

SIEMENS

SIMATIC

S7-400 Programmable Controller
CPU 412, 414, 416, 417

Instruction List

6ES7 498-8AA03-8BN0

Edition 10/2000

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

© Siemens AG 1999
Subject to change without prior notice

Siemens Aktiengesellschaft

6ES7498-8AA03-8BN0
Printed in the Federal Republic of Germany



Contents

Address Identifier and Parameter Ranges	2
Abbreviations and Mnemonics	6
Registers	8
Examples of Addressing	10
Execution Times with Indirect Addressing	14
List of Instructions	20
Bit Logic Instructions	20
Bit Logic Instructions with Parenthetical Expressions ..	26
ORing of AND Instructions	28
Logic Instructions with Timers and Counters	30
Word Logic Instructions with the Contents of Accumulator 1	34
Evaluating Conditions Using AND, OR and EXCLUSIVE OR	36
Edge-Triggered Instructions	40
Setting/Resetting Bit Addresses	42
Instructions Directly Affecting the RLO	46
Timer Instructions	48
Counter Instructions	52
Load Instructions	56
Load Instructions for Timers and Counters	66
Transfer Instructions	68
Load and Transfer Instructions for the Address Register	74
Load and Transfer Instructions for the Status Word	78
Load Instructions for DB Number and DB Length	80
Integer Math (16 Bits)	82
Integer Math (32 Bits)	84
Floating-Point Math (32 Bits)	86
Square Root and Square Instructions (32 Bits)	88
Logarithmic Function (32 Bits)	90
Trigonometrical Functions (32 Bits)	92
Adding Constants	94
Adding Using Address Registers	96
Comparison Instructions (16-Bit Integers)	98
Comparison Instructions (32-Bit Integers)	100
Comparison Instructions (32-Bit Real Numbers)	102
Shift Instructions	104
Rotate Instructions	106

Accumulator Transfer Instructions, Incrementing and Decrementing	108
Program Display and Null Operation Instructions	110
Data Type Conversion Instructions	112
Forming the Ones and Twos Complements	116
Block Call Instructions	118
Block End Instructions	122
Exchanging Shared Data Block and Instance Data Block	122
Jump Instructions	124
Instructions for the Master Control Relay (MCR)	132
Organization Blocks (OB)	134
Function Blocks (FB), Functions (FC) and Data Blocks	140
System Functions	142
System Function Blocks	178
Sublist of the System Status List (SSL)	186

Alphabetical Index of Instructions	190
---	------------

Applicability

This list of instructions applies to the CPUs listed below.

Name	Order number	subsequently described as*
CPU 412-1	6ES7 412-1XF03-0AB0	CPU 412
CPU 412-2	6ES7 412-2XG00-0AB0	
CPU 412-2 PCI	6ES7 612-2QH00-0AB4	
CPU 414-2	6ES7 414-2XG03-0AB0	CPU 414
CPU 414-3	6ES7 414-3XJ00-0AB0	
CPU 416-2	6ES7 416-2XK02-0AB0	CPU 416
CPU 416-2 PCI	6ES7 616-2QL00-0AB4	
CPU 416-3	6ES7 416-3XL00-0AB0	
CPU 417-4	6ES7 417-4XL00-0AB0	CPU 417
CPU 417-4 H	6ES7 417-4HL00-0AB0	

* except in the tables, where a detailed differentiation is necessary

Address Identifier and Parameter Ranges

Addr. ID	Parameter Range				Description
	CPU 412	CPU 414	CPU 416	CPU 417	
Q ²⁾	0.0 to 127.7	0.0 to 255.7	0.0 to 511.7	0.0 to 1023.7	Output (in PIQ)
QB ²⁾	0 to 127	0 to 255	0 to 511	0 to 1023	Output byte (in PIQ)
QW ²⁾	0 to 126	0 to 254	0 to 510	0 to 1022	Output word (in PIQ)
QD ²⁾	0 to 124	0 to 252	0 to 508	0 to 1020	Output double word (in PIQ)
DBX	0.0 to 65533.7 ¹⁾	0.0 to 65533.7	0.0 to 65533.7	0.0 to 65533.7	Data bit in data block
DB	1 to 511	1 to 4095	1 to 4095	1 to 8191	Data block
DBB	0 to 65533 ¹⁾	0 to 65533	0 to 65533	0 to 65533	Data byte in DB
DBW	0 to 65532 ¹⁾	0 to 65532	0 to 65532	0 to 65532	Data word in DB
DBD	0 to 65530 ¹⁾	0 to 65530	0 to 65530	0 to 65530	Data double word in DB
DIX	0.0 to 65533.7 ¹⁾	0.0 to 65533.7	0.0 to 65533.7	0.0 to 65533.7	Data bit in instance DB
DI	1 to 511	1 to 4095	1 to 4095	1 to 8191	Instance data block
DIB	0 to 65533 ¹⁾	0 to 65533	0 to 65533	0 to 65533	Data byte in instance DB

¹⁾ Also restricted by the size of the working memory.

²⁾ Default setting can be changed, see Technical Specifications

Addr. ID	Parameter Range				Description
	CPU 412	CPU 414	CPU 416	CPU 417	
DIW	0 to 65532 ¹⁾	0 to 65532	0 to 65532	0 to 65532	Data word in instance DB
DID	0 to 65530 ¹⁾	0 to 65530	0 to 65530	0 to 65530	Data double word instance DB
I ²⁾	0.0 to 127.7	0.0 to 255.7	0.0 to 511.7	0.0 to 1023.7	Input bit (in PII)
IB ²⁾	0 to 127	0 to 255	0 to 511	0 to 1023	Input byte (in PII)
IW ²⁾	0 to 126	0 to 254	0 to 510	0 to 1022	Input word (in PII)
ID ²⁾	0 to 124	0 to 252	0 to 508	0 to 1020	Input double word (in PII)
L ²⁾	0.0 to 4095.7	0.0 to 8191.7	0.0 to 16383.7	0.0 to 32767.0	Local data
LB ²⁾	0 to 4095	0 to 8191	0 to 16383	0 to 32767	Local data byte
LW ²⁾	0 to 4094	0 to 8190	0 to 16382	0 to 32766	Local data word
LD ²⁾	0 to 4092	0 to 8188	0 to 16380	0 to 32764	Local data double word

Addr. ID	Parameter Range				Description
	CPU 412	CPU 414	CPU 416	CPU 417	
M	0.0 to 4095.7	0.0 to 8191.7	0.0 to 16383.7	0.0 to 16383.7	Bit memory
MB	0 to 4095	0 to 8191	0 to 16383	0 to 16383	memory byte
MW	0 to 4094	0 to 8190	0 to 16382	0 to 16382	memory word
MD	0 to 4092	0 to 8188	0 to 16380	0 to 16380	memory double word
PQB	0 to 4095	0 to 8191	0 to 16383	0 to 16383	Peripheral output byte (direct I/O access)
PQW	0 to 4094	0 to 8190	0 to 16382	0 to 16382	Peripheral output word (direct I/O access)
PQD	0 to 4092	0 to 8188	0 to 16380	0 to 16380	Peripheral output double word (direct I/O access)
PIB	0 to 4095	0 to 8191	0 to 16383	0 to 16383	Peripheral input byte (direct I/O access)

Addr. ID	Parameter Range				Description
	CPU 412	CPU 414	CPU 416	CPU 417	
PIW	0 to 4094	0 to 8190	0 to 16382	0 to 16382	Peripheral input word (direct I/O access)
PID	0 to 4092	0 to 8188	0 to 16380	0 to 16380	Peripheral input double word (direct I/O access)
T	0 to 255	0 to 255	0 to 511	0 to 511	Timer
C	0 to 255	0 to 255	0 to 511	0 to 511	Counter

Constant	Range	Description
B(b1,b2) B(b1,b2,b3,b4)	–	Constant, 2 or 4 bytes
D# Date	–	IEC date constant
L# Integer	–	32-bit integer constant
P# Bit pointer	–	Pointer constant
S5T# Time	–	S7 time constant ¹⁾
T# Time	–	Time constant
TOD# Time	–	IEC time constant
C# Count	–	Counter constant (BCD code)
2#n	–	Binary constant
W#16# DW#16#	–	Hexadecimal constant

¹⁾ for loading of S7 timers

Abbreviations and Mnemonics

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

Abbrev.	Description	Example
k8	8-bit constant 0 to 255	32
k16	16-bit constant 256 to 32 767	28 131
k32	32-bit constant 32 768 to 999 999 999	127 624
i8	8-bit integer -128 to +127	-113
i16	16-bit integer -32768 to +32767	+6523
i32	32-bit integer -2 147 483 648 to +2 147 483 647	-2 222 222
m	Pointer constant	P#240.3
n	Binary constant	1001 1100
p	Hexadecimal constant	EA12
LABEL	Symbolic jump address (max. 4 characters)	DESTINATION
a	Byte address	
b	Bit address	
c	Address area	I, Q, M, L, DBX, DIX
d	Address in: MD, DBD, DID or LD	
e	Number in: MW, DBW, DIW or LW	
f	Timer/counter No.	

Abbrev.	Description	Example
g	Address area	IB, QB, PIB, PQB, MB, LB, DBB, DIB
h	Address area	IW, QW, PIW, PQW, MW, LW, DBW, DIW
i	Address area	ID, QD, PID, PQD, MD, LD, DBD, DID
q	Block No.	

Registers

ACCU1 to ACCU4 (32 Bits)

The accumulators are registers for processing bytes, words or double words. The address identifiers are loaded into the accumulators, where they are logically gated. The result of the logic operation (RLO) is in ACCU1 and can be transferred from there to a memory cell.

The accumulators are 32 bits long.

Accumulator designations :

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

Address Registers AR1 and AR2 (32 Bits)

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

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

- Area-internal pointer

00000000 00000bbb bbbbbbbb bbbbxxxx

- Area-crossing pointer

yyyyyyyy 00000bbb bbbbbbbb bbbbxxxx

Legend: b Byte address
x Bit number
y Area identifier (see page 10 "Examples of Addressing")

Status Word (16 Bits)

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

The status word is 16 bits long.

Bit	Assignment	Description
0	\overline{FC}	First check bit
1	RLO	Result of logic operation
2	STA	Status
3	OR	Or (AND before OR)
4	OS	Stored overflow
5	OV	Overflow
6	CC 0	Condition code 0
7	CC 1	Condition code 1
8	BR	Binary result
9 to 15	Unassigned	–

Examples of Addressing

Addressing Examples	Description
Direct 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#6#A0F0BCFD	Load hexadecimal constant into ACCU1
L 'ENDE'	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
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

Addressing Examples	Description
Indirect Addressing of Timers/Counters	
SP T [LW 8]	Start timer; the timer number is in local data word 8
CU C [LW 10]	Count upwards; 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 open DB as pointer
A I [DID 12]	AND operation: The address of the output is in data double word 12 of the open instance DB as pointer
A I [MD 12]	AND operation: The address of the output is in memory double word 12 as pointer
Area-Internal Register-Indirect Addressing	
A I [AR1,P#12.2]	AND operation: The address of the input is calculated as follows: "pointer value contained in address register 1 + pointer P#12.2"

Addressing Examples	Description																																				
Area-Crossing Register-Indirect Addressing																																					
For area-crossing register-indirect addressing, the address must also contain an area identifier. The address is in the address register. The area identifiers are as follows:																																					
<table border="1"> <thead> <tr> <th>Area identifier</th> <th>Coding (binary)</th> <th>hex.</th> <th>Area</th> </tr> </thead> <tbody> <tr> <td>P</td> <td>1000 0000</td> <td>80</td> <td>I/O area</td> </tr> <tr> <td>I</td> <td>1000 0001</td> <td>81</td> <td>Input area</td> </tr> <tr> <td>Q</td> <td>1000 0010</td> <td>82</td> <td>Output area</td> </tr> <tr> <td>M</td> <td>1000 0011</td> <td>83</td> <td>Bit memory area</td> </tr> <tr> <td>DB</td> <td>1000 0100</td> <td>84</td> <td>Data area</td> </tr> <tr> <td>DI</td> <td>1000 0101</td> <td>85</td> <td>Instance data area</td> </tr> <tr> <td>L</td> <td>1000 0110</td> <td>86</td> <td>Local data area</td> </tr> <tr> <td>VL</td> <td>1000 0111</td> <td>87</td> <td>Predecessor local data area (access to local data of invoking block)</td> </tr> </tbody> </table>	Area identifier	Coding (binary)	hex.	Area	P	1000 0000	80	I/O area	I	1000 0001	81	Input area	Q	1000 0010	82	Output area	M	1000 0011	83	Bit memory area	DB	1000 0100	84	Data area	DI	1000 0101	85	Instance data area	L	1000 0110	86	Local data area	VL	1000 0111	87	Predecessor local data area (access to local data of invoking block)	
Area identifier	Coding (binary)	hex.	Area																																		
P	1000 0000	80	I/O area																																		
I	1000 0001	81	Input area																																		
Q	1000 0010	82	Output area																																		
M	1000 0011	83	Bit memory area																																		
DB	1000 0100	84	Data area																																		
DI	1000 0101	85	Instance data area																																		
L	1000 0110	86	Local data area																																		
VL	1000 0111	87	Predecessor local data area (access to local data of invoking block)																																		
L B [AR1,P#8.0]	Load byte into ACCU1: The address is calculated from the "pointer value in AR 1 + P#8.0"																																				
A [AR1,P#32.3]	AND operation: The address of the operand is calculated from the "pointer value in AR 1 + P#32.3"																																				
Addressing Via Parameters																																					
A Parameter	Addressing via parameters																																				

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 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 over)

Execution Times with Indirect Addressing

When using indirect addresses statement consists of two parts:

Part 1: Load the address of the instruction

Part 2: Execute the instruction

In other words, when working with indirect addresses, you must calculate the execution time of an instruction 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
=	<u>Total execution time of the instruction</u>

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 following Table).

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			
	CPU 412	CPU 414	CPU 416	CPU 417
Bit memory area M				
Word	0.2	0.2	0.16	0.2
Double word	0.3	0.3	0.24	0.3
Data block DB/DX				
Word	0.2	0.2	0.16	0.2
Double word	0.3	0.3	0.24	0.3
Local data area L				
Word	0.2	0.2	0.16	0.2
Double word	0.3	0.3	0.24	0.3
AR1/AR2 (area-internal)	0.0 ¹⁾	0.0 ¹⁾	0.0 ¹⁾	0.0 ¹⁾
AR1/AR2 (area-crossing)	0.0 ¹⁾	0.0 ¹⁾	0.0 ¹⁾	0.0 ¹⁾
Parameter (word) ... for:				
• Timers	0.2	0.2	0.16	0.2
• Counters	0.2	0.2	0.16	0.2
• Block calls	0.2	0.2	0.16	0.2
Parameter (double word) ... for				
Bits, bytes, words and double words	0.3	0.3	0.24	0.3

¹⁾ Address registers AR1/AR2 do not need to be loaded in separate cycles for addressing.

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

Examples of Calculations

You will find a few examples here for calculating the execution times for the various methods of indirect addressing.

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

Example: A I [DBD 12] with CPU 414

Step 1: Load the contents of DBD 12 (time required is listed in the table on page 15)

Address is in ...	Execution Time in μs
Bit memory area M	
Word	0.2
Double word	0.3
Data block DB/DX	
Word	0.2
Double word	0.3

Step 2: AND the input addressed in this way (you will find the execution time in the tables in the chapter entitled "List of Instructions" on page 21)

Typical Execution Time in μs	
Direct Addressing	Indirect Addressing
0.2	Time for AI 0.1+
:	:

Total execution time:

0.3 ms
 + 0.1 ms
 = 0.4 ms

Execution Time for Area-Crossing Register-Indirect Addressing

Example: A [AR1, P#23.1] ... with I 1.0 in AR1 with CPU 416

Step 1: Load the contents of AR1, and increment them by the offset 23.1 (the time required is in the table on page 15)

Address is in ...	Execution Time in μs
:	:
AR1/AR2 (area-crossing)	0.00
:	:

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
0.08	Time for AI 0.08+
:	:

Total execution time:

0.00 ms
 + 0.08 ms
 = 0.08 ms

Execution Time for Addressing Via Parameters

Example: A Parameter ... with I 0.5 in the block parameter list with CPU 414

Step 1: Load input I 0.5 addressed via the parameter (the time required is in the table on page 15)

Address is in ...	Execution Time in μs
:	:
:	:
Parameter (double word)	0.3

Step 2: AND the input addressed in this way (you will find the execution time in the table in the chapter entitled "List of Instructions" on page 21)

Typical Execution Time in μs	
Direct Addressing	Indirect Addressing
0.2 :	Time for AI \rightarrow 0.1+ :

Total execution time:

	0.3 ms
+	0.1 ms
=	<u>0.4 ms</u>

List of Instructions

This chapter contains the complete list of instructions for the S7-400 CPUs. 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 11), you must add the time required for loading the address of the particular instruction to the execution times listed (see page 15).

Instr.	Address ID	Description	Length in Words	
A/AN	I/Q a.b	AND/AND NOT Input/output	1*/2	
	M a.b	Bit memory	1**/2	
	L a.b	Local data bit	2	
	DBX a.b	Data bit	2	
	DIX a.b	Instance data bit	2	
	c [d]	Memory-indirect, area-internal	2	
	c [AR1,m]	Register-ind., area-internal (AR1)	2	
	c [AR2,m]	Register-ind., area-internal (AR2)	2	
	[AR1,m]	Area-crossing (AR1)	2	
	[AR2,m]	Area-crossing (AR2)	2	
	Parameter	Via parameter	2	
	Status word for:	A/AN	BR	CC 1
	Instruction depends on:		–	–
Instruction affects:		–	–	

+ Plus time required for loading the address of the instruction (see page 15)

* With direct instruction addressing;
Address area 0 to 127

** With direct instruction addressing;
Address area 0 to 255

Bit Logic Instructions

All logic instructions generate a result (new RLO). The first instruction in a logic string generates the new RLO from the signal state scanned. The subsequent logic instructions generate the new RLO from the signal state scanned and the old RLO. The logic string ends with an instruction which limits the RLO (e.g. a memory instruction); that is, the /FC bit is set to zero.

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.2/0.3	0.1	0.08	0.1	0.08	0.1	0.1
0.2/0.3	0.1	0.08	0.1	0.08	0.1	0.1
0.3	0.1	0.08	0.1	0.08	0.1	0.1
0.3	0.1	0.08	0.1	0.08	0.1	0.1
0.3+	0.1+	0.08+	0.1+	0.08+	0.1+	0.1+
0.3+	0.1+	0.08+	0.1+	0.08+	0.1+	0.1+
0.3+	0.1+	0.08+	0.1+	0.08+	0.1+	0.1+
0.3+	0.1+	0.08+	0.1+	0.08+	0.1+	0.1+
0.3+	0.1+	0.08+	0.1+	0.08+	0.1+	0.1+
0.3+	0.1+	0.08+	0.1+	0.08+	0.1+	0.1+
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	Yes	–	Yes	Yes
–	–	–	Yes	Yes	Yes	1

Bit Logic Instructions, continued

Instr.	Address ID	Description	Length in Words
O/ON	I/Q a.b	Input/output	1*/2
	M a.b	Bit memory	1**/2
	L a.b	Local data bit	2
	DBX a.b	Data bit	2
	DIX a.b	Instance data bit	2
	c [d]	Memory-indirect, area-internal	2
	c [AR1,m]	Register-ind., area-internal (AR1)	2
	c [AR2,m]	Register-ind., area-internal (AR2)	2
	[AR1,m]	Area-crossing (AR1)	2
	[AR2,m]	Area-crossing (AR2)	2
	Parameter	Via parameter	2
	Status word for:	O, ON	BR
Instruction evaluates:		–	–
Instruction affects:		–	–

- + Plus time required for loading the address of the instruction
- * With direct instruction addressing
Address area 0 to 127
- ** With direct instruction addressing
Address area 0 to 255
(see page 15)

Execution Time in μ s						
CC0	OV	OS	OR	STA	RLO	\overline{FC}
0.2/0.3	0.1	0.08	0.1	0.08	0.1	0.1
0.2/0.3	0.1	0.08	0.1	0.08	0.1	0.1
0.3	0.1	0.08	0.1	0.08	0.1	0.1
0.3	0.1	0.08	0.1	0.08	0.1	0.1
0.3+	0.1+	0.08+	0.1+	0.08+	0.1+	0.1+
0.3+	0.1+	0.08+	0.1+	0.08+	0.1+	0.1+
0.3+	0.1+	0.08+	0.1+	0.08+	0.1+	0.1+
0.3+	0.1+	0.08+	0.1+	0.08+	0.1+	0.1+
0.3+	0.1+	0.08+	0.1+	0.08+	0.1+	0.1+
0.3+	0.1+	0.08+	0.1+	0.08+	0.1+	0.1+

Bit Logic Instructions, continued

Instr.	Address ID	Description	Length in Words
X/XN		Exclusive OR/ Exclusive OR NOT	
	I/Q a.b	Input/output	2
	M a.b	Bit memory	2
	L a.b	Local data bit	2
	DBX a.b	Data bit	2
	DIX a.b	Instance data bit	2
	c [d]	Memory-indirect, area-internal	2
	c [AR1,m]	Register-ind., area-internal (AR1)	2
	c [AR2,m]	Register-ind., area-internal (AR2)	2
	[AR1,m]	Area-crossing (AR1)	2
	[AR2,m]	Area-crossing (AR2)	2
	Parameter	Via parameter	2
Status word for: X, XN		BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	–

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
	0.3	0.1	0.08	0.1		
	0.3	0.1	0.08	0.1		
	0.3	0.1	0.08	0.1		
	0.3	0.1	0.08	0.1		
	0.3+	0.1+	0.08+	0.1+		
	0.3+	0.1+	0.08+	0.1+		
	0.3+	0.1+	0.08+	0.1+		
	0.3+	0.1+	0.08+	0.1+		
	0.3+	0.1+	0.08+	0.1+		
	0.3+	0.1+	0.08+	0.1+		
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	Yes	Yes
–	–	–	0	Yes	Yes	1

+ Plus time required for loading the address of the instruction (see page 15)

Bit Logic Instructions with Parenthetical Expressions

Saving the RLO and OR bits and the relevant function identifier (A, AN, ...) to the nesting stack. Seven nesting levels are possible per block. After the right parenthesis, the logic operation indicated by the

Instruction	Address ID	Description	Length in Words
A(AND left parenthesis	1
AN(AND NOT left parenthesis	1
O(OR left parenthesis	1
ON(OR NOT left parenthesis	1
X(Exclusive OR left parenthesis	1
XN(Exclusive OR NOT left parenthesis	1
Status word for:	A(, AN(, O(, ON(, X(, XN(BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	–
)		Right parenthesis, removing an entry from the nesting stack.	1
Status word for:)	BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	–

function identifier is performed on the saved RLO and the current RLO; the current OR is overwritten with the saved OR.

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	/FC
–	–	–	Yes	–	Yes	Yes
–	–	–	0	1	–	0
0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	Yes	–
–	–	–	Yes	1	Yes	1

ORing of AND Instructions

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

Instruction	Address ID	Description	Length in Words
O		ORing of AND operations according to the rule: AND before OR	1
Status word for:	O	BR	CC1
Instruction evaluates:	–	–	–
Instruction affects:	–	–	–

Execution Time in μs						
CPU 412		CPU 414		CPU 416		CPU 417
0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	Yes	Yes
–	–	–	Yes	1	–	Yes

Logic Instructions with Timers and Counters

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

Instruction	Address ID	Description	Length in Words
A/AN	T f	AND/AND NOT Timer	1 ¹⁾ /2
	T [e]	Timer, memory-indirect addressing	2
	C f	Counter	1 ¹⁾ /2
	C [e]	Counter, memory-indirect addressing	2
	Timer para. Counter para.	Timer/counter (addressing via parameter)	2
Status word for: A, AN		BR	CC1
Instruction evaluates:		-	-
Instruction affects:		-	-

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0,3	0,3+	0.1	0.1+	0.08	0.08+	0.1
0.1	0.1+	0.08	0.08+	0.1	0.1+	0.1
0.3+	0.3+	0.1+	0.1+	0.08+	0.08+	0.1+
0.1+	0.1+	0.08+	0.08+	0.1	0.1	0.1
0.3+	0.3+	0.1+	0.1+	0.08+	0.08+	0.1+
CC0	OV	OS	OR	STA	RLO	\overline{FC}
-	-	-	Yes	-	Yes	Yes
-	-	-	Yes	Yes	Yes	1

+ Plus time required for loading the address of the instruction (see page 15)

1) With direct instruction addressing
Address area 0 to 255

Logic Instructions with Timers and Counters, continued

Instruction	Address ID	Description	Length in Words
O/ON	T f	OR/OR NOT Timer	1 ¹⁾ /2
	T [e]	Timer, memory-indirect addr.	2
	C f	Counter	1 ¹⁾ /2
	C [e]	Counter, memory-indirect addressing	2
	Timer para. Counter para.	Timer/counter (addressing via parameter)	2
X/XN	T f	EXCLUSIVE OR/EXCLUSIVE OR NOT Timer	2
	T [e]	Timer, memory-indirect addr.	2
	C f	Counter	2
	C [e]	Counter, mem.-indirect addr.	2
	Timer para. Counter para.	EXCLUSIVE OR timer/counter (addressing via parameter)	2
Status word for:	O, ON, X, XN	BR	CC1
Instruction evaluates:		-	-
Instruction affects:		-	-

+ Plus time required for loading the address of the instruction (see page 15)

1) With direct instruction addressing
Address area 0 to 255

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.3	0.1	0.08	0.1	0.08	0.1	0.1
0.3+	0.1+	0.08+	0.1+	0.08	0.1	0.1+
0.3	0.1	0.08	0.1	0.08	0.1	0.1
0.3+	0.1+	0.08+	0.1+	0.08+	0.1+	0.1+
0.3+	0.1+	0.08+	0.1+	0.08+	0.1+	0.1+
0.3+	0.1+	0.08+	0.1+	0.08+	0.1+	0.1+
0.3+	0.1+	0.08+	0.1+	0.08+	0.1+	0.1+
0.3+	0.1+	0.08+	0.1+	0.08+	0.1+	0.1+
CC0	OV	OS	OR	STA	RLO	\overline{FC}
-	-	-	-	-	Yes	Yes
-	-	-	0	Yes	Yes	1

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

Instruction	Address ID	Description	Length in Words
AW		AND ACCU2-L	1
AW	W#16#p	AND 16-bit constant	2
OW		OR ACCU2-L	1
OW	W#16#p	OR 16-bit constant	2
XOW		EXCLUSIVE OR ACCU2-L	1
XOW	W#16#p	EXCLUSIVE OR 16-bit constant	2
AD		AND ACCU2	1
AD	DW#16#p	AND 32-bit constant	3
OD		OR ACCU2	1
OD	DW#16#p	OR 32-bit constant	3
XOD		EXCLUSIVE OR ACCU2	1
XOD	DW#16#p	EXCLUSIVE OR 32-bit constant	3
Status word for:	AW, OW, XOW, UD, OD, XOD	BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	Yes

double word is either specified in the instruction as an address or is in ACCU2. The result is in ACCU1 and/or ACCU1-L.

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.1	0.1	0.08	0.08	0.1	0.1	0.1
0.2	0.1	0.08	0.08	0.1	0.1	0.1
0.1	0.1	0.08	0.08	0.1	0.1	0.1
0.2	0.1	0.08	0.08	0.1	0.1	0.1
0.1	0.1	0.08	0.08	0.1	0.1	0.1
0.2	0.1	0.08	0.08	0.1	0.1	0.1
0.1	0.1	0.08	0.08	0.1	0.1	0.1
0.3	0.15	0.12	0.12	0.15	0.15	0.15
0.1	0.1	0.08	0.08	0.1	0.1	0.1
0.3	0.15	0.12	0.12	0.15	0.15	0.15
0.1	0.1	0.08	0.08	0.1	0.1	0.1
0.3	0.15	0.12	0.12	0.15	0.15	0.15
CC0	OV	OS	OR	STA	RLO	FC
–	–	–	–	–	–	–
0	0	–	–	–	–	–

Evaluating Conditions Using AND, OR and EXCLUSIVE OR

All logic instructions generate a result (new RLO). The first instruction in a logic string generates the new RLO from the signal state scanned. The subsequent logic instructions generate the new RLO

Instruction	Address ID	Description	Length in Words	
A/AN O/ON X/XN	==0	AND/AND NOT OR/OR-NOT EXCLUSIVE OR/ EXCLUSIVE-OR-NOT Result=0 (A1=0 and A0=0)	1	
	>0	Result>0 (CC1=1 and CC0=0)	1	
	<0	Result<0 (CC1=0 and CC0=1)	1	
	<>0	Result≠0 ((CC1=0 and CC0=1) or (CC1=1 and CC0=0))	1	
	<=0	Result<=0 ((CC1=0 and CC0=1) or (CC1=0 and CC0=0))	1	
	>=0	Result>=0 ((CC1=1 and CC0=0) or (CC1=0 and CC0=0))	1	
Status word for:	A/AN/O/ON/X/XN		BR	CC1
Instruction evaluates:			–	Yes
Instruction affects:			–	–

from the signal state scanned and the old RLO. The logic string ends with an instruction which limits the RLO (e.g. a memory instruction); that is, the \overline{FC} bit is set to zero.

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	\overline{FC}
Yes	–	–	Yes	–	Yes	Yes
–	–	–	Yes	Yes	Yes	1

Evaluating Conditions Using AND, OR and EXCLUSIVE OR, continued

Instruction	Address ID	Description	Length in Words
A/AN O/ON X/XN	UO	AND/AND-NOT OR/OR-NOT EXCLUSIVE-OR/ EXCLUSIVE-OR-NOT Unordered math instruction (CC1=1 and CC0=1)	1
	OS	AND OS=1	1
	BR	AND BR=1	1
	OV	AND OV=1	1
Status word for: A/AN/O/ON/X/XN		BR	CC1
Instruction evaluates:		Yes	Yes
Instruction affects:		–	–

Execution Time in μs						
CPU 412		CPU 414		CPU 416		CPU 417
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	FC
Yes	Yes	Yes	Yes	–	Yes	Yes
–	–	–	Yes	Yes	Yes	1

Edge-Triggered Instructions

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

Instruction	Address ID	Description	Length in Words	
FP/FN	I/Q a.b	The positive/negative edge is indicated by RLO = 1. The bit addressed in the instruction is the auxiliary edge bit memory.	2	
	M a.b		2	
	L a.b ¹⁾		2	
	DBX a.b		2	
	DIX a.b		2	
	c [d]		2	
	c [AR1,m]		2	
	c [AR2,m]		2	
	[AR1,m]		2	
	[AR2,m]		2	
	Parameter		2	
Status word for: FP, FN			BR	CC1
Instruction evaluates:			-	-
Instruction affects:			-	-

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.4		0.2		0.16		0.2
0.4		0.2		0.16		0.2
0.4		0.2		0.16		0.2
0.4		0.2		0.16		0.2
0.4		0.2		0.16		0.2
0.4+		0.2+		0.16+		0.2+
0.4+		0.2+		0.16+		0.2+
0.4+		0.2+		0.16+		0.2+
0.4+		0.2+		0.16+		0.2+
0.4+		0.2+		0.16+		0.2+
0.4+		0.2+		0.16+		0.2+
CC0	OV	OS	OR	STA	RLO	FC
-	-	-	-	-	Yes	-
-	-	-	0	Yes	Yes	1

- + Plus time required for loading the address of the instruction (see page 15)
- 1) Unnecessary if the bit being monitored is in the process image (local data of a block are only valid while the block is running).

Setting/Resetting Bit Addresses

Assigning the value "1" or "0" to the addressed instruction when RLO = 1. The instructions can be dependent on the MCR (see page 133).

Instruction	Address ID	Description	Length in Words
S R		Set addressed bit to "1"	
		Set addressed bit to "0"	
	I/Q	a.b Input/output	1 ¹⁾ /2
	M	a.b Bit memory	1 ²⁾ /2
	L	a.b Local data bit	2
	DBX	a.b Data bit	2
	DIX	a.b Instance data bit	2
	c [d]	Memory-indirect, area-internal	2
	c [AR1,m]	Register-indirect, area-internal (AR1)	2
	c [AR2,m]	Register-indirect, area-internal (AR2)	2
[AR1,m]	Area-crossing (AR1)	2	
[AR2,m]	Area-crossing (AR2)	2	
Parameter	Via parameter	2	
Status word for: S, R			BR CC1
Instruction evaluates:			- -
Instruction affects:			- -

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
	0.3/0.4		0.2		0.16	0.2
	0.3/0.4		0.2		0.16	0.2
	0.4		0.2		0.16	0.2
	0.4		0.2		0.16	0.2
	0.4		0.2		0.16	0.2
	0.4+		0.2+		0.16+	0.2+
	0.4+		0.2+		0.16+	0.2+
	0.4+		0.2+		0.16+	0.2+
	0.4+		0.2+		0.16+	0.2+
	0.4+		0.2+		0.16+	0.2+
CC0	OV	OS	OR	STA	RLO	FC
-	-	-	-	-	Yes	-
-	-	-	0	Yes	-	0

+ Plus time required for loading the address of the instruction (see page 15)

- 1) With direct instruction addressing
Address area 0 to 127
- 2) With direct instruction addressing
Address area 0 to 255

Setting/Resetting Bit Addresses, continued

The RLO is written to the address of the instruction. The instructions can be dependent on the MCR (see page 133).

In- struc- tion	Address ID	Description	Length in Words	
=	I/Q a.b	Assign RLO To input/output	1 ¹⁾ /2	
	M a.b	To bit memory	1 ²⁾ /2	
	L a.b	To local data bit	2	
	DBX a.b	To data bit	2	
	DIX a.b	To instance data bit	2	
	c [d]	Memory-indirect, area-internal	2	
	c [AR1,m]	Register-indirect, area-internal (AR1)	2	
	c [AR2,m]	Register-indirect, area-internal (AR2)	2	
	[AR1,m]	Area-crossing (AR1)	2	
	[AR2,m]	Area-crossing (AR2)	2	
Parameter	Via parameter	2		
Status word for: =			BR	CC1
Instruction evaluates:			-	-
Instruction affects:			-	-

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.3/0.4		0.2		0.16		0.2
0.3/0.4		0.2		0.16		0.2
0.4		0.2		0.16		0.2
0.4		0.2		0.16		0.2
0.4		0.2		0.16		0.2
0.4+		0.2+		0.16+		0.2+
0.4+		0.2+		0.16+		0.2+
0.4+		0.2+		0.16+		0.2+
0.4+		0.2+		0.16+		0.2+
0.4+		0.2+		0.16+		0.2+
CC0	OV	OS	OR	STA	RLO	FC
-	-	-	-	-	Yes	-
-	-	-	0	Yes	-	0

+ Plus time required for loading the address of the instruction
(see page 15)

- 1) With direct instruction addressing
Address area 0 to 127
- 2) With direct instruction addressing
Address area 0 to 255

Instructions Directly Affecting the RLO

The following instructions have a direct effect on the RLO.

Instruction	Address ID	Description	Length in Words	
CLR		Set RLO to "0"	1	
Status word for:	CLR	BR	CC1	
Instruction evaluates:		–	–	
Instruction affects:		–	–	

SET		Set RLO to "1"	1	
Status word for:	SET	BR	CC1	
Instruction evaluates:		–	–	
Instruction affects:		–	–	

NOT		Negate RLO	1	
Status word for:	NOT	BR	CC1	
Instruction evaluates:		–	–	
Instruction affects:		–	–	

SAVE		Save RLO to the BR bit	1	
Status word for:	SAVE	BR	CC1	
Instruction evaluates:		–	–	
Instruction affects:		Yes	–	

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	–	–
–	–	–	0	0	0	0

0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	–	–
–	–	–	0	1	1	0

0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	Yes	–	Yes	–
–	–	–	–	1	Yes	–

0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	Yes	–
–	–	–	–	–	–	–

Timer Instructions

Starting or resetting a timer. The time value must be in ACCU1-L. The instructions are triggered by an edge transition in the RLO; that is, when the status of the RLO has changed between two calls.

In-struction	Address ID	Description	Length in Words
SP	T f T [e]	Start timer as pulse on edge change from "0" to "1"	1 ¹⁾ /2
	Timer para.		2
SE	T f T [e]	Start timer as extended pulse on edge change from "0" to "1"	1 ¹⁾ /2
	Timer para.		2
SD	T f T [e]	Start timer as ON delay on edge change from "0" to "1"	1 ¹⁾ /2
	Timer para.		2
SS	T f T [e]	Start timer as retentive ON delay on edge change from "0" to "1"	1 ¹⁾ /2
	Timer para.		2
SF	T f T [e]	Start timer as OFF delay on edge change from "0" to "1"	1 ¹⁾ /2
	Timer para.		2
Status word for SP, SE, SD, SS, SF		BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	–

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.3/0.4 0.3+0.4+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.2 0.2+
0.4+	0.2+	0.16+	0.2+	0.16+	0.2+	0.2+
0.3/0.4 0.4+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.2 0.2+
0.4+	0.2+	0.16+	0.2+	0.16+	0.2+	0.2+
0.3/0.4 0.4+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.2 0.2+
0.4+	0.2+	0.16+	0.2+	0.16+	0.2+	0.2+
0.3/0.4 0.4+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.2 0.2+
0.4+	0.2+	0.16+	0.2+	0.16+	0.2+	0.2+
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	Yes	–
–	–	–	0	–	–	0

+ Plus time required for loading the address of the instruction (see page 15)

1) With direct instruction addressing
Timer No.: 0 to 255

Timer Instructions, continued

Instruction	Address ID	Description	Length in Words
FR	T f T [e]	Enable timer for restarting on edge change from "0" to "1" (reset edge bit memory for starting timer)	1 ¹⁾ /2
	Timer para.		2
R	T f T [e]	Reset timer	1 ¹⁾ /2
	Timer para.		2
Status word for: FR, R			BR
Instruction evaluates:			CC1
Instruction affects:			–

+ Plus time required for loading the address of the instruction
(see page 15)

1) With direct instruction addressing
Timer No.: 0 to 255

Execution Time in μ s		Execution Time in μ s		Execution Time in μ s		Execution Time in μ s	
CPU 412		CPU 414		CPU 416		CPU 417	
0.3/0.4 0.4+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.2 0.2+	0.2 0.2+
0.4+	0.2+	0.16+	0.2+	0.16+	0.2+	0.2+	0.2+
0.3/0.4 0.4+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.2 0.2+	0.2 0.2+
0.4+	0.2+	0.16+	0.2+	0.16+	0.2+	0.2+	0.2+
CC0	OV	OS	OR	STA	RLO	FC	
–	–	–	–	–	Yes	–	–
–	–	–	0	–	–	0	0

Counter Instructions

The count value must be in ACCU1-L in the form of a BCD number (0 - 999).

In-struction	Address ID	Description	Length in Words
S	C f C [e]	Presetting of counter on edge change from "0" to "1"	1 ¹⁾ /2
	Counter para.		2
R	C f C [e]	Reset counter to "0" when RLO = "1"	1 ¹⁾ /2
	Counter para.		2
CU	C f C [e]	Increment counter by 1 on edge change from "0" to "1"	1 ¹⁾ /2
	Counter para.		2
CD	C f C [e]	Decrement counter by 1 on edge change from "0" to "1"	1 ¹⁾ /2
	Counter para.		2
Status word for: S, R, CU, CD			BR CC1
Instruction evaluates:			- -
Instruction affects:			- -

+ Plus time required for loading the address of the instruction (see page 15)

1) With direct instruction addressing
Counter No.: 0 to 255

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.3/0.4 0.4+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.2 0.2+
0.4+	0.2+	0.16+	0.2+	0.16+	0.2+	0.2+
0.3/0.4 0.4+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.2 0.2+
0.4+	0.2+	0.16+	0.2+	0.16+	0.2+	0.2+
0.3/0.4 0.4+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.16 0.16+	0.2 0.2+	0.2 0.2+
0.4+	0.2+	0.16+	0.2+	0.16+	0.2+	0.2+
CC0	OV	OS	OR	STA	RLO	\overline{FC}
-	-	-	-	-	Yes	-
-	-	-	0	-	-	0

Counter Instructions, continued

In-struction	Address ID	Description	Length in Words
FR	C f C [e]	Enable counter on edge change from "0" to "1" (reset edge bit memory for up and down counting and setting the counter)	1 ¹⁾ /2
	Counter para.		2
Status word for: FR		BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	–

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.3/0.4 0.4+		0.2 0.2+		0.16 0.16+		0.2 0.2+
0.4+		0.2+		0.16+		0.2+
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	Yes	–
–	–	–	0	–	–	0

+ Plus time required for loading the address of the instruction (see page 15)

1) With direct instruction addressing
Counter No.: 0 to 255

Load Instructions

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

In-struction	Address ID	Description	Length in Words
L	IBa	Load ... Input byte	1 ¹⁾ /2
	QB a	Output byte	1 ¹⁾ /2
	PIB a	Peripheral input byte ²⁾	2
	MB a	Bit memory byte	1 ³⁾ /2
	LB a	Local data byte	2
	DBB a	Data byte	2
	DIB a	Instance data byte ... into ACCU1	2
	g [d]	Memory-indirect, area-internal	2
	g [AR1.m]	Register-indirect, area-internal (AR1)	2
	g [AR2,m]	Register-indirect, area-internal (AR2)	2
B[AR1.m]	Area-crossing (AR1)	2	
B[AR2,m]	Area-crossing (AR2)	2	
Parameter	Via parameter	2	

Execution Time in μ s			
CPU 412	CPU 414	CPU 416	CPU 417
0.2/0.3	0.1	0.08	0.1
0.2/0.3	0.1	0.08	0.1
0.3	0.1	0.08	0.1
0.2/0.3	0.1	0.08	0.1
0.3	0.1	0.08	0.1
0.3	0.1	0.08	0.1
0.3	0.1	0.08	0.1
0.3+	0.1+	0.08+	0.1+
0.3+	0.1+	0.08+	0.1+
0.3+	0.1+	0.08+	0.1+
0.3+	0.1+	0.08+	0.1+
0.3+	0.1+	0.08+	0.1+

+ Plus time required for loading the address of the instruction (see page 15)

- 1) With direct instruction addressing
Address area 0 to 127
- 2) Plus response time of the I/O module (> 1 μ s)
with CPU 417-4H: solo 34 μ s, redundant 64 μ s
- 3) With direct instruction addressing
Address area 0 to 255

Load Instructions, continued

With direct addressing, access to odd word addresses takes an additional 0.08 μ s for all CPU 416 operations and an additional 0.1 μ s for all CPU 414 operations.

In-struction	Address ID	Description	Length in Words
L	IW a	Load ... Input word	1 ¹⁾ /2
	QW	Output word	1 ¹⁾ /2
	PIW a	Peripheral input word ²⁾	2
	MW a	Bit memory word	1 ³⁾ /2
	LW a	Local data word	2
	DBW a	Data word	2
	DIW a	Instance data word ... into ACCU1-L	2
	h [d]	Memory-indirect, area-internal	2
	h [AR1.m]	Register-indirect, area-internal (AR1)	2
	h [AR2.m]	Register-indirect, area-internal (AR2)	2
W[AR1.m]	Area-crossing (AR1)	2	
W[AR2.m]	Area-crossing (AR2)	2	
Parameter	Via parameter	2	

Execution Time in μ s			
CPU 412	CPU 414	CPU 416	CPU 417
0.2/0.3	0.1	0.08	0.1
0.2/0.3	0.1	0.08	0.1
0.3	0.1	0.08	0.1
0.2/0.3	0.1	0.08	0.1
0.3	0.1	0.08	0.1
0.3	0.1	0.08	0.1
0.3	0.1	0.08	0.1
0.3+	0.1+	0.08+	0.1+
0.3+	0.1+	0.08+	0.1+
0.3+	0.1+	0.08+	0.1+
0.3+	0.1+	0.08+	0.1+
0.3+	0.1+	0.08+	0.1+

+ Plus time required for loading the address of the instruction (see page 15)

- 1) With direct instruction addressing
Address area 0 to 127
- 2) Plus response time of the I/O module (> 1 μ s)
with CPU 417-4H: solo 37 μ s, redundant 67 μ s
- 3) With direct instruction addressing
Address area 0 to 255

Load Instructions, continued

The execution times given in the table for direct addressing are extended by 0.1 μs (CPU 412), 0.1 μs (CPU 414), 0.8 μs (CPU 416) or 0.1 μs (CPU 417) when accessing odd-numbered addresses.

In-struction	Address ID	Description	Length in Words
L	IDa	Load ... Input double word	1 ¹⁾ /2
	QD a	Output double word	1 ¹⁾ /2
	PID a	Peripheral input double word ²⁾	2
	MD a	Bit memory double word	1 ³⁾ /2
	LD a	Local data double word	2
	DBD a	Data double word	2
	DID a	Instance data double word ... in ACCU1	2
	i [d]	Memory-indirect, area internal	2
	i [AR1,m]	Register-ind., area internal (AR1)	2
	i [AR2,m]	Register-ind., area internal (AR2)	2
	i [AR2,m]	Area-crossing (AR1)	2
	i [AR2,m]	Area-crossing (AR2)	2
	D[AR1,m]	Via parameter	2
	D[AR2,m]		2
Parameter		2	

Execution Time in μs			
CPU 412	CPU 414	CPU 416	CPU 417
0.3/04	0.2	0.16	0.2
0.3/04	0.2	0.16	0.2
0.3/04	0.2	0.16	0.2
0.3/04	0.2	0.16	0.2
0.4	0.2	0.16	0.2
0.4	0.2	0.16	0.2
0.4	0.2	0.16	0.2
0.4+	0.2+	0.16+	0.2+
0.4+	0.2+	0.16+	0.2+
0.4+	0.2+	0.16+	0.2+
0.4+	0.2+	0.16+	0.2+
0.4+	0.2+	0.16+	0.2+
0.4+	0.2+	0.16+	0.2+

+ Plus time required for loading the address of the instruction
(see page 15)

- 1) With direct instruction addressing
Address area 0 to 127
- 2) Plus response time of the I/O module (> 1 μs)
with CPU 417-4H: solo 41 μs , redundant 71 μs
- 3) With direct instruction addressing
Address area 0 to 255

Load Instructions, continued

In-struction	Address ID	Description	Length in Words
L	k8	Load ... 8-bit constant into ACCU1-LL	2
	k16	16-bit constant into ACCU1-L	2
	k32	32-bit constant into ACCU1	3
	Parameter	Load constant into ACCU1 (addressed via parameter)	2
L	2#n	Load 16-bit binary constant into ACCU1-L	2
		Load 32-bit binary constant into ACCU1	3
	B#16#p	Load 8-bit-hexadecimal constant into ACCU1-L	1
L	W#16#p	Load 16-bit hexadecimal constant into ACCU1-L	2
	DW#16#p	Load 32-bit hexadecimal constant into ACCU1	3

Execution Time in μs			
CPU 412	CPU 414	CPU 416	CPU 417
0.2	0.1	0.08	0.1
0.2	0.1	0.08	0.1
0.3	0.15	0.12	0.15
0.2/03+	0.1+	0.08+	0.1+
0.2	0.1	0.08	0.1
0.3	0.15	0.12	0.15
0.1	0.1	0.08	0.1
0.2	0.1	0.08	0.1
0.3	0.15	0.12	0.15

+ Plus time required for loading the address of the instruction (see page 15)

Load Instructions, continued

In-struction	Address ID	Description	Length in Words
L	'x'	Load 1 character	2
	'xx'	Load 2 characters	2
	'xxx'	Load 3 characters	3
	'xxxx'	Load 4 characters	3
L	D# time value	Load IEC date	3
L	S5T# time value	Load S7 time constant (16 bits)	2
L	TOD# time value	Load IEC time constant	3
L	T# time value	Load 16-bit time constant	2
		Load 32-bit time constant	3
L	C# count value	Load counter constant (BCD code)	2
L	B# (b1, b2)	Load constant as byte (b1, b2)	2
	B# (b1, b2, b3, b4)	Load constant as 4 bytes (b1, b2, b3, b4)	3
L	P# bit pointer	Load bit pointer	3
L	L# integer	Load 32-bit integer constant	3
L	Real number	Load floating-point number	3

Execution Time in μ s			
CPU 412	CPU 414	CPU 416	CPU 417
0.2	0.1	0.08	0.1
0.2	0.1	0.08	0.1
0.3	0.15	0.12	0.15
0.3	0.15	0.12	0.15
0.3	0.15	0.12	0.15
0.2	0.1	0.08	0.1
0.3	0.15	0.12	0.15
0.2	0.1	0.08	0.1
0.3	0.15	0.12	0.15
0.3	0.15	0.12	0.15
0.3	0.15	0.12	0.15
0.3	0.15	0.12	0.15

Load Instructions for Timers and Counters

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

In-struction	Address ID	Description	Length in Words
L	T f T (e)	Load time value	1 ¹⁾ /2 2
	Timer para.	Load time value (addressed via parameter)	2
L	C f C (e)	Load count value	1 ¹⁾ /2 2
	Counter para.	Load count value (addressed via parameter)	2
LC	T f T (e)	Load time value in BCD	1 ¹⁾ /2 2
	Timer para.	Load time value in BCD (addressed via parameter)	2
LC	C f C (e)	Load count value in BCD	1 ¹⁾ /2 2
	Counter para.	Load count value in BCD (addressed via parameter)	2

Execution Time in μ s			
CPU 412	CPU 414	CPU 416	CPU 417
0.2/0.3 0.3+	0.1 0.1+	0.08 0.08+	0.1 0.1+
0.3+	0.1+	0.08+	0.1+
0.2/0.3 0.3+	0.1 0.1+	0.08 0.08+	0.1 0.1+
0.3+	0.1+	0.08+	0.1+
0.3 0.3+	0.3 0.3+	0.24 0.24+	0.3 0.3+
0.3+	0.3+	0.24+	0.3+
0.3 0.3+	0.3 0.3+	0.24 0.24+	0.3 0.3+
0.3+	0.3+	0.24+	0.3+

+ Plus time required for loading the address of the instruction (see page 15)

1) With direct instruction addressing
Timer/counter No.: 0 to 255

Transfer Instructions

Transferring the contents of ACCU1 to the addressed operand. Note that some instructions are affected by the MCR (see page 133). The status word is not affected.

In-struction	Address ID	Description	Length in Words
T	IBa	Transfer contents of ACCU1-LL to ... input byte	1 ¹⁾ /2
	QB a	output byte	1 ¹⁾ /2
	PQB a	peripheral output byte ²⁾	2
	MB a	bit memory byte	1 ³⁾ /2
	LB a	local data byte	2
	DBB a	data byte	2
	DIB a	instance data byte	2
	g [d]	Memory-indirect, area internal	2
	g [AR1,m]	Register-ind., area internal (AR1)	2
	g [AR2,m]	Register-ind., area internal (AR2)	2
	B[AR1,m]	Area-crossing (AR1)	2
	B[AR2,m]	Area-crossing (AR2)	2
	Parameter	Via parameter	2

Execution Time in μs			
CPU 412	CPU 414	CPU 416	CPU 417
0.2/0.3	0.1	0.08	0.1
0.2/0.3	0.1	0.08	0.1
0.3	0.1	0.08	0.1
0.2/0.3	0.1	0.08	0.1
0.3	0.1	0.08	0.1
0.3	0.1	0.08	0.1
0.3	0.1	0.08	0.1
0.3+	0.1+	0.08+	0.1+
0.3+	0.1+	0.08+	0.1+
0.3+	0.1+	0.08+	0.1+
0.3+	0.1+	0.08+	0.1+
0.3+	0.1+	0.08+	0.1+
0.3+	0.1+	0.08+	0.1+

+ Plus time required for loading the address of the instruction (see page 15)

- 1) With direct instruction addressing
Address area 0 to 127
- 2) Peripheral acknowledgement time must be observed,
with CPU 417-4H: solo 29 μs , redundant 58 μs
- 3) With direct instruction addressing
Address area 0 to 255

Transfer Instructions, continued

With direct addressing, access to add word addresses requires an additional 0.1 µs for all CPU 417 operations, an additional 0.08 µs for

all CPU 416 operations, an additional 0.1 µs for all CPU 414 operations, an additional 0.1 µs for all CPU 412 operations.

Instruction	Address ID	Description	Length in Words
T	IW a	Transfer contents of ACCU1-L to ... input word	1 ¹⁾ /2
	QW a	output word	1 ¹⁾ /2
	PQW a	peripheral output word ²⁾	2
	MW a	bit memory word	1 ³⁾ /2
	LW a	local data word	2
	DBW a	data word	2
	DIW a	instance data word	2
	h [d]	Memory-indirect, area internal	2
	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

Execution Time in µs			
CPU 412	CPU 414	CPU 416	CPU 417
0.2/0.3	0.1	0.08	0.1
0.2/0.3	0.1	0.08	0.1
0.3	0.1	0.08	0.1
0.2/0.3	0.1	0.08	0.1
0.3	0.1	0.08	0.1
0.3	0.1	0.08	0.1
0.3	0.1	0.08	0.1
0.3+	0.1+	0.08+	0.1+
0.3+	0.1+	0.08+	0.1+
0.3+	0.1+	0.08+	0.1+
0.3+	0.1+	0.08+	0.1+
0.3+	0.1+	0.08+	0.1+

+ Plus time required for loading the address of the instruction (see page 15)

- 1) With direct instruction addressing
Address area 0 to 127
- 2) Peripheral acknowledgement time must be observed,
with CPU 417-4H: solo 32 µs, redundant 61 µs
- 3) With direct instruction addressing
Address area 0 to 255

Transfer Instructions, continued

The execution times given in the table for direct addressing are extended by 0.1 μ s (CPU 412), 0.1 μ s (CPU 414), 0.08 μ s (CPU 416), or 0.1 μ s (CPU 417) when accessing odd-numbered

addresses. Note that some instructions are affected by the MCR (see page 133).

Instruction	Address ID	Description	Length in Words
T		Transfer contents of ACCU1 to ...	
	IDa	Input double word	1 ¹ /2
	QD a	Output double word	1 ¹ /2
	PQD a	periph. output double word ²⁾	2
	MD a	Bit memory double word	1 ³ /2
	LD a	Local data double word	2
	DBD a	Data double word	2
	DID a	Instance data double word	2
T	i [d]	Memory-indirect, area internal	2
	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

Execution Time in μ s			
CPU 412	CPU 414	CPU 416	CPU 417
0.3/0.4	0.2	0.16	0.2
0.3/0.4	0.2	0.16	0.2
0.4	0.2	0.16	0.2
0.3/0.4	0.2	0.16	0.2
0.4	0.2	0.16	0.2
0.4	0.2	0.16	0.2
0.4	0.2	0.16	0.2
0.4+	0.2+	0.16+	0.2+
0.4+	0.2+	0.16+	0.2+
0.4+	0.2+	0.16+	0.2+
0.4+	0.2+	0.16+	0.2+
0.4+	0.2+	0.16+	0.2+
0.4+	0.2+	0.16+	0.2+

+ Plus time required for loading the address of the instruction (see page 15)

- 1) With direct instruction addressing
Address area 0 to 127
- 2) Peripheral acknowledgement time must be observed,
with CPU 417-4H: solo 36 μ s, redundant 65 μ s
- 3) With direct instruction addressing
Address area 0 to 255

Load and Transfer Instructions for Address Registers

Loading a double word from a memory area or register into address register 1 (AR1) or address register 2 (AR2). The status word is not affected.

Instruction	Address ID	Description	Length in Words
LAR1	–	Load contents from ... ACCU1	1
	AR2	Address register 2	1
	DBD a	Data double word	2
	DID a	Instance data double word	2
	m	32-bit constant as pointer	3
	LD a	Local data double word	2
	MD a	Bit memory double word ... into AR1	2
LAR2	–	Load contents from ... ACCU1	1
	DBD a	Data double word	2
	DID a	Instance data double word	2
	m	32-bit constant as pointer	3
	LD a	Local data double word	2
	MD a	Bit memory double word ... into AR2	2

Execution Time in μ s			
CPU 412	CPU 414	CPU 416	CPU 417
0.2	0.2	0.16	0.2
0.2	0.2	0.16	0.2
0.4	0.3	0.24	0.3
0.4	0.3	0.24	0.3
0.3	0.2	0.16	0.2
0.4	0.3	0.24	0.3
0.4	0.3	0.24	0.3
0.2	0.2	0.16	0.2
0.4	0.3	0.24	0.3
0.4	0.3	0.24	0.3
0.3	0.2	0.16	0.2
0.4	0.3	0.24	0.3
0.4	0.3	0.24	0.3

Load and Transfer Instructions for Address Registers, continued

Transferring a double word from address register 1 (AR1) or address register 2 (AR2) to a memory area or register. The contents of ACCU1 are first saved to ACCU2. The status word is not affected.

Instruction	Address ID	Description	Length in Words
TAR1	–	Transfer contents from AR1 in ... ACCU1	1
	AR2	Address register 2	1
	DBD a	Data double word	2
	DID a	Instance data double word	2
	LD a	Local data double word	2
	MD a	Bit memory double word	2
TAR2	–	Transfer contents from AR2 in ... ACCU1	1
	DBD a	Data double word	2
	DID a	Instance data double word	2
	LD a	Local data double word	2
	MD a	Bit memory double word	2
	CAR		Exchange the contents of AR1 and AR2

Execution Time in μ s			
CPU 412	CPU 414	CPU 416	CPU 417
0.1	0.1	0.08	0.1
0.2	0.2	0.16	0.2
0.4	0.2	0.16	0.2
0.4	0.2	0.16	0.2
0.4	0.2	0.16	0.2
0.4	0.2	0.16	0.2
0.1	0.1	0.08	0.1
0.4	0.2	0.16	0.2
0.4	0.2	0.16	0.2
0.4	0.2	0.16	0.2
0.4	0.2	0.16	0.2
0.4	0.4	0.32	0.4

Load and Transfer Instructions for the Status Word

Instruction	Address ID	Description	Length in Words	
L	STW	Load status word into ACCU1		
Status word for:		L STW	BR	CC1
Instruction evaluates:			Yes	Yes
Instruction affects:			–	–
T	STW	Transfer ACCU1 (bits 0 to 8) to the status word		
Status word for:		T STW	BR	CC1
Instruction evaluates:			–	–
Instruction affects:			Yes	Yes

Execution Time in μs						
CPU 412		CPU 414		CPU 416	CPU 417	
0.1		0.1		0.08	0.1	
CC0	OV	OS	OR	STA	RLO	\overline{FC}
Yes	Yes	Yes	Yes	Yes	Yes	Yes
–	–	–	–	–	–	–
0.1		0.1		0.08	0.1	
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	–	–
Yes	Yes	Yes	Yes	Yes	Yes	Yes

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 status word is not affected.

Instruction	Address ID	Description	Length in Words
L	DBNO	Load number of data block	1
L	DINO	Load number of instance data block	1
L	DBLG	Load length of data block into byte	1
L	DILG	Load length of instance data block into byte	1

Execution Time in μ s			
CPU 412	CPU 414	CPU 416	CPU 417
0.1	0.1	0.08	0.1
0.1	0.1	0.08	0.1
0.1	0.1	0.08	0.1
0.1	0.1	0.08	0.1

Integer Math (16 Bits)

Math instructions on two 16-bit words. The result is written to ACCU1 and/or ACCU1-L. ACCU3 and ACCU4 are then transferred to ACCU2 and ACCU3.

Instruction	Address ID	Description	Length in Words
+I		Add 2 integers (16 bits) (ACCU1-L)=(ACCU1-L)+ (ACCU2-L)	1
-I		Subtract 1 integer from another (16 bits) (ACCU1-L)=(ACCU2-L)- (ACCU1-L)	1
I		Multiply 1 integer by another (16 bits) (ACCU1)=(ACCU2-L) (ACCU1-L)	1
/I		Divide 1 integer by another (16 bits) (ACCU1-L)=(ACCU2-L): (ACCU1-L) The remainder is in ACCU1-H	1
Status word for: +I, -I,*I,/I			BR CC1
Instruction evaluates:			- -
Instruction affects:			- Yes

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.8		0.8		0.64		0.8
0.8		0.8		0.64		0.8
CC0	OV	OS	OR	STA	RLO	\overline{FC}
-	-	-	-	-	-	-
Yes	Yes	Yes	-	-	-	-

Integer Math (32 Bits)

Math instructions on two 32-bit words. The result is written to ACCU1. ACCU3 and ACCU4 are then transferred to ACCU2 and ACCU3.

Instruction	Address ID	Description	Length in Words
+D		Add 2 integers (32-bit) (ACCU1)=(ACCU2)+(ACCU1)	1
-D		Subtract 1 integer from another (32 bits) (ACCU1)=(ACCU2)-(ACCU1)	1
D		Multiply 1 integer by another (32 bits) (ACCU1)=(ACCU2)(ACCU1)	1
/D		Divide 1 integer by another (32 bits) (ACCU1)=(ACCU2):(ACCU1)	1
MOD		Divide 1 integer by another (32 bits) and load the remainder into ACCU1: (ACCU1)=remainder of [(ACCU2):(ACCU1)]	1
Status word for:	+D, -D,*D, /D, MOD	BR	CC1
Instruction evaluates:		-	-
Instruction affects:		-	Yes

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
1.3		1.3		1.04		1.3
1.3		1.3		1.04		1.3
1.3		1.3		1.04		1.3
CC0	OV	OS	OR	STA	RLO	\overline{FC}
-	-	-	-	-	-	-
Yes	Yes	Yes	-	-	-	-

Floating-Point Math (32 Bits)

The result of the math instruction is in ACCU1. ACCU3 and ACCU4 are then transferred to ACCU2 and ACCU3.

Instruction	Address ID	Description	Length in Words
+R		Add 2 real numbers (32 bits) (ACCU1)=(ACCU2)+(ACCU1)	1
-R		Subtract 1 real number from another (32 bits) (ACCU1)=(ACCU2)-(ACCU1)	1
R		Multiply 1 real number by another (32 bits) (ACCU1)=(ACCU2)(ACCU1)	1
/R		Divide 1 real number by another (32 bits) (ACCU1)=(ACCU2):(ACCU1)	1
Status word for: +R, -R, *R, /R			BR CC1
Instruction evaluates:			- -
Instruction affects:			- Yes

NEGR		Negate the real number in ACCU1	1
ABS		Form the absolute value of the real number in ACCU1	1
Status word for: NEGR, ABS			BR CC1
Instruction evaluates:			- -
Instruction affects:			- -

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.6		0.6		0.48		0.6
0.6		0.6		0.48		0.6
1.4		1.4		1.12		1.4
2.1		2.1		1.68		2.1
CC0	OV	OS	OR	STA	RLO	FC
-	-	-	-	-	-	-
Yes	Yes	Yes	-	-	-	-
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	FC
-	-	-	-	-	-	-
-	-	-	-	-	-	-

Square Root and Square Instructions (32 Bits)

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

Instruction	Address ID	Description	Length in Words
SQRT		Calculate the square root of a real number in ACCU1	1
SQR		Form the square of the real number in ACCU1	1
Status word for:		SQRT, SQR	BR CC1
Instruction evaluates:		–	–
Instruction affects:		–	Yes

Execution Time in μs						
CPU 412		CPU 414		CPU 416		CPU 417
72		40		37 - 39		40
1.4		1.4		1.12		1.4
CC0	OV	OS	OR	STA	RLO	FC
–	–	–	–	–	–	–
Yes	Yes	Yes	–	–	–	–

Logarithmic Function (32 Bits)

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

Instruction	Address ID	Description	Length in Words
LN		Form the natural logarithm of a real number in ACCU1	1
EXP		Calculate the exponential value of a real number in ACCU1 to the base e (= 2.71828)	1
Status word for:	LN, EXP	BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	Yes

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
63		35		33		35
63		35		32 - 34		35
CC0	OV	OS	OR	STA	RLO	FC
–	–	–	–	–	–	–
Yes	Yes	Yes	–	–	–	–

Trigonometrical Functions (32 Bits)

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

Instruction	Address ID	Description	Length in Words
SIN		Calculate the sine of a real number	1
ASIN		Calculate the arcsine of a real number	1
COS		Calculate the cosine of a real number	1
ACOS		Calculate the arccosine of a real number	1
TAN		Calculate the tangent of a real number	1
ATAN		Calculate the arctangent of a real number	1
Status word for:	SIN, ASIN, COS, ACOS, TAN, ATAN		BR CC1
Instruction evaluates:			– –
Instruction affects:			– Yes

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
56		31		30		31
117 - 133		65 - 74		62 - 70		65 - 74
58		32		30		32
122 - 139		68 - 77		65 - 72		68 - 77
58 - 63		32 - 35		30 - 33		32 - 35
43 - 58		24 - 32		23 - 30		24 - 32
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	–	–
Yes	Yes	Yes	–	–	–	–

Adding Constants

Adding integer constants and storing the result in ACCU1. The status word is not affected.

Instruction	Address ID	Description	Length in Words
+	i8	Add an 8-bit integer constant	1
+	i16	Add a 16-bit integer constant	2
+	i32	Add a 32-bit integer constant	3

Execution Time in μs			
CPU 412	CPU 414	CPU 416	CPU 417
0.1	0.1	0.08	0.1
0.2	0.1	0.08	0.1
0.3	0.15	0.12	0.15

Adding Using Address Registers

Adding a 16-bit integer to the contents of the address register. The value is either specified as an address in the instruction or is in ACCU1-L. The status word is not affected.

Instruction	Address ID	Description	Length in Words
+AR1		Add the contents of ACCU1-L to those of AR1	1
+AR1	m (0 to 4095)	Add a pointer constant to the contents of AR1	2
+AR2		Add the contents of ACCU1-L to those of AR2	1
+AR2	m (0 to 4095)	Add pointer constant to the contents of AR2	2

Execution Time in μ s			
CPU 412	CPU 414	CPU 416	CPU 417
0.2	0.2	0.16	0.2
0.2	0.2	0.16	0.2
0.2	0.2	0.16	0.2
0.2	0.2	0.16	0.2

Comparison Instructions (16-Bit Integers)

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

Instruction	Address ID	Description	Length in Words
==I		ACCU2-L=ACCU1-L	1
<>I		ACCU2-L≠ACCU1-L	1
<I		ACCU2-L<ACCU1-L	1
<=I		ACCU2-L<=ACCU1-L	1
>I		ACCU2-L>ACCU1-L	1
>=I		ACCU2-L>=ACCU1-L	1
Status word for:	==I, <>I, <I, <=I, >I, >=I	BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	Yes

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	FC
–	–	–	–	–	–	–
Yes	0	–	0	Yes	Yes	1

Comparison Instructions (32-Bit Integers)

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

Instruction	Address ID	Description	Length in Words
==D		ACCU2=ACCU1	1
<>D		ACCU2≠ACCU1	1
<D		ACCU2<ACCU1	1
<=D		ACCU2<=ACCU1	1
>D		ACCU2>ACCU1	1
>=D		ACCU2>=ACCU1	1
Status word for:	==D,< >D, <D, <=D, >D, >=D	BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	Yes

Execution Time in μ s						
CPU 412		CPU 414		CPU 416	CPU 417	
0.1		0.1		0.08	0.1	
0.1		0.1		0.08	0.1	
0.1		0.1		0.08	0.1	
0.1		0.1		0.08	0.1	
0.1		0.1		0.08	0.1	
0.1		0.1		0.08	0.1	
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	–	–
Yes	0	–	0	Yes	Yes	1

Comparison Instructions (32-Bit Real Numbers)

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

Instruction	Address ID	Description	Length in Words
==R		ACCU2=ACCU1	1
<>R		ACCU2≠ACCU1	1
<R		ACCU2<ACCU1	1
<=R		ACCU2<=ACCU1	1
>R		ACCU2>ACCU1	1
>=R		ACCU2>=ACCU1	1
Status word for:	==R, <>R, <R, <=R, >R, >=R	BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	Yes

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	–	–
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,

Instruction	Address ID	Description	Length in Words
SLW ¹⁾		Shift the contents of ACCU1-L to the left. Positions that become free are provided with zeros.	1
SLW	0 ... 15		
SLD		Shift the contents of ACCU1 to the left. Positions that become free are provided with zeros.	1
SLD	0 ... 32		
SRW ¹⁾		Shift the contents of ACCU1-L to the right. Positions that become free are provided with zeros.	1
SRW	0 ... 15		
SRD		Shift the contents of ACCU1 to the right. Positions that become free are provided with zeros.	1
SRD	0 ... 32		
SSI ¹⁾		Shift the contents of ACCU1-L with sign to the right. Positions that become free are provided with with the sign (bit 15).	1
SSI	0 ... 15		
SSD		Shift the contents of ACCU1 with sign to the right. Positions that become free are provided with with the sign (bit 31).	1
SSD	0 ... 32		
Status word for:	SLW, SLD, SRW, SRD, SSI, SSD	BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	Yes

¹⁾ No. of places shifted: 0 to 16

the contents of ACCU2-LL are used as the number of places. The last bit shifted is loaded into condition code bit CC 1.

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	–	–
0	0	–	–	–	–	–

Rotate Instructions

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

Instruction	Address ID	Description	Length in Words
RLD		Rotate the contents of ACCU1 to the left	1
RLD	0 ... 32		
RRD		Rotate the contents of ACCU1 to the right	1
RRD	0 ... 32		
Status word for:		RLD, RRD	BR CC1
Instruction evaluates:		–	–
Instruction affects:		–	Yes

RLDA		Rotate the contents of ACCU1 one bit position to the left through condition code bit CC 1	
RRDA		Rotate the contents of ACCU1 one bit position to the right through condition code bit CC 1	
Status word for:		RLDA, RRDA	BR CC1
Instruction evaluates:		–	–
Instruction affects:		–	Yes

ACCU2-LL are used as the number of places. The last bit shifted is loaded into condition code bit CC1.

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	–	–
Yes	Yes	–	–	–	–	–
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	–	–
0	0	–	–	–	–	–

Accumulator Transfer Instructions, Incrementing and Decrementing

The status word is not affected.

Instruction	Address ID	Description	Length in Words
CAW		Reverse the order of the bytes in ACCU1-L.	1
CAD		Reverse the order of the bytes in ACCU1.	1
TAK		Swap the contents of ACCU1 and ACCU2	1
ENT		The contents of ACCU2 and ACCU3 are transferred to ACCU3 and ACCU4.	1
LEAVE		The contents of ACCU3 and ACCU4 are transferred to ACCU2 and ACCU3.	1
PUSH		The contents of ACCU1, ACCU2 and ACCU3 are transferred to ACCU2, ACCU3 and ACCU4	1
POP		The contents of ACCU2, ACCU3 and ACCU4 are transferred to ACCU1, ACCU2 and ACCU3	1
INC	k8	Increment ACCU1-LL	1
DEC	k8	Decrement ACCU1-LL	1

Execution Time in μ s			
CPU 412	CPU 414	CPU 416	CPU 417
0.1	0.1	0.08	0.1
0.1	0.1	0.08	0.1
0.1	0.1	0.08	0.1
0.1	0.1	0.08	0.1
0.1	0.1	0.08	0.1
0.1	0.1	0.08	0.1
0.1	0.1	0.08	0.1
0.1	0.1	0.08	0.1
0.1	0.1	0.08	0.1

Program Display and Null Operation Instructions

The status word is not affected.

Instruction	Address ID	Description	Length in Words
BLD	k8	Program display instruction: Is treated by the CPU as a null operation instruction.	1
NOP	0 1	Null operation instruction	1

Execution Time in μ s			
CPU 412	CPU 414	CPU 416	CPU 417
0.1	0.1	0.08	0.1
0.1	0.1	0.08	0.1

Data Type Conversion Instructions

The results of the conversion are in ACCU1.

Instruction	Addr. ID	Description	Length in Words
BTI		Convert contents of ACCU1-L from BCD (0 to +/- 999) to integer (16 bits) (BCD To Int)	1
BTD		Convert contents of ACCU1 from BCD (0 to +/-9 999 999) to double integer (32 bits) (BCD To Doubleint)	1
DTR		Convert contents of ACCU1 from double integer (32 bits) to real number (32 bits) (Doubleint To Real)	1
ITD		Convert contents of ACCU1 from integer (16 bits) to double integer (32 bits) (Int To Doubleint)	1
Status word for:		BTI, BTD, DTR, ITD	BR CC1
Instruction evaluates:		–	–
Instruction affects:		–	–

ITB		Convert contents of ACCU1-L from integer (16 bits) to BCD from 0 to +/- 999 (Int To BCD)	1
DTB		Convert contents of ACCU1 from double integer (32 bits) to BCD from 0 to +/- 9 999 999 (Doubleint To BCD)	1
Status word for:		ITB, DTB	BR CC1
Instruction evaluates:		–	–
Instruction affects:		–	–

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
0.3		0.3		0.24		0.3
0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	FC
–	–	–	–	–	–	–
–	–	–	–	–	–	–

0.1		0.1		0.08		0.1
0.2		0.2		0.16		0.2
CC0	OV	OS	OR	STA	RLO	FC
–	–	–	–	–	–	–
–	Yes	Yes	–	–	–	–

Data Type Conversion Instructions, continued

The real number to be converted is in ACCU1.

Instruction	Address ID	Description	Length in Words
RND		Convert a real number into a 32-bit integer.	1
RND-		Convert a real number into a 32-bit integer. The number is rounded down to the next whole number.	1
RND+		Convert a real number into a 32-bit integer. The number is rounded up to the next whole number.	1
TRUNC		Convert a real number into a 32-bit integer. The places after the decimal point are truncated.	1
Status word for:	RND, RND-, RND+, TRUNC	BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	–

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.4		0.4		0.32		0.4
0.4		0.4		0.32		0.4
0.4		0.4		0.32		0.4
0.4		0.4		0.32		0.4
CC0	OV	OS	OR	STA	RLO	FC
–	–	–	–	–	–	–
–	Yes	Yes	–	–	–	–

Forming the Ones and Twos Complements

Instruction	Address ID	Description	Length in Words
INVI		Form the ones complement of ACCU1-L	1
INVD		Form the ones complement of ACCU1	1
Status word for: INVI, INVD		BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	–

NEGI		Form the twos complement of ACCU1-L (integer)	1
NEGD		Form the twos complement of ACCU1 (double integer)	1
Status word for: NEGI, NEGD		BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	Yes

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	–	–
–	–	–	–	–	–	–

0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	–	–
Yes	Yes	Yes	–	–	–	–

Block Call Instructions

The runtimes of the SFCs are specified in the chapter entitled "System Functions" as of page 142.

In-struction	Address ID	Description	Length in Words	
CALL	FB q, DB q	Unconditional call of an FB, with parameter transfer	1 ¹⁾ /2	
CALL	SFB q, DB q	Unconditional call of an SFB, with parameter transfer	2	
CALL	FC q	Unconditional call of a function, with parameter transfer	1 ¹⁾ /2	
CALL	SFC q	Unconditional call of an SFC, with parameter transfer	2	
UC	FB q	Unconditional call of blocks, without parameter transfer	1 ¹⁾ /2	
	FC q	Memory-indirect FB call	2	
	FB [e]	Memory-indirect FC call	2	
	FC [e]	FB/FC call via parameter	2	
	Parameter	FB/FC call via parameter	2	
CC	FB q	Conditional call of blocks, without parameter transfer	1 ¹⁾ /2	
	FC q	Memory-indirect FB call	2	
	FB [e]	Memory-indirect FC call	2	
	FC [e]	FB/FC call via parameter	2	
	Parameter	FB/FC call via parameter	2	
Status word for: CALL, UC, CC ²⁾			BR	CC1
Instruction evaluates:			–	–
Instruction affects:			–	–

+ Plus time required for loading the address of the instruction (see page 15)

- 1) With direct instruction (DB) addressing
Block No. 0 to 255
- 2) Depending on RLO, sets RLO = 1
- 3) Plus time required for supplying parameters
- 4) If call is not executed

The information on the status word only relates to the block call itself and not to the commands called in this block.

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
8.2 ³⁾		3.2 ³⁾		2.56 ³⁾		XX
8.2 ³⁾		3.2 ³⁾		2.56 ³⁾		XX
4.6 ³⁾		1.8 ³⁾		1.44 ³⁾		XX
4.6 ³⁾		1.8 ³⁾		1.44 ³⁾		XX
2.1/2.2		1.4		1.12		1.4
2.1/2.2		1.4		1.12		1.4
2.2+		1.4+		1.12+		1.4+
2.2+		1.4+		1.12+		1.4+
2.2+		1.4+		1.12+		1.4+
2.2+		1.4+		1.12+		1.4+
2.1/2.2/0.4 ⁴⁾		1.4/0.4 ⁴⁾		1.12/0.32 ⁴⁾		1.4/0.4 ⁴⁾
2.1/2.2/0.4 ⁴⁾		1.4/0.4 ⁴⁾		1.12/0.32 ⁴⁾		1.4/0.4 ⁴⁾
2.2+/0.4 ⁴⁾		1.4+/0.4 ⁴⁾		1.12+/0.32 ⁴⁾		1.4+/0.4 ⁴⁾
2.2+/0.4 ⁴⁾		1.4+/0.4 ⁴⁾		1.12+/0.32 ⁴⁾		1.4+/0.4 ⁴⁾
2.2+/0.4 ⁴⁾		1.4+/0.4 ⁴⁾		1.12+/0.32 ⁴⁾		1.4+/0.4 ⁴⁾
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	–	–
–	–	0	0	1	–	0

Block Call Instructions, continued

Instruction	Address ID	Description	Length in Words
OPN	DB q DI q DB [e] DI [e] Parameter	Open: Data block Instance data block Data block, memory-indirect Instance DB, memory-indirect Data block using parameters	1 ¹⁾ /2
Status word for: OPN		BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	–

Execution Time in μ s Direct Addressing						
CPU 412		CPU 414		CPU 416		CPU 417
0.6/0.7		0.3		0.24		0.3
0.7		0.3		0.24		0.3
0.7+		0.3+		0.24+		0.3+
0.7+		0.3+		0.24+		0.3+
0.7+		0.3+		0.24+		0.3+
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	–	–
–	–	–	–	–	–	–

+ Plus time required for loading the address of the instruction
(see page 15)

1) With direct instruction (DB) addressing
Block No. 0 to 255

Block End Instructions

Instruction	Address ID	Description	Length in Words
BE		End block	1
BEU		End block unconditionally	1
Status word for: BE, BEU		BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	–

BEC		End block conditionally if RLO = "1"	
Status word for: BEC		BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	–

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 status word is not affected.

Instruction	Address ID	Description	Length in Words
CDB		Exchange shared data block and instance data block	1

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
2.8		2.0		1.60		2.0
2.8		2.0		1.60		2.0
CC0	OV	OS	OR	STA	RLO	FC
–	–	–	–	–	–	–
–	–	0	0	1	–	0

3.0 0.4 ¹⁾		2.2 0.4 ¹⁾		1.76 0.32 ¹⁾		2.2 0.4 ¹⁾
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	Yes	–
–	–	Yes	0	1	1	0

Execution Time in μ s			
CPU 412	CPU 414	CPU 416	CPU 417
0.2	0.2	0.16	

1) If jump is not executed

Jump Instructions

Jumping as a function of conditions.

Instruction	Address ID	Description	Length in Words
JU	LABEL	Jump unconditionally	1 ¹⁾ /2
Status word for: JU		BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	–

JC	LABEL	Jump if RLO = "1"	1 ¹⁾ /2
JCN	LABEL	Jump if RLO = "0"	2
Status word for: JC, JCN		BR	CC1
Instruction evaluates:		–	–
Instruction affects:		–	–

JCB	LABEL	Jump if RLO = "1". Save the RLO in the BR bit	2
JNB	LABEL	Jump if RLO = "0". Save the RLO in the BR bit	2
Status word for: JCB, JNB		BR	CC1
Instruction evaluates:		–	–
Instruction affects:		Yes	–

1) 1 word long for jump widths between -128 and +127

2) If jump is not executed

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.5/0.6		0.5		0.4		0.5
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	–	–
–	–	–	–	–	–	–

0.5/0.6 ²⁾		0.5/0.2 ²⁾		0.4/0.16 ²⁾		0.5/0.2 ²⁾
0.6/0.2 ²⁾		0.5/0.2 ²⁾		0.4/0.16 ²⁾		0.5/0.2 ²⁾
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	Yes	–
–	–	–	0	1	1	0

0.6/0.2 ²⁾		0.5/0.2 ²⁾		0.4/0.16 ²⁾		0.5/0.2 ²⁾
0.6/0.2 ²⁾		0.5/0.2 ²⁾		0.4/0.16 ²⁾		0.5/0.2 ²⁾
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	Yes	–
–	–	–	0	1	1	0

Jump Instructions, continued

Instruction	Address ID	Description	Length in Words	
JBI	LABEL	Jump if BR = "1"	2	
JNBI	LABEL	Jump if BR = "0"	2	
Status word for: JBI, JNBI			BR	CC1
Instruction evaluates:			Yes	–
Instruction affects:			–	–
JO	LABEL	Jump on stored overflow (OV = "1")	1 ¹⁾ /2	
Status word for: JO			BR	CC1
Instruction evaluates:			–	–
Instruction affects:			–	–
JOS	LABEL	Jump on stored overflow (OS = "1")	2	
Status word for: JOS			BR	CC1
Instruction evaluates:			–	–
Instruction affects:			–	–

1) 1 word long for jump widths between -128 and +127

2) If jump is not executed

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.6/0.2 ²⁾		0.5/0.2 ²⁾		0.4/0.16 ²⁾		0.5/0.2 ²⁾
0.6/0.2 ²⁾		0.5/0.2 ²⁾		0.4/0.16 ²⁾		0.5/0.2 ²⁾
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	–	–
–	–	–	0	1	–	0
0.5/0.6/0.2 ²⁾		0.5/0.2 ²⁾		0.4/0.16 ²⁾		0.5/0.2 ²⁾
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	Yes	–	–	–	–	–
–	–	–	–	–	–	–
0.6/0.2 ²⁾		0.5/0.2 ²⁾		0.4/0.16 ²⁾		0.5/0.2 ²⁾
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	Yes	–	–	–	–
–	–	0	–	–	–	–

Jump Instructions, continued

Instruction	Address ID	Description	Length in Words
JUO	LABEL	Jump if "unordered math instruction" (CC1=1 and CC0=1)	2
JZ	LABEL	Jump if result = 0 (CC1=0 and CC0=0)	1 ¹⁾ /2
JP	LABEL	Jump if result > 0 (CC1=1 and CC0=0)	1 ¹⁾ /2
JM	LABEL	Jump if result < 0 (CC1=0 and CC0=1)	1 ¹⁾ /2
JN	LABEL	Jump if result ≠ 0 (CC1=1 and CC0=0) or (CC1=0 and CC0=1)	1 ¹⁾ /2
JMZ	LABEL	Jump if result ≤ 0 (CC1=0 and CC0=1) or (CC1=0 and CC0=0)	2
JPZ	LABEL	Jump if result ≥ 0 (CC1=1 and CC0=0) or (CC1=0 and CC0=0)	2
Status word for:	JUO, JZ, JP, JM, JN, JMZ, JPZ		BR CC1
Instruction evaluates:	-		Yes
Instruction affects:	-		-

Execution Time in μs						
CPU 412		CPU 414		CPU 416		CPU 417
0.6/0.2 ²⁾		0.5/0.2 ²⁾		0.4/0.16 ²⁾		0.5/0.2 ²⁾
0.5/0.6/0.2 ²⁾		0.5/0.2 ²⁾		0.4/0.16 ²⁾		0.5/0.2 ²⁾
0.5/0.6/0.2 ²⁾		0.5/0.2 ²⁾		0.4/0.16 ²⁾		0.5/0.2 ²⁾
0.5/0.6/0.2 ²⁾		0.5/0.2 ²⁾		0.4/0.16 ²⁾		0.5/0.2 ²⁾
0.5/0.6/0.2 ²⁾		0.5/0.2 ²⁾		0.4/0.16 ²⁾		0.5/0.2 ²⁾
0.6/0.2 ²⁾		0.5/0.2 ²⁾		0.4/0.16 ²⁾		0.5/0.2 ²⁾
0.6/0.2 ²⁾		0.5/0.2 ²⁾		0.4/0.16 ²⁾		0.5/0.2 ²⁾
CC0	OV	OS	OR	STA	RLO	\overline{FC}
Yes	-	-	-	-	-	-
-	-	-	-	-	-	-

1) 1 word long for jump widths between -128 and +127

2) If jump is not executed

Jump Instructions, continued

Instruction	Address ID	Description	Length in Words
JL	LABEL	Jump distributor This instruction is followed by a list of jump instructions. The address identifier is a jump label to subsequent instructions in this list. ACCU1-LL contains the number of the jump instruction to be executed (max. 254). The number of the first jump instruction is 0.	2
LOOP	LABEL	Decrement ACCU1-L and jump if ACCU1-L \neq 0 (loop programming)	2
Status word for:		JL, LOOP	BR CC1
Instruction evaluates:		–	–
Instruction affects:		–	–

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.8		0.7		0.56		0.7
0.6/0.2 ¹⁾		0.5/0.2 ¹⁾		0.4/0.08 ¹⁾		0.5/0.2 ¹⁾
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	–	–
–	–	–	–	–	–	–

¹⁾ If jump is not executed

Instructions for the Master Control Relay (MCR)

MCR=1→MCR is deactivated

MCR=0→MCR is activated; "T" and "=" instructions write zeros to the

Instruction	Address ID	Description	Length in Words	
MCR(Open an MCR zone. Save the RLO to the MCR stack.	1	
Status word for:	MCR(CC1	BR	
Instruction evaluates:		–	–	
Instruction affects:		–	–	

)MCR		Close an MCR zone. Pop an entry off the MCR stack.	1	
Status word for:)MCR	BR	CC1	
Instruction evaluates:		–	–	
Instruction affects:		–	–	

MCRA		Activate the MCR	1	
MCRD		Deactivate the MCR	1	
Status word for:	MCRA, MCRD	BR	CC1	
Instruction evaluates:		–	–	
Instruction affects:		–	–	

corresponding address identifiers if RLO = "0"; "S" and "R" instructions leave the memory contents unchanged.

Execution Time in μ s						
CPU 412		CPU 414		CPU 416		CPU 417
0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	FC
–	–	–	–	–	Yes	–
–	–	–	0	1	–	0

0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	\overline{FC}
–	–	–	–	–	–	–
–	–	–	0	1	–	0

0.1		0.1		0.08		0.1
0.1		0.1		0.08		0.1
CC0	OV	OS	OR	STA	RLO	FC
–	–	–	–	–	–	–
–	–	–	–	–	–	–

Organization Blocks (OB)

A user program for the S7-400 is made up of blocks containing the statements, parameters and data for the relevant CPU. The number of blocks you can create or which are provided by the operating

Organization Blocks	CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H
Free cycle					
OB 1	x	x	x	x	x
Time-of-day interrupts					
OB 10	x	x	x	x	x
OB 11	x	x	x	x	x
OB 12		x	x	x	x
OB 13		x	x	x	x
OB 14			x	x	x
OB 15			x	x	x
OB 16			x	x	x
OB 17			x	x	x
Time-delay interrupts					
OB 20	x	x	x	x	x
OB 21	x	x	x	x	x
OB 22		x	x	x	x
OB 23		x	x	x	x

system is different for each of the S7-400 CPUs. You will find a detailed description of the OBs and their use in the *STEP 7 Programming Manual*.

Start Events (Hexadecimal Values)
1101, 1102, 1103, 1104
1111
1112
1113
1114
1115
1116
1117
1118
1121
1122
1123
1124

Organization Blocks (OB), continued

Organization Blocks	CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H
Timed interrupts					
OB 30			x	x	x
OB 31			x	x	x
OB 32	x	x	x	x	x
OB 33		x	x	x	x
OB 34		x	x	x	x
OB 35	x	x	x	x	x
OB 36			x	x	x
OB 37			x	x	x
OB 38			x	x	x
Hardware interrupts					
OB 40	x	x	x	x	x
OB 41	x	x	x	x	x
OB 42		x	x	x	x
OB 43		x	x	x	x
OB 44			x	x	x
OB 45			x	x	x
OB 46			x	x	x
OB 47			x	x	x
Multicomputing interrupts					
OB 60	x	x	x	x	

Start Events (Hexadecimal Values)
1131
1132
1133
1134
1135
1136
1137
1138
1139
1141, 1142, 1143, 1144
1141, 1142, 1143, 1144
1141, 1142, 1143, 1144
1141, 1142, 1143, 1144
1141, 1142, 1143, 1144
1141, 1142, 1143, 1144
1141, 1142, 1143, 1144
1141, 1142, 1143, 1144
1161, 1162

Organization Blocks (OB), continued

Organization Blocks	CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H
Redundancy error interrupts:					
OB 70					x
OB 72					x
Asynchronous error interrupts:					
OB 80	x	x	x	x	x
OB 81	x	x	x	x	x
OB 82	x	x	x	x	x
OB 83	x	x	x	x	x
OB 85	x	x	x	x	
OB 86	x	x	x	x	x
OB 87	x	x	x	x	x
Background:					
OB 90	x	x	x	x	
Warm restart:					
OB 100	x	x	x	x	x
Hot restart:					
OB 101	x	x	x	x	
Cold restart:					
OB 102	x	x	x	x	x
Synchronous error interrupts:					
OB 121	x	x	x	x	x
OB 122	x	x	x	x	x

Start Events (Hexadecimal Values)
73A2, 73A3, 72A3
7301, 7302, 7303, 7320, 7321, 7322, 7323, 7331, 7333, 7334, 7335, 7340, 7341, 7342, 7343, 7344, 7950, 7951, 7952, 7852, 7953, 7954, 7955, 7855, 7956, 73C1, 73C2
3501, 3502, 3505, 3506, 3507
3821, 3822, 3823, 3825, 3826, 3827, 3831, 3832, 3833, 3921, 3922, 3923, 3925, 3926, 3927, 3931, 3932, 3933
3842, 3942
3861, 3863, 3864, 3865, 3961
35A1, 35A2, 35A3, 38B3, 38B4, 39B1, 39B2, 39B3, 39B4
38C1, 38C2, 39C1, 38C6, 38C7, 38C8 38C4 ¹⁾ , 38C5 ¹⁾ , 39C3 ¹⁾ , 39C4 ¹⁾ , 39C5 ¹⁾
35D2, 35D3, 35D4, 35D5, 35E1, 35E2, 35E3, 35E4, 35E5, 35E6
1191, 1192, 1193, 1195
1381, 1382, 138A, 138B
1383, 1384
1385, 1386, 1387, 1388
2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 253A, 253C, 253D, 253E, 253F
2942, 2943, 2944, 2945

¹⁾ not for CPU 412-1

Function Blocks (FB), Functions (FC) and Data Blocks

The following tables list the quantities, numbers and maximum sizes of the function blocks, functions and data blocks you can create for the various S7-400 CPUs.

Function Blocks	CPU 412-1	CPU 412-2	CPU 414	CPU 416	CPU 417
Quantity	256	256	2048	2048	6144
Permissible numbers	0 to 255	0 to 255	0 to 2047	0 to 2047	0 to 6143
Maximum size of a function block (code required for execution)	48 Kbytes	64 Kbytes	64 Kbytes	64 Kbytes	64 Kbytes

Functions	CPU 412-1	CPU 412-2	CPU 414	CPU 416	CPU 417
Quantity	256	256	2048	2048	6144
Permissible numbers	0 to 255	0 to 255	0 to 2047	0 to 2047	0 to 6143
Maximum size of a function (code required for execution)	48 Kbytes	64 Kbytes	64 Kbytes	64 Kbytes	64 Kbytes

Data Blocks	CPU 412-1	CPU 412-2	CPU 414	CPU 416	CPU 417
Quantity	511	511	4095	4095	8191
Permissible numbers	1 to 511	1 to 511	1 to 4095	1 to 4095	1 to 8191
Maximum size of a data block (number of data bytes)	48 Kbytes	64 Kbytes	64 Kbytes	64 Kbytes	64 Kbytes

System Functions

Operating system of the S7-400 CPUs and the execution times for the various CPUs.

SFC No.	SFC Name	Function
0	SET_CLK	Set clock
1	READ_CLK	Read clock
2	SET_RTM	Set run-time meter
3	CTRL_RTM	Start and stop run-time meter
4	READ_RTM	Read run-time meter
5	GADR_LGC	Find logical address of a channel Rack 0
		internal DP
6	RD_SINFO	Read start information of current OB
9	EN_MSG	Enable block-related, symbol-related, and group status messages. First call, REQ = 1
		Last call
10	DIS_MSG	Disable block-related, symbol-related, and group status messages. First call, REQ = 1
		Last call

Execution Time in μ s					
CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
340	249	215	249	289	288
40	29	23	29	29	54
35	26	20	26	25	26
30	23	18	23	22	22
41	30	23	30	29	58
55	39	31	39	38	38
66	46	36	46	46	46
54	38	30	38	39	39
176	122	97	122	128	232
61	44	34	44	39	62
176	122	97	122	128	232
61	44	34	44	39	63

System Functions, continued

SFC No.	SFC Name	Function	Execution Time in μs					
			CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
11	DPSYC_FR	Synchronize groups of DP Slaves First call, internal DP interface, REQ = 1	170	110	90	110	--	--
		First call, external DP interface, REQ=1	94	71	60	71	--	--
		Intermediate call, internal DP interface, BUSY = 1 ¹⁾	$51 + n^* 4$	$36 + n^* 3$	$28 + n^* 2$	$36 + n^* 3$	--	--
		Intermediate call, external DP interface, BUSY = 1 ¹⁾	$64 + n^* 4$	$50 + n^* 3$	$39 + n^* 2$	$50 + n^* 3$	--	--
		Last call, internal DP interface, BUSY=0 ¹⁾	$51 + n^* 4$	$36 + n^* 3$	$28 + n^* 2$	$36 + n^* 3$	--	--
		Last call, external DP interface, BUSY= 0 ¹⁾	$64 + n^* 4$	$50 + n^* 3$	$39 + n^* 2$	$50 + n^* 3$	--	--
13	DP_NRMDG	Read slave diagnostic data First call	300	200	165	200	210	290
		Intermediate call	--	--	--	--	79	79
		Last call (28 bytes)	180	125	100	125	101	101
14	DPRD_DAT	Read consistent user data (n bytes) via integrated DP interface 3 bytes	80	57	45	57	70	97
		32 bytes	92	67	54	67	87	122
15	DPWR_DAT	Read consistent user data (n bytes) via integrated DP interface 3 bytes	80	57	45	57	70	93
		32 bytes	92	67	54	67	86	109
17	ALARM_SQ	Generate acknowledgeable block-related messages. First call, SIG = 0 -> 1	440	305	240	305	266	358
		Empty call	130	90	72	90	90	156

1) n = number of active jobs with identical logical addresses

System Functions, continued

SFC No.	SFC Name	Function	Execution Time in μ s					
			CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
18	ALARM_S	Generate unacknowledgeable block-related messages. First call, SIG = 0 \rightarrow 1	460	310	250	310	275	365
		Empty call	140	90	75	90	97	163
19	ALARM_SC	Acknowledgment status of the last ALARM_SQ entering state message.	85	60	46	60	56	82
20	BLKMOV	Copy variable within the work memory (n = number of bytes to be copied)	60 + n * 0.3	41 + n * 0.13	32 + n * 0.23	41 + n * 0.13	42 + n * 0.17	42 + n * 0.17
		Source = Load memory	1400 + n * 1.0	1160 + n * 0.7	1100 + n * 0.7	1160 + n * 0.7	1124 + n * 1.0	2065 + n * 1.98
21	FILL	Set array default variables within the work memory (n = length of target variables in bytes)	60 + n * 0.15	44 + n * 0.13	34 + n * 0.1	44 + n * 0.13	45 + n * 0.12	45 + n * 0.12
		Source = Load memory	70 + n * 1.2	50 + n * 0.85	38 + n * 0.7	50 + n * 0.85	49 + n * 0.82	49 + n * 0.82
22	CREAT_DB	Create data block n = DB length [bytes]	142	94	72	94	155 + n * 0.1	424 + n * 0.1
		Occupy last free DB No. from a field of 100 DBs	606	400	320	400	2877	13601
23	DEL_DB	Delete data block	122	81	64	81	179	625
24	TEST_DB	Test data block	47	32	25	32	68	248
25	COMPRESS	Compress user memory First call (trigger)	112	78	63	78	93	173
		Intermediate call (active)	32	23	18	23	22	22
26	UPDAT_PI	Update process image input table (run-time entry for 1 DI 32 in the central rack)	45	35	29	35	67	103
		AI 8* 13Bit	70	59	51	59	155	192

System Functions, continued

SFC No.	SFC Name	Function
27	UPDAT_PO	Update process image output table (run-time entry for 1 DO 32 in the central rack)
		AO 8* 13Bit
28	SET_TINT	Set time-of-day interrupt
29	CAN_TINT	Cancel time-of-day interrupt
30	ACT_TINT	Activate time-of-day interrupt
31	QRY_TINT	Query time-of-day interrupt
32	SRT_DINT	Start time-delay interrupt
33	CAN_DINT	Cancel time-delay interrupt
34	QRY_DINT	Query time-delay interrupt
35	MP_ALM	Trigger multicomputing interrupt
36	MSK_FLT	Mask synchronous faults
37	DMSK_FLT	Demask synchronous faults
38	READ_ERR	Read error register
39	DIS_IRT	Discard new events Block all events (MODE = 0)
		Block all events of a priority class (MODE = 1)
		Block one event (MODE = 2)
40	EN_IRT	Stop discarding events Enable all events (MODE = 0)
		Enable all events in a priority class (MODE = 1)
		Enable an event (MODE = 2)

Execution Time in μ s					
CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
45	35	29	35	54	81
66	55	48	55	122	149
108	75	60	75	74	98
40	29	22	29	34	34
73	53	41	53	51	75
44	34	27	34	33	34
65	46	36	46	44	44
41	30	23	30	36	36
43	33	26	33	32	32
240	171	138	171	--	--
30	22	17	22	21	21
31	23	18	23	22	23
32	23	18	23	23	23
555	535	580	535	731	732
70 - 190	50 - 145	40 - 160	50 - 145	42 - 194	42 - 194
40 - 50	30 - 37	24 - 28	30 - 37	31 - 36	31 - 37
555	535	580	535	736	737
70 - 190	50 - 145	40 - 160	50 - 145	42 - 197	42 - 197
40 - 50	30 - 37	24 - 28	30 - 37	31 - 37	31 - 37

System Functions, continued

SFC No.	SFC Name	Function
41	DIS_AIRT	Delay interrupt events the first time delay is activated ¹⁾
		if the delay is already activated
42	EN_AIRT	Stop delaying interrupt events when canceling the last delay ²⁾
		if other delays are present
43	RE_TRIGR	Retrigger watchdog monitoring
44	REPL_VAL	Transfer substitute value to ACCU1
46	STP	Force CPU into STOP mode cannot be measured
47	WAIT	Delay program execution in addition to waiting time
48	SNC_RTCB	Synchronize slave clocks
49	LGC_GADR	Find slot with logical address
50	RD_LGADR	Find all logical addresses of a block (run-time entry for 1 DI 32 in the central rack)

- 1) The run time of SFC41 the first time the delay is activated depends on the priority class within which SFC41 is called. This time increases with decreasing priority class number.
- 2) The run time of SFC42 when the last delay is canceled depends on the priority class within which the SFC42 is called. This time increases with decreasing priority class number.

Execution Time in μ s					
CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
248	166	132	166	165	165
26	19	14	19	18	18
26	19	14	19	18	18
435	320	272	320	343	343
155	104	84	104	118	307
30	21	16	21	20	20
—	—	—	—	—	—
13 - 18	7 - 15	4 - 11	7 - 15	6 - 13	6 - 13
25	19	14	19	18	41
55	40	31	40	41	41
146	101	80	101	104	104

System Functions, continued

SFC No.	SFC Name	Function	Execution Time in μ s					
			CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
51	RDSYSST	List of all system status list information (0000)	618	493	395	493	477	477
		List of all system status list information (0F00)	140	97	77	97	97	97
51	RDSYSST	"Module Identification" partial list Display all data records (0011)	224	170	135	170	169	168
		Display one data record (0111)	175	125	100	125	123	122
		Display header information (0F11)	145	100	80	100	99	99
51	RDSYSST	"CPU Characteristics" partial list Display all data records (0012)	317	235	187	235	233	232
		Display one data record (0112)	190 - 215	135 - 155	108 - 123	135 - 155	135 - 155	135 - 154
		Display header information (0F12)	145	100	80	100	99	98
51	RDSYSST	"Save" partial list Display all data records (0013)	185	134	105	134	134	133
		Display one data record (0113)	185	134	105	134	134	133
		Display header information (0F13)	145	100	80	100	100	99
51	RDSYSST	"System Areas" partial list Display all data records (0014)	220	145	120	145	145	144
		Display one data record (0114)	170	117	93	117	117	117
		Display header information (0F14)	745	480	480	99	99	99
51	RDSYSST	"Block Types" partial list Display all data records (0015)	196	425	425	145	145	144
		Display one data record (0115)	165 - 185	118 - 128	94 - 102	118 - 128	119 - 128	118 - 127
		Display header information (0F15)	142	100	78	100	98	98

System Functions, continued

SFC No.	SFC Name	Function
51	RDSYSST	"Priority Classes" partial list Display all data records (0016)
		Display one data record (0116)
		Display header information (0F16)
51	RDSYSST	"Status of Module LEDs" partial list Display status of all LEDs (0019)
		Display status of one LED (0119)
		Display header information (0F19)
51	RDSYSST	"Interrupt/Error Assignment" partial list Display all data records (0021)
		Display all data records of one interrupt class (0121)
		Display one data record (0221)
		Display all assigned interrupts of one class (0921)
		Alternative: n= number of loaded OBs (0921)
51	RDSYSST	"Interrupt/Error Assignment" partial list Display all assigned interrupts (0A21)
		Display header information (0F21)

Execution Time in μ s					
CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
858	740	765	740	947	947
196 - 347	110 - 250	110 - 135	110 - 250	137 - 291	137 - 290
153	106	85	106	107	106
322	216	175	216	225	--
225	150	120	150	151	--
206	136	110	136	136	--
1225	1010	1055	1010	1298	1297
210 - 590	145 - 410	115 - 330	145 - 410	145 - 365	144 - 364
195 - 215	135 - 150	110 - 120	135 - 150	135 - 152	135 - 151
225 - 640	155 - 440	125 - 390	155 - 440	155 - 485	155 - 485
(225/375)+ n*34	(155/260)+ n*23	(125/245)+ n*18	(155/260)+ n*23	(155/305)+ n*23	(155/305)+ n*23
930 - 1510	795 - 1285	835 - 1390	795 - 1285	1037 - 1697	1037 - 1697
155	107	85	107	108	107

System Functions, continued

SFC No.	SFC Name	Function	Execution Time in μ s					
			CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
51	RDSYSST	"Interrupt Status" partial list Display all data records of one interrupt class (0122)	225 - 660	160 - 490	125 - 390	160 - 490	157 - 432	160 - 450
		Display one data record (0222)	210 - 225	148 - 158	118 - 128	148 - 158	148 - 155	148 - 158
		Display all assigned interrupts of one class (0822)	235 - 720	165 - 515	130 - 470	165 - 515	165 - 560	165 - 560
		Alternative: n = number of loaded OBs	(235/ 375)+ n*45	(165/ 260)+ n*35	(130/ 245)+ n*25	(165/ 260)+ n*35	(165/ 305)+ n*35	(165/ 305)+ n*35
		Display header information (0F22)	158	110	87	110	44	108
51	RDSYSST	"Status of Priority Classes" partial list Display one data record (0123)	210	147	117	147	147	147
		All priority classes in process (0223) (n= number of priority classes)	535 + n*52	450 + n*35	443 + n*28	450 + n*35	540 + n*36	540 + n*36
		Display header information (0F23)	145	100	80	100	101	100
51	RDSYSST	"Operating Mode" partial list Display the last operating mode transition (0124)	200	140	111	140	139	138
		Display the current operating mode	175	125	100	125	125	125
51	RDSYSST	"Status Information Communication" partial list Display status information of a communication unit (0132) INDEX = 5	205	150	120	150	157	181
51	RDSYSST	"Status Information Communication" partial list Display status information of a communication unit (0232) INDEX = 4	-	-	-	-	235	425

System Functions, continued

SFC No.	SFC Name	Function	Execution Time in μ s					
			CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
51	RDSYSST	"Start Information List" partial list All synchronization error start information of one priority class (0281)	190 - 225	128 - 155	102 - 135	128 - 155	127 - 168	127 - 167
		All start information of one priority class (0381)	210 - 395	128 - 305	102 - 255	128 - 305	128 - 318	127 - 317
		All synchronization error start information of one priority class before processing (0681)	190 - 225	128 - 155	102 - 135	128 - 155	127 - 168	127 - 167
		All start information of one priority class before processing (0781)	190 - 390	145 - 295	115 - 235	145 - 295	142 - 293	141 - 293
		All synchronization error start information of one priority class in process (0A81)	190 - 225	130 - 160	102 - 135	130 - 160	129 - 170	128 - 169
		All start information of one priority class in process (0B81)	190 - 240	130 - 170	102 - 145	130 - 170	129 - 179	129 - 179
		Display one header information (0F81)	160	112	90	112	112	111
51	RDSYSST	"Start Information List" partial list All synchronization error start events of one priority class (0282)	190 - 220	128 - 150	102 - 125	128 - 150	128 - 156	128 - 155
		All start events of one priority class (0382)	210 - 305	128 - 225	102 - 185	128 - 225	129 - 230	128 - 229
		All synchronization error start events of one priority class before processing (0682)	190 - 220	128 - 150	102 - 125	128 - 150	128 - 156	128 - 155
		All start events of one priority class before processing (0782)	210 - 310	145 - 225	115 - 180	145 - 225	143 - 225	142 - 223
		All synchronization error start events of one priority class in process (0A82)	190 - 220	130 - 150	102 - 125	130 - 150	130 - 158	129 - 161
		All start events of one priority class in process (0B82)	190 - 225	130 - 155	102 - 130	130 - 155	130 - 162	130 - 161
		Display one header information (0F82)	160	112	90	112	113	112

System Functions, continued

SFC No.	SFC Name	Function	Execution Time in μ s					
			CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
51	RDSYSST	"Module Status Information" partial list Display the status information of all inserted modules (n = number of the data records) (0091)	660 + n* 22	508 + n* 19	408 + n* 16	508 + n* 19	--	--
		Display the status information: of all modules/racks with incorrect type ID (0191)	570 + n* 70	427 + n* 60	365 + n* 40	405 + n* 24	--	--
		of all faulty modules (0291)	580 + n* 138	428 + n* 22	344 + n* 18	428 + n* 22	--	--
		of all unavailable modules (0391)	585 + n* 72	430 + n* 60	370 + n* 40	430 + n* 60	--	--
		Display the status information: of all submodules of the host module in the specified rack (0991)	354 + n* 30	250 + n* 26	200 + n* 21	250 + n* 26	--	--
		centralized of one module with logical base address (0C91)	200 - 315	180	145	180	177	242
		distributed of one module with logical base address (0C91)	315	225	180	225	224	289
51	RDSYSST	"Module Status Information" partial list of a module (distributed) with a logical base address (4C91) first call	200 - 315	145 - 240	130 - 190	145 - 240	242	305
		"Module Status Information" partial list of a module (distributed) with a logical base address (4C91) intermediate call	--	--	--	--	148	148
		"Module Status Information" partial list of a module (distributed) with a logical base address (4C91) last call	--	--	--	--	167	167

System Functions, continued

SFC No.	SFC Name	Function
51	RDSYSST	local of all modules in the specified rack (n = number of the DS) (0D91)
		distributed of all modules in the specified distributed I/O station (0D91)
		Display one header information (0F91)
51	RDSYSST	“Rack/Station Status Information” partial list local Display the setpoint status of the racks 0 (0092)
		distributed Display the setpoint status of distributed I/O system 1 (0092)
		local Display the actual status of the racks 0 (0292)
		distributed Display the actual status of distributed I/O system 1 (0292)
		Display header information (0F92)

Execution Time in μ s					
CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
377 + n* 13	275 + n* 16	240 + n* 10	275 + n* 16	260 + n* 20	405 + n* 23
330 - 390	250 - 300	200 - 240	250 - 300	305	408 - 420
560	435	350	435	--	--
180	127	100	127	130	154
900	725	585	725	712	743
180	127	103	127	131	155
940	745	600	745	725	757
160	113	90	113	113	113

System Functions, continued

SFC No.	SFC Name	Function
51	RDSYSST	"Diagnostic Buffer" partial list Display all available current operating mode event information (max. 23) (00A0)
		Display the n newest entries (n = 1-23) (01A0)
		Display the standard OB start information (04A0). Max value of -04A0 is calculated
		Display all communication information (05A0) Display all object management system information (06A0) Display all test and startup information (07A0) Display all operating mode information (08A0) Display all asynchronous error start information (09A0) Display all synchronous error start information (0AA0) Display all STOP/cancel/operating mode transition information (0BA0) Display all fault-tolerant/failsafe information (0CA0) Display all diagnostic information (0DA0) Display all user information (0EA0)
		Display header information (0FA0)

Execution Time in μ s					
CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
195 - 525	138 - 410	110 - 330	138 - 410	140 - 412	140 - 412
195 + n* 14.5	138 + n* 12	110 + n* 9.5	138 + n* 12	140 + n* 12	140 + n* 12
195 - 1270	138 - 1530	110 - 1095	138 - 1530	140 - 1540	140 - 1540
195 - 1270	138 - 1530	110 - 1095	138 - 1530	140 - 1540	140 - 1540
167	--	90	--	114	114

System Functions, continued

SFC No.	SFC Name	Function
51	RDSYSST	"Diagnostic Data DS 0", partial list Display via logical address (00B1) local
		distributed First call
		distributed Intermediate call, REQ = 0
		distributed Last call
51	RDSYSST	"Diagnostic Data DS 1" partial list Display via graphical address (00B2) Display a 16-byte long DS 1
51	RDSYSST	"Diagnostic Data DS 1" partial list Display via logical address (00B3) Display a 16-byte long DS 1 local
		distributed First call
		distributed Intermediate call
		distributed Last call
51	RDSYSST	"Diagnostic Data DP Slave" partial list Display via configured diagnostic address (00B4) First call
		Intermediate call, REQ = 0
		Last call (6 - 240 bytes)

Execution Time in μ s					
CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
406	286	233	286	300	360
392	270	217	270	278	356
215	150	120	150	153	152
405	165	132	165	170	169
408	300	250	300	313	375
408	300	250	300	313	375
395	270	218	270	272	356
218	150	120	150	153	153
257	178	142	178	182	182
385	266	213	266	272	351
--	--	115	--	149	148
246	170	135	170	174	173

DS = Data record

System Functions, continued

SFC No.	SFC Name	Function
52	WR_USMSG	Write user entry in diagnostic buffer write with message
		without message
54	RD_DPARAM	Read dynamic parameters local AI 8*13 bits
		distributed AI 8*12 bits (DS1 = 14 bytes)
55	WR_PARM	Write dynamic parameters local AI 8*13 bits
		distributed First call AI 8*12 bits (14 - 240 bytes)
		distributed Intermediate/last call, REQ = 0
56	WR_DPARAM	Write predefined dynamic parameters AI 8*13 bits local
		distributed First call AI 8*12 bits (2 - 240 bytes)
		Intermediate/last call
57	PARAM_MOD	Assign module parameters local Module/DS number/DS lengths in bytes AI 8*13 bits
		distributed AO 8*12 bits First call (16 - 240 bytes)
		distributed Intermediate/last call

Execution Time in μ s					
CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
186	128	102	128	75	100
107	75	60	75	74	98
180	125	95	125	126	153
200	135	105	135	121	121
485	345	280	345	360	418
370	260	210	260	268	347
175	115	90	115	122	122
445	336	280	336	353	411
300	205	165	205	217	296
145	100	80	100	106	106
770	580	490	580	609	695
300	205	165	205	215	295
145	100	80	100	104	104

System Functions, continued

SFC No.	SFC Name	Function	Execution Time in μ s					
			CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
58	WR_REC	Write parameter data record local (n = number of bytes)	390 + n* 2.87	267 + n* 2.71	217 + n* 2.52	267 + n* 2.71	282 + n* 2.68	311 + n* 2.71
		First call, integrated DP interface module (n = number of bytes)	334 + n* 0.42	228 + n* 0.35	182 + n* 0.30	228 + n* 0.35	222 + n* 0.39	276 + n* 0.32
		Intermediate call, REQ = 0 integrated DP interface module	138	90	70	90	94	94
		Last call, integrated DP interface module	138	90	70	90	95	94
		First call, external DP interface module (n = number of bytes)	332 + n* 0.32	215 + n* 0.26	171 + n* 0.23	215 + n* 0.26	208 + n* 0.26	208 + n* 0.29
		Intermediate call, REQ = 0 external DP interface module	139	90	72	90	95	94
		Last call, external DP interface module	140	91	72	91	95	95
59	RD_REC	Read data record local (n = number of bytes)	390 + n* 3.13	267 + n* 2.90	218 + n* 2.71	267 + n* 2.90	282 + n* 2.97	342 + n* 3.13
		First call, integrated DP interface module	322	217	172	217	212	264
		Intermediate call, REQ = 0 integrated DP interface module	138	90	70	90	95	94
		Last call, integrated DP interface module (n = number of bytes)	198 + n* 0.35	132 + n* 0.33	106 + n* 0.27	132 + n* 0.33	138 + n* 0.33	138 + n* 0.33
		First call, external DP interface module	304	204	163	204	198	197
		Intermediate call, REQ = 0 external DP interface module	139	91	72	91	95	94
		Last call, external DP interface module (n = number of bytes)	200 + n* 0.33	132 + n* 0.2	105 + n* 0.2	132 + n* 0.2	136 + n* 0.33	136 + n* 0.27
60	GD_SND	Send GD packet 1 byte	295	215	175	215	--	--
		32 bytes	910	640	515	640	--	--

System Functions, continued

SFC No.	SFC Name	Function
61	GD_RCV	Receive GD package (1 - 32 Byte)
62	CONTROL	Check status of the connection belonging to a local communication-SFB-instance
64	TIME_TCK	Display millisecond timer
65	X_SEND	Transmit data to external partner First call, establish a connection (1 - 76 bytes) REQ = 1
		First call, connection present (1-76 bytes)
		Intermediate call (1-76 bytes)
		Last call, BUSY = 0
66	X_RCV	Receive data from external partner Test reception (1-76 bytes)
		Read data (1-76 bytes)
67	X_GET	Read data from external partner First call, establish a connection (1-76 bytes) REQ = 1
		First call, connection present (1-76 bytes)
		Intermediate call (1-76 bytes)
		Last call BUSY = 0

Execution Time in μ s					
CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
145	105	85	105	--	--
116	87	69	87	107	136
24	19	15	19	19	47
860 - 910	710 - 740	765 - 795	710 - 740	--	--
590 - 635	400 - 430	320 - 345	400 - 430	--	--
180	130	100	130	--	--
285	195	155	195	--	--
92	65	55	65	--	--
275 - 315	190 - 220	150 - 175	190 - 220	--	--
760	645	715	645	--	--
490	335	265	335	--	--
195	135	110	135	--	--
450 - 490	310 - 340	245 - 270	310 - 340	--	--

System Functions, continued

SFC No.	SFC Name	Function
68	X_PUT	Write data to external partner First call, establish a connection (1-76 bytes) REQ = 1
		First call, connection present (1-76 bytes)
		Intermediate call (1-76 bytes)
		Last call, BUSY = 0
69	X_ABORT	Abort connection to external partner First call, REQ = 1
		Intermediate call
		Last call, BUSY = 0
72	I_GET	Read data from internal partner First call, establish a connection (1-76 bytes) REQ = 1
		First call, connection present (1-76 bytes)
		Intermediate call (1-76 bytes)
		Last call, BUSY = 0
73	I_PUT	Write data to internal partner First call, establish a connection (1-76 bytes) REQ = 1
		First call, connection present (1-76 bytes)
		Intermediate call (1-76 bytes)
		Last call, BUSY = 0

Execution Time in μ s					
CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
880 - 925	725 - 755	780 - 810	725 - 755	--	--
610 - 655	415 - 445	330 - 360	415 - 445	--	--
195	135	110	135	--	--
300	205	162	205	--	--
220	160	125	160	--	--
125	90	70	90	--	--
365	375	75 - 500	375	--	--
815	680	745	680	--	--
505	345	275	345	--	--
205	145	115	145	--	--
460 - 505	315 - 345	250 - 275	315 - 345	--	--
690 - 980	430 - 800	340 - 840	430 - 800	--	--
625 - 665	425 - 455	340 - 365	425 - 455	--	--
205	145	115	145	--	--
310	215	170	215	--	--

System Functions, continued

SFC No.	SFC Name	Function
74	I_ABORT	Abort connection to internal partner First call, REQ = 1
		Intermediate call
		Last call, without / with connection BUSY = 0
79	SET ¹⁾	Set bit array in I/O area n = number of bits to set at 1
80	RSET ¹⁾	Delete bit array in I/O area n = number of bits to set at 0
81	UBLKMOV	Copy variable without interruption n = number of bytes to copy
90	H_CTRL	Influence processes involving fault-tolerant systems

Execution Time in μ s					
CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
225	160	125	160	--	--
125	90	75	90	--	--
365	380	70 / 503	380	--	--
43 + n * 0.39	28 + n * 0.32	23 + n * 0.26	28 + n * 0.32	53 + n * 1.35	80 + n * 1.32
43 + n * 0.39	28 + n * 0.32	23 + n * 0.26	28 + n * 0.32	53 + n * 1.35	80 + n * 1.32
62 + n * 0.30	44 + n * 0.17	33 + n * 0.17	44 + n * 0.17	43 + n * 0.17	42 + n * 0.17
--	--	--	--	19 - 21	19 - 21

¹⁾ Measured with I/O modules of the type "Binary Simulator C79459-A1002-A1, Release 1" in the central rack

System Function Blocks

The following table lists the system function blocks provided with the operating system of the S7-400 CPUs as well as the execution times

SFC No.	SFC Name	Function
0	CTU	Count up
1	CTD	Count down
2	CTUD	Count up and down
3	TP	Generate pulse
4	TON	Generate on-delay
5	TOF	Generate off-delay
8	USEND	Send data without coordination (one send parameter supplied) JOB activated (1 - 440 bytes)
		JOB checked
		JOB finished (DONE = 1)
9	URCV	Receive data without coordination (one receive parameter supplied) JOB activated
		JOB checked
		JOB finished (NDR = 1; 1 - 440 bytes)
12	BSEND	Send data block by block JOB activated (1 - 3000 bytes)
		JOB checked
		JOB finished (DONE = 1)

of the individual CPUs (X: function exists, execution times were not available when manual was printed).

Execution Time in μ s					
CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
26	16	13	16	17	16
25	17	13	17	17	17
29	19	15	19	19	19
34	23	18	23	24	52
34	23	18	23	24	52
36	24	19	24	20	53
473 - 737	318 - 509	253 - 407	317 - 509	330 - 436	425 - 542
159	107	86	108	115	145
152	103	82	104	107	137
137	93	74	94	100	130
137	93	74	94	100	130
345 - 610	232 - 421	186 - 337	233 - 421	243 - 363	314 - 435
386	258	207	258	264	323
171	115	92	116	122	152
165	110	88	111	115	145

System Function Blocks, continued

SFC No.	SFC Name	Function	Execution Time in μ s					
			CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
13	BRCV	Receive data block by block JOB activated (1 - 3000 bytes)	203	138	110	139	145	175
		JOB checked	161	110	88	111	117	147
		JOB finished	162	109	87	110	113	143
14	GET	Read data from remote CPU (one area specified) JOB activated	336	227	183	228	227	297
		JOB checked	161	109	87	110	116	146
		JOB finished (NDR = 1; 1 - 450 bytes)	344 - 626	231 - 431	185 - 345	232 - 432	243 - 369	314 - 441
15	PUT	Write data to remote CPU JOB activated (1 - 404 bytes)	498 - 748	337 - 513	269 - 410	337 - 515	349 - 458	443 - 552
		JOB checked	161	108	87	109	116	146
		JOB finished (DONE = 1)	154	104	83	105	108	138
16	PRINT	Send data to a printer JOB activated, REQ = 1	513 - 757	338 - 516	271 - 414	339 - 518	354 - 462	449 - 545
		JOB checked	160	107	86	108	115	145
		JOB finished, DONE = 1	153	103	82	104	107	137
19	START	Start remote device JOB activated, REQ = 1	497	333	265	333	339	408
		JOB checked	169	114	91	115	121	151
		JOB finished, DONE = 1	164	110	88	111	115	146
20	STOP	Stop remote device JOB activated, REQ = 1	472	314	251	314	322	384
		JOB checked	169	114	91	115	121	151
		JOB finished, DONE = 1	164	110	88	111	115	146

System Function Blocks, continued

SFC No.	SFC Name	Function	Execution Time in μ s					
			CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
21	RESUME	Restart remote device JOB activated, REQ = 1	496	334	265	332	339	399
		JOB checked	169	114	91	115	121	151
		JOB finished, DONE = 1	164	110	88	111	115	145
22	STATUS	Query status of remote partner JOB activated, REQ = 1	268	183	146	184	188	258
		JOB checked	161	108	87	109	116	146
		JOB finished, NDR = 1	604	404	323	404	415	486
23	USTATUS	Receive status of remote device without coordination JOB activated, NDR = 1	137	93	74	94	100	131
		JOB checked	137	93	74	94	100	130
		JOB finished	604	404	323	404	415	486
32	DRUM	Implement sequencer	52	33	26	33	35	62
33	ALARM	Generate block-related message with acknowledgment JOB activated, SIG = 0→ 1 (1 - 420 bytes)	581 - 843	386 - 587	307 - 470	385 - 589	392 - 518	527 - 652
		JOB checked	205	136	109	137	141	171
		JOB finished, DONE = 1	207	137	110	138	136	166
34	ALARM_8	Generate block-related message without accompanying values for 8 signals JOB activated, SIG = 0→ 1 (1 - 420 bytes)	416	278	222	279	278	372
		JOB checked	203	135	108	136	140	170
		JOB finished, DONE = 1	206	137	109	138	135	166

System Function Blocks, continued

SFB No.	SFB Name	Function
35	ALARM_8P	Generate block-related message with accompanying values for 8 signals JOB activated, SIG = 0→ 1 (1 - 420 bytes)
		JOB checked
		JOB finished, DONE = 1
36	NOTIFY	Generate block-related message without acknowledgment JOB activated, SIG = 0→ 1 (1 - 420 bytes)
		JOB checked
		JOB finished, DONE = 1
37	AR_SEND	Send archive data JOB activated, REQ = 1 (1 - 3000 bytes)
		JOB checked
		JOB finished, DONE = 1

Execution Time in μ s					
CPU 412	CPU 414	CPU 416	CPU 417	CPU 417-4H (solo)	CPU 417-4H (redundant)
580 - 842	384 - 587	308 - 469	385 - 597	392 - 517	526 - 651
204	136	108	137	140	170
207	137	110	138	135	166
561 - 823	373 - 580	301 - 462	379 - 578	384 - 510	519 - 644
186	125	100	126	133	163
191	128	102	129	130	160
388	258	208	258	265	328
173	116	92	116	123	155
167	111	88	112	115	147

Sublist of the System Status List (SSL)

SSL-ID	Information Functions
	Module Identification
0111	One ident. data record only
	CPU Characteristics
0012	CPU features, all features
0112	Features of a group
	MC7 processing unit
	Time system
	System behavior
	MC7 language description
0F12	Only partial list header information
	User Memory Area
0113	Data record for specified memory area
	Work memory
0F13	Only partial list header information
	System Areas
0014	System areas, all system areas
0F14	Only partial list header information
	Block Types
0015	Block types, data records for all block types
	Status Module LEDs
0019	Status of all module LEDs
0F19	Only partial list header information

SSL-ID	Information Functions
	Interrupt Status
0222	Data record for specified interrupt
	Communication Status Data
0132	Status data for a communication unit
	Diagnostics
	Time system
0232	Status data for a communication unit
	CPU protection level and operator switch settings
	H CPU Group Information
0071	Information on the current status of the H system
0F71	Only partial list header information
	Status of the Module LEDs
0174	Status of one LED
	Module Status Information (A maximum of 27 data records are supplied)
0091	Module status information of all inserted modules/submodules
0191	Status information of all modules/racks with incorrect type IDs.
0291	Module status information of all faulty modules
0391	Module status information of all unavailable modules
0591	Module status information of all submodules of the host module

Sublist of System Status List, continued

SSL-ID	Information Functions
0991	Status information of all submodules in the host module in the rack
0A91	Module status information of all DP master systems
0C91	Status information of a module in the central rack or connected to an integrated DP interface module via the logical base address
4C91	Status information of a module connected to an external DP interface module via the logical base address
0D91	Status information of all modules in the specified rack
0E91	Status information of all assigned modules
	Rack/Station Status Information
0092	Expected status of the central racks/stations of a DP master system
4092	Expected status of the stations of a DP master system which is connected via an external DP interface module
0292	Actual status of the central racks/stations of a DP master system
4292	Actual status of the stations of a DP master system which is connected via an external DP interface module

SSL-ID	Information Functions
0692	OK status of the expansion units in the central configuration/stations of a DP master system which is connected via an integrated DP interface module.
4692	OK status of the stations of a DP master system which is connected via an external DP interface module.
0492	Status of the total backup for the racks in the central configuration
	Diagnostic Buffer (A maximum of 21 data records are supplied)
00A0	All current diagnostic entries available in current operating mode
01A0	Last x entries. X is listed in index
0FA0	Only partial list header information
	Module Diagnostic Data
00B1	First four diagnostic bytes of a module (DS0)
00B2	All diagnostic data of a module (≤ 220 bytes, DS1) (no DP module)
00B3	All diagnostic data of a module (≤ 220 bytes, DS1)
00B4	Diagnostic data of a DP slave with logical base address

Alphabetical Index of Instructions

Instruction	Page	Instruction	Page
)	26	==R	102
)MCR	132	<=D	100
+	94	<=I	98
+AR1	96	<=R	102
+AR2	96	<D	100
+D	84	<I	98
+I	82	<R	102
+R	86	<>D	100
-D	84	<>I	98
-I	82	<>R	102
-R	86	>=D	100
*D	84	>=I	98
*I	82	>=R	102
*R	86	>D	100
/D	84	>I	98
/I	82	>R	102
/R	86	A	20, 30, 36, 38
=	44	A(26
==D	100	ABS	86
==I	98	ACOS	92

Instruction	Page	Instruction	Seite
AD	34	CU	52
AN	20, 30, 36, 38	DEC	108
AN(26	DTB	112
ASIN	92	DTR	112
ATAN	92	ENT	108
AW	34	EXP	90
BE	122	FN	40
BEC	122	FP	40
BEU	122	FR	50, 54
BLD	110	INC	108
BTD	112	INVD	116
BTI	112	INVI	116
CAD	108	ITB	112
CALL	118	ITD	112
CAR	76	JBI	126
CAW	108	JC	124
CC	118	JCB	124
CD	52	JCN	124
CDB	122	JL	130
CLR	46	JM	128
COS	92	JMZ	128

Alphabetical Index of Instructions, continued

Instruction	Page	Instruction	Page
JN	128	MOD	84
JNB	124	NEGD	116
JNBI	126	NEGI	116
JO	126	NEGR	86
JOS	126	NOP	110
JP	128	NOT	46
JPZ	128	O	22, 28, 32, 36
JU	124	O(26
JUO	128	OD	34
JZ	128	ON	22, 32, 36
L	56 - 66, 80	ON(26
LAR1	74	OPN	120
LAR2	74	OW	34
LC	66	POP	108
LEAVE	108	PUSH	108
LN	90	R	42, 50, 52
LOOP	130	RLD	106
MCR(132	RLDA	106
MCRA	132	RND	114
MCRD	132	RND+	114

Instruction	Page	Instruction	Page
RND-	114	SS	48
RRD	106	SSD	104
RRDA	106	SSI	104
S	42, 52	T	68 - 72, 78
SAVE	46	TAK	108
SD	48	TAR1	76
SE	48	TAR2	76
SET	46	TAN	92
SF	48	TRUNC	114
SIN	92	UC	118
SLD	104	X	24, 32, 36
SLW	104	X(26
SP	48	XN	24, 32, 36
SQR	88	XN(26
SQRT	88	XOD	34
SRD	104	XOW	34
SRW	104		

Siemens AG
A&D AS E 81
Oestl. Rheinbrueckenstr. 50
D-76181 Karlsruhe
Federal Republic of Germany

Fax: 0721-595-2022

From:

Your Name: _ _ _ _ _

Your Title: _ _ _ _ _

Company Name: _ _ _ _ _

Street: _ _ _ _ _

City, Zip Code: _ _ _ _ _

Country: _ _ _ _ _

Phone: _ _ _ _ _

Please check any industry that applies to you:

- Automotive
- Chemical
- Electrical Machinery
- Food
- Instrument and Control
- Nonelectrical Machinery
- Petrochemical
- Pharmaceutical
- Plastic
- Pulp and Paper
- Textiles
- Transportation
- Other _ _ _ _ _

Remarks Form

Your comments and recommendations will help us to improve the quality and usefulness of our publications. Please take the first available opportunity to fill out this questionnaire and return it to Siemens.

Title of Instruction List: _ _ _ _ _
Order No. of Instruction List: _ _ _ _ _

Please give each of the following questions your own personal mark within the range from 1 (very good) to 5 (poor).

- 1. Do the contents meet your requirements?
- 2. Is the information you need easy to find?
- 3. Is the text easy to understand?
- 4. Does the level of technical detail meet your requirements?
- 5. Please rate the quality of the graphics/tables?

Additional comments:

