S7-300 Instruction List

CPU 31xC, CPU 31x,
IM 151-7 CPU, IM 151-8 CPU, IM 154-8 CPU, BM 147-1 CPU, BM 147-2 CPU

This instruction list is part of the
documentation package with the order number:

6ES7398-8FA10-8BA0
6ES7198-8FA01-8BA0

06/2008

A5E00105517-10

Siemens AG
Industry Sector
Postfach 4848
90437 NÜRNBERG / GERMANY

Disclaim of Liability

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

A5E00105517-10

Contents

Validity Range of the Instructions List

CPU	As of order no.	As of Version	In the following referred to as
		Firmware	
CPU 312	6ES7 312-1AE13-0AB0	V2.6	312
CPU 312C	6ES7 312-5BE03-0AB0		
CPU 313C	6ES7 313-5BF03-0AB0	V2.6	31x
CPU 313C-2 PtP	6ES7 313-6BF03-0AB0		
CPU 313C-2 DP	6ES7 313-6CF03-0AB0		
CPU 314	6ES7 314-1AG13-0AB0		
CPU 314C-2 PtP	6ES7 314-6BG03-0AB0		
CPU 314C-2 DP	6ES7 314-6CG03-0AB0		
CPU 315-2 DP	6ES7 315-2AG10-0AB0	V2.6	31x or 315
CPU 315-2 PN/DP	6ES7 315-2EH13-0AB0	V2.6	315 or 315 PN
CPU 315T-2 DP	6ES7 315-6TG10-0AB0	V2.4	315 or 315T
CPU 317-2 DP	6ES7 317-2AJ10-0AB0	V2.6	31x, 317
CPU 317-2 PN/DP	6ES7 317-2EK13-0AB0	V2.6	317 or 317 PN
CPU 317T-2 DP	6ES7 317-6TJ10-0AB0	V2.4	317 or 317T
CPU 319–3 PN/DP	6ES7 318–3EL00–0AB0	V2.7	319 or 319 PN

CPU	As of order no.	As of Version	In the following referred to as
		Firmware	
BM 147-1 CPU	6ES7 147-1AA10-0AB0	V2.1.0	147
BM 147-2 CPU	6ES7 147-2AA00-0XB0	V2.1.0	147
IM 151-7 CPU	6ES7 151-7AA20-0AB0	V2.6	151-7[1]
IM 151-8 CPU	6ES7 151-8AB00-0AB0	V2.7	151-8[1]
IM 154-8 CPU	6ES7 154-8AB00-0AB0	V2.5	154

[1] If the values are effective for the IM151-7 CPU and the IM151-8 CPU. you will only see “151” in the operation list.

Address Identifiers and Parameter Ranges

Addr. ID	Parameter Ranges			Description
	31x, 147, 151, 154	317	319	
Q	0.0 to 127.7 (can be set up 2047.7[1])	0.0 to 255.7 (can be set up 2047.7[1])	0.0 to 255.7 (can be set up 4095.7)	Output (in PIQ)
QB	0 to 127 (can be set up 2047[1])	0 to 255 (can be set up 2047[1])	0 to 255 (can be set up 4095)	Output byte (in PIQ)
QW	0 to 126 (can be set up 2046[1])	0 to 254 (can be set up 2046[1])	0 to 254 (can be set up 4094)	Output word (in PIQ)
QD	0 to 124 (can be set up 2044[1])	0 to 252 (can be set up 2044[1])	0 to 252 (can be set up 4092)	Output double word (in PIQ)

[1] only CPU 315-2 PN, CPU 317-2 DP, CPU 317-2 PN/DP, IM 151-8 CPU and IM 154-8 CPU

Addr. ID	Parameter Ranges					Description
	31xC, 312, 314, 147, 151-7	315, 154	151-8	317	319	
DBX	0.0 to 16383.7	0.0 to 16383.7	0.0 to 65535.7	0.0 to 65535.7	0.0 to 65535.7	Data bit in data block
DB	1 to 511	1 to 1023	1 to 511	1 to 2047	1 to 4095	Data block
DBB	0 to 16383	0 to 16383	0 to 65535	0 to 65535	0 to 65535	Data byte in DB
DBW	0 to 16382	0 to 16382	0 to 65534	0 to 65534	0 to 65534	Data word in DB
DBD	0 to 16380	0 to 16380	0 to 65532	0 to 65532	0 to 65532	Data double word in DB
DIX	0.0 to 16383.7	0.0 to 16383.7	0.0 to 65535.7	0.0 to 65535.7	0.0 to 65535.7	Data bit in instance DB
DI	1 to 511	1 to 1023	1 to 511	1 to 2047	1 to 2047	Instance data block
DIB	0 to 16383	0 to 16383	0 to 65535	0 to 65535	0 to 65535	Data byte in instance DB
DIW	0 to 16382	0 to 16382	0 to 65535	0 to 65534	0 to 65534	Data word in instance DB
DID	0 to 16380	0 to 16380	0 to 65532	0 to 65532	0 to 65532	Data double word in instance DB

Addr. ID	Parameter Ranges			Description
	31x, 147, 151, 154	317	319	
I	0.0 to 127.7 (can be set up 2047.7[1])	0.0 to 255.7 (can be set up 2047.7[1])	0.0 to 255.7 (can be set up 4095.7)	Inputs (in PII)
IB	0 to 127 0 to 255 (can be set up 2047[1])	0 to 255 (can be set up 2047[1])	0.0 to 255.7 (can be set up 4095)	Input byte (in PII)
IW	0 to 126 0 to 254 (can be set up 2046[1])	0 to 254 (can be set up 2046[1])	0.0 to 255.7 (can be set up 4094)	Input word (in PII)
ID	0 to 124 0 to 252 (can be set up 2044[1])	0 to 252 (can be set up 2044[1])	0.0 to 255.7 (can be set up 4092)	Input double word (in PII)

Addr. ID	Parameter Ranges			Description
	312	313C, 314, 314C, 147, 151-7, 151-8, 154	317 / 319	
L	0.0 to 255.7	0.0 to 509.7	0.0 to 1023.7	Local data bit
LB	0 to 255	0 to 509	0 to 1023	Local data byte
LW	0 to 254	0 to 508	0 to 1022	Local data word
LD	0 to 252	0 to 506	0 to 1020	Local data double word

[1] only CPU 315-2 PN, CPU 317-2 DP, CPU 317-2 PN/DP, IM 151-8 CPU and IM 154-8 CPU

Addr. ID	Parameter Ranges					Description
	312	313C, 314, 314C, 147, 151	315, 154	317	319	
M	0.0 to 127.7	0.0 to 255.7	0.0 to 2047.7	0.0 to 4095.7	0.0 to 8191.7	Bit memory bit
MB	0 to 127	0 to 255	0 to 2047	0 to 4095	0 to 8191	Bit memory byte
MW	0 to 126	0 to 254	0 to 2046	0 to 4094	0 to 8190	Bit memory word
MD	0 to 124	0 to 252	0 to 2044	0 to 4092	0 to 8188	Bit memory double word

Addr. ID	Except for CPU 315, 151-8, 154, 317 and 319	315, 151-8, 154	317	319	Description
PQB	0 to 1023	0 to 2047	0 to 8191	0 to 8191	Peripheral output byte (direct I/O access)
PQW	0 to 1022	0 to 2046	0 to 8190	0 to 8190	Peripheral output word (direct I/O access)
PQD	0 to 1020	0 to 2044	0 to 8188	0 to 8188	Peripheral output double word (direct I/O access)
PIB	0 to 1023	0 to 2047	0 to 8191	0 to 8191	Peripheral input byte (direct I/O access)
PIW	0 to 1022	0 to 2046	0 to 8190	0 to 8190	Peripheral input word (direct I/O access)
PID	0 to 1020	0 to 2044	0 to 8188	0 to 8188	Peripheral input double word (direct I/O access)

Addr. ID	Parameter Ranges				Description
	312	31x, 147, 151, 154	317	319	
T	0 – 127	0 – 255	0 to 511	0 to 2047	Timer
Z	0 – 127	0 – 255	0 to 511	0 to 2047	Counter
Parameter	–	–	–	–	Instruction adressed via parameter
B#16# W#16# DW#16#	–	–	–	–	Byte Word Double word hexadecimal
D#	–	–	–	–	IEC date constant
L#	–	–	–	–	32-bit integer constant
P#	–	–	–	–	Pointer constant
S5T#Time	–	–	–	–	S5 time constant [1] (16 bits), T#1D_5H-3M_1S_2MS
T#Time	–	–	–	–	IEC time constant, T#1D_5H-3M_1S_2MS
TOD#Time	–	–	–	–	time constant (16-/32-Bit), T#1D_5H-3M_1S_2MS
C#	–	–	–	–	Counter constant (BCD-codiert)

[1] for loading of S5 timers

Addr. ID	Parameter Ranges				Description
	312	31x, 147, 151, 154	317	319	
2#	–	–	–	–	Binary constant
B (b1,b2) B (b1,b2; b3,b4)	–	–	–	–	Constant, 2 or 4 Byte

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

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.

Accumulator designations:

ACCU	Bits
ACCUx (x = 1 to 2)	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	$\overline{\text{FC}}$	First check bit , Bit cannot be written and evaluated in the user program since it is not updated at program runtime
1	RLO	Result of (previous) logic operation
2	STA	Status, Bit cannot be written and evaluated in the user program since it is not updated at program runtime
3	OR	Or, Bit cannot be written and evaluated in the user program 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

<table>
<tr><th>Addressing Examples</th><th>Description</th></tr>
<tr><td colspan="2"><b>Area-Internal Register-Indirect Addressing</b></td></tr>
<tr><td>A I [AR1,P#12.2]</td><td>AND operation: The address of the input is calculated from the “pointer value in AR1+ P#12.2”</td></tr>
<tr><td colspan="2"><b>Area-Crossing Register-Indirect Addressing</b></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 0<b>000</b></td><td>80</td><td>I/O area</td></tr>
<tr><td>I</td><td>1000 0<b>001</b></td><td>81</td><td>Input area</td></tr>
<tr><td>Q</td><td>1000 0<b>010</b></td><td>82</td><td>Output area</td></tr>
<tr><td>M</td><td>1000 0<b>011</b></td><td>83</td><td>Bit memory area</td></tr>
<tr><td>DB</td><td>1000 0<b>100</b></td><td>84</td><td>Data area</td></tr>
<tr><td>DI</td><td>1000 0<b>101</b></td><td>85</td><td>Instance data area</td></tr>
<tr><td>L</td><td>1000 0<b>110</b></td><td>86</td><td>Local data area</td></tr>
<tr><td>VL</td><td>1000 0<b>111</b></td><td>87</td><td>Predecessor local data (access to local data of invoking block)</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"><b>Addressing Via Parameters</b></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	**31x, 147, 151, 154**	**317**	**319**
Bit memory area M Word (for times, counters and block calls) Double word	 0.7 1.6	 0.4 0.9	 0.08 0.21	 0.02 0.05
Data block DB/DX Word (for times, counters and block calls) Double word	 1.5 3.7	 0.8 2.0	 0.20 0.25	 0.02 0.05
Local data area L Word (for times, counters and block calls) Double word	 0.9 2.2	 0.5 1.2	 0.08 0.20	 0.02 0.05
AR1/AR2 (area-internal)	1.0	0.5	0.20	0.02[1)]
AR1/AR2 (area-crossing)	3.0	1.6	0.31	0.05
Parameter (word) ... for: • Timers • Counters • Block calls	2.0	1.0	0.08	0.02
Parameter (double word) ... for Bits, bytes, words and double words	4.0	2.0	0.26	0.01

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

[1)] For the adress areas I/Q/M/L 0.05 μs

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

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

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

Example: A I [DBD 12]

Step 1: Load the contents of DBD 12 (time required is listed in the table on page 24)

Address is in ...	Execution Time in μs
Bit memory area M	
Word	0.4
Double word	0.9
Data block DB/DI	
Word	0.8
Double word	**2.0**

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**
0.1	Time for A I **1.6+**
:	**:**

Total execution time:

2.0 µs
+ 1.6 µs
= 3.6 µs

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 24)

Address is in ...	Execution Time in µs
:	:
AR1/AR2 (area-internal)	**0.5**
:	:

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**
0.1 :	Time for A I **1.6+** **:**

Total execution time:

  0.5 µs
+ 1.6 µs
= 2.1 µs

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 24)

Address is in ...	Execution Time in µs
:	:
AR1/AR2 (area-crossing)	**1.6**
:	:

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

Typical Execution Time in µs	
Direct Addressing	**Indirect Addressing**
0.1	Time for **1.6+**
:	A I **:**

Total execution time:

1.6 µs
+ 1.6 µs
= 3.2 µs

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 24).

Address is in ...	Execution Time in µs
:	:
:	:
Parameter (double word)	**2.0**

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**
0.1	Time for A I **1.6+**
:	**:**

Total execution time:

```
        2.0 µs
+       1.6 µs
=       3.6 µs
```

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 19), you must add the time required for loading the address of the particular instruction to the execution times listed (see page 24).

Bit Logic Instructions

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

Instruction	Address Identifier	Description	Length in Words 2)	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing 1)			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
A		AND									
	I/Q a.b	Input/output	1/2	0.2	0.1	0.05	0.01	3.0+	1.6+	0.09+	0.01+
	M a.b	Bit memory	1/2	0.4	0.2	0.05	0.01	3.2+	1.7+	0.09+	0.01+
	L a.b	Local data bit	2	0.7	0.3	0.06	0.02	3.7+	2.0+	0.07+	0.01+
	DBX a.b	Data bit	2	2.9	1.4	0.17	0.02	4.5+	2.4+	0.08+	0.01+
	DIX a.b	Instance data bit	2	2.9	1.4	0.17	0.02	4.5+	2.4+	0.07+	0.01+
	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 24)
2) With direct instruction addressing/ with indirect instruction adressing

Instruction	Address Identifier	Description	Length in Words 2)	Typical Execution Time in μs							
				Direct Addressing				Indirect Addressing 1)			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
AN		AND NOT									
	I/Q a.b	Input/output	1/2	0.3	0.2	0.05	0.01	3.2+	1.7+	0.09+	0.01+
	M a.b	Bit memory	1/2	0.4	0.2	0.05	0.01	3.4+	1.8+	0.09+	0.01+
	L a.b	Local data bit	2	0.8	0.4	0.06	0.02	3.9+	2.1+	0.08+	0.01+
	DBX a.b	Data bit	2	3.0	1.5	0.17	0.02	4.7+	2.5+	0.09+	0.01+
	DIX a.b	Instance data bit	2	3.0	1.5	0.17	0.02	4.7+	2.5+	0.07+	0.01+
	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 24)
[2] With direct instruction addressing/ with indirect instruction adressing

Instruction	Address Identifier	Description	Length in Words 2)	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing 1)			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
O		OR									
	I/Q a.b	Input/output	1/2	0.2	0.1	0.05	0.01	3.0+	1.6+	0.11+	0.01+
	M a.b	Bit memory	1/2	0.3	0.2	0.05	0.01	3.2+	1.7+	0.11+	0.01+
	L a.b	Local data bit	2	0.7	0.3	0.06	0.02	3.7+	2.0+	0.10+	0.01+
	DBX a.b	Data bit	2	2.9	1.4	0.20	0.02	4.6+	2.4+	0.11+	0.01+
	DIX a.b	Instance data bit	2	2.9	1.4	0.20	0.02	4.6+	2.4+	0.09+	0.01+
	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	RLO2	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	Yes
Instruction affects:	–	–	–	–	–	0	Yes	Yes	1

[1] Plus time required for loading the address of the instruction (see page 24)

[2] With direct instruction addressing/ with indirect instruction adressing

Instruction	**Address Identifier**	**Description**	**Length in Words** [2]	**Typical Execution Time in µs**							
				Direct Addressing				**Indirect Addressing** [1]			
				312	**31x, 147, 151, 154**	**317**	**319**	**312**	**31x, 147, 151, 154**	**317**	**319**
ON		OR NOT									
	I/Q a.b	Input/output	1/2	0.3	0.2	0.05	0.01	3.2+	1.7+	0.11+	0.01+
	M a.b	Bit memory	1/2	0.4	0.2	0.05	0.01	3.5+	1.8+	0.11+	0.01+
	L a.b	Local data bit	2	0.8	0.4	0.06	0.02	3.9+	2.1+	0.10+	0.01+
	DBX a.b	Data bit	2	3.0	1.5	0.20	0.02	4.7+	2.5+	0.11+	0.01+
	DIX a.b	Instance data bit	2	3.0	1.5	0.20	0.02	4.7+	2.5+	0.09+	0.01+
	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 24)

[2] With direct instruction addressing/ with indirect instruction adressing

Instruction	Address Identifier	Description	Length in Words [2]	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
X		EXCLUSIVE OR									
	I/Q a.b	Input/output	1/2	0.2	0.1	0.05	0.01	2.9+	1.6+	0.11+	0.01+
	M a.b	Bit memory	1/2	0.3	0.2	0.05	0.01	3.2+	1.7+	0.11+	0.01+
	L a.b	Local data bit	2	0.7	0.3	0.06	0.02	3.7+	2.0+	0.10+	0.01+
	DBX a.b	Data bit	2	2.9	1.4	0.20	0.02	4.5+	2.4+	0.11+	0.01+
	DIX a.b	Instance data bit	2	2.9	1.4	0.20	0.02	4.5+	2.4+	0.09+	0.01+
	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 24)

[2] With direct instruction addressing/ with indirect instruction adressing

Instruction	**Address Identifier**	**Description**	**Length in Words** [2]	**Typical Execution Time in µs**							
				Direct Addressing				**Indirect Addressing** [1]			
				312	**31x, 147, 151, 154**	**317**	**319**	**312**	**31x, 147, 151, 154**	**317**	**319**
XN		EXCLUSIVE OR NOT									
	I/Q a.b	Input/output	1/2	0.3	0.2	0.05	0.01	3.2+	1.7+	0.11+	0.01+
	M a.b	Bit memory	1/2	0.4	0.2	0.05	0.01	3.5+	1.8+	0.11+	0.01+
	L a.b	Local data bit	2	0.8	0.4	0.06	0.02	3.9+	2.1+	0.10+	0.01+
	DBX a.b	Data bit	2	3.0	1.5	0.20	0.02	4.7+	2.5+	0.11+	0.01+
	DIX a.b	Instance data bit	2	3.0	1.5	0.20	0.02	4.7+	2.5+	0.10+	0.01+
	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	$\overline{\text{FC}}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	Yes
Instruction affects:	–	–	–	–	–	0	Yes	Yes	1

[1] Plus time required for loading the address of the instruction (see page 24)

[2] With direct instruction addressing/ with indirect instruction adressing

Bit Logic Instructions with Parenthetical Expressions

Saving the BR, RLO and OR bits and a function identifier (A, AN, ...) to the nesting stack. Seven nesting levels are possible per block. The listed parenthesese also apply to the “right parenthesis”-Instructions.

1

Instruction	**Address Identifier**	**Description**	**Length in Words**	**Typical Execution Time in μs[1]**			
				312	**31x, 147, 151, 154**	**317**	**319**
A(		AND left parenthesis	1	3.2	1.6	0.18	0.02
AN(		AND NOT left parenthesis	1	3.3	1.6	0.18	0.02
O(		OR left parenthesis	1	3.0	1.5	0.11	0.02
ON(		OR NOT left parenthesis	1	3.0	1.5	0.11	0.02
X(		EXCLUSIVE OR left parenthesis	1	3.0	1.5	0.11	0.02
XN(		EXCLUSIVE OR NOT left parenthesis	1	3.0	1.5	0.11	0.02

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

[1] also applies to “right parenthesis”- Instructions

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in μs			
				312	31x, 147, 151, 154	317	319
)		Right parenthesis, popping an entry off the nesting stack, gating the RLO with the current RLO in the processor	1	1.0	1.0	0.1	0.02

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

ORing of AND Operations

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
O		ORing of AND operations according to the rule: AND before OR	1	0.2	0.1	0.04	0.01

Status word for: **O**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
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.

Instruction	Address Identifier	Description	Length in Words 2)	Typical Execution Time in μs							
				Direct Addressing				Indirect Addressing 1)			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
A		AND									
	T f	Timer	1/2+	0.6	0.3	0.36	0.13	2.1+	1.1+	0.42+	0.13+
	C f	Counter	1/2+	0.3	0.2	0.10	0.09	2.0+	1.1+	0.13+	0.09+
	Timer para.	Timer/counter (adressed via parameter)	2	–	–	–	–	+	+	+	+
	Counter p.			–	–	–	–	+	+	+	+

Status word for: **A**	CC 1	BR	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
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 24)

2) With direct instruction addressing/ with indirect instruction adressing

Instruction	Address Identifier	Description	Length in Words [2]	Typical Execution Time in μs							
				Direct Addressing				Indirect Addressing [1]			
				312	**31x, 147, 151, 154**	**317**	**319**	**312**	**31x, 147, 151, 154**	**317**	**319**
AN		AND NOT									
	T f	Timer	1/2	0.8	0.4	0.36	0.13	2.3+	1.2+	0.42+	0.13+
	C f	Counter	1/2	0.5	0.3	0.10	0.09	2.2+	1.2+	0.13+	0.09+
	Timer para. Counter p.	Timer/counter (addressed via parameter)	2	– –	– –	– –	– –	+ +	+ +	+ +	+ +

Status word for: **AN**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
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 24)

[2] With direct instruction addressing/ with indirect instruction adressing

Instruction	Address Identifier	Description	Length in Words 2)	Typical Execution Time in μs							
				Direct Addressing				Indirect Addressing 1)			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
O	T f C f	OR timer OR counter	1/2 1/2	0.6 0.3	0.3 0.2	0.36 0.10	0.13 0.09	2.1+ 2.0+	1.1+ 1.0+	0.42+ 0.13+	0.13+ 0.09+
	Timerpara. Counter p.	OR timer/counter (adressed via parameter)	2	– –	– –	– –	– –	+ +	+ +	+ +	+ +
ON	T f C f	OR NOT timer OR NOT counter	1/2 1/2	0.8 0.5	0.4 0.3	0.36 0.10	0.13 0.09	2.3+ 2.2+	1.2+ 1.1+	0.42+ 0.13+	0.13+ 0.09+
	Timerpara. Counter p.	OR NOT timer/ counter (addressed via parameter)	2	– –	– –	– –	– –	+ +	+ +	+ +	+ +

[1] Plus time required for loading the address of the instruction (see page 24)
[2] With direct instruction addressing/ with indirect instruction adressing

Instruction	Address Identifier	Description	Length in Words [2]	Typical Execution Time in μs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
X		EXCLUSIVE OR									
	T f C f	timer counter	1/2 1/2	0.6 0.4	0.3 0.2	0.36 0.10	0.13 0.09	2.1+ 2.0+	1.1+ 1.1+	0.42+ 0.13+	0.13+ 0.09+
	Timerpara. Counter p.	EXCLUSIVE OR timer/counter (addressed via parameter)	2	– –	– –	– –	– –	+ +	+ +	+ +	+ +

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

[1] Plus time required for loading the address of the instruction (see page 24)
[2] With direct instruction addressing/ with indirect instruction adressing

Instruction	**Address Identifier**	**Description**	**Length in Words** [2]	**Typical Execution Time in µs**							
				Direct Addressing				**Indirect Addressing** [1]			
				312	**31x, 147, 151, 154**	**317**	**319**	**312**	**31x, 147, 151, 154**	**317**	**319**
XN	T f C f	EXCLUSIVE OR timer counter	1/2 1/2	0.8 0.5	0.4 0.3	0.36 0.10	0.13 0.09	2.3+ 2.2+	1.2+ 1.2+	0.42+ 0.13+	0.13+ 0.09+
	Timerpara. Counter p.	EXCLUSIVE OR NOT timer/counter (addressed via parameter)	2	– –	– –	– –	– –	+ +	+ +	+ +	+ +

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

[1] Plus time required for loading the address of the instruction (see page 24)

[2] With direct instruction addressing/ with indirect instruction adressing

Word Logic Instructions with the Contents of Accumulator 1

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
AW		AND ACCU2-L	1	0.6	0.3	0.21	0.02
AW	k16	AND 16-bit constant	2	0.6	0.3	0.19	0.02
OW		OR ACCU2-L	1	0.6	0.3	0.18	0.02
OW	k16	OR 16-bit constant	2	0.6	0.3	0.18	0.02
XOW		EXCLUSIVE OR ACCU2-L	1	0.6	0.3	0.21	0.02
XOW	k16	EXCLUSIVE OR 16-bit constant	2	0.6	0.3	0.21	0.02
AD		AND ACCU2	1	1.9	1.0	0.13	0.02
AD	k32	AND 32-bit constant	3	2.1	1.0	0.18	0.02

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
OD		OR ACCU2	1	1.9	1.0	0.13	0.02
OD	k32	OR 32-bit constant	3	2.1	1.0	0.18	0.02
XOD		EXCLUSIVE OR ACCU2	1	1.9	1.0	0.13	0.02
XOD	k32	EXCLUSIVE OR 32-bit constant	3	2.1	1.0	0.18	0.02

Status word for: **OD, XOD**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
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.

Instruction	**Address Identifier**	**Description**	**Length in Words**	**Typical Execution Time in µs**			
				312	**31x, 147, 151, 154**	**317**	**319**
A/ O/ X	==0	AND, OR, EXCLUSIVE OR Result=0 (CC 1=0)and (CC 0=0)	1	0.3	0.2	0.03	0.03
	>0	Result>0 (CC 1=1) and (CC 0=0)	1	0.5	0.3	0.05	0.03
	<0	Result<0 (CC 1=0)and (CC 0=1)	1	0.5	0.3	0.05	0.03
	<>0	Result≠0 ((CC1=0)and(CC 0=1)or (CC1=1)and(CC 0=0))	1	0.3	0.2	0.05	0.03
	<=0	R<=0((CC 1=0) and (CC 0=1) or (CC1=0) and (CC 0=0))	1	0.3	0.2	0.03	0.03
	>=0	R>=0((CC 1=1) and (CC 0=0) or (CC1=0) and (CC 0=0))	1	0.3	0.2	0.03	0.03
	UO	AND unordered math instruction (CC 1=1) and (CC 0=1)	1	0.3	0.2	0.03	0.03
	OS	AND OS=1	1	0.2	0.1	0.03	0.03
	BR	AND BR=1	1	0.2	0.1	0.03	0.03
	OV	AND OV=1	1	0.2	0.1	0.03	0.03

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

In-struction	**Address Identifier**	**Description**	**Length in Words**	**Typical Execution Time in µs**			
				312	**31x, 147, 151, 154**	**317**	**319**
AN/ ON/ XN	==0	AND NOT, OR NOT, EXCLUSIVE OR NOT Result=0 (CC 1=0) and (CC 0=0)	1	0.3	0.2	0.03	0.03
	>0	Result>0 (CC 1=1) and (CC 0=0)	1	0.5	0.3	0.05	0.03
	<0	Result<0 (CC 1=0) and (CC 0=1)	1	0.5	0.3	0.05	0.03
	<>0	Result≠0 ((CC 1=0) and (CC 0=1) or (CC 1=1) and (CC 0=0))	1	0.5	0.3	0.05	0.03
	<=0	Result<=0 ((CC 1=0) and (CC 0=1) or (CC 1=0) and (CC 0=0))	1	0.2	0.1	0.03	0.03
	>=0	Result>=0 ((CC 1=1) and (CC 0=0) or (CC 1=0) and (CC 0=0))	1	0.2	0.1	0.03	0.03
	UO	unordered math instruction (CC 1=1) and (CC 0=1)	1	0.5	0.3	0.03	0.03
	OS	OS=1	1	0.3	0.2	0.03	0.03
	BR	BR=1	1	0.3	0.2	0.03	0.03
	OV	OV=1	1	0.3	0.2	0.03	0.03

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

Edge-Triggered Instructions

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

Instruction	**Address Identifier**		**Description**	**Length in Words**	**Typical Execution Time in µs**							
					Direct Addressing				**Indirect Addressing** [1]			
					312	**31x, 147, 151, 154**	**317**	**319**	**312**	**31x, 147, 151, 154**	**317**	**319**
FP	I/Q	a.b	Detecting the positive edge in the RLO. The bit addressed in the instruction is the auxiliary edge bit memory.	2	0.5	0.3	0.13	0.04	3.3+	1.8+	0.10+	0.02+
	M	a.b		2	1.0	0.5	0.29	0.04	3.6+	1.9+	0.10+	0.02+
	L	a.b		2	1.2	0.6	0.30	0.04	4.0+	2.1+	0.08+	0.02+
	DBX	a.b		2	3.6	1.8	0.20	0.04	5.2+	2.7+	0.11+	0.02+
	DIX	a.b		2	3.6	1.8	0.20	0.04	5.2+	2.7+	0.09+	0.02+
	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	$\overline{\text{FC}}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	–	0	Yes	Yes	1

[1] Plus time required for loading the address of the instruction (see page 24)

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
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	0.7	0.3	0.13	0.04	3.5+	1.9+	0.10+	0.02+
	M a.b		2	1.1	0.5	0.13	0.04	3.8+	2.0+	0.10+	0.02+
	L a.b		2	1.3	0.7	0.14	0.04	4.2+	2.2+	0.08+	0.02+
	DBX a.b		2	3.7	1.9	0.20	0.04	5.2+	2.8+	0.11+	0.02+
	DIX a.b		2	3.7	1.9	0.20	0.04	5.2+	2.8+	0.09+	0.02+
	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	$\overline{\text{FC}}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	–	0	Yes	Yes	1

[1] Plus time required for loading the address of the instruction (see page 24)

Setting/Resetting Bit Addresses

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

In-struc-tion	Address Identifier	Description	Length in Words 2)	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing 1)			
				312	31x,147 151,154	317	319	312	31x,147 151,154	317	319
S	I/Q a.b	Set input/output to “1”	1/2	0.2	0.1	0.11	0.02	3.1+	1.7+	0.08+	0.02+
		(MCR-dependent)		0.3	0.2	0.13	0.06	3.3+	1.8+	0.10+	0.06+
	M a.b	Set bit memory to “1”	1/2	0.4	0.2	0.11	0.02	3.4+	1.8+	0.11+	0.02+
		(MCR-dependent)		1.8	0.9	0.13	0.06	3.7+	2.0+	0.12+	0.06+
	L a.b	Set local data bit to “1”	2	0.9	0.4	0.12	0.02	3.8+	2.0+	0.07+	0.02+
		(MCR-dependent)		2.0	1.0	0.14	0.06	3.9+	2.1+	0.09+	0.06+
	DBX a.b	Set data bit to “1”	2	3.4	1.7	0.19	0.02	4.8+	2.6+	0.10+	0.02+
		(MCR-dependent)		3.5	1.7	0.19	0.06	5.0+	2.7+	0.11+	0.06+
	DIX a.b	Set instance data bit to “1”	2	3.4	1.7	0.19	0.02	4.8+	2.6+	0.09+	0.02+
		(MCR-dependent)		3.5	1.7	0.19	0.06	5.0+	2.7+	0.11+	0.06+
	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	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	–	0	Yes	–	0

[1] Plus time required for loading the address of the instruction (see page 24)
[2] With direct instruction addressing/ with indirect instruction addressing

Instruction	Address Identifier	Description	Length in Word [2]	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
R	I/Q a.b	Reset input/output to “0”	1/2	0.3	0.1	0.12	0.02	3.2+	1.7+	0.08+	0.02+
		(MCR-dependent)		0.3	0.2	0.13	0.06	3.5+	1.8+	0.11+	0.06+
	M a.b	Set bit memory to “0”	1/2	0.5	0.3	0.12	0.02	3.5+	1.8+	0.11+	0.02+
		(MCR-dependent)		1.8	0.9	0.13	0.06	3.6+	1.9+	0.13+	0.06+
	L a.b	Set local data bit to“ 0”	2	0.9	0.4	0.12	0.02	3.9+	2.1+	0.10+	0.02+
		(MCR-dependent)		2.0	1.0	0.14	0.06	4.0+	2.1+	0.12+	0.06+
	DBX a.b	Set data bit to “0”	2	3.4	1.7	0.23	0.02	5.0+	2.6+	0.14+	0.02+
		(MCR-dependent)		3.6	1.8	0.25	0.06	5.1+	2.7+	0.16+	0.06+
	DIX a.b	Set instance data bit to “0”	2	3.4	1.7	0.23	0.02	5.0+	2.6+	0.13+	0.02+
		(MCR-dependent)		3.6	1.8	0.25	0.06	5.1+	2.7+	0.16+	0.06+
	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	$\overline{\text{FC}}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	–	0	Yes	–	0

[1] Plus time required for loading the address of the instruction (see page 24)

[2] With direct instruction addressing/ with indirect instruction addressing

In-struction	Address Identifier		Description	Length in Words [2]	Typical Execution Time in µs							
					Direct Addressing				Indirect Addressing [1]			
					312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
=	I/Q	a.b	Assign RLO to input/output	1/2	0.2	0.1	0.08	0.02	3.2+	1.7+	0.10+	0.02+
			(MCR-dependent)		0.3	0.2	0.10	0.06	3.4+	1.8+	0.11+	0.06+
	M	a.b	Assign RLO to bit memory	1/2	0.6	0.3	0.08	0.02	3.5+	1.8+	0.13+	0.02+
			(MCR-dependent)		1.8	0.9	0.10	0.06	3.7+	2.0+	0.13+	0.06+
	L	a.b	Assign RLO to local data bit	2	0.8	0.4	0.09	0.02	3.9+	2.0+	0.12+	0.02+
			(MCR-dependent)		2.1	1.0	0.11	0.06	4.1+	2.2+	0.12+	0.06+
	DBX	a.b	Assign RLO to data bit	2	3.4	1.7	0.23	0.02	5.0+	2.6+	0.16+	0.02+
			(MCR-dependent)		3.6	1.8	0.23	0.06	5.1+	2.7+	0.16+	0.06+
	DIX	a.b	Assign RLO to instance data bit	2	3.4	1.7	0.23	0.02	5.0+	2.6+	0.15+	0.02+
			(MCR-dependent)		3.6	1.8	0.23	0.06	5.1+	2.7+	0.16+	0.06+
	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	$\overline{FC}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	–	0	Yes	–	0

[1] Plus time required for loading the address of the instruction (see page 24)

[2] With direct instruction addressing/ with indirect instruction addressing

Instructions Directly Affecting the RLO

The following instructions have a direct effect on the RLO.

<table>
<tr><th rowspan="2">Instruction</th><th rowspan="2">Address Identifier</th><th colspan="2" rowspan="2">Description</th><th colspan="3" rowspan="2">Length in Words</th><th colspan="10">Typical Execution Time in µs</th></tr>
<tr><th colspan="2">312</th><th colspan="3">31x, 147, 151, 154</th><th colspan="3">317</th><th colspan="2">319</th></tr>
<tr><td>CLR</td><td></td><td colspan="2">Set RLO to ”0”</td><td colspan="3">2</td><td colspan="2">0.2</td><td colspan="3">0.1</td><td colspan="3">0.03</td><td colspan="2">0.01</td></tr>
<tr><td colspan="3">Status word for: <b>CLR</b></td><td colspan="2">BR</td><td>CC 1</td><td colspan="2">CC 0</td><td colspan="2">OV</td><td>OS</td><td colspan="2">OR</td><td>STA</td><td colspan="2">RLO</td><td>$\overline{\text{FC}}$</td></tr>
<tr><td colspan="3">Instruction depends on:</td><td colspan="2">–</td><td>–</td><td colspan="2">–</td><td colspan="2">–</td><td>–</td><td colspan="2">–</td><td>–</td><td colspan="2">–</td><td>–</td></tr>
<tr><td colspan="3">Instruction affects:</td><td colspan="2">–</td><td>–</td><td colspan="2">–</td><td colspan="2">–</td><td>–</td><td colspan="2">0</td><td>0</td><td colspan="2">0</td><td>0</td></tr>
<tr><td>SET</td><td></td><td colspan="2">Set RLO to ”1”</td><td colspan="3">2</td><td colspan="2">0.2</td><td colspan="3">0.1</td><td colspan="3">0.03</td><td colspan="2">0.01</td></tr>
<tr><td colspan="3">Status word for: <b>SET</b></td><td colspan="2">BR</td><td>CC 1</td><td colspan="2">CC 0</td><td colspan="2">OV</td><td>OS</td><td colspan="2">OR</td><td>STA</td><td colspan="2">RLO</td><td>$\overline{\text{FC}}$</td></tr>
<tr><td colspan="3">Instruction depends on:</td><td colspan="2">–</td><td>–</td><td colspan="2">–</td><td colspan="2">–</td><td>–</td><td colspan="2">–</td><td>–</td><td colspan="2">–</td><td>–</td></tr>
<tr><td colspan="3">Instruction affects:</td><td colspan="2">–</td><td>–</td><td colspan="2">–</td><td colspan="2">–</td><td>–</td><td colspan="2">0</td><td>1</td><td colspan="2">1</td><td>0</td></tr>
<tr><td>NOT</td><td></td><td colspan="2">Negate RLO</td><td colspan="3">2</td><td colspan="2">0.2</td><td colspan="3">0.1</td><td colspan="3">0.03</td><td colspan="2">0.01</td></tr>
<tr><td colspan="3">Status word for: <b>NOT</b></td><td colspan="2">BR</td><td>CC 1</td><td colspan="2">CC 0</td><td colspan="2">OV</td><td>OS</td><td colspan="2">OR</td><td>STA</td><td colspan="2">RLO</td><td>$\overline{\text{FC}}$</td></tr>
<tr><td colspan="3">Instruction depends on:</td><td colspan="2">–</td><td>–</td><td colspan="2">–</td><td colspan="2">–</td><td>–</td><td colspan="2">Yes</td><td>–</td><td colspan="2">Yes</td><td>–</td></tr>
<tr><td colspan="3">Instruction affects:</td><td colspan="2">–</td><td>–</td><td colspan="2">–</td><td colspan="2">–</td><td>–</td><td colspan="2">–</td><td>1</td><td colspan="2">Yes</td><td>–</td></tr>
</table>

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in μs			
				312	31x, 147, 151, 154	317	319
SAVE		Retain the RLO in the Bit BR	1	0.2	0.1	0.03	0.01

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.

Instruction	Address Identifier	Description	Length in Words[2]	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
SP	T f	Start timer as pulse on edge change from “0” to “1”	4/6	4.4	2.3	0.91	0.20	5.4+	2.9+	0.84+	0.20+
	Timer para.		2	–	–	–	–	+	+	+	+
SE	T f	Start timer as exded pulse on edge change from “0” to “1”	4/6	2.2	1.1	0.91	0.18	2.2+	1.2+	0.84+	0.18+
	Timer para.		2	–	–	–	–	+	+	+	+
SD	T f	Start timer as ON delay on edge change from “0” to “1”	4/6	4.6	2.4	0.91	0.23	5.5+	3.0+	0.85+	0.23+
	Timer para.		2	–	–	–	–	+	+	+	+
SS	T f	Start timer as retive ON delay on edge change from “0” to “1”	4/6	4.7	2.4	0.91	0.20	5.7+	3.0+	0.86+	0.20+
	Timer para.		2	–	–	–	–	+	+	+	+

Status word for: **SP, SE, SD, SS**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
Instruction depends on:	–	–	–	–	–	–	–	Yes	–
Instruction affects:	–	–	–	–	–	0	–	–	0

[1] Plus time required for loading the address of the instruction (see page 24)

[2] With direct instruction addressing/ with indirect instruction addressing

Instruction	Address Identifier	Description	Length in Words [2]	Typical Execution Time in μs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
SA	T f	Start timer as off-delay timer when the edge changes from "1" to "0".	4/6	4.9	2.5	0.97	0.24	5.9+	3.2+	0.88+	0.24+
	Timer para.		2	–	–	–	–	+	+	+	+
FR	T f	Enable timer for restarting on edge change from "0" to "1" (reset edge bit memory for starting timer)	4/6	2.3	1.2	0.79	0.10	2.8+	1.5+	0.70	0.10+
	Timer para.		2	–	–	–	–	+	+	+	+
R	T f	Reset timer	4/6	2.3	1.1	0.44	0.12	2.8+	1.5+	0.41	0.12+
	Timer para.		2	–	–	–	–	+	+	+	+

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

[1] Plus time required for loading the address of the instruction (see page 24)
[2] With direct instruction addressing

Counter Instructions

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

Instruction	Address Identifier	Description	Length in Words 2)	Typical Execution Time in µs							
				Direct Addressing				**Indirect Addressing 1)**			
				312	**31x, 147, 151, 154**	**317**	**319**	**312**	**31x, 147, 151, 154**	**317**	**319**
S	C f	Presetting of counter on edge change from “0” to “1”	4/6	3.3	1.7	0.33	0.14	4.5+	2.4+	0.29+	0.14+
	Counter p.		2	–	–	–	–	+	+	+	+
R	C f	Reset counter to “0”	4/6	1.3	0.6	0.17	0.10	2.1+	1.1+	0.13+	0.10+
	Counter p.		2	–	–	–	–	+	+	+	+
CU	C f	Increment counter by 1 on edge change from “0” to “1”	4/6	1.9	1.0	0.20	0.10	2.9+	1.6+	0.17+	0.10+
	Counter p.		2	–	–	–	–	+	+	+	+
CD	C f	Decrement counter by 1 on edge change from “0” to “1”	4/6	1.9	0.9	0.20	0.10	2.9+	1.5+	0.17+	0.10+
	Counter p.		2	–	–	–	–	+	+	+	+

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

[1] Plus time required for loading the address of the instruction (see page 24)

[2] With direct instruction addressing/ with indirect instruction addressing

Instruction	Address Identifier	Description	Length in Words [2]	Typical Execution Time in μs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
FR	C f	Enable counter on edge change from “0” to “1” (reset edge bit memory for up and down counting)	2	1.6	0.8	0.20	0.10	2.6+	1.4	0.17+	0.10+
	Counter p.		2	–	–	–	–	+	+	–	–

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

[1] Plus time required for loading the address of the instruction (see page 24)
[2] With direct instruction addressing/ with indirect instruction addressing

Load Instructions

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

In-struction	Address Identifier	Description	Length in Words [2]	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
L		Load ...									
	IB a	Input byte	1/2	0.4	0.2	0.05	0.01	2.7+	1.4+	0.14+	0.01+
	QB a	Output byte	1/2	0.4	0.2	0.05	0.01	2.7+	1.4+	0.14+	0.01+
	PIB a	Peripheral input byte for 31x	1/2	70.2	43.3	15.01	13.1	108.4+	44.6+	15.08+	13.1+
	PIB a	... for 147	1/2	–	50.5	–	–	–	51.8+	–	–
	PIB a	... for 151-7 (Bus <= 1m)	1/2	–	104.8	–	–	–	105.0+	–	–
	PIB a	... for 151-7 (Bus > 1m)	1/2	–	136.4	–	–	–	138.2+	–	–
	PIB a	... for 151-8 (Bus <= 1m)	1/2	–	68.3	–	–	–	69.6+	–	–
	PIB a	... for 151-8 (Bus > 1m)	1/2	–	88.8	–	–	–	90.5+	–	–
	PIB a	... for 154	1/2	–	68.3	–	–	–	69.6+	–	–
	PIB a	Digital Onboard I/O [3]	1/2	51.5	48.3	–	–	65.2+	55.6+	–	–
	PIB a	Analog Onboard I/O [4]	1/2	–	162.1	–	–	–	169.4+	–	–
	MB a	Bit memory byte	1/2	0.5	0.2	0.05	0.01	2.6+	1.4+	0.14+	0.01+
	LB a	Local data byte	2	0.9	0.5	0.05	0.02	3.3+	1.7+	0.13+	0.01+
	DBB a	Data byte	2	3.0	1.5	0.17	0.02	4.7+	2.5+	0.12+	0.01+
	DIB a	Instance data byte into ACCU1	2	3.0	1.5	0.17	0.02	4.7+	2.5+	0.12+	0.01+

[1] Plus time required for loading the address of the instruction (see page 24)
[2] With direct instruction addressing/ with indirect instruction addressing
[3] Access to digital onboard I/O
[4] Access to analog onboard I/O

Instruction	Address Identifier	Description	Length in Words [2]	Typical Execution Time in μs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
L	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 24)

[2] With direct instruction addressing/ with indirect instruction addressing

In-struction	Address Identifier	Description	Length in Words 2)	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing 1)			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
L		Load ...									
	IW a	Input word	1/2	0.6	0.3	0.10	0.01	2.9+	1.6+	0.15+	0.01+
	QW a	Output word	1/2	0.6	0.3	0.10	0.01	2.9+	1.6+	0.15+	0.01+
	PIW a	Peripheral input word for 31x	2	76.7	47.4	20.71	16.7	131.1+	48.9+	20.75+	16.7+
	PIW a	... for 147	2	–	56.2	–	–	–	57.8+	–	–
	PIW a	... for 151-7 (Bus <= 1m)	2	–	105.8	–	–	–	108.4+	–	–
	PIW a	... for 151-7 (Bus > 1m)	2	–	141.7	–	–	–	142.5+	–	–
	PIW a	... for 151-8 (Bus <= 1m)	2	–	72.9	–	–	–	74.2+	–	–
	PIW a	... for 151-8 (Bus > 1m)	2	–	97.7	–	–	–	99.4+	–	–
	PIW a	... for 154	2	–	72.9	–	–	–	74.2+	–	–
	PIW a	Digital Onboard I/O 3)	2	61.4	57.6	–	–	77.6+	66.3+	–	–
	PIW a	Analog Onboard I/O 4)	2	–	170.5	–	–	–	179.2+	–	–
	MW a	Bit memory word	1/2	0.8	0.4	0.10	0.01	3.2+	1.7+	0.15+	0.01+
	LW a	Local data word	2	1.1	0.6	0.10	0.02	3.8+	2.0+	0.16+	0.01+
	DBW a	Data word	1/2	3.5	1.8	0.24	0.02	5.6+	3.0+	0.16+	0.01+
	DIW a	Instance data word...into ACCU1	1/2	3.5	1.8	0.24	0.02	5.6+	3.0+	0.16+	0.01+
	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 24)
[2] With direct instruction addressing/ with indirect instruction addressing
[3] Access to digital onboard I/O
[4] Access to analog onboard I/O

In-struction	Address Identifier	Description	Length in Words 2)	Typical Execution Time in μs							
				Direct Addressing				Indirect Addressing 1)			
				312	31x, 147, 151, 154	317	319	312x	31x, 147, 151, 154	317	319
L		Load ...									
	ID a	Input double word	1/2	0.8	0.4	0.20	0.02	3.1+	1.6+	0.17+	0.01+
	QD a	Output double word	1/2	0.8	0.4	0.20	0.02	3.1+	1.6+	0.17+	0.01+
	PID a	Peripheral input double word	2	95.9	60.2	27.58	24.9	150.6+	61.9+	27.65+	24.9+
	PID a	... for 147	2	–	68.7	–	–	–	70.8+	–	–
	PID a	... for 151-7 (Bus <= 1m)	2	–	120.2	–	–	–	21.8+	–	–
	PID a	... for 151-7 (Bus > 1m)	2	–	161.0	–	–	–	163.6+	–	–
	PID a	... for 151-8 (Bus <= 1m)	2	–	81.6	–	–	–	82.9+	–	–
	PID a	... for 151-8 (Bus > 1m)	2	–	109.3	–	–	–	111.1+	–	–
	PID a	... for 154	2	–	81.6	–	–	–	82.9+	–	–
	PID a	Analog Onboard I/O 3)	2	–	303.0	–	–	–	323.0+	–	–
	MD a	Bit memory double word	1/2	1.0	0.5	0.19	0.02	3.8+	2.0+	0.17+	0.01+
	LD a	Local data double word	2	1.5	0.7	0.19	0.02	4.4+	2.3+	0.19+	0.01+
	DBD a	Data double word	2	4.7	2.3	0.33	0.02	6.9+	3.7+	0.19+	0.01+
	DID a	Instance data double word	2	4.7	2.3	0.33	0.02	6.9+	3.7+	0.19+	0.01+
		... into ACCU1									

[1] Plus time required for loading the address of the instruction (see page 24)

[2] With direct instruction addressing/ with indirect instruction addressing

[3] Access to analog onboard I/O

In-struction	Address Identifier	Description	Length in Words [2]	Typical Execution Time in μs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x, 147, 151, 154	317	319	312x	31x, 147, 151, 154	317	319
L	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 24)
[2] With direct instruction addressing/ with indirect instruction addressing

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs Direct Addressing 312	 31x, 147, 151, 154	 317	 319	 Indirect Addressing [1)] 312	 31x, 147, 151, 154	 317	 319
L		Load ...									
	k8	8-bit constant into ACCU1-LL	1	0.4	0.2	0.05	0.01	–	–	–	–
	k16	16-bit constant into ACCU1-L	2	0.4	0.2	0.05	0.01	–	–	–	–
	k32	32-bit constant into ACCU1	3	0.5	0.3	0.05	0.01	–	–	–	–
	Parameter	Load constant into ACCU1 (addressed via parameter)	2	–	–	–	–	+	+	+	+
L	2#n	Load 16-bit binary constant into ACCU1-L	2	0.4	0.2	0.05	0.01	–	–	–	–
		Load 32-bit binary constant into ACCU1	3	0.5	0.3	0.05	0.01	–	–	–	–
L	B#8#p	Load 8-bit hexadecimal constant into ACCU1-L	1	0.4	0.2	0.05	0.01	–	–	–	–
	W#16#p	Load 16-bit hexadecimal constant into ACCU1-L	2	0.4	0.2	0.05	0.01	–	–	–	–
	DW#16#p	Load 32-bit hexadecimal constant into ACCU1-L	3	0.5	0.3	0.05	0.01	–	–	–	–

[1)] Plus time required for loading the address of the instruction (see page 24)

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs 312	 31x, 147, 151, 154	 317	 319
L	’x’	Load 1 characters		0.4	0.2	0.05	0.01
L	’xx’	Load 2 characters	2	0.4	0.2	0.05	0.01
L	’xxx’	Load 3 characters		0.5	0.3	0.08	0.01
L	’xxxx’	Load 4 characters	3	0.5	0.3	0.08	0.01
L	D# date	Load IEC date (BCD)	3	0.5	0.3	0.08	0.01
L	S5T# time value	Load S5 time constant (16 bits)	2	0.5	0.3	0.05	0.01
L	TOD# time value	Load 32-bit time constant IEC – daytime	3	0.5	0.3	0.08	0.01
L	T# time value	Load 16-bit timer constant	2	0.4	0.2	0.05	0.01
		Load 32-bit timer constant	3	0.5	0.3	0.08	0.01
L	C# count value	Load 16-bit counter constant	2	0.4	0.2	0.05	0.01
L	P# bit pointer	Load bit pointer	3	0.5	0.3	0.08	0.01
L	L# integer	Load 32 bit integer constant	3	0.5	0.3	0.08	0.01
L	Real number	Load real number	3	0.5	0.3	0.08	0.01

Load Instructions for Timers and Counters

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

Instruction	Operand	Description	Length in Words 2)	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing 1)			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
L	T f	Load time value	1/2	1.7	0.8	0.43	0.19	2.0+	1.1+	0.39+	0.19+
	Timer para.	Load time value (adressed via parameter)	2	–	–	–	–	+	+	+	+
L	C f	Load count value	1/2	1.4	0.7	0.14	0.08	2.3+	1.2+	0.11+	0.08+
	Counter para.	Load count value (adressed via parameter)	2	–	–	–	–	+	+	+	+
LD	T f	Load time value in BCD	1/2	4.2	2.2	0.87	0.30	5.0+	2.5+	0.84+	0.30+
	Timer para.	Load time value in BCD (adressed via parameter)	2	–	–	–	–	+	+	+	+
LD	C f	Load count value in BCD	1/2	4.4	2.2	0.56	0.19	5.4+	2.9+	0.53+	0.19+
	Counter para.	Load count value (adressed via parameter)	2	–	–	–	–	+	+	+	+

[1] Plus time required for loading the address of the instruction (see page 24)
[2] With direct instruction addressing/ with indirect instruction addressing

Transfer Instructions

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

In-struction	Address Identifier	Description	Length in Words [2]	Typical Execution Time in μs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x,147, 151,154	317	319	312	31x,147, 151,154	317	319
T		Transfer contents of ACCU1-LL to ...									
	IB a	input byte	1/2	0.2	0.1	0.06	0.01	2.4+	1.3+	0.13+	0.01+
		(MCR-dependent)		1.1	0.5	0.12	0.05	2.7+	1.5+	0.15+	0.05+
	QB a	output byte	1/2	0.2	0.1	0.06	0.01	2.4+	1.3+	0.12+	0.01+
		(MCR-dependent)		1.1	0.5	0.12	0.05	2.7+	1.5+	0.15+	0.05+
	PQB a	peripheral output byte for 31x	1/2	58.7	35.9	13.10	10.3	104.8+	37.5+	13.11+	10.3+
		(MCR–dependent)		58.8	36.1	13.53	10.3	105.2+	37.8+	13.51+	10.3+
	PQB a	... for 147	1/2	–	45.1	–	–	–	46.6+	–	–
		... for 147 (MCR–dependent)		–	45.3	–	–	–	46.8+	–	–

[1] Plus time required for loading the address of the instruction (see page 24)

[2] With direct instruction addressing/ with indirect instruction addressing

Instruction	Address Identifier	Description	Length in Words [2]	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x,147, 151,154	317	319	312	31x,147, 151,154	317	319
T	PQB a	... for 151-7 (Bus <= 1m)	1/2	–	93.1	–	–	–	94.9+	–	–
		... for 151-7 (MCR–dependent)		–	93.6	–	–	–	95.4+	–	–
	PQB a	... for 151-7 (Bus > 1m)	1/2	–	118.9	–	–	–	121.2+	–	–
		... for 151-7 (MCR–dependent)		–	119.2	–	–	–	121.4+	–	–
	PQB a	... for 151-8 (Bus <= 1m)	1/2	–	63.7	–	–	–	65.0+	–	–
		... for 151-8 (MCR–dependent)		–	64.6	–	–	–	65.9+	–	–
	PQB a	... for 151-8 (Bus > 1m)	1/2	–	81.4	–	–	–	83.0+	–	–
		... for 151-8 (MCR–dependent)		–	82.3	–	–	–	83.9+	–	–
	PQB a	... for 154	1/2	–	63.7	–	–	–	65.0+	–	–
		... for 154 (MCR–dependent)		–	64.6	–	–	–	65.9+	–	–
T	PQB a	Digital Onboard I/O [3]	1/2	57.3	53.9	–	–	70.6+	61.0+	–	–
		(MCR-dependent)		58.2	54.4	–	–	71.2+	61.3+	–	–
	PQB a	Analog Onboard I/O [4]	1/2	–	49.2	–	–	–	56.3+	–	–
		(MCR-dependent)		–	49.7	–	–	–	56.8+	–	–

[1] Plus time required for loading the address of the instruction (see page 24)

[2] With direct instruction addressing/ with indirect instruction addressing

[3] Access to digital onboard I/O

[4] Access to analog onboard I/O

In-struction	Address Identifier	Description	Length in Words [2]	Typical Execution Time in μs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
T	MB a	bit memory byte	1/2	0.2	0.1	0.06	0.01	2.4+	1.3+	0.13+	0.01+
		(MCR-dependent)		1.2	0.6	0.12	0.05	2.7+	1.5+	0.15+	0.05+
	LB a	local data byte	2	0.4	0.2	0.06	0.02	3.3+	1.7+	0.11+	0.01+
		(MCR-dependent)		1.5	0.8	0.14	0.05	2.9+	1.5+	0.16+	0.05+
	DBB a	data byte	2	2.7	1.3	0.24	0.02	4.1+	2.2+	0.13+	0.01+
		(MCR-dependent)		2.7	1.3	0.16	0.05	4.5+	2.4+	0.16+	0.05+
	DIB a	instance data byte	2	2.4	1.3	0.24	0.02	4.1+	2.2+	0.14+	0.01+
		(MCR-dependent)		2.7	1.3	0.16	0.05	4.5+	2.4+	0.16+	0.05+
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 24)
[2] With direct instruction addressing/ with indirect instruction addressing

Instruction	Address Identifier	Description	Length in Words 2)	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing 1)			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
T		Transfer contents of ACCU1-L to...									
		input word	1/2	0.4	0.2	0.13	0.01	2.6+	1.4+	0.14+	0.01+
	IW	(MCR-dependent)		1.1	0.6	0.13	0.05	2.9+	1.5+	0.16+	0.05+
		output word	1/2	0.4	0.2	0.13	0.01	2.6+	1.4+	0.14+	0.01+
	QW	(MCR-dependent)		1.1	0.6	0.13	0.05	2.9+	1.5+	0.16+	0.05+
		peripheral output word									
	PQW	(MCR-dependent)	1/2	64.4	40.4	15.04	11.6	121.6+	41.8+	14.99+	11.6+
		... for 147		64.6	40.6	15.32	11.6	120.5+	42.1+	15.43+	11.6+
	PQW	... for 147 (MCR–dependent)	1/2	–	52.8	–	–	–	53.9+	–	–
				–	53.1	–	–	–	54.1+	–	–

[1] Plus time required for loading the address of the instruction (see page 24)
[2] With direct instruction addressing/ with indirect instruction addressing

In-struction	Address Identifier	Description	Length in Words [2]	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
T	PQW	... for 151-7 (Bus <= 1m)	1/2	–	98.9	–	–	–	100.3+	–	–
		... for 151-7 (MCR–dependent)		–	99.0	–	–	–	100.6+	–	–
	PQW	... for 151-7 (Bus > 1m)	1/2	–	126.9	–	–	–	128.1+	–	–
		... for 151-7 (MCR–dependent)		–	126.4	–	–	–	128.4+	–	–
	PQW	... for 151-8 (Bus <= 1m)	1/2	–	67.8	–	–	–	69.1+	–	–
		... for 151-8 (MCR–dependent)		–	69.6	–	–	–	70.9+	–	–
	PQW	... for 151-8 (Bus > 1m)	1/2	–	86.6	–	–	–	88.3+	–	–
		... for 151-8 (MCR–dependent)		–	87.5	–	–	–	89.2+	–	–
	PQW	... for 154	1/2	–	67.8	–	–	–	69.1+	–	–
		... for 154 (MCR–dependent)		–	69.6	–	–	–	70.9+	–	–
		Digital Onboard I/O [3]	1/2	70.5	66.1	–	–	85.8+	74.2+	–	–
		(MCR-dependent)		71.1	66.4	–	–	86.4+	74.8+	–	–
		Analog Onboard I/O [4]	1/2	–	66.1	–	–	–	74.2+	–	–
		(MCR-dependent)		–	66.4	–	–	–	74.8+	–	–

[1] Plus time required for loading the address of the instruction (see page 24)

[2] With direct instruction addressing/ with indirect instruction addressing

[3] Access to digital onboard I/O

[4] Access to analog onboard I/O

In-struction	Address Identifier	Description	Length in Words [2]	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
T	MW	bit memory word	1/2	0.4	0.2	0.18	0.01	3.2+	1.7+	0.16+	0.01+
		(MCR-dependent)		1.5	0.7	0.15	0.05	3.5+	1.9+	0.18+	0.05+
	LW	local data word	2	0.5	0.2	0.12	0.02	3.8+	2.0+	0.15+	0.01+
		(MCR-dependent)		1.6	0.8	0.15	0.05	3.3+	1.8+	0.22+	0.05+
	DBW	data word	2	3.2	1.6	0.30	0.02	4.8+	2.6+	0.17+	0.01+
		(MCR-dependent)		3.2	1.6	0.16	0.05	5.2+	2.8+	0.19+	0.05+
	DIW	Instanz-data word	2	3.2	1.5	0.30	0.02	4.8+	2.6+	0.17+	0.01+
		(MCR-dependent)		3.2	1.6	0.15	0.05	5.2+	2.8+	0.19+	0.05+
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	–	–	–	–	+	+	+	+

[1] Plus time required for loading the address of the instruction (see page 24)
[2] With direct instruction addressing/ with indirect instruction addressing

In-struction	Address Identifier	Description	Length in Words [2]	Typical Execution Time in μs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
T		Transfer contents of ACCU1 to ...									
	ID	input double word	1/2	0.6	0.3	0.22	0.01	2.8+	1.5+	0.16+	0.01+
		(MCR-dependent)		1.4	0.7	0.16	0.05	3.2+	1.7+	0.18+	0.05+
	QD	output double word	1/2	0.6	0.3	0.22	0.01	2.8+	1.5+	0.16+	0.01+
		(MCR-dependent)		1.4	0.7	0.16	0.05	3.2+	1.7+	0.18+	0.05+
	PQD	peripheral output double word	1/2	73.1	45.4	18.43	15.1	130.1+	46.8+	18.44+	15.1+
		(MCR-dependent)		73.4	45.5	18.87	15.1	128.0+	47.0+	19.07+	15.1+
	PQD	... for 147	1/2	–	63.7	–	–	–	65.0+	–	–
		... for 147 (MCR–dependent)		–	63.7	–	–	–	65.3+	–	–
	PQD	... for 151-7 (Bus <= 1m)	1/2	–	111.7	–	–	–	113.5+	–	–
		... for 151-7 (MCR–dependent)		–	111.8	–	–	–	113.8+	–	–
	PQD	... for 151-7 (Bus > 1m)	1/2	–	148.9	–	–	–	150.7+	–	–
		... for 151-7 (MCR–dependent)		–	149.4	–	–	–	151.1+	–	–
	PQD	... for 151-8 (Bus <= 1m)	1/2	–	76.1	–	–	–	77.4+	–	–
		... for 151-8 (MCR–dependent)		–	86.4	–	–	–	87.7+	–	–
	PQD	... for 151-8 (Bus > 1m)	1/2	–	101.5	–	–	–	103.2+	–	–
		... for 151-8 (MCR–dependent)		–	115.2	–	–	–	116.9+	–	–
	PQD	... for 154	1/2	–	76.1	–	–	–	77.4+	–	–
		... for 154 (MCR–dependent)		–	86.4	–	–	–	87.7+	–	–
	PQD	Analog Onboard I/O [3]	1/2	–	91.3	–	–	–	100.4+	–	–
		(MCR-dependend)		–	91.9	–	–	–	101.3+	–	–

[1] Plus time required for loading the address of the instruction (see page 24)
[2] With direct instruction addressing/ with indirect instruction addressing
[3] Access to digital onboard I/O

Instruction	Address Identifier	Description	Length in Words [2]	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
T	MD	bit memory double word	1/2	0.6	0.3	0.27	0.01	3.8+	2.0+	0.19+	0.01+
		(MCR-dependent)		1.7	0.8	0.18	0.05	4.2+	2.3+	0.22+	0.05+
	LD	local data double word	2	0.9	0.4	0.22	0.02	4.4+	2.4+	0.18+	0.02+
		(MCR-dependent)		2.0	1.0	0.18	0.05	4.0+	2.1+	0.25+	0.05+
	DBD	Data double word	2	4.5	2.2	0.19	0.02	5.7+	3.0+	0.20+	0.02+
		(MCR-dependend)		4.4	2.2	0.21	0.05	6.1+	3.3+	0.23+	0.05+
	DID	Instanz data double word	2	4.5	2.2	0.18	0.02	5.7+	3.0+	0.19+	0.02+
		(MCR-dependend)		4.4	2.2	0.20	0.05	6.1+	3.3+	0.22+	0.05+
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 24)
[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.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
LAR1		Load contents from ...					
	–	ACCU1	1	0.2	0.1	0.03	0.02
	AR2	Address register 2	1	0.2	0.1	0.03	0.04
	DBD a	Data double word	2	4.6	2.3	0.20	0.06
	DID a	Instance data double word	2	4.6	2.3	0.20	0.06
	m	32-bit constant as pointer	3	0.3	0.2	0.05	0.03
	LD a	Local data double word	2	1.5	0.7	0.20	0.06
	MD a	Bit memory double word	2	1.0	0.5	0.20	0.06
		... into AR1					
LAR2		Load contents from ...					
	–	ACCU1	1	0.2	0.1	0.03	0.02
	DBD a	Data double word	2	4.6	2.3	0.20	0.06
	DID a	Instance data double word	2	4.6	2.3	0.20	0.06
	m	32-bit constant as pointer	3	0.3	0.2	0.05	0.03
	LD a	Local data double word	2	1.5	0.7	0.20	0.06
	MD a	Bit memory double word	2	1.0	0.5	0.20	0.06
		... into AR2					

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in μs			
				312	31x, 147, 151, 154	317	319
TAR1		Transfer contents of AR1 to ...					
	–	ACCU1	1	0.3	0.2	0.04	0.04
	AR2	Address register 2	1	0.2	0.1	0.03	0.04
	DBD a	Data double word	2	4.4	2.2	0.20	0.06
	DID a	Instance data double word	2	4.4	2.2	0.20	0.06
	m	32-bit constant as pointer	2	0.9	0.4	0.22	0.06
	LD a	Local data double word	2	0.6	0.3	0.22	0.06
	MD a	Bit memory double word					
TAR2		Transfer contents of AR2 to ...					
	–	ACCU1	1	0.3	0.2	0.04	0.04
	DBD a	Data double word	2	0.2	0.1	0.20	0.06
	DID a	Instance data double word	2	4.4	2.2	0.20	0.06
	LD a	Local data double word	2	4.4	2.2	0.20	0.06
	MD a	Bit memory double word	2	0.9	0.4	0.20	0.06
TAR	–	Exchange the contents of AR1 and AR2	1	0.6	0.3	0.06	0.02

Load and Transfer Instructions for the Status Word

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
L	STW	Load status word [1] into ACCU1		1.1	0.6	0.09	0.03

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

Instruction	Address Identifier	Description	Length in Words	312	31x, 147, 151, 154	317	319
T	STW	Transfer ACCU1 (bits 0 to 8) to the status word [1]		1.1	0.6	0.23	0.02

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

[1] For the structure of the status word see page 17

Load Instructions for DB Number and DB Length

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
L	DBNO	Load number of data block	1	2.4	1.3	0.18	0.03
L	DINO	Load number of instance data block	1	2.4	1.3	0.18	0.03
L	DBLG	Load length of data block into byte	1	0.5	0.3	0.04	0.03
L	DILG	Load length of instance data block into byte	1	0.5	0.3	0.04	0.03

Integer Math (16 Bits)

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

Instruction	**Address Identifier**	**Description**	**Length in Words**	**Typical Execution Time in µs**			
				312	**31x, 147, 151, 154**	**317**	**319**
+I	–	Add 2 integers (16 bits) (ACCU1-L)=(ACCU1-L)+ (ACCU2-L)	1	1.3	0.6	0.20	0.02
–I	–	Subtract 1 integer from another (16 bits) (ACCU1-L)=(ACCU2-L)– (ACCU1-L)	1	1.5	0.7	0.17	0.02
*I	–	Multiply 1 integer by another (16 bits) (ACCU1)=(ACCU2-L)* (ACCU1-L)	1	2.2	1.1	0.22	0.02
/I	–	Divide 1 integer by another (16 bits) (ACCU1-L)= (ACCU2-L):(ACCU1-L) The remainder is in ACCU1-H	1	2.6	1.3	0.35	0.06

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

Integer Math (32 Bits)

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in μs			
				312	31x, 147, 151, 154	317	319
+D	–	Add 2 integers (32 bits) (ACCU1)=(ACCU2)+ (ACCU1)	1	1.6	0.8	0.16	0.01
–D	–	Subtract 1 integer from another (32 bits) (ACCU1)=(ACCU2)– (ACCU1)	1	2.2	1.1	0.18	0.01
*D	–	Multiply 1 integer by another (32 bits) (ACCU1)=(ACCU2)* (ACCU1)	1	7.1	3.5	0.17	0.01
/D	–	Divide 1 integer by another (32 bits) (ACCU1)=(ACCU2): (ACCU1)	1	5.7	2.8	0.43	0.06
MOD	–	Divide 1 integer by another (32 bits) and load the remainder into ACCU1: (ACCU1)=remainder of [(ACCU2):(ACCU1)]	1	3.8	1.9	0.15	0.06

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

Floating-Point Math (32 Bits)

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
+R	–	Add 2 real numbers (32 bits) (ACCU1)=(ACCU2)+(ACCU1)	1	5.5	2.7	0.98	0.04
–R	–	Subtract 1 real number from another (32 bits) (ACCU1)=(ACCU2)–(ACCU1)	1	5.5	2.7	0.98	0.04
*R	–	Multiply 1 real number by another (32 bits) (ACCU1)=(ACCU2)*(ACCU1)	1	6.4	3.2	0.55	0.04
/R	–	Divide 1 real number by another (32 bits) (ACCU1)=(ACCU2):(ACCU1)	1	6.1	3.0	1.46	0.06

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in μs			
				312	31x, 147, 151, 154	317	319
NEGR	–	Negate the real number in ACCU1	1	0.8	0.4	0.03	0.01
ABS	–	Form the absolute value of the real number in ACCU1	1	0.8	0.4	0.03	0.01

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

Square Root and Square Instructions (32 Bits)

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

Instruction	**Address Identifier**	**Description**	**Length in Words**	**Typical Execution Time in μs**			
				312	**31x, 147, 151, 154**	**317**	**319**
SQRT	–	Calculate the square root of a real number in ACCCU1	1	643	322	30.03	0.64
SQR	–	Form the square of a real number in ACCU1	1	177	89	5.02	0.04

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

Logarithmic Function (32 Bits)

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
LN	–	Form the natural logarithm of a real number in ACCU1	1	455	227	14.97	0.69
EXP	–	Calculate the exponential value of a real number in ACCU1 to the base e (= 2.71828)	1	898	449	33.71	0.67

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

Trigonometrical Functions (32 Bits)

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
SIN [1]	–	Calculate the sine of a real number	1	545	272	21.52	0.48
ASIN [2]	–	Calculate the arcsine of a real number	1	1584	792	61.07	0.73
COS [1]	–	Calculate the cosine of a real number	1	606	303	23.54	0.50
ACOS [2]	–	Calculate the arccosine of a real number	1	1762	881	67.47	0.73
TAN [1]	–	Calculate the tangent of a real number	1	549	274	21.39	0.62
ATAN [2]	–	Calculate the arctangent of a real number	1	595	297	22.09	0.54

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

[1] Specify the angle in radians; the angle must be given as a floating point value in ACCU 1.
[2] The result is an angle in radians.

Adding Constants

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
+	i8	Add an 8-bit integer constant	1	0.2	0.1	0.08	0.01
+	i16	Add a 16-bit integer constant	2	0.2	0.1	0.08	0.01
+	i32	Add a 32-bit integer constant	3	0.3	0.2	0.08	0.01

Adding Using Address Registers

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in μs			
				312	31x, 147, 151, 154	317	319
+AR1	–	Add the contents of ACCU1-L to those of AR1	1	0.2	0.1	0.1	0.02
+AR1	m	Add a pointer constant to the contents of AR1	2	0.4	0.2	0.1	0.02
+AR2	–	Add the contents of ACCU1-L to those of AR2	1	0.2	0.1	0.1	0.02
+AR2	m	Add pointer constant to the contents of AR2	2	0.4	0.2	0.1	0.02

Comparison Instructions with Integers (16 Bits)

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

Identifier	Address Instruction	Description	Length in Words	Typical Execution Time in μs			
				312	**31x, 147, 151, 154**	**317**	**319**
==I	–	ACCU2-L=ACCU1-L	1	1.4	0.7	0.14	0.03
<>I	–	ACCU2-L≠ACCU1-L	1	1.6	0.8	0.14	0.03
<I	–	ACCU2-L<ACCU1-L	1	1.6	0.7	0.14	0.03
<=I	–	ACCU2-L<=ACCU1-L	1	1.4	0.7	0.14	0.03
>I	–	ACCU2-L>ACCU1-L	1	1.3	0.7	0.14	0.03
>=I	–	ACCU2-L>=ACCU1-L	1	1.4	0.7	0.14	0.03

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

Comparison Instructions with Integers (32 Bits)

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
==D	–	ACCU2=ACCU1	1	1.4	0.7	0.10	0.03
<>D	–	ACCU2≠ACCU1	1	1.4	0.7	0.10	0.03
<D	–	ACCU2<ACCU1	1	1.4	0.7	0.10	0.03
<=D	–	ACCU2<=ACCU1	1	1.4	0.7	0.10	0.03
>D	–	ACCU2>ACCU1	1	1.3	0.7	0.10	0.03
>=D	–	ACCU2>=ACCU1	1	1.3	0.7	0.10	0.03

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.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
== R	–	ACCU2=ACCU1	1	6.3	3.1	0.50	0.06
<> R	–	ACCU2≠ACCU1	1	6.3	3.1	0.47	0.06
< R	–	ACCU2<ACCU1	1	6.4	3.2	0.47	0.06
< = R	–	ACCU2<=ACCU1	1	6.3	3.1	0.47	0.06
> R	–	ACCU2>ACCU1	1	6.3	3.1	0.49	0.06
> = R	–	ACCU2>=ACCU1	1	6.4	3.2	0.48	0.06

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

Shift Instructions

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
SLW	–	Shift the contents of ACCU1-L to the left. Positions that become free are provided with zeros.	1	1.9	1.0	0.19	0.03
	0 ... 15			0.6	0.3	0.19	0.03
SLD	–	Shift the contents of ACCU1 to the left. Positions that become free are provided with zeros.	1	2.5	1.2	0.22	0.03
	0 ... 32			2.5	1.3	0.26	0.03
SRW	–	Shift the contents of ACCU1-L to the right. Positions that become free are provided with zeros.	1	1.9	0.9	0.23	0.03
	0 ... 15			0.6	0.3	0.33	0.03
SRD	–	Shift the contents of ACCU1 to the right. Positions that become free are provided with zeros.	1	2.5	1.2	0.24	0.03
	0 ... 32			2.5	1.3	0.28	0.03

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

Instruction	**Address Identifier**	**Description**	**Length in Words**	**Typical Execution Time in µs**			
				312	**31x, 147, 151, 154**	**317**	**319**
SSI	–	Shift the contents of ACCU1-L with sign to the right. Positions that become free are provided with the sign (bit 15).	1	1.8	0.9	0.22	0.03
	0 ... 15			0.6	0.3	0.33	0.03
SSD	–	Shift the contents of ACCU1 with sign to the right	1	2.5	1.2	0.24	0.03
	0 ... 32			2.5	1.3	0.28	0.03

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

Rotate Instructions

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in μs			
				312	31x, 147, 151, 154	317	319
RLD	–	Rotate the contents of ACCU1 to the left	1	2.2	1.1	0.18	0.03
	0 ... 32			3.2	1.6	0.24	0.03
RRD	–	Rotate the contents of ACCU1 to the right	1	2.2	1.1	0.23	0.03
	0 ... 32			2.4	1.2	0.28	0.03

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	–	–	–	–	–

RLDA	–	Rotate the contents of ACCU1 one bit position to the left t	1	1.7	0.8	0.14	0.02
RRDA	–	Rotate the contents of ACCU1 one bit position to the right	1	1.7	0.8	0.14	0.02

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.

Instruc-tion	Address Identifier	Description	Length in Words	Typical Execution Time in μs			
				312	31x, 147, 151, 154	317	319
CAW	–	Reverse the order of the bytes in ACCU1-L. LL, LH becomes LH, LL.	1	0.2	0.1	0.10	0.01
CAD	–	Reverse the order of the bytes in ACCU1. LL, LH, HL, AA becomes HH, HL, LH, LL.	1	0.4	0.2	0.23	0.01
TAK	–	Swap the contents of ACCU1 and ACCU2	1	0.5	0.3	0.06	0.01
PUSH	–	The contents of ACCU1 are transferred to ACCU2.	1	0.2	0.1	0.03	0.01
POP	–	The contents of ACCU2 are transferred to ACCU1:	1	0.2	0.1	0.03	0.01
INC	0 ... 255	Increment ACCU1-LL	1	0.2	0.1	0.10	0.01
DEC	0 ... 255	Decrement ACCU1-LL	1	0.2	0.1	0.10	0.01

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	31x, 147, 151, 154	317	319
BLD	0 ... 255	Program display instruction: Is treated by the CPU like a null operation instruction.	1	0.2	0.1	0.04	0
NOP	0 1	Null Operation instruction:	1	0.2 0.2	0.1 0.1	0.04 0.04	0

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	31x, 147, 151, 154	317	319
BTI	–	Conv. cont. of ACCU1 from BCD to integer (16 bits) (**B**CD **T**o **I**nt)	1	3.9	1.9	0.32	0.03
BTD	–	Conv. cont. of ACCU1 from BCD to double int. (32 bits) (**B**CD **T**o **D**oubleint)	1	8.6	4.3	0.68	0.05
DTR	–	Convert contents of ACCU1 from double integer to real (32 bits) (**D**oubleint **T**o **R**eal)	1	5.5	2.7	0.33	0.02
ITD	–	Convert contents of ACCU1 from integer (16 bits) to double int. (32 bits) (**I**nt **T**o **D**oubleint)	1	0.2	0.1	0.03	0.02

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in μs			
				312	31x, 147, 151, 154	317	319
ITB	–	Conv. cont. of ACCU1 from int. (16 bits) to BCD from 0 to +/– 999 (**I**nt **T**o **B**CD)	1	4.4	2.2	0.57	0.13
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	10.0	5.0	1.38	0.33

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	**31x, 147, 151, 154**	**317**	**319**
RND	–	Convert a real number into a 32-bit integer.	1	6.5	3.2	0.41	0.02
RND–	–	Convert a real number into a 32-bit integer. The number is rounded to the next whole number.	1	6.5	3.3	0.41	0.02

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

RND+	–	Convert a real number into a 32-bit integer. The number is rounded to the next whole number.	1	6.7	3.3	0.42	0.02
TRUNC	–	Convert a real number into a 32-bit integer. The places after the decimal point are truncated.	1	6.3	3.1	0.41	0.02

Status word for: **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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
INVI	–	Form the ones complement of ACCU1-L	1	0.2	0.1	0.05	0.01
INVD	–	Form the ones complement of ACCU1	1	0.2	0.1	0.08	0.01

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

Instruction	Address Identifier	Description	Length in Words	312	31x, 147, 151, 154	317	319
NEGI	–	Form the twos complement of ACCU1-L (integer)	1	1.4	0.7	0.19	0.01
NEGD	–	Form the twos complement of ACCU1 (double integer)	1	1.6	0.8	0.16	0.01

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

Block Call Instructions

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
CALL	FB q, DB q	Unconditional call of an FB, with parameter transfer	1	16.4	8.8	1.9	0.68	–	–	–	–
CALL	SFB q, DB q	Unconditional call of an SFB, with parameter transfer	2	[2]	[2]	[2]	[2]	–	–	–	–
CALL	FC q	Unconditional call of a function, with parameter transfer	1	15.6	7.5	1.72	0.61	–	–	–	–
CALL	SFC q	Unconditional call of an SFC, with parameter transfer	2	[2]	[2]	[2]	[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 24)
[2] See chapter System Functions (SFCs)/ see chapter System Function Blocks (SFBs)

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs							
				Direct Addressing				Indirect Addressing [1]			
				312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
UC	FB q FC q Parameter	Unconditional call of blocks without parameter transfer FB/FC call via parameter	1[3]	9.1 9.1 9.1	6.0 6.0 6.0	1.47 1.55	0.59 0.59 0.59	9.8+ 9.8+ 9.8+	6.4+ 6.4+ 6.4+	1.63+ 1.70+	0.59+ 0.59+ 0.59+
CC	FB q FC q Parameter	Conditional call of blocks without parameter transfer FB/FC call via parameter	1[3]	9.4 9.4 9.4	6.2 6.2 6.2	1.53 1.59	0.59 0.59 0.59	9.9+ 9.9+ 9.9+	6.6+ 6.6+ 6.6+	1.65+ 1.73+	0.59+ 0.59+ 0.59+

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

Instruction	Address Identifier	Description	Length in Words	312	31x, 147, 151, 154	317	319	312	31x, 147, 151, 154	317	319
OPN	DB q DI q Parameter	Open: Data block Instance data block Data block using parameters	1/2[2] 2 2	0.7	0.7	0.15	0.03	1.2+	1.2+	0.25+	0.03+

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 24)
[2] Block No. > 255
[3] With direct instruction addressing

Block End Instructions

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in μs			
				312	31x, 147, 151, 154	317	319
BE	–	End block	1	4.4	2.2	0.05	0.07
BEU	–	End block unconditionally	1	4.4	2.2	0.05	0.07

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

Instruction	Address Identifier	Description	Length in Words	312	31x, 147, 151, 154	317	319
BEC	–	End block conditionally if RLO = “1”	1	1.2	0.6	0.14	0.07

Status word for: **BEC**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
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.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
CDB		Exchange shared data block and instance data block	1	0.2	0.1	0.18	0.06

Jump Instructions

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

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
JU	LABEL	Jump unconditionally	1 [1]/2	3.6	1.8	0.43	0.03

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

Instruction	Address Identifier	Description	Length in Words	312	31x, 147, 151, 154	317	319
JC	LABEL	Jump if RLO = “1”	1 [1]/2	3.8	1.9	0.51	0.03
JCN	LABEL	Jump if RLO = “0”	2	3.8	1.9	0.51	0.03

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

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
JCB	LABEL	Jump if RLO = “1”. Save the RLO in the BR bit	2	3.8	1.9	0.51	0.06
JNB	LABEL	Jump if RLO = “0”. Save the RLO in the BR bit	2	3.8	1.9	0.51	0.06

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

Instruction	Address Identifier	Description	Length in Words	312	31x, 147, 151, 154	317	319
JBI	LABEL	Jump if BR = “1”	2	3.8	1.9	0.51	0.06
JNBI	LABEL	Jump if BR = “0”	2	3.8	1.9	0.51	0.06

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

Instruction	Address Identifier	Description	Length in Words	312	31x, 147, 151, 154	317	319
JO	LABEL	Jump on stored overflow (OV = “1”)	1 [1]/2	3.8	1.9	0.51	0.06

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

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

Instruction	**Address Identifier**	**Description**	**Length in Words**	**Typical Execution Time in µs**			
				312	**31x, 147, 151, 154**	**317**	**319**
JOS	LABEL	Jump on stored overflow (OS = “1”)	2	3.8	1.9	0.51	0.06

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	3.8	1.9	0.51	0.06
JZ	LABEL	Jump if result=0 (CC 1=0 and CC 0=0)	1 [1]/2	3.8	1.9	0.51	0.06
JP	LABEL	Jump if result>0 (CC 1=1 and CC 0=0)	1 [1]/2	3.8	1.9	0.51	0.06
JM	LABEL	Jump if result<0 (CC 1=0 and CC 0=1)	1 [1]/2	3.8	1.9	0.51	0.06

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

Instruction	**Address Identifier**	**Description**	**Length in Words**	**Typical Execution Time in μs**			
				312	**31x, 147, 151, 154**	**317**	**319**
JN	LABEL	Jump if result ≠ 0 (CC 1=1 and CC 0=0) or (CC 1=0) and (CC 0=1)	1 [1]/2	3.8	1.9	0.51	0.06
JMZ	LABEL	Jump if result ≤ 0 (CC 1=0 and CC 0=1) or (CC 1=0 and CC 0=0)	2	3.8	1.9	0.51	0.06
JPZ	LABEL	Jump if result ≥ 0 (CC 1=1 and CC 0=0) or (CC 1=0) and (CC 0=0)	2	3.8	1.9	0.51	0.06

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

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	31x, 147, 151, 154	317	319
JL	LABEL	Jump distributor This instruction is followed by a list of jump instructions. The operand is a jump label to subsequent instructions in this list. ACCU1-L contains the number of the jump instruction to be executed.	2	5.0	2.5	0.78	0.04
LOOP	LABEL	Decrement ACCU1-L and jump if ACCU1-L≠0 (loop programming)	2	3.5	1.8	0.30	0.03

Status word for: **JL, LOOP**	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	$\overline{\text{FC}}$
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.

Instruction	Address Identifier	Description	Length in Words	Typical Execution Time in µs			
				312	**31x, 147, 151, 154**	**317**	**319**
MCR(		Open an MCR zone. Save the RLO to the MCR stack.	1	1.3	0.8	0.24	0.06
)MCR		Close an MCR-Zone. Pop an entry off the MCR-Stack.	1	1.3	0.8	0.24	0.06

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

MCRA		Activate the MCR	1	0.2	0.1	0.02	0.05
MCRD		Deactivate the MCR	1	0.2	0.1	0.02	0.03

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

Organisation Blocks (OB)

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

Organisation Blocks	**312**	**31x, 147, 151, 154**	**317**	**319**	**Starting Events (Hexadecimal Values)**	
Cycle:						
OB 1	x	x	x	x	1101_H 1103_H	OB1 starting event Running OB1 start event (conclusion of the free cycle)
Time-of-day interrupt:						
OB 10	x	x	x	x	1111_H	Time-of-day interrupt event
Delay Interrupt:						
OB 20	x	x	x	x	1121_H	Delay interrupt event
OB 21	–	–	x	x	1122_H	Delay interrupt event

Organisation Blocks	312	31x, 147, 151, 154	317	319	Starting Events (Hexadecimal Values)	
Cyclic interrupt:						
OB 32	–	–	x	x	1133_H	Cyclic interrupt event
OB 33	–	–	x	x	1134_H	Cyclic interrupt event
OB 34	–	–	x	x	1135_H	Cyclic interrupt event
OB 35 [1]	x	x	x	x	1136_H	Cyclic interrupt event
Process interrupt:						
OB 40	x	x	x	x	1141_H	Process interrupt
DPV1-Interrupt (only DP-CPUs)						
OB 55	–	x	x	x	1155_H	Status interrupt
OB 56	–	x	x	x	1156_H	Update-interrupt
OB 57	–	x	x	x	1157_H	Manufacture-specific interrupt
Synchronous cycle interrupt						
OB 61	–	X [2]	X [3]	x	1164_H	Synchronous cycle interrupt

[1] For CPU 319: in addition to the ms granular setting of the OB35 call interval, you can also select a µs granular setting in STEP 7 for the OB35. This makes it possible for you to also configure the shortest alarm cycle of 500µs and multiples thereof (value range of 500µs to 60000ms can be set).

[2] for CPU315-2 PN/DP with firmware as of V2.5 and IM154-8 CPU

[3] for all CPUs 317 with firmware as of V2.5

Organisa-tion Blocks	312	31x, 147, 151, 154	317	319	Starting Events (Hexadecimal Values)	
Technology synchronous interrupt (only Technology CPU)						
OB 65	–	only 315T	only 317T	-	$116A_H$	Technology synchronous interrupt
Error responses:						
OB 80	x	x	x	x	3501_H 3502_H 3505_H 3507_H	Cycle time violation OB or FB request error Time-of-day interrupt elapsed due to time jump Multiple OB request error caused start info buffer overflow

Organisation Blocks	312	31x, 147, 151, 154	317	319	Starting Events (Hexadecimal Values)	
Diagnostic interrupt:						
OB 82	x	x	x	x		3842_H Module o. k. 3942_H Module fault
OB 83	–	151-7 [1], 151-8 [3], 315 PN[2] IM 154 [3]	only 317 PN [2]	x [2]	3854_H	PROFINET IO-Submodule plugged in and is proportional to a parameteterized submodule
					3855_H	PROFINET IO-Submodule plugged in and is not proportional to a parameteterized submodule
					3861_H	Module is inserted
					3951_H	Pull out PROFINET IO-Module
					3961_H	Module is removed

[1] only for central IO
[2] only for PROFINET IO
[3] for central IO and PROFINET IO

Organisation Blocks	**312**	**31x, 147, 151, 154**	**317**	**319**	**Starting Events (Hexadecimal Values)**	
OB 85	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)
					$38B4_H$	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	31x, 147, 151, 154	317	319	Starting Events (Hexadecimal Values)	
OB 86	–	only DP, PN IO	only DP, PN IO	only DP, PN IO	$38C4_H$ $38CB_H$ $39C4_H$ $39CB_H$	Distributed I/O: station failed, outgoing PROFINET I/O: Station restart Distributed I/O: station failed, incoming PROFINET I/O: Station failure
OB 87	x	x	x	x	$35E1_H$ $35E2_H$ $35E6_H$	Incorrect frame identifier in GD $35E2_H$ GD packet status cannot be entered in DB GD whole status cannot be entered in DB
Restart:						
OB 100	x	x	x	x	1381_H 1382_H	Manual restart requests Automatic restart requests

Organisation Blocks	312	31x, 147, 151, 154	317	319	Starting Events (Hexadecimal Values)	
Synchronous error responses:						
OB 121	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
OB 122	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	31x, 147, 151-7, 315, 154	151-8	317, 319
Quantity [1]	1024	1024	2048
Admissible numbers	0 to 2047	0 to 2047	0 to 2047
Maximal size of an FB (process-relevant code)	16 kByte	64 kByte	64 kByte

Functions (FC)

Blocks	31x, 147, 151-7, 315, 154	151-8	317, 319
Quantity [1]	1024	1024	2048
Admissible numbers	0 to 2047	0 to 2047	0 to 2047
Maximal size of an FC (process-relevant code)	16 kByte	64 kByte	64 kByte

[1] Entire number FB, FC, DB: 1024
CPU 317: 2048
CPU 319: 4096

Data Blocks

Blocks	31x, (except 315), 147, 151-7	315, 154	151-8	317	319
Quantity [1]	511	1023	511	2047	4095
Admissible numbers	1 to 511	1 to 1023	1 to 511	1 to 2047	1 to 4095
Maximal size of an FB (process-relevant code)	16 kByte	16 kByte	64 kByte	64 kByte	64 kByte

[1] Entire number FB, FC, DB: 1024
CPU 317: 2048
CPU 319: 4096

Memory required by the SFBs for the integrated inputs and outputs

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

System Functions (SFC)

The following tables show the system functions offered by the

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

SFC No.	SFC Name	Description	Execution Time in μs			
			312	31x, 147, 151, 154	317	319
0	SET_CLK	Sets the clock time	235	195	44	3.0
1	READ_CLK	Reads the clock time	70	60	17	1.4
2	SET_RTM	Sets the operating hours counter	75	65	14	1.1
3	CTRL_RTM	Starts/stops the operating hours counter	70	60	12	1.0
4	READ_RTM	Reads the operating hours counter	105	90	16	1.3
5	GADR_LGC	Determine logical channel address	160	135	23	2.3
6	RD_SINFO	Reads start information of the current OB.	135	110	19	1.9
7	DP_PRAL [1)2)]	Triggers a process interrupt from the user program of the CPU as DP slave through to DP master.	–	90	19	9.0
		concurrent running requests, max.	–	34 requests together with SFB 75 requests		

[1)] only DP-CPUs
[2)] SFC 7 is not supported by the IM151-8

SFC No.	SFC Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
11	SYC_FR [1]	Synchronizes groups of DP slaves	–	300	63	16.0
		concurrent running requests, max.	–	2 requests		
12	D_ACT_DP [2]	Activates or deactivates DP slaves	–	410	90	13.0
		concurrent running requests, max.	–	4 requests[3]		8 requests
13	DPNRM_DG [1]	Reads the DP-compliant slave diagnosis (CPU31)	–	150	32	30.0
		concurrent running requests, max.	–	4 requests		
14	DPRD_DAT [1]	Reads/writes consistent data (n bytes)	–	150	30	25.0
15	DPWR_DAT [1]	Reads/writes consistent data (n bytes)	–	150	32	10.5

[1] only DP-CPUs

[2] only DP-CPUs and PROFINET-CPUs

[3] The IM151-8 as of V2.7 can handle 8 jobs simultaneously.

SFC No.	SFC Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
17	ALARM_SQ	Generates block-related messages that can be acknowledged	250	250	52	12.0
18	ALARM_S	Generates block-related messages that can not be acknowledged	250	250	50	9.0
19	ALARM_SC	Acknowledgment state of the last ALARM_SQ received message	110	110	23	8.0
20	BLKMOV	Copies variables within the working memory	90 µs + 2µs/ Byte	75 µs+1.6µs/ Byte	16 µs+0.05µs/ Byte	1.6µs+0.0015µs/ Byte
21	FILL	Sets array default variables within the working memory	90µs+2.6µs/ Byte	75 µs+2.2µs/ Byte	16 µs+0.08µs/ Byte	1.6µs+0.013µs/ Byte

SFC No.	SFC Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
22	CREAT_DB	Generates a data block	110µs+3.5µs/ DB in the specified areas	110µs+3.5µs/ DB in the specified areas	23.1µs+0.75µs/ DB in the specified areas	10.0
23	DEL_DB	Deletes a data block	402	402	80	13.0
		concurrent running requests, max.	21 requests			
24	TEST_DB	Tests a data block	130	110	18	2.1
28	SET_TINT	Sets the times of a time-of-day interrupt	190	160	40	2.5
29	CAN_TINT	Cancels a time-of-day interrupt	85	70	2	0.8
30	ACT_TINT	Activates a time-of-day interrupt	140	120	28	1.7
31	QRY_TINT	Queries the status of a time–of-day interrupt	90	75	12	1.3
32	SRT_DINT	Starts a delay interrupt	90	75	22	3.8
33	CAN_DINT	Cancels a delay interrupt	60	50	11	3.2

SFC No.	SFC Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
34	QRY_DINT	Queries started delay interrupts	85	71	13	1.4
36	MSK_FLT	Masks sync faults	132	110	17	1.8
37	DMSK_FLT	Enables sync faults	143	120	18	1.9
38	READ_ERR	Reads event status register	140	120	18	1.9
39	DIS_IRT	Disables the handling of new interrupts	180	155	64	3.5
40	EN_IRT	Enables the handling of new interrupt events	125	105	31	3.0
41	DIS_AIRT	Delays the handling of interrupts	50	45	9	1.0
42	EN_AIRT	Enables the handling of interrupts	55	45	9	1.0
43	RE_TRIGR	Re-triggers the scan time monitor	50	40	23	4.7
44	REPL_VAL	Copies a substitute value into accumulator 1	60	50	39	3.9
46	STP	Forces the CPU into the STOP mode	–			
47	WAIT	Delays program execution in addition to waiting times	250	250	198	193

SFC No.	SFC Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
49	LGC_GADR	Converts a free address to the slot and rack for a module	250	210	33	2.3
50	RD_LGADR	Reads all the declared free addresses for a module	500	420	59	3.7
51	RDSYSST	Reads out the information from the system state list. SFC 51 is not interruptible through interrupts.	250µs + 10µs / Byte	224µs + 10µs / Byte	44µs + 2 µs / Byte	3.6µs+0.013µs / Byte
		concurrent running requests, max.	4 requests			
52	WR_USMSG	Writes specific diagnostic information in the diagnostic buffer	280	235	66	3.0
55	WR_PARM	Writes dynamic parameters to a module	2000	1700	349	130
		concurrent running requests, max.	1 request			
56	WR_DPARM	Writes predefined dynamic parameters to a module	1750	1750	346	130
		concurrent running requests, max.	1 request			

SFC No.	SFC Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
57	PARM_MOD	Assigns a module’s parameters	<1650	<1400	<190	< 160
		concurrent running requests to different modules,	1 request			
58	WR_REC	Writes a module-specific data record	1400µs +32µs / Byte	1400µs+32µs / Byte	278µs + 6.5µs / Byte	180µs + 5.11µs / Byte
		concurrent running requests to different modules to different modules, max	4 requests together with SFB 53 requests		8 requests together with SFB 53 requests	
59	RD_REC	Reads a module-specific data record	500	500	275µs + 6.4µs / Byte	212µs + 6.25µs / Byte
		concurrent running requests to different modules, max.	4 requests together with SFB 52 requests		8 requests together with SFB 52 requests	
64	TIME_TICK	Reads out the system time	55	50	9	0.8

SFC No.	SFC Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
65	X_SEND[1]	Sends data to a communication partner external to your own S7 station	310	310	155	40.0
		The maximum number of simultaneous SFC65, SFC67, SFC68, SFC72 or SFC73 jobs to different remote communication partners (Note: only one SFC65, SFC67, SFC68, SFC72 or SFC73 job at a time is possible to a remote communication partner).	4 requests	[2]	30 requests	
66	X_RCV[1]	Receives data from a communication partner external to your own S7 station	120	120	24	9.0

[1] SFC 7 is not supported by the IM151-8

[2] CPU 313: 6 requests
CPU 314 and IM 151-7: 10 requests
CPU 315 and IM 154-8: 14 requests

SFC No.	SFC Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
67	X_GET[1]	Reads data from a communication partner external to your own S7 station	190	190	38	10.0
		The maximum number of simultaneous SFC65, SFC67, SFC68, SFC72 or SFC73 jobs to different remote communication partners (Note: only one SFC65, SFC67, SFC68, SFC72 or SFC73 job at a time is possible to a remote communication partner).	4 requests	[2]	30 requests	

[1] SFC 7 is not supported by the IM151-8

[2] CPU 313: 6 requests
CPU 314 and IM 151-7: 10 requests
CPU 315 and IM 154-8: 14 requests

SFC No.	SFC Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
68	X_PUT[1]	Writes data to a communication partner external to your own S7 station	190	190	38	10.0
		The maximum number of simultaneous SFC65, SFC67, SFC68, SFC72 or SFC73 jobs to different remote communication partners (Note: only one SFC65, SFC67, SFC68, SFC72 or SFC73 job at a time is possible to a remote communication partner).	4 requests	[2]	30 requests	
69	X_ABORT[1]	Aborts connection to a communication partner external to your own S7 station	100	100	20	5.0

[1] SFC 7 is not supported by the IM151-8

[2] CPU 313: 6 requests
CPU 314 and IM 151-7: 10 requests
CPU 315 and IM 154-8: 14 requests

SFC No.	SFC Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
70	GEO_LOG [1]	Determine module start address	135	100	17	8.0
71	LOG_GEO [1]	Querying the module slot belonging to a logical address	275	116	20	10.0
72	I_GET	Reads data from a communication partner within your own S7 station	190	190	38	10.0
		The maximum number of simultaneous SFC65, SFC67, SFC68, SFC72 or SFC73 jobs to different remote communication partners (Note: only one SFC65, SFC67, SFC68, SFC72 or SFC73 job at a time is possible to a remote communication partner).	4 requests	[2]	30 requests	

[1] only CPUs with firmware as of V 2.3.0

[2] CPU 313: 6 requests
CPU 314 and IM 151-7: 10 requests
CPU 315 and IM 154-8: 14 requests

SFC No.	SFC Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
73	I_PUT	Writes data to a communication partner within your own S7 station	190	190	38	10.0
		The maximum number of simultaneous SFC65, SFC67, SFC68, SFC72 or SFC73 jobs to different remote communication partners (Note: only one SFC65, SFC67, SFC68, SFC72 or SFC73 job at a time is possible to a remote communication partner).	4 requests	[1]	30 requests	
74	I_ABORT	Aborts connection to a communication partner within your own S7 station	100	100	20	5.0

[1] CPU 313: 6 requests
CPU 314 and IM 151-7: 10 requests
CPU 315 and IM 154-8: 14 requests

SFC No.	SFC Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
81	UBLKMOV	Copy the variable without interruption, length of the data to be copied up to 32 bytes	90µs+ 2µs / Byte	75µs + 2µs / Byte	16µs+0.05µs / Byte	1.6µs + 0.013µs / Byte
82	CREA_DBL	Create data block in load memory.	<1250	<1050	<320	<100
		concurrent running requests, max.	3 requests			
83	READ_DBL	Read from a data block in load memory	<1100	<950	<300	<300
		concurrent running requests, max.	3 requests			
84	WRIT_DBL	Write to a data block in load memory.	<1100	<900	<300	<300
		concurrent running requests, max.	3 requests			

SFC No.	SFC Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
101	RTM	Handling the Run-time meter	170	150	35	4.0
102	RD_DPARA	Read predefined parameter.	<1750	<1500	<320	<150
		concurrent running requests, max.	1 request			
103	DP_TOPOL	Detemine bus topology in a DP Master system fist call	–	250.0[1)][2)]	19.0[2)]	3.0
105	READ_SI[2)]	Read dynamically assigned system resources	2122.0 + 40.5 per alarm	2122.0 + 37.0per alarm	125.0 + 1.0 per alarm	30.0 + 0.2 per alarm
106	DEL_SI[2)]	Enable dynamically assigned system resources	2040.0 + 57.0 per alarm	2040.0 + 29.0 per alarm	246.0 + 2.6 per alarm	56.0 + 0.2 per alarm
107	ALARM_DQ[2)]	Acknowledgeable block-related messages create first call	354.0	354.0	33.0	9.0
108	ALARM_D[2)]	Not acknowledgeable block-related messages create first call	344.0	344.0	35.0	11.0
109	PROTECT[2)]	Activate write protection	45	45	7	3

[1)] only DP-CPUs

[2)] only CPUs with firmware as of V 2.5.0

SFC No.	SFC Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
112	PN_IN [1]	Update inputs of the PROFINET component user program interface	–	<20200	<20200	<6000
113	PN_OUT [1]	Update outputs of the PROFINET component interface	–	<21400	<21400	<6000
114	PN_DP [1]	Update DP interconnection	–	<4000	<4000	<5000

[1] only CPU 315-2 PN/DP / 317-2 PN/DP. / 319-3 PN/DP / IM 151-8 CPU / IM 154-8 CPU.
The runtimes of these blocks depend on their respective interconnection configuration. See also manual CPU 31xC and CPU 31x, technical data: chapter *cycle and response times, extending the OB1 cycle for cyclical PROFINET interconnections.*

SFC No.	SFC Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
126	SYNC_PI	Update the process image partition of the inputs in a synchronous cycle	–	230µs + 20µs/ Byte [1)2)]	80µs + 10µs/ Byte [2)]	7µs + 2µs / Byte
		concurrent running requests, max.	–	1 request[1)2)]	1 request	
127	SYNC_PO	Update the process image partition of the outputs in a synchronous cycle	–	230µs + 20µs/ Byte [1)2)]	80µs + 10µs/ Byte [2)]	7µs + 2µs/ Byte
		concurrent running requests, max.	–	1 request[1)2)]	1 request	

[1)] only CPU 315-2DP, 315-2 PN/DP, IM 154-8 CPU

[2)] availiable as of V 2.5

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	31x, 147, 151, 154	317	319
0	CTU	Counts up	101	90	19	3.0
1	CTD	Counts down	101	90	19	3.0
2	CTUD	Counts up and counts down	109	100	21	3.0
3	TP	Generates a pulse	135	115	26	3.0
4	TON	Delays a leading edge	120	101	20	3.0
5	TOF	Delays a falling edge	120	100	21	3.0
32	DRUM	Implements a sequence processor with a maximum of 16 s	90	80	16	3.0

SFB No.	SFB Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
SFBs for the integrated inputs/outputs (only CPU 31xC)						
41	CONT_C	Continuous control	–	3300	–	–
42	CONT_S	Step control	–	2800	–	–
43	PULSEGEN	Pulse generation	–	1500	–	–
44	ANALOG[1]	positioning with analog output	–			
		idle run		880	–	–
		start positioning run		2900	–	–
		request		1300	–	–
46	DIGITAL[1]	positioning with digital outputs	–			
		idle run		810	–	–
		start positioning run		2200	–	–
		request		1200	–	–
SFBs for the integrated inputs/outputs (only CPU 31xC)						
47	COUNT	counting	1222		–	–
48	FREQUENC	frequency measurement	1240		–	–
49	PULSE	pulse width modulation	1101		–	–

[1] only CPU 314C-2

SFB No.	SFB Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
52	RDREC	Read Data set from DP slave, PROFINET IO-Device or central module	500		272 µs + 6.4 µs / Bytes	214µs+6.25 µs / Byte
		concurrent running requests to different modules, max.	4 requests together with SFC 59 requests		8 requests together with SFC 59 requests	
53	WRREC	Write Data set to DP slave, PROFINET IO-Device or central module	1400 µs + 32 µs / Byte		248 µs+5.25 µs / Byte	181 µs+5.11 µs / Byte
		concurrent running requests to different modules, max.	4 requests together with SFC 58 requests		8 requests together with SFC 58 requests	
54	RALRM	Read out interrupt status information from interrupts of a DP slave, PROFINET IO-Device or of a central module in the respective OB	650		137	25.0
60	SEND_PTP[1]	send data (n characters) idle run operationalmode	–	 405 600+n*11 (1≤n≤1024)	–	–

[1] only CPU 31xC-2 PtP

SFB No.	SFB Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
61	RCV_PTP [1]	receive data (n characters)	–		–	–
		idle run		430		
		operationalmode		600+n*7 (1≤n≤1024)		
62	RES_RCVB [1]	clear input buffer	–		–	–
		idle run		390		
		operational mode		700		
63	SEND_RK [2]	send data (n characters, data exceeding a length of 128 characters are transferred in blocks with a maximum length of 128 characters)	–		–	–
		idle run		450		
		operational mode		1210+n*11 (1≤n≤128)		

[1] only CPU 31xC-2 PtP

[2] only CPU 314C-2PtP

SFB No.	SFB Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
64	FETCH_RK [1]	send data (n characters, data exceeding a length of 128 characters are transferred in blocks with a maximum length of 128 characters)	–		–	–
		idle run		620		
		operational mode		1680+n*7 (1≤n≤128)		
65	SERVE_RK [1]	receive/provide data (n characters, data exceeding a length of 128 characters are transferred in blocks with a maximum length of 128 characters)	–		–	–
		idle run		510		
		operational mode		1320+n*7 (1≤n≤128)		

[1] only CPU 31xC-2 PtP
[2] only CPU 314C-2 PtP

SFB No.	SFB Name	Description	Execution Time in µs			
			312	31x, 147, 151, 154	317	319
75	SALRM [1)2)]	Set desired interrupts of I-slaves	–	90	19	9.0
		concurrent running requests, max.	–	34 requests together with SFC 7 requests		
81	RD_DPAR	Reading predefined parameters	< 1500	< 1500	< 300	< 200
		concurrent running requests, max.	4 requests			

[1)] only DP-CPUs

[2)] SFC 7 is not supported by the IM151-8

Standard Function Blocks for S7-Communication via CP or Integrated PROFINET Interface

For some communication services, pre-fabricated blocks are available as an interface your STEP7 user program. See also STEP7 (as of version V5.3), Standard-Library, Communication Blocks.

FB No.	FB Name	Description	may be used with			
			31x, 315 (without PROFINET-Interface)	**147, 151-7**	**31x, 317, 319**	**151-8, 154**
8	USEND	Uncoordinated data sending	Communication via CP	–	Communication via CP or integrated PROFINET-Interface	Communication via integrated PROFINET-Interface
9	URCV	Uncoordinated data reception		–		
12	BSEND	Block-oriented data sending		–		
13	BRCV	Block-oriented data reception		–		
14	GET	Read data from a remote CPU		–		
15	PUT	Write data from a remote CPU		–		

FC No.	FC-Name	Description	may be used with		
			31x (without PROFINET-Interface)	147, 151	317, 319, 154
62	C_CNTRL	Request connection status which belongs to a local connection.	Communication via CP	–	Communication via CP or integrated PROFINET-Interface

See also STEP7, Standard-Library, Communication Blocks

Function blocks for open system interconnection over Industrial Ethernet

In order to be able to exchange data via user programs with other TCP/IP–capable communication partners, STEP7 places FBs and UDTs at your disposal. These blocks are saved in the Standard-Library,Communication Blocks.

FB-Nr.	FB-Name	Description	IM 151-8	IM 154-8	315 PN, 317 PN	319 PN	Communication protocol
63 1)2)	TSEND	Sending of data	with firmware as of V 2.7.0	with firmware as of V 2.5.0	with firmware as of V 2.3.0	with firmware as of V 2.4.0	TCP, ISO-on-TCP
64 1)2)	TRCV	Receiving of data	with firmware as of V 2.7.0	with firmware as of V 2.5.0	with firmware as of V 2.3.0	with firmware as of V 2.4.0	TCP, ISO-on-TCP
65 1)2)	TCON	Establishing a communication link	with firmware as of V 2.7.0	with firmware as of V 2.5.0	with firmware as of V 2.3.0	with firmware as of V 2.4.0	TCP, ISO-on-TCP, UDP
66 1)2)	TDI-SCON	Disconnecting a communication link	with firmware as of V 2.7.0	with firmware as of V 2.5.0	with firmware as of V 2.3.0	with firmware as of V 2.4.0	TCP, ISO-on-TCP, UDP
67 2)	TUSEND	Sending of data	with firmware as of V 2.7.0	with firmware as of V 2.5.0	with firmware as of V 2.5.0	with firmware as of V 2.4.0	UDP
68 2)	TURCV	Receiving of data	with firmware as of V 2.7.0	with firmware as of V 2.5.0	with firmware as of V 2.5.0	with firmware as of V 2.4.0	UDP

1) as of STEP 7, V5.3, SP1
You can find blocks for **UDP protocol** on the internet at: http://support.automation.siemens.com/ww/view/en/22146612

2) as of STEP 7, V5.4

IEC Functions

You can use the following functions in STEP 7:

These blocks are saved in the Standard Library, IEC Function-Blocks in STEP 7.

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

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

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

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

see also STEP 7 Online Help

System Status Sublist

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

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
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)
	Block types		
0015_H	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)
	State of module LEDs		–
0019_H	Status of each LED		
0074_H			
0174_H		0001_H	SF-LED
		0004_H	RUN-LED
		0005_H	STOP-LED
		0006_H	FRCE-LED
		$001B_H$	BF1-LED
		$001C_H$	BF2-LED
		0014_H	BF3-LED
		0015_H	MAINT-LED
$0F19_H$ $0F74_H$	Header information only		

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
001C$_H$	**All Records for the Component Identifications**	–	Station name Module name Module plant identification Copyright spezification Module serial number MMC serial number OEM identification
011C$_H$	**Component-Identification**	0001$_H$ 1) 0002$_H$ 1) 0003$_H$ 1) 0004$_H$ 1) 0005$_H$ 1) 0008$_H$ 1) 000A$_H$ 1)	Station name Module name Module plant identification Copyright spezification Module serial number MMC serial number OEM identification

[1] as Firmware V2.2.0

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
0132_H	**Communications status** on the communications type specified	0004_H	CPU protection level, position of the key switch, version identification of the user program and configuration
		0005_H	Diagnostic status data
		0006_H	PBK state parameter (only CPU 317-2 PN/DP)
		0008_H	Target system, correction factor, Run–time meter, Date/Time
		$000B_H$	Run–time meter (32 bits) 0 to 7
		$000C_H$	Run–time meter (32 bits) 8 to 15
0222_H	**Interrupt status** Record for the specified interrupt	OB number	–

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
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
	Status information of module racks		
0092_H	Expected status of the module rack in the central configuration	0000_H	Information about the status of the module rack in the central configuration
0292_H	Actual status of module rack in the central configuration		
0692_H	OK status of the expansion devices in the central configuration		

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
	Status information of module racks		
0094_H	Expected status of the module rack in the central configuration	0000_H	Information about the status of the module rack in the central configuration
0294_H	Actual status of module rack in the central configuration	0000_H	
0694_H	Faulty status of the rack in a central configuration	0000_H	
0794_H	Faulty and/or maintenance status of the rack in a central configuration	0000_H	
$0F94_H$	Header information only		

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
0591_H	**Module status information** of all submodules of the host module		
$0C91_H$	**Module status information** of a module in the central rack or connected to an integrated DP interface module	any logic address of a module	Features/parameters of the module plugged in
$0D91_H$	**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)
	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)
	Module status data in the CPU		
0591_H [1)	Module status information of all submodules		
$0A91_H$	Status information of all DP subsystems and DP masters		
$0C91_H$	Module status information of a module	any logic address of a module	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

1) only CPUs with firmware as of V 2.3.0

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
	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	0000_H	Information on the state of the mounting rack in the central configuration
0292_H	Actual status of racks in central configuration or of stations in a subnet	DP master system ID	Information of status of stations in subnet
0692_H	OK status of expansion racks in central configuration or of stations in a subnet		
	Station status in a DP subnet		Status of the devices in a DP subnet
0094_H [1)]	Expected status of the stations in a subnet	DP master system ID	
0294_H [1)]	Current status of the stations	DP master system ID	
0694_H [1)]	all faulty or non-existing stations	DP master system ID	
$0F94_H$ [1)]	only header information		

[1)] only CPUs with firmware as of V 2.3.0

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
	Module status information for PROFIBUS DP		
$0C96_H$ [1]	Module status information of a submodule	any logic address of a module/submodule	Status of the devices in a PROFIBUS subnet
$00B4_H$	**Module diagnostics** All standard diagnostic data of a station (only with DP master)	Module start address (Diagnostic address)	Module-dependent diagnostic information

[1] only CPUs with firmware as of V 2.3.0

S7 Communication Sublists and PROFINET Sublists

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
	Module status information in PROFINET IO		
0591_H	Module status information of all submodules		
$0A91_H$	Module status information of all PN I/O subsystems		
$0C91_H$	Module status information of a module	any logic address of a module/submodule [1)]	Module status data of inserted modules

[1)] When specifying logical output addresses the most significant bit (bit 15) must be set in the INDEX parameter (Example: Output address 10dez=>INDEX:=W#16#800A)

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
	Module status information		
$0D91_H$	in the specified station	Slot number of the PROFINET I/O Device Bit 15: is always = 1 Bit 11-14: PN I/O-Subsystem ID (Value range 100-115; where only 0 to 15 is specified) Bit 0-10: Station number of the PROFINET I/O Device	Module status information of all modules in the corresponding PROFINET I/O Device

SZL_ID	Sublist	Index (= ID of the individual records of the sublist)	Record Contents (Sublist Excerpt)
	Station status in PROFINET IO		Status of the PROFINET devices in a PROFINET subnet
0094_H	Expected status of the stations in a subnet	PN IO Subsystem number	
0294_H	Current status of the stations	PN IO Subsystem number	
0694_H	all faulty or non-existing stations	PN IO Subsystem number	
0794_H	Faulty and/or maintenance status of the stations	PN IO Subsystem number	
$0F94_H$	only header information		
	Module status information for PROFINET IO		Status of the PROFINET devices in a PROFINET-I/O subnet
0696_H	Module status information of all configured submodules of a module	any logic address of a module/submodule	
$0C96_H$	Module status information of a submodule	any logic address of a module/submodule	
$0xB3_H$	**Read diagnostic data record 1**		

Alphabetical Index of Instructions