory	105 + 3,2**	90 + 2**			75 + 2**			43 + 0,17**

* µs per request

** µs per byte

SFC No.	SFC Name	Description	Execution Time in µs							
			312*	313	314	314*	315	315-2	316-2	318-2
21	FILL	Sets array default variables within the working memory	126+ 3,2*	90 + 3,2*			75 + 2*			45 + 0,12*
22	CREAT_DB	Generates a data block	126+ 3,5**	110 + 3,5**			110 + 3,5**			27
23	DEL_DB	Deletes a data block	–	615		650	615	805		22
24	TEST_DB	Tests a data block	–	130			130			30
25	COMPRESS	Compresses a user program	–	–			–			22
26	UPDAT_PI	Updates process image of the inputs	–	–			–			32 + 4,2***
27	UPDAT_PO	Updates process image of the outputs	–	–			–			30+ 3,5***
28	SET_TINT	Sets the times of a time-of-day interrupt	–	190			190			51
29	CAN_TINT	Cancels a time-of-day interrupt	–	50			50			22
30	ACT_TINT	Activates a time-of-day interrupt	–	50			50			19
31	QRY_TINT	Queries the status of a time-of-day interrupt	–	85			75			30

* µs per byte

** µs per DB in stated area

*** µs per module

SFC No.	SFC Name	Description	Execution Time in µs							
			312*	313	314	314*	315	315-2	316-2	318-2
32	SRT_DINT	Starts a delay interrupt	–	85			80			45
33	CAN_DINT	Cancels a delay interrupt	–	50			50			29
34	QRY_DINT	Queries started delay interrupts	–	80			80			32
36	MSK_FLT	Masks sync faults	185	150			110			21
37	DMSK_FLT	Enables sync faults	205	160			130			23
38	READ_ERR	Reads event status register	205	160			115			23
39	DIS_IRT	Disables the handling of new interrupts	300	215			300			42
40	EN_IRT	Enables the handling of new interrupt events	490	305			280			42
41	DIS_AIRT	Delays the handling of interrupts	55	35			35			18
42	EN_AIRT	Enables the handling of interrupts	55	35			35			18
43	RE_TRIGR	Re-triggers the scan time monitor	40	30			30			98
44	REPL_VAL	Copies a substitute value into accumulator 1	–	45			45			20
46	STP	Forces the CPU into the STOP mode	–	–			–			–
47	WAIT	Delays program execution in addition to waiting times	200	200			200			5
48	SNC_RTCB	Sychronizes slave clocks	–	–			–			17

SFC No.	SFC Name	Description	Execution Time in µs							
			312*	313	314	314*	315	315-2	316-2	318-2
49	LGC_GADR	Converts a free address to the slot and rack for a module	140	140			140			38
50	RD_LGADR	Reads all the declared free addresses for a module	190	190			190			77
51	RDSYSST	Reads out the information from the system state list. SFC 51 is not interruptible through interrupts.	350+ 10**	280 + 10**			270 + 10**			150
52	WR_USMSG	Writes specific diagnostic information in the diagnostic buffer	140	110			110			82
54	RD_DPARAM	Reads dynamic parameters from a module	1300	1300			1300			116
55	WR_PARM	Writes dynamic parameters to a module	1000	1600			1600			118
56	WR_DPARM	Writes predefined dynamic parameters to a module	1600	1750			1750			101
57	PARM_MOD	Assigns a module's parameters	1920	2200			2200			87

** µs per byte of a data record

SFC No.	SFC Name	Description	Execution Time in µs							
			312*	313	314	314*	315	315-2	316-2	318-2
58	WR_REC	Writes a module-specific data record	1400 + 32*	1400 + 32*			1400 + 32			720 + 15*
59	RD_REC	Reads a module-specific data record	500	500			500			810 + 15*
60	GD_SND	Sends a GD packet	–	–			–			200+ 9,4*
61	GD_RCV	Receives a GD packet	–	–			–			56
64	TIME_TICK	Reads out the system time	56	45			45			18
65	X_SEND	Sends data to a communication partner external to your own S7 station	510	420			310			300
66	X_RCV	Receives data from a communication partner external to your own S7 station	190	160			120			220
67	X_GET	Reads data from a communication partner external to your own S7 station	310	250			190			130+ 8,3*
68	X_PUT	Writes data to a communication partner external to your own S7 station	310	250			190			130+ 8,3*
69	X_ABORT	Aborts connection to a communication partner external to your own S7 station	150	120			100			138

* µs per byte

SFC No.	SFC Name	Description	Execution Time in µs							
			312*	313	314	314*	315	315-2	316-2	318–2
72	I_GET	Reads data from a communication partner within your own S7 station	300	250			190			140+ 9,8*
73	I_PUT	Writes data to a communication partner within your own S7 station	300	250			190			150+ 10,6*
74	I_ABORT	Aborts connection to a communication partner within your own S7 station	150	120			100			138
79	SET	Sets a bit field in the I/O area (n = number of bits to be set to 1)	–	–			–			56
80	RSET	Deletes a bit field in the I/O area (n = number of bits to be set to 0)	–	–			–			56
81	UBLKMOV	Copies a variable uninterruptibly. Data to be copied may be up to 32 bytes long.	–	140		148	148			42 + 0,17*

* µs per byte

System Function Blocks (SFB)

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

SFB No.	SFB Name	Description	Execution Time in µs							
			312*	313	314	314*	315	315-2	316-2	318-2
0	CTU	Counts up	120	80			70			16
1	CTD	Counts down	120	80			70			16
2	CTUD	Counts up and counts down	150	95			80			19
3	TP	Generates a pulse	140	100			90			23
4	TON	Delays a leading edge	140	100			90			23
5	TOF	Delays a falling edge	145	100			90			18
32	DRUM	Implements a sequence processor with a maximum of 16 sequences	480	360			80			19
SFBs for the integrated inputs/outputs										
29	HS_COUNT	Counts pulses at the special inputs of the integrated inputs/outputs	about 300			about 300	–			
30	FREQ_MES	Measures frequency via a special input of the integrated inputs/outputs	about 220			about 220	–			

SFB No.	SFB Name	Description	Execution Time in µs							
			312*	313	314	314*	315	315-2	316-2	318-2
38	HSC_A_B	Counts pulses with 2 counters A and B at the special inputs of the integrated inputs/outputs	–			about 230		–		
39	POS	Controlled positioning of axes in cooperation with the user program	–			about 150		–		
41	CONT_C	Continuous control	–			about 3300		–		
42	CONT_S	Step control	–			about 2800		–		
43	PULSEGEN	Pulse generation	–			about 1500		–		

IEC Functions

You can use the following functions in STEP 7:

FC No.	FC Name	Description	Execution Time in µs
DATE_AND_TIME			
3	D_TOD_DT	Concatenates the data formats DATE and TIME_OF_DAY (TOD) and converts to data format DATE_AND_TIME.	680
6	DT_DATE	Extracts the DATE data format from the DATE_AND_TIME data format.	230
7	DT_DAY	Extracts the day of the week from the data format DATE_AND_TIME.	230
8	DT_TOD	Extracts the TIME_OF_DAY data format from the DATE_AND_TIME data format.	200
Time Formats			
33	S5TI_TIM	Converts S5 TIME data format to TIME data format	80
40	TIM_S5TI	Converts TIME data format to S5 TIME data format	160
Duration			
1	AD_DT_TM	Adds a duration in the TIME format to a time in the DT format. The result is a new time in the DT format.	750
35	SB_DT_TM	Subtracts a duration in the TIME format from a time in the DT format. The result is a new time in the DT format.	750
34	SB_DT_DT	Subtracts two times in the DT format. The result is a duration in the TIME format.	700

FC No.	FC Name	Description	Execution Time in µs
Compare DATE_AND_TIME			
9	EQ_DT	Compares the contents of two variables in the DATE_AND_TIME format for equal to.	190
12	GE_DT	Compares the contents of two variables in the DATE_AND_TIME format for greater than or equal to.	190
14	GT_DT	Compares the contents of two variables in the DATE_AND_TIME format for greater than.	190
18	LE_DT	Compares the contents of two variables in the DATE_AND_TIME format for less than or equal to.	190
23	LT_DT	Compares the contents of two variables in the DATE_AND_TIME format for less than.	190
28	NE_DT	Compares the contents of two variables in the DATE_AND_TIME format for not equal to.	190

FC No.	FC Name	Description	Execution Time in µs
Compare STRING			
10	EQ_STRNG	Compares the contents of two variables in the STRING format for equal to.	150 + (n × 32)
13	GE_STRNG	Compares the contents of two variables in the STRING format for greater than or equal to.	150 + (n × 32)
15	GT_STRNG	Compares the contents of two variables in the STRING format for greater than.	150 + (n × 32)
19	LE_STRNG	Compares the contents of two variables in the STRING format for less than or equal to.	150 + (n × 32)
24	LT_STRNG	Compares the contents of two variables in the STRING format for less than.	150 + (n × 32)
29	NE_STRNG	Compares the contents of two variables in the STRING format for not equal to.	150 + (n × 32)

n = number of characters

FC-Nr.	FC-Name	Description	Execution Time in µs
STRING Variable Processing			
21	LEN	Reads the length of a STRING variable.	90
20	LEFT	Reads the first L characters of a STRING variable.	150 + (L × 26)
32	RIGHT	Reads the last L characters of a STRING variable.	150 + (L × 26)
26	MID	Reads the middle L characters of a STRING variable (starting at the defined character).	150 + (L × 26)
2	CONCAT	Concatenates two STRING variables in one STRING variable.	180 + (n × 28)
17	INSERT	Inserts a STRING variable into another STRING variable at a defined point.	250 + (n × 26)
4	DELETE	Deletes L characters of a STRING variable.	300 + ((L + P) × 27)
31	REPLACE	Replaces L characters of a STRING variable with a second STRING variable.	300 + ((L + P) × 27)
11	FIND	Finds the position of the second STRING variable in the first STRING variable.	k × 50

L, P = block parameters (if I + P = 0, then the execution time L + P = 254 µs)
n = number of characters
k = number of characters in parameter IN1

FC No.	FC Name	Description	Execution Time in µs
Format Conversions with STRING			
16	I_STRNG	Converts a variable from INTEGER format to STRING format.	1110
5	DI_STRNG	Converts a variable from INTEGER (32-bit) format to STRING format.	1500
30	R_STRNG	Converts a variable from REAL format to STRING format.	1720
38	STRNG_I	Converts a variable from STRING format to INTEGER format.	500
37	STRNG_DI	Converts a variable from STRING format to INTEGER (32-bit) format.	840
39	STRNG_R	Converts a variable from STRING format to REAL format.	200
Number Processing			
22	LIMIT	Limits a number to a defined limit value.	450
25	MAX	Selects the largest of three numeric variables.	450
27	MIN	Selects the smallest of three numeric variables.	450
36	SEL	Selects one of two variables.	450

System Status Sublist

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)	Remarks
	CPU identification		CPU type and version number	–
0111_H	One record of the sublist	0001_H		
	CPU features			–
0012_H	All records of the sublist	0000_H		
0112_H	Only those records of a group of features	0100_H	STEP 7 processing	
		0300_H	Time system in the CPU	
$0F12_H$	Header information only		STEP 7 operation set	
0013_H	**User memory areas**	–	Work memory	–

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)	Remarks
0014_H	**Operating system areas**	–	Process image of the inputs (number in bytes) Process image of the outputs (number in bytes) Number of memory markers Number of timers Number of counters Size of the I/O address area Entire local data area of the CPU (in bytes)	–
0015_H	**Block types** All records of the sublist	–	OBs (number and size) DBs (number and size) SDBs (number and size) FCs (number and size) FBs (number and size)	–

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)	Remarks
	State of module LEDs		–	–
0019_H	Status of each LED			
0074_H				
0174_H				
$0F19_H$	Header information only			
$0F74_H$				
0132_H	**Communications status** on the communications type specified	0001_H	Number and type of connections	–
		0004_H	CPU protection level, position of the key switch, version identification of the user program and configuration	
		0005_H	Diagnostics status data	
		0008_H	Time system, correction factor, operating hours counter, date/time of day	

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)	Remarks
0222_H	**Interrupt status** Record for the specified interrupt	OB number	–	–
0232_H	**CPU Protection Level**	0004_H	CPU protection level and position of the key switch, version identification of the user program and hardware configuration	–
0692_H	**Status information of module racks** for all racks of an S7-300	–	OK status of individual racks	–
$0D91_H$	**Module status information** of all modules in the specified rack (all CPUs)	 0000_H 0001_H 0002_H 0003_H	Features/parameters of the module plugged in Rack 0 Rack 1 Rack 2 Rack 3	–

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)	Remarks
	Diagnostic buffer	–	Event information	–
$00A0_H$	All entered event information		The information in each case depends on the event	
$01A0_H$	The x latest information entries			
	Module diagnostics			
$00B1_H$	Data record 0 of the module diagnostics information	Module starting address	Module-dependent diagnostics information	
$00B2_H$	Complete module-dependent record of the module diagnostics information	Module rack and slot number		
$00B3_H$	Complete module-dependent record of the module diagnostics information	Module starting address		

PROFIBUS-DP Sublists

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)	Remarks
	Module status information in the CPU			
$0A91_H$	Status information of all DP subsystems and DP masters			not 318-2
$0C91_H$	Module status information of a module	Module starting address	Features/parameters of the module plugged in	
	Module status information			–
$0D91_H$	In the station named (for CPU 315-2 DP)	$xxyy_H$	All modules of station *yy* in the DP subnet *xx* As DP slave: Status data for transfer memory areas	

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)	Remarks
	Status information of module racks or stations in the DP network			–
0092_H	Target status of racks in central configuration or of stations in a subnet			
0292_H	Actual status of racks in central configuration or of stations in a subnet	0000_H	Information on the state of the mounting rack in the central configuration	
0692_H	OK status of expansion racks in central configuration or of stations in a subnet	Subnet ID	Information of status of stations in subnet	
	Module diagnostics			–
$00B4_H$	All standard diagnostic data of a station (only with DP master)	Module start address (Diagnostic address)	Module-dependent diagnostic information	

Alphabetical Index of Instructions