SIEMENS

SIMATIC

PCS 7 process control system Instruction List CPU 410-5H Process Automation

Parameter Manual

05/2013
A5E31664440-AA

Legal information

Warning notice system

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

⚠DANGER

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

⚠WARNING

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

⚠CAUTION

indicates that minor personal injury can result if proper precautions are not taken.

NOTICE

indicates that property damage can result if proper precautions are not taken.

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

Qualified Personnel

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

Proper use of Siemens products

Note the following:

⚠WARNING

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

Trademarks

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

Disclaimer of Liability

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

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E31664440-AA
℗ 06/2013 Technical data subject to change

Table of contents

Overview

1

1.1 Validity Range of the Instructions List

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

Name	Order number
CPU 410–5H Process Automation	6ES7 410-5HX08-0AB0

1.2 Address Identifiers and Parameter Ranges

Address	Parameter range	Description
Q	0.0 to 16383.7	Output (in PIQ)
QB	0 to 16383	Output byte (in PIQ)
QW	0 to 16382	Output word (in PIQ)
QD	0 to 16380	Output double word (in PIQ)
DB	1 to 16000	Data block
DBX	0.0 to 65533.7	Data bit in DB
DBB	0 to 65533	Data byte in DB
DBW	0 to 65532	Data word in DB
DBD	0 to 65530	Data double word in DB
DIX	0.0 to 65533.7	Data bit in instance DB
DI	1 to 16000	Instance data block
DIB	0 to 65533	Data byte in instance DB
DIW	0 to 65532	Data word in instance DB
DID	0 to 65530	Data double word in instance DB

Address	Parameter ranges	Description
I	0.0 to 16383.7	Input (in PII)
IB	0 to 16383	Input byte (in PII)

Address	Parameter ranges	Description
IW	0 to 16382	Input word (in PII)
ID	0 to 16380	Input double word (in PII)
L	0.0 to 65535.7	Local data
LB	0.0 to 65535	Local data byte
LW	0.0 to 65534	Local data word
LD	0.0 to 65532	Local data double word
M	0.0 to 16383.7	Bit memory
MB	0 to 16383	Memory byte
MW	0 to 16382	Memory word
MD	0 to 16380	Bit memory double word

Address	Parameter ranges	Description
PQB	0 to 16383	Peripheral output byte (direct I/O access)
PQW	0 to 16382	Peripheral output word (direct I/O access)
PQD	0 to 16380	Peripheral output double word (direct I/O access)
PIB	0 to 16383	Peripheral input byte (direct I/O access)
PIW	0 to 16382	Peripheral input word (direct I/O access)
PID	0 to 16380	Peripheral input double word (direct I/O access)
T	0 to 2047	Timer
C	0 to 2047	Counter

1.3 Constants

Table 1- 2 The following constants are used:

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

1) For loading the S7 timers

1.4 Abbreviations

Table 1- 3 The abbreviations are used:

Abbreviation	... 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 4 294 967 295	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 letters)	DEST
a	Byte address	

Abbreviation	... Description	Example
b	Bit address	
c	Address area (bit)	I, Q, M, L, DBX, DIX
d	Address is in: MD, DBD, DID, or LD	
e	Number is in: MW, DBW, DIW, or LW	
f	Timer/counter number	
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, PAD MD, LD, DBD, DID
q	Block number	

1.5 Registers

ACCU1 and ACCU2 (32 bit)

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

The accumulators are 32 bit long.

Table 1- 4 Designations:

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

Address Registers AR1 and AR2 (32 bit)

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

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

- Areainternal address:

  00000000 00000bbb bbbbbbbb bbbbbxxx

- Areacrossing address:

  10000yyy 00000bbb bbbbbbbb bbbbbxxx

Legend for structure of addresses:

- b: Byte address
- x: Bit number
- y: Area identifier (see section: Examples of Addressing (Page 13))

1.6 Status Word

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	/FC	First input bit scan
1	RLO	Result of logic operation
2	STA	Status
3	OR	Or (AND before OR)
4	OS	Stored overflow
5	OV	Overflow
6	CC0	Condition code
7	CC1	Condition code 1
8	BR	Binary result
9 to 15	Unassigned	-

Addressing

2

2.1 Address types

Table 2- 1 The following address types are used:

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

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

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

Definition of abbreviations

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

See also

Abbreviations (Page 7)

Examples of addressing (Page 13)

2.2 Examples of addressing

Examples of Addressing	Description
Immediate Addressing	
L +27	Load 16bit integer constant "27" into ACCU1
L L#–1	Load 32bit integer constant "-1" into ACCU1
L 2#1010101010101010	Load binary constant into ACCU1
L DW#16#A0F0BCFD	Load hexadecimal constant into ACCU1
L 'END'	Load ASCII character into ACCU1
L T#500 ms	Load time value into ACCU1
L C#100	Load count value into ACCU1
L B#(100,12)	Load 2-byte constant
L B#(100,12,50,8)	Load 4-byte constant
L P#10.0	Load areainternal pointer into ACCU1
L P#E20.6	Load cross-area pointer into ACCU1
L -2.5	Load real number into ACCU1
L D#1995–01–20	Load date
L TOD#13:20:33.125	Load time of day
Direct Addressing	
A I 0.0	ANDing of input bit 0.0
L IB1	Load input byte 1 into ACCU1
L IW 0	Load input word 0 into ACCU1
L ID 0	Load input double word 0 into ACCU1
Indirect Addressing of Timers/Counters	
SP T [LW 8]	Start timer; the timer number is in local data word 8
CU C [LW 10]	Start counter; the counter number is in local data word 10
AreaInternal MemoryIndirect Addressing	
A I [LD 12] **Example:** L P#22.2 T LD 12 A I [LD 12]	AND instruction: The address of the input is in local data double word 12 as pointer
A I [DBD 1]	AND instruction: The address of the input is in data double word 1 of the open DB as pointer
A Q [DID 12]	AND instruction: The address of the output is in data double word 12 of the open instance DB as pointer
A Q [MD 12]	AND instruction: The address of the output is in memory double word 12 as pointer

Examples of Addressing	Description			
Area-Internal Register-Indirect Addressing				
A I [AR1,P#12.2]	AND operation: The address of the input is calculated from the “pointer value in address register 1+ pointer P#12.2"			
Cross-area, register-indirect addressing [1)]				
	For cross-area registerindirect addressing, bit 24 to 26 of the address must also contain an area identifier. The address is in the address register.			
	Area ID	**Coding binary**	**Coding hexadecimal**	**Area**
	P	1000 0**000**	80	I/O area
	I	1000 0**001**	81	Input area
	Q	1000 0**010**	82	Output area
	M	1000 0**011**	83	Bit memory area
	DB	1000 0**100**	84	Data area
	DI	1000 0**101**	85	Instance data area
	L	1000 0**110**	86	Local data area
	VL	1000 0**111**	87	Predecessor local data area (access to local data of invoking block)
L B [AR1,P#8.0]	Load byte into ACCU1: The address is calculated from the "pointer value in AR1 + pointer P#8.0"			
A [AR1,P#32.3]	AND operation: The address is calculated from the "pointer value in address register 1+ pointer P#32.3"			
Addressing Via Parameters				
A Parameter	Addressing via parameters			

[1)] Logic Instructions with timers and counters (Page 22)

2.3 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)

2.4 Execution Times with Indirect Addressing

An instruction using indirect addressing consists of two parts:
Part 1: Loading the instruction address
Part 2: Executing the instruction
This means that when you are working with indirect addresses, you must also calculate the execution time of an instruction from these two parts.

Calculating the Execution Time

The total execution time is calculated as follows:
Execution time for loading the instruction address
+ Execution time of the instruction
= Total execution time of the instruction

The execution times given in chapter "List of instructions" are the execution times for 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 to this execution time (see following table).

The following table indicates the execution times for loading the address depending on the location of the address.

Address is in ...	Execution time in ns
Bit memory address area M	
Word	15
Double word	15
Data block DB/DI	
Word	18.75
Double word	18.75
Local data area L	
Word	15
Double word	15
AR1/AR2 (area-internal)	0.0 [1]
AR1/AR2 (cross-area)	0.0 [1]
Parameter (word) for:	
• Timers	18.75
• Counters	18.75
• Block calls	18.75
Parameter (double word) for	
Bits, bytes, words, and double words	18.75

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

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

- Calculating the execution time for cross-area registerindirect addressing
  Example: U [AR1, P#23.1] ... with I 1.0 in AR1 with CPU 410

Step 1: Loading of the content of DBD 12 (time required is in the above table)

Address is in ...	Execution time in ns
Bit memory address area M	
Word	18.75
Double word	18.75
Data block DB/DI	
Double word	46.88

Step 2: ANDing of the addressed input

Execution time in ns	
Direct addressing	Indirect addressing
18.75 :	Time for AI 46.88+

Total execution time
46.88 ns
+ 18.75 ns
= 65.63 ns

- Calculating the execution time for area-internal memory-indirect addressing
  Example: A I [DBD 12] with CPU 410

Step 1: Loading of the content of DBD 12 (time required is in table above)

Address is in ...	Execution time in ns
:	:
AR1/AR2 (cross-area)	0.00
:	:

Step 2: ANDing of the addressed input

Execution time in ns	
Direct addressing	Indirect addressing
13,25 :	Time for AI 0+

Total execution time
0 ns
+ 13.25 ns
= 13.25 ns

See also

Bit logic instructions (Page 18)

Instruction list

3

This section contains the complete list of CPU 410-5H Process Automation instructions. The descriptions have been kept as concise as possible.

Note

Execution times

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

See also:

- Examples of addressing (Page 13)
- Address types (Page 11)

Additional information

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

3.1 Logic instructions

3.1.1 Bit logic instructions

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

Instruction	Address	Description	Length in words	Execution time in ns
A/ AN		AND/AND-NOT		
	I/Q a.b	Input/output	1 [2] /2	7.5
	M a.b	Bit memory	1 [3] /2	7.5
	L a.b	Local data bit	2	7.5
	DBX a.b	Data bit	2	11.25
	DIX a.b	Instance data bit	2	11.25
	c [d]	Memory-indirect, area-internal [1]	2	7.5+/11.25+
	c [AR1,m]	Register-indirect, area-internal (AR1) [1]	2	7.5+/11.25+
	c [AR2,m]	Register-indirect, area-internal (AR2) [1]	2	7.5+/11.25+
	[AR1,m]	Cross-area (AR1) [1]	2	7.5+/11.25+
	[AR2,m]	Cross-area (AR2) [1]	2	7.5+/11.25+
	Parameter	Via parameter [1]	2	7.5+/11.25+

[1] I, Q, M, L / DB, DI

[2] With direct addressing; address range 0 to 127

[3] With direct addressing; address range 0 to 255

+ Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

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

Instruction	Address	Description	Length in words	Execution time in ns
O/ON		OR/OR-NOT		
	I/Q a.b	Input/output	1 [2] /2	7.5
	M a.b	Bit memory	1 [3] /2	7.5
	L a.b	Local data bit	2	7.5
	DBX a.b	Data bit	2	11.25
	DIX a.b	Instance data bit	2	11.25
	c [d]	Memory-indirect, areainternal [1]	2	7.5+/11.25+
	c [AR1,m]	Register-indirect, areainternal (AR1) [1]	2	7.5+/11.25+
	c [AR2,m]	Register-indirect, areainternal (AR2) [1]	2	7.5+/11.25+
	[AR1,m]	Cross-area (AR1) [1]	2	7.5+/11.25+
	[AR2,m]	Cross-area (AR2) [1]	2	7.5+/11.25+
	Parameter	Via parameter [1]	2	7.5+/11.25+

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

[1] I, Q, M, L / DB, DI

[2] With direct addressing; address range 0 to 127

[3] With direct addressing; address range 0 to 255

+ Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

Instruction	Address	Description	Length in words	Execution time in ns
X/XN		EXCLUSIVE-OR/ EXCLUSIVE-OR-NOT		
	I/Q a.b	Input/output	2	7.5
	M a.b	Bit memory	2	7.5
	L a.b	Local data bit	2	7.5
	DBX a.b	Data bit	2	11.25
	DIX a.b	Instance data bit	2	11.25
	c [d]	Memory-indirect, areainternal [1]	2	7.5+/11.25+
	c [AR1,m]	Register-indirect, areainternal (AR1) [1]	2	7.5+/11.25+
	c [AR2,m]	Register-indirect, areainternal (AR2) [1]	2	7.5+/11.25+
	[AR1,m]	Cross-area (AR1) [1]	2	7.5+/11.25+
	[AR2,m]	Cross-area (AR2) [1]	2	7.5+/11.25+
	Parameter	Via parameter [1]	2	7.5+/11.25+

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

[1] I, Q, M, L / DB, DI

+ Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

See also

Address types (Page 11)

Logic Instructions with Timers and Counters (Page 22)

3.1.2 Bit logic instructions with parenthetical expressions

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

Instruction	Address	Description	Length in words	Execution time in ns
A(		AND left parenthesis	1	7.5
AN(		AND NOT left parenthesis	1	7.5
O(		OR left parenthesis	1	7.5
ON(		OR NOT left parenthesis	1	7.5
X(		EXCLUSIVE OR left parenthesis	1	7.5
XN(		EXCLUSIVE OR NOT left parenthesis	1	7.5

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

Instruction	Address	Description	Length in words	Execution time in ns
)		Right parenthesis, removing an entry from the nesting stack.	1	7.5

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

3.1.3 ORing of AND functions

ORing of AND operations according to the rule: AND before OR

Instruction	Address	Description	Length in words	Execution time in ns
O		ORing of AND operations according to the rule: AND before OR	1	7.5

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

3.1.4 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	Description	Length in words	Execution time in ns
A/AN		AND/AND-NOT		
	T f	Timer	1[1)]/2	7.5
	T [e]	Timer, memory-indirect addressing	2	7.5+
	C f	Counter	1[1)]/2	7.5
	C [e]	Counter, memory-indirect addressing	2	7.5+
	Timer para. Counter para.	Timer/counter (addressing via parameter)	2	7.5+ 7.5+

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

[1)] With direct addressing; address range 0 to 255

[+] Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

Instruction	Address	Description	Length in words	Execution time in ns
O/ON		OR/OR-NOT	1[1)]/2	
	T f	Timer	2	7.5
	T [e]	Timer, memory-indirect addressing	1[1)]/2	7.5[+]
	C f	Counter	2	7.5
	C [e]	Counter, memory-indirect addressing	2	7.5[+]
	Timer para. Counter para.	Timer/counter (addressing via parameter)	2	7.5 7.5
X/XN		EXCLUSIVE-OR/EXCLUSIVE-OR-NOT		
	T f	Timer	2	7.5
	T [e]	Timer, memory-indirect addressing	2	7.5[+]
	C f	Counter	2	7.5
	C [e]	Counter, memory-indirect addressing	2	7.5[+]
	Timer para. Counter para.	EXCLUSIVE-OR timer/counter (addressing via parameter)	2	7.5[+] 7.5[+]

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

[1)] With direct addressing; address range 0 to 255

[+] Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

See also

Address types (Page 11)

3.1.5 Word Logic Instructions with the Contents of Accumulator 1

Gating the contents of ACCU1 and/or ACCU1L with a word or double word according to the appropriate function. The word or double word
is either specified in the instruction as an address or is in ACCU2. The result is in ACCU1 or ACCU1-L.

Instruction	Address	Description	Length in words	Execution time in ns
AW		AND ACCU2L	1	7.5
AW	W#16#p	AND 16-bit constant	2	7.5
OW		OR ACCU2L	1	7.5
OW	W#16#p	OR 16-bit constant	2	7.5
XOW		EXCLUSIVE-OR ACCU2L	1	7.5
XOW	W#16#p	EXCLUSIVE-OR 16-bit constant	2	7.5

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

Instruction	Address	Description	Length in words	Execution time in ns
AD		AND ACCU2	1	7.5
AD	DW#16#p	AND 32-bit constant	3	7.5
OD		OR ACCU2	1	7.5
OD	DW#16#p	OR 32-bit constant	3	7.5
XOD		EXCLUSIVE-OR ACCU2	1	7.5
XOD	DW#16#p	EXCLUSIVE-OR 32-bit constant	3	7.5

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

3.1.6 Logic Instructions Using AND, OR and EXCLUSIVE OR

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

Instruction	Address	Description	Length in words	Execution time in ns
A/AN O/ON X/XN	==0	AND/AND-NOT OR/OR-NOT EXCLUSIVE-OR/ EXCLUSIVE-OR-NOT Result=0 (CC1=0 and CC0=0)	1	7.5
	>0	Result>0 (CC1=1 and CC0=0)	1	7.5
	<0	Result<0 (CC1=0 and CC0=1)	1	7.5
	<>0	Result≠0 ((CC1=0 and CC0=1) or (CC1=1 and CC0=0))	1	7.5
	<=0	Result<=0 ((CC1=0 and CC0=1) or (CC1=0 and CC0=0))	1	7.5
	>=0	Result>=0 ((CC1=1 and CC0=0) or (CC1=0 and CC0=0))	1	7.5

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

Instruction	Address	Description	Length in words	Execution time in ns
A/AN O/ON X/XN	<=0	Result<=0 ((CC1=0 and CC0=1) or (CC1=0 and CC0=0))	1	7.5
	>=0	Result>=0 ((CC1=1 and CC0=0) or (CC1=0 and CC0=0))	1	7.5

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

Instruction	Address	Description	Length in words	Execution time in ns
A/AN O/ON X/XN	AO	AND/AND-NOT OR/OR-NOT EXCLUSIVE-OR/ EXCLUSIVE-OR-NOT unordered/invalid math instruction (CC1=1 and CC0=1)	1	7.5
	OS	AND OS=1	1	7.5
	BR	AND BR=1	1	7.5
	OV	AND OV=1	1	7.5

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

3.2 EdgeTriggered Instructions

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

Instruction	Address	Description	Length in words	Execution time in ns
FP/FN		The positive/negative edge is indicated by RLO=1. The bit addressed in the instruction is the auxiliary edge bit memory.		
	I/Q a.b		2	7.5
	M a.b		2	7.5
	L a.b [1)]		2	7.5
	DBX a.b		2	18.75
	c [d] [2)]		2	18.75
	c [AR1,m] [2)]		2	7.5+/18.75+
	c [AR2,m] [2)]		2	7.5+/18.75+
	[AR1,m] [2)]		2	7.5+/18.75+
	[AR2,m] [2)]		2	7.5+/18.75+
	Parameter [2)]		2	7.5+/18.75+

[1)] Unnecessary if the edge bit memory is in the process image (local data of a block are only valid while the block is running).
[2)] I, Q, M, L / DB, DI
+ Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

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

See also

Address types (Page 11)

3.3 Setting/Resetting Bit Addresses

Assigning the value "1" or "0" to the specified address, if RLO = 1. The instructions can be dependent on the MCR.

Instruction	Address	Description	Length in words	Execution time in ns
S		Set addressed bit to "1"		
R		Set addressed bit to "0"		
	I/Q a.b	Input/output	1[2)]/2	7.5
	M a.b	Bit memory	1[3)]/2	7.5
	L a.b	Local data bit	2	7.5
	DBX a.b	Data bit	2	18.75
	DIX a.b	Instance data bit	2	18.75
	c [d]	Memory-indirect, area-internal[1)]	2	7.5+/18.75+
	c [AR1,m]	Register-indirect, areainternal (AR1) [1)]	2	7.5+/18.75+
	c [AR2,m]	Register-indirect, areainternal (AR2) [1)]	2	7.5+/18.75+
	[AR1,m]	Cross-area (AR1) [1)]	2	7.5+/18.75+
	[AR2,m]	Cross-area (AR2) [1)]	2	7.5+/18.75+
	Parameter	Via parameter [1)]	2	7.5+/18.75+

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

[1)] I, Q, M, L / DB, DI
[2)] With direct addressing; address range 0 to 127
[3)] With direct addressing; address range 0 to 255
+ Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

The RLO value is written to the specified address identifier. Note the MCR dependency (see Instructions for the Master Control Relay (MCR) (Page 58)).

Instruction	Address	Description	Length in words	Execution time in ns
=		Assign RLO		
	I/Q a.b	Input/output	1[2)]/2	7.5
	M a.b	Bit memory	1[3)]/2	7.5
	L a.b	Local data bit	2	7.5
	DBX a.b	Data bit	2	18.75
	DIX a.b	Instance data bit	2	18.75
	c [d]	Memory-indirect, area-internal[1)]	2	7.5+/18.75+
	c [AR1,m]	Register-indirect, areainternal (AR1) [1)]	2	7.5+/18.75+
	c [AR2,m]	Register-indirect, areainternal (AR2) [1)]	2	7.5+/18.75+
	[AR1,m]	Cross-area (AR1) [1)]	2	7.5+/18.75+
	[AR2,m]	Cross-area (AR2) [1)]	2	7.5+/18.75+
	Parameter	Via parameter [1)]	2	7.5+/18.75+

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

[1)] I, Q, M, L / DB, DI
[2)] With direct addressing; address range 0 to 127
[3)] With direct addressing; address range 0 to 255
+ Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

See also

Address types (Page 11)

3.4 Instructions Directly Affecting the RLO

The following instructions have a direct effect on the RLO.

Instruction	Address	Description	Length in words	Execution time in ns
CLR		Set RLO to "0"	1	7.5

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

Instruction	Address	Description	Length in words	Execution time in ns
SET		Set RLO to "1"	1	7.5

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

Instruction	Address	Description	Length in words	Execution time in ns
NOT,		Negate RLO	1	7.5

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

Instruction	Address	Description	Length in words	Execution time in ns
SAVE		Retain the RLO in the Bit BR	1	7.5

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

3.5 Timer Instructions

Starting or resetting a timer. The time value must be in ACCU1L. The instructions are triggered by an edge transition in the RLO. That is, when the status of the RLO has changed between two calls, the instruction is initiated.

Instruction	Address	Description	Length in words	Execution time in ns
SP	T f T [e]	Start timer as pulse on edge change from "0" to "1"	1[1)]/2	18.75 18.75[+]
	Timer para.		2	18.75[+]
SE	T f T [e]	Start timer as extended pulse on edge change from "0" to "1"	1[1)]/2	18.75 18.75[+]
	Timer para.		2	18.75[+]
SD	T f T [e]	Start timer as ON delay on edge change from "0" to "1"	1[1)]/2	18.75 18.75[+]
	Timer para.		2	18.75[+]

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

[1)] With direct addressing; timer no.: 0 to 255
[+] Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

Instruction	Address	Description	Length in words	Execution time in ns
SS	T f T [e]	Start timer as retentive ON delay on edge change from "0" to "1"	1[1)]/2	18.75 18.75[+]
	Timer para.		2	18.75[+]
SF	T f T [e]	Start timer as OFF delay on edge change from "1" to "0"	1[1)]/2	18.75 18.75[+]
	Timer para.		2	18.75[+]

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

[1)] With direct addressing; timer no.: 0 to 255
[+] Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

Instruction	Address	Description	Length in words	Execution time in ns
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	18.75 18.75[+]
	Timer para.		2	18.75[+]
R	T f T [e]	Reset timer	1[1)]/2	18.75 18.75[+]
	Timer para.		2	18.75[+]

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

[1)] With direct addressing; timer no.: 0 to 255
[+] Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

See also

Address types (Page 11)

3.6 Counter Instructions

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

Instruction	Address	Description	Length in words	Execution time in ns
S	C f C [e]	Preset counter on edge change from "0" to "1"	1[1)]/2	15 15[+]
	Counter para.		2	15[+]
R	C f C [e]	Reset of counter to "0" when RLO = "1"	1[1)]/2	15 15[+]
	Counter para.		2	15[+]
CU	C f C [e]	Increment counter by 1 on edge change from "0" to "1"	1[1)]/2	18.75 18.75[+]
	Counter para.		2	18.75[+]

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

[1)] With direct addressing, counter no.: 0 to 255
[+] Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

Instruction	Address	Description	Length in words	Execution time in ns
CD	C f C [e]	Decrement counter by 1 on edge change from "0" to "1"	1[1)]/2	18.75 18.75[+]
	Counter para.		2	18.75[+]
FR	C f C [e]	Enable counter on edge change from "0" to "1" (reset edge bit memory for up and down counting and set a counter)	1[1)]/2	18.75 18.75[+]
	Counter para.		2	18.75[+]

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

[1)] With direct addressing; counter no.: 0 to 255
[+] Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

See also

Address types (Page 11)

3.7 Load Instructions

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

Instruction	Address	Description	Length in words	Execution time in ns
L		Load ...		
	IB a	Input byte	1[2]/2	7.5
	QB a	Output byte	1[2]/2	7.5
	PIB a	Peripheral input byte [1]	1[2]/2	7.5
	MB a	Memory byte	1[3]/2	7.5
	LB a	Local data byte	2	7.5
	DBB a	Data byte	2	7.5
	DIB a	Instance data byte	2	7.5
		... in ACCU1		
	g [d]	Memory-indirect, areainternal [4]	2	7.5+
	g [AR1,m]	Register-indirect, areainternal (AR1) [4]	2	7.5+
	g [AR2,m]	Register-indirect, areainternal (AR2) [4]	2	7.5+
	B[AR1,m]	Cross-area (AR1) [4]	2	7.5+
	B[AR2,m]	Cross-area (AR2) [4]	2	7.5+
	Parameter	Via parameter [4]	2	7.5+

[1] Plus acknowledgment time of the I/O module (> 1 µs), bus runtimes and synchronization time in redundant mode
[2] With direct addressing; address range 0 to 127
[3] With direct addressing; address range 0 to 255
[4] I, Q, P, M, L / DB, DI
+ Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

If there is remainder of 3 following an integral division of the used address by 4, the execution times for instructions specified on this page are doubled.

Instruction	Address	Description	Length in words	Execution time in µs
L		Load ...		
	IW a	Input word	1[2)]/2	7.5
	QW a	Output word	1[2)]/2	7.5
	PIW a	Peripheral input word 1)	1[2)]/2	7.5
	MW a	Memory word	1[3)]/2	7.5
	LW a	Local data word	2	7.5
	DBW a	Data word	2	7.5
	DIW a	Instance data word	2	7.5
		... in ACCU1-L		
	h [d]	Memory-indirect, areainternal [4)]	2	7.5+
	h [AR1,m]	Register-indirect, areainternal (AR1) [4)]	2	7.5+
	h [AR2,m]	Register-indirect, areainternal (AR2) [4)]	2	7.5+
	W[AR1,m]	Cross-area (AR1) [4)]	2	7.5+
	W[AR2,m]	Cross-area (AR2) [4)]	2	7.5+
	Parameter	Via parameter [4)]	2	7.5+

[1)] Plus acknowledgment time of the I/O module (> 1 µs), bus runtimes and synchronization time in redundant mode
[2)] With indirect addressing; address range 0 to 127
[3)] With direct addressing; address range 0 to 255
[4)] I, Q, P, M, L / DB, DI
+ Plus time for loading the address (see Auto-Hotspot)

If the address used cannot be evenly divided by 4, the execution times for instructions specified on this page are doubled.

Instruction	Address	Description	Length in words	Execution time in µs
L		Load ...		
	ID a	Input double word	1[2)]/2	7.5
	QD a	Output double word	1[2)]/2	7.5
	PID a	Peripheral input double word [1)]	2	7.5
	MD a	Memory double word	1[3)]/2	7.5
	LD a	Local data double word	2	7.5
	DBD a	Data double word	2	11.25
	DID a	Instance data double word	2	11.25
		... in ACCU1		
	i [d]	Memory-indirect, areainternal [4)]	2	7.5+
	i [AR1,m]	Register-indirect, areainternal (AR1) [4)]	2	7.5+
	i [AR2,m]	Register-indirect, areainternal (AR2) [4)]	2	7.5+
	D[AR1,m]	Cross-area (AR1) [4)]	2	7.5+
	D[AR2,m]	Cross-area (AR2) [4)]	2	7.5+
	Parameter	Via parameter [4)]	2	7.5+

[1] Plus acknowledgment time of the I/O module (> 1 µs), bus runtimes and synchronization time in redundant mode
[2] With indirect addressing; address range 0 to 127
[3] With direct addressing; address range 0 to 255
[4] I, Q, P, M, L / DB, DI
+ Plus time for loading the address (see Constants (Page 7))

Instruction	Address	Description	Length in words	Execution time in µs
L		Load ...		
	k8	8bit constant in ACCU1-LL	2	7.5
	k16	16bit constant in ACCU1-L	2	7.5
	k32	32bit constant in ACCU1	3	7.5
	Parameter	Load constant into ACCU1 (from parameter)	2	11.25+
L	2#n	Load 16bit binary constant into ACCU1-L	2	7.5
		Load 32bit binary constant into ACCU1	3	7.5
	B#16#p	Load 8bit hexadecimal constant into ACCU1-L	1	7.5
L	W#16#p	Load 16bit hexadecimal constant into ACCU1-L	2	7.5
	DW#16#p	Load 32bit hexadecimal constant into ACCU1	3	7.5

+ Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

Instruction	Address	Description	Length in words	Execution time in µs
L	'x'	Load 1 character	2	7.5
	'xx'	Load 2 characters	2	7.5
	'xxx'	Load 3 characters	3	7.5
	'xxxx'	Load 4 characters	3	7.5
L	D# time value	Load IEC date constant	3	7.5
L	S5T# time value	Load S7 time constant (16 bits)	2	7.5
L	TOD# time value	Load IEC time constant	3	7.5
L	T# time value	Load 16-bit time constant	2	7.5
		Load 32-bit time constant	3	7.5
L	C# Count value	Load counter constant (BCD code)	2	7.5
L	B# (b1, b2)	Load constant as bytes (b1, b2)	2	7.5
	B# (b1, b2, b3, b4)	Load constant as 4 bytes (b1, b2, b3, b4)	3	7.5

Instruction	Address	Description	Length in words	Execution time in µs
L	P# bit pointer	Load bit pointer	3	7.5
L	L# integer	Load 32-bit integer constant	3	7.5
L	Real number	Load floating-point number	3	7.5

See also

Address types (Page 11)

3.8 Load Instructions for Timers and Counters

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

Instruction	Address	Description	Length in words	Execution time in ns
L	T f T [e]	Load time value	1[1)]/2 2	7.5 7.5+
	Timer para.	Load time value (addressed via parameter)	2	7.5+
L	C f C [e]	Load count value	1[1)]/2 2	7.5 7.5+
	Counter para.	Load count value (addressed via parameter)	2	7.5+
LC	T f T [e]	Load time value BCD-coded	1[1)]/2 2	7.5 7.5+
	Timer para.	Load time value in BCD (addressed via parameter)	2	7.5+
LC	C f C [e]	Load count value in BCD	1[1)]/2 2	7.5 7.5+
	Counter para.	Load count value in BCD (addressed via parameter)	2	7.5+

[1)] With direct addressing, timer/counter no.: 0 to 255
+ Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

See also

Address types (Page 11)

3.9 Transfer Instructions

Transferring the contents of ACCU1 to the specified address identifier. Note that some instructions are affected by the MCR. The status word is not affected.

Instruction	Address	Description	Length in words	Execution time in ns
T		Transfer contents of ACCU1-LL to ...		
	IB a	Input byte	1[2)]/2	7.5
	QB a	Output byte	1[2)]/2	7.5
	PQB	Peripheral output byte [1)]	1[2)]/2	7.5
	MB a	Memory byte	1[3)]/2	7.5
	LB a	Local data byte	2	7.5
	DBB a	Data byte	2	11.25
	DIB a	Instance data byte	2	11.25
	g [d]	Memory-indirect, area-internal [4)]	2	7.5+/11.25+
	g [AR1,m]	Register-indirect, areainternal (AR1) [4)]	2	7.5+/11.25+
	g [AR2,m]	Register-indirect, areainternal (AR2) [4)]	2	7.5+/11.25+
	B[AR1,m]	Cross-area (AR1) [4)]	2	7.5+/11.25+
	B[AR2,m]	Cross-area (AR2) [4)]	2	7.5+/11.25+
	Parameter	Via parameter [4)]	2	7.5+/11.25+

[1)] Plus acknowledgment time of the I/O module (> 1 µs), bus runtimes and synchronization time in redundant mode
[2)] With direct addressing; address range 0 to 127
[3)] With direct addressing; address range 0 to 255
[4)] I, Q, P, M, L / DB, DI
+ Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

If there is remainder of 3 following an integral division of the used address by 4, the execution times for instructions specified on this page are doubled.

Instruction	Address	Description	Length in words	Execution time in ns
T		Transfer contents of ACCU1-L to ...		
	IW a	Input word	1[2)]/2	7.5
	QW a	Output word	1[2)]/2	7.5
	PQW a	Peripheral output word [1)]	1[2)]/2	7.5
	MW a	Memory word	1[3)]/2	7.5
	LW a	Local data word	2	7.5
	DBW a	Data word	2	22.5
	DIW a	Instance data word	2	22.5
	h [d]	Memory-indirect, area-internal [4)]	2	7.5+/22.5+
	h [AR1,m]	Register-indirect, areainternal (AR1) [4)]	2	7.5+/22.5+
	h [AR2,m]	Register-indirect, areainternal (AR2) [4)]	2	7.5+/22.5+
	W[AR1,m]	Cross-area (AR1) [4)]	2	7.5+/22.5+
	W[AR2,m]	Cross-area (AR2) [4)]	2	7.5+/22.5+
	Parameter	Via parameter [4)]	2	7.5+/22.5+

[1)] Plus acknowledgment time of the I/O module (> 1 µs), bus runtimes and synchronization time in redundant mode
[2)] With direct addressing; address range 0 to 127
[3)] With direct addressing; address range 0 to 255
[4)] I, Q, P, M, L / DB, DI
+ plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

If the address used cannot be evenly divided by 4, the execution times for instructions specified on this page are doubled.

Instruction	Address	Description	Length in words	Execution time in µs
T		Transfer contents of ACCU1-L to ...		
	ID a	Input double word	1[2)]/2	7.5
	QD a	Output double word	1[2)]/2	7.5
	PQD a	Peripheral output double word [1)]	2	7.5
	MD a	Memory double word	1[3)]/2	7.5
	LD a	Local data double word	2	7.5
	DBD a	Data double word	2	22.5
	DID a	Instance data double word	2	22.5
T	i [d]	Memory-indirect, area-internal [4)]	2	7.5+/22.5+
	i [AR1,m]	Register-indirect, areainternal (AR1) 4)	2	7.5+/22.5+
	i [AR2,m]	Register-indirect, areainternal (AR2) 4)	2	7.5+/22.5+
	D[AR1,m]	Cross-area (AR1) 4)	2	7.5+/22.5+
	D[AR2,m]	Cross-area (AR2) 4)	2	7.5+/22.5+
	Parameter	Via parameter 4)	2	7.5+/22.5+

[1] Plus acknowledgment time of the I/O module (> 1 µs), bus runtimes and synchronization time in redundant mode
[2] With direct addressing; address range 0 to 127
[3] With direct addressing; address range 0 to 255
[4] I, Q, P, M, L / DB, DI
[+] Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))

See also

Address types (Page 11)

3.10 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	Description	Length in words	Execution time in ns
LAR1		Load contents from ...		
	-	ACCU1	1	7.5
	AR2	Address register 2	1	7.5
	DBD	Data double word	2	11.25
	DID a	Instance data double word	2	11.25
	m	32bit constant as pointer	3	7.5
	LD a	Local data double word	2	7.5
	MD a	Memory double word	2	7.5
		... in AR1		
LAR2		Load contents from ...		
	-	ACCU1	1	7.5
	DBD a	Data double word	2	11.25
	DID a	Instance data double word	2	11.25
	m	32bit constant as pointer	3	7.5
	LD a	Local data double word	2	7.5
	MD a	Memory double word	2	7.5
		... in AR2		

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

Instruction	Address	Description	Length in words	Execution time in ns
TAR1		Transfer contents from AR1 to ...		
	-	ACCU1	1	7.5
	AR2	Address register 2	1	7.5
	DBD a	Data double word	2	22.5
	DID a	Instance data double word	2	22.5
	LD a	Local data double word	2	7.5
	MD a	Memory double word	2	7.5
TAR2		Transfer contents from AR2 to ...		
	-	ACCU1	1	7.5
	DBD a	Data double word	2	22.5
	DID a	Instance data double word	2	22.5
	LD a	Local data double word	2	7.5
	MD a	Memory double word	2	7.5
TAR		Exchange the contents of AR1 and AR2	1	7.5

3.11 Load and Transfer Instructions for the Status Word

Instruction	Address	Description	Length in words	Execution time in ns
L	STW	Load status word into ACCU1	1	7.5

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

Instruction	Address	Description	Length in words	Execution time in ns
	STW	Transfer ACCU1 (bits 0 to 8) to the status word	1	7.5

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

See also

Status Word (Page 9)

3.12 Load Instructions for DB Number and DB Length

Loading the number/length of a data block into ACCU1. The old contents of ACCU1 are saved to ACCU2. The status word is not affected.

Instruction	Address	Description	Length in words	Execution time in ns
L	DBNO	Load number of data block	1	7.5
L	DINO	Load number of instance data block	1	7.5
L	DBLG	Load length of data block in bytes	1	7.5
L	DILG	Load length of instance data block in bytes	1	7.5

3.13 Fixedpoint arithmetic (16/32 bit) / Floatingpoint arithmetic (32 bit)

Integer Math (16 bits)

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

Instruction	Address	Description	Length in words	Execution time in ns
+I		Add 2 integers (16 bits) (ACCU1-L)=(ACCU1-L)+(ACCU2-L)	1	7.5
-I		Subtract 2 integers (16 bits) (ACCU1-L)=(ACCU2-L)-(ACCU1-L)	1	7.5

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

Instruction	Address	Description	Length in words	Execution time in ns
*I		Multiply 2 integers (16 bits) (ACCU1)=(ACCU2-L)*(ACCU1-L)	1	7.5
/I		Divide 2 integers (16 bits) (ACCU1-L)=(ACCU2-L):(ACCU1-L) The remainder is in ACCU1-H.	1	30

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

Integer Math (32 bits)

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

Instruction	Address	Description	Length in words	Execution time in ns
+D		Add 2 integers (32 bits) (ACCU1)=(ACCU2)+(ACCU1)	1	7.5
-D		Subtract 2 integers (32 bits) (ACCU1)=(ACCU2)-(ACCU1)	1	7.5
*D		Multiply 2 integers (32 bits) (ACCU1)=(ACCU2)*(ACCU1)	1	7.5

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

Instruction	Address	Description	Length in words	Execution time in ns
/D		Divide 2 integers (32 bits) (ACCU1)=(ACCU2):(ACCU1)	1	45
MOD		Divide 2 integers (32 bits) and load the remainder into ACCU1: (ACCU1)=remainder of [(ACCU2):(ACCU1)]	1	45

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

Floating-Point Math (32 bits)

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

Instruction	Address	Description	Length in words	Execution time in ns
+R		Add 2 real numbers (32 bits) (ACCU1)=(ACCU2)+(ACCU1)	1	15
-R		Subtract 2 real numbers (32 bits) (ACCU1)=(ACCU2)-(ACCU1)	1	15

Instruction	Address	Description	Length in words	Execution time in ns
*R		Multiply 2 real numbers (32 bits) (ACCU1)=(ACCU2)*(ACCU1)	1	15
/R		Divide 2 real numbers (32 bits) (ACCU1)=(ACCU2):(ACCU1)	1	45

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

Instruction	Address	Description	Length in words	Execution time in ns
NEGR		Negate the real number in ACCU1	1	7.5
ABS		Form the absolute value of the real number in ACCU1	1	7.5

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

3.14 Square root, Square (32bit) / Logarithm function (32bit)

Square Root, Square (32 Bits)

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

Instruction	Address	Description	Length in words	Execution time in ns
SQRT		Calculate the square root of a real number in ACCU1	1	60
SQR		Form the square of a real number in ACCU1	1	15

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

Logarithmic Functions (32-bit)

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

Instruction	Address	Description	Length in words	Execution time in ns
LN		Form the natural logarithm of a real number in ACCU1	1	157.5
EXP		Calculate the exponential value of a real number in ACCU1 to the base e (= 2.71828)	1	157.5

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

3.15 Trigonometrical Functions (32 Bits)

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

Instruction	Address	Description	Length in words	Execution time in ns
SIN		Calculate the sine of a real number	1	150
ASIN		Calculate the arcsine of a real number	1	487.5
COS		Calculate the cosine of a real number	1	150
ACOS		Calculate the arccosine of a real number	1	495
TAN		Calculate the tangent of a real number	1	202.5
ATAN		Calculate the arctangent of a real number	1	142.5

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

3.16 Adding Constants

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

Instruction	Address	Description	Length in words	Execution time in ns
+	i8	Add an 8bit integer constant	1	7.5
+	i16	Add a 16bit integer constant	2	7.5
+	i32	Add a 32bit integer constant	3	7.5

3.17 Adding Using Address Registers

Adding a 16bit 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	Description	Length in words	Execution time in ns
+AR1		Add the contents of ACCU1L to those of AR1	1	7.5
+AR1	m (0 to 4095)	Add a pointer constant to the contents of AR1	2	7.5
+AR2		Add the contents of ACCU1L to those of AR2	1	7.5
+AR2	m (0 to 4095)	Add a pointer constant to the contents of AR2	2	7.5

3.18 Comparison Instructions with Integers (16/32 bit) or with 32-bit real numbers

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	Description	Length in words	Execution time in ns
==I		ACCU2-L=ACCU1-L	1	7.5
<>I		ACCU2-L≠ACCU1-L	1	7.5
<I		ACCU2-L<ACCU1-L	1	7.5
<=I		ACCU2-L<=ACCU1-L	1	7.5
>I		ACCU2-L>ACCU1-L	1	7.5
>=I		ACCU2-L>=ACCU1-L	1	7.5

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

Comparison Instructions (32-bit Integers)

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

Instruction	Address	Description	Length in words	Execution time in ns
==D		ACCU2=ACCU1	1	7.5
<>D		ACCU2≠ACCU1	1	7.5
<D		ACCU2<ACCU1	1	7.5
<=D		ACCU2<=ACCU1	1	7.5
>D		ACCU2>ACCU1	1	7.5
>=D		ACCU2>=ACCU1	1	7.5

Status word for: ==I, < >I, <I, <=I, >I, >=I,	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	-	-
Instruction affects:	-	Yes	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	Description	Length in words	Execution time in ns
==R		ACCU2=ACCU1	1	7.5
<>R		ACCU2≠ACCU1	1	7.5
<R		ACCU2<ACCU1	1	7.5
<=R		ACCU2<=ACCU1	1	7.5
>R		ACCU2>ACCU1	1	7.5
>=R		ACCU2>=ACCU1	1	7.5

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

3.19 Shift Instructions

Shifting the contents of ACCU1 and ACCU1L to the left or right by the specified number of places. If no address is specified, the contents of ACCU2-LL are used as the number of places. The last bit shifted is loaded into condition code bit CC 1.

Instruction	Address	Description	Length in words	Execution time in ns
SLW [1)]		Shift the contents of ACCU1L to the left. Positions that become free are provided with zeros.	1	7.5
SLW	0 ... 15			
SLD		Shift the contents of ACCU1 to the left. Positions that become free are provided with zeros.	1	7.5
SLD	0 ... 32			
SRW [1)]		Shift the contents of ACCU1L to the right. Positions that become free are provided with zeros.	1	7.5
SRW	0 ... 15			

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

[1)] Number of places shifted: 0 to 16

Instruction	Address	Description	Length in words	Execution time in ns
SRD		Shift the contents of ACCU1 to the right. Positions that become free are provided with zeros.	1	7.5
SRD	0 ... 32			
SSI [1]		Shift the contents of ACCU1L with sign to the right. Positions that become free are provided with the sign (bit 15).	1	7.5
SSI	0 ... 15			
SSD		Shift the contents of ACCU1 with sign to the right. Positions that become free are provided with the sign (bit 31).	1	7.5
SSD	0 ... 32			

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

[1] Number of places shifted: 0 to 16

3.20 Rotate Instructions

Rotate the contents of ACCU1 to the left or right by the specified number of places. If no address is specified, the contents of ACCU2-LL are used as the number of places. The last bit shifted is loaded into condition code bit CC 1.

Instruction	Address	Description	Length in words	Execution time in ns
RLD		Rotate the contents of ACCU1 to the left	1	7.5
RLD	0 ... 32			
RRD		Rotate the contents of ACCU1 to the right	1	7.5
RRD	0 ... 32			
RLDA		Rotate the contents of ACCU1 one bit position to the left via condition code bit CC1	1	7.5
RRDA		Rotate the contents of ACCU1 one bit position to the right via condition code bit CC1	1	7.5

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

3.21 ACCU-transfer instructions, incrementing and decrementing

The status word is not affected.

Instruction	Address	Description	Length in words	Execution time in ns
CAW		Reverse the order of the bytes in ACCU1-L.	1	7.5
CAD		Reverse the order of the bytes in ACCU1.	1	7.5
TAK		Swap the contents of ACCU1 and ACCU2.	1	7.5
ENT		The contents of ACCU2 and ACCU3 are transferred to ACCU3 and ACCU4.	1	7.5
LEAVE		The contents of ACCU3 and ACCU4 are transferred to ACCU2 and ACCU3.	1	7.5
PUSH		The contents of ACCU1, ACCU2, and ACCU3 are transferred to ACCU2, ACCU3, and ACCU4.	1	7.5
POP		The contents of ACCU2, ACCU3, and ACCU4 are transferred to ACCU1, ACCU2, and ACCU3.	1	7.5
INC	k8	Increment ACCU1LL	1	7.5
DEC	k8	Decrement ACCU1LL	1	7.5

3.22 Program Display and Null Operation Instructions

The status word is not affected.

Instruction	Address	Description	Length in words	Execution time in ns
BLD	k8	Program display instruction: Is treated by the CPU as a null operation instruction.	1	3.75
NOP	0 1	Null operation	1	3.75

3.23 Data Type Conversion Instructions

The results of the conversion are in ACCU1.

Instruction	Address	Description	Length in words	Execution time in ns
BTI		Convert contents of ACCU1-L from BCD (0 to +/- 999) to integer (16 bits) (BCD To Int)	1	7.5
BTD		Convert contents of ACCU1 from BCD (0 to +/- 9 999 999) to double integer (32 bits) (BCD To Doubling)	1	7.5
DTR		Convert contents of ACCU1 from double integer (32 bits) to real number (32 bits) (Doubleint To Real)	1	7.5
ITD		Convert contents of ACCU1 from integer (16 bits) to double integer (32 bits) (Int To Doubleint)	1	7.5

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

Instruction	Address	Description	Length in words	Execution time in ns
ITB		Convert contents of ACCU1-L from integer (16 bits) to BCD (0 to +/- 999) (Int To BCD)	1	7.5
DTB		Convert contents of ACCU1 from double integer (32 bits) to BCD 0 to +/- 9 999 999) (Doubleint To BCD)	1	7.5

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

Instruction	Address	Description	Length in words	Execution time in ns
RND		Convert a real number into a 32-bit integer.	1	7.5
RND-		Convert a real number into a 32-bit integer. The number is rounded down to the next whole number.	1	7.5
RND+		Convert a real number into a 32-bit integer. The number is rounded up to the next whole number.	1	7.5
TRUNC		Convert a real number into a 32-bit integer. The places after the decimal point are truncated.	1	7.5

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

3.24 Forming the Ones and Twos Complements

Instruction	Address	Description	Length in words	Execution time in ns
INVI		Form the ones complement of ACCU1L	1	7.5
INVD		Form the ones complement of ACCU1	1	7.5

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

Instruction	Address	Description	Length in words	Execution time in ns
NEGI		Form the twos complement of ACCU1L (integer)	1	7.5
NEGD		Form the twos complement of ACCU1 (double integer)	1	7.5

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

3.25 Block Call Instructions

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

Instruction	Address	Description	Length in words	Execution time in ns
CALL	FB q, DB q	Unconditional call of an FB, with parameter passing	15/17 1)	292.5 2)
CALL	SFB q, DB q	Unconditional call of an SFB, with parameter passing	16/17 1)	292.5 2)
CALL	FC q	Unconditional call of a function, with parameter passing	7/8 1)	255 2)
CALL	SFC q	Unconditional call of an SFC, with parameter passing	8	255 2)

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

1) Die instruction length depends on the block number (0...255 or more).
2) Plus time required for supplying parameters

Instruction	Address	Description	Length in words	Execution time in ns
UC	FB q FC q	Unconditional call of blocks without parameter transfer	1 1)/2	165 165
	FB [e]	Memory-indirect FB call	2	165+
	FC [e]	Memory-indirect FC call	2	165+
	Parameter	FB/FC call via parameter	2	165+
CC	FB q FC q	Conditional call of blocks without parameter transfer	1 1)/2	180/37.5 2) 180/37.5 2)
	FB [e]	Memory-indirect FB call	2	180 +/37.5 + 2)
	FC [e]	Memory-indirect FC call	2	180 +/37.5 + 2)
	Parameter	FB/FC call via parameter	2	180 +/37.5 + 2)

Status word for: UC, CC,	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Instruction depends on:	-	-	-	-	-	-	-	- 3)	-
Instruction affects:	-	-	-	-	0	0	1	- 3)	0

1) With direct addressing, block no. 0 to 255
+ Plus time for loading the address (see Execution Times with Indirect Addressing (Page 15))
2) If call is not executed
3) CC instruction: Depending on RLO, sets RLO = 1

Instruction	Address	Description	Length in words	Execution time in ns	
				1st opening	2nd - n-th opening [1]
OPN		Open a data block			
	DB q DI q	Direct data block Direct instance DB	1[2]/2	30	7.5
	DB [e] DI [e]	Data block, memory-indirect Bit memory address area M Local data area L Data block DB/DI	2	30 30 45	15 15 22.5
	Param.	Data block via parameter	2		22.5

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

[1] If the same DB or DI is already open
[2] Direct data block, DB no. 1 to 255

See also

System Functions (SFC) (Page 63)
System Function Blocks (SFB) (Page 74)

3.26 Block End Instructions

Instruction	Address	Description	Length in words	Execution time in ns
BE		End block	1	142.5
BEU		End block absolute	1	142.5

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

Instruction	Address	Description	Length in words	Execution time in ns
BEC		End block conditionally if RLO="1"		157.6 30 [1]

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

[1] If jump is not executed

3.27 Exchange Data Blocks

Exchanging the two current data blocks. The current data block becomes the current instance data block and vice versa. The status word is not affected.

Instruction	Address	Description	Length in words	Execution time in ns
CDB		Exchange data blocks	1	15

Instruction	Address	Description	Length in words	Execution time in ns
JU	LABEL	Jump unconditionally	2	52.5

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

Instruction	Address	Description	Length in words	Execution time in ns
JC	LABEL	Jump if RLO="1"	2	52.5/7.5 [1]
JCN	LABEL	Jump if RLO="0"	2	52.5/7.5 [1]

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

Instruction	Address	Description	Length in words	Execution time in ns
JCB	LABEL	Jump if RLO="1"; Save the RLO in the BR bit	2	52.5/7.5 [1]
JNB	LABEL	Jump if RLO="0"; Save the RLO in the BR bit	2	52.5/7.5 [1]

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

[1] If jump is not executed

Instruction	Address	Description	Length in words	Execution time in ns
JBI	LABEL	Jump if BR="1"	2	52.5/7.5 [1]
JNBI	LABEL	Jump if BR="0"	2	52.5/7.5 [1]

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

[1] If jump is not executed

Instruction	Address	Description	Length in words	Execution time in ns
JO	LABEL	Jump on stored overflow (OV="1")	2	52.5/7.5 [1]

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

Instruction	Address	Description	Length in words	Execution time in ns
JOS	LABEL	Jump on stored overflow (OS="1")	2	52.5/7.5[1]

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

[1] If jump is not executed

Instruction	Address	Description	Length in words	Execution time in ns
JUO	LABEL	Jump if "unordered math instruction" (CC1=1 and CC0=1)	2	52.5/7.5[1]
JZ	LABEL	Jump if result=0 (CC1=0 and CC0=0)	2	52.5/7.5[1]
JP	LABEL	Jump if result>0 (CC1=1 and CC0=0)	2	52.5/7.5[1]
JM	LABEL	Jump if result<0 (CC1=0 and CC0=1)	2	52.5/7.5[1]
JN	LABEL	Jump if result ≠ 0 (CC1=1 and CC0=0) or (CC1=0 and CC0=1)	2	52.5/7.5[1]
JMZ	LABEL	Jump if result ≤0 (CC1=0 and CC0=1) or (CC1=0 and CC0=0)	2	52.5/7.5[1]
JPZ	LABEL	Jump if result ≥0 (CC1=1 and CC0=0) or (CC1=0 and CC0=0)	2	52.5/7.5[1]

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

[1] If jump is not executed

Instruction	Address	Description	Length in words	Execution time in ns
JL	LABEL	Jump distributor The instruction is followed by a list of jump instructions. The address is a jump label to the subsequent instruction in the 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	52.5
LOOP	LABEL	Decrement ACCU1-L and jump if ACCU1-L ≠ 00 (loop programming)	2	45/7.5 [1]

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

[1] If jump is not executed

3.28 Instructions for the Master Control Relay (MCR)

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

Instruction	Address	Description	Length in words	Execution time in ns
MCR(		Open an MCR zone. Save the RLO to the MCR stack.	1	7.5

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

Instruction	Address	Description	Length in words	Execution time in ns
)MCR		Close an MCR zone. Remove an entry from the MCR stack.	1	7.5

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

Instruction	Address	Description	Length in words	Execution time in ns
MCRA		Activate the MCR	1	7.5
MCRD		Deactivate the MCR	1	7.5

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

3.29 Organization Blocks (OB)

A user program for an S7-400 is made up of blocks containing the instructions, parameters, and data for the respective CPU. You will find a detailed description of the OBs and their use in the manual "Programming with STEP 7 V5.5".

Organization blocks	CPU 410-5H Process Automation	Start events (hexadecimal value)
Free cycle:		
OB 1	x	1101, 1103, 1104, 1105
Time-of-day interrupts:		
OB 10	x	1111
OB 11	x	1112
OB 12	x	1113
OB 13	x	1114
OB 14	x	1115
OB 15	x	1116
OB 16	x	1117
OB 17	x	1118

Organization blocks	CPU 410-5H Process Automation	Start events (hexadecimal value)
Time-delay interrupts:		
OB 20	x	1121
OB 21	x	1122
OB 22	x	1123
OB 23	x	1124
Cyclic interrupts: [1]		
OB 30	x	1131
OB 31	x	1132
OB 32	x	1133
OB 33	x	1134
OB 34	x	1135
OB 35	x	1136
OB 36	x	1137
OB 37	x	1138
OB 38	x	1139
Hardware interrupts:		
OB 40	x	1141
OB 41	x	1141
OB 42	x	1141
OB 43	x	1141
OB 44	x	1141
OB 45	x	1141
OB 46	x	1141
OB 47	x	1141
Interrupt OBs for DPV1:		
OB 55	x	1155
OB 56	x	1156
OB 57	x	1157
Redundancy error interrupts:		
OB 70	x	73A2, 73A3, 72A3
OB 72	x	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
Asynchronous error interrupts:		
OB 80	x	3501, 3502, 3505, 3506, 3507, 3508, 3509, 350A
OB 81	x	3821, 3822, 3823, 3825, 3826, 3827, 3831, 3832, 3833, 3921, 3922, 3923, 3925, 3926, 3927, 3931, 3932, 3933
OB 82	x	3842, 3942

Organization blocks	CPU 410-5H Process Automation	Start events (hexadecimal value)
OB 83	x	3951, 3954, 3854, 3855, 3856, 3858, 3861, 3961, 3863, 3864, 3865, 3866, 3966, 3267, 3367, 3968
OB 84	x	3582, 3583, 3986, 3587
OB 85	x	35A1, 35A2, 35A3, 34A4, 35A4, 39B1, 39B2, 38B3, 39B3, 38B4, 39B4
OB 86	x	38C1, 39C1, 38C2, 39C3, 38C4, 39C4, 38C5, 39C5, 38C6, 38C7, 38C8, 39CA, 38CB, 39CB, 38CC, 39CD, 39CE
OB 87	x	35D2, 35D3, 35D4, 35D5, 35E5
OB 88	x	3573, 3575, 3576
Restart (warm restart):		
OB 100	x	1381, 1382, 138A, 138B
Cold restart:		
OB 102	x	1385, 1386, 1387, 1388
Synchronous error interrupts:		
OB 121	x	2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 253A, 253C, 253D, 253E, 253F
OB 122	x	2942, 2943, 2944, 2945

[1] Additional start event of the H-CPUs for OB 30 to OB 38: 1130H

3.30 Function Blocks (FBs)

The following table lists the quantity, number, and maximum size of the function blocks that you can create for the CPU 410-5H PN/DP.

Function blocks	CPU 410-5H Process Automation
Number	8000
Permissible numbers	0 to 7999
Maximum size (runtime-relevant code)	64 KB

3.31 Functions (FC)

The following table lists the quantity, number, and maximum size of the functions that you can create for the CPU 410-5H PN/DP.

Functions	CPU 410-5H Process Automation
Number	8000
Permissible numbers	0 to 7999
Maximum size (runtime-relevant code)	64 KB

3.32 Data blocks (DB)

The following table lists the quantity, number, and maximum size of the data blocks that you can create for the CPU 410-5H PN/DP.

Data blocks	CPU 410-5H Process Automation
Number	16000
Permissible numbers	1 to 16000
Maximum size (runtime-relevant code)	64 KB

3.33 System Functions (SFC)

The following tables show the system functions provided by the operating system of the CPU 410-5H PN/DP and the execution times on the respective CPU.

SFC no.	SFC name	Description	Execution time in µs	
			CPU 410-5H Process Automation solo	CPU 410-5H Process Automation redundant
0	SET_CLK	Set time of day	38	96
1	READ_CLK	Read time of day	3	14
2	SET_RTM	Set operating hours counter	2	2
3	CTRL_RTM	Starts/stops operating hours counter	2	2
4	READ_RTM	Read operating hours counter	2	10
5	GADR_LGC	Find logical address of a channel Rack 0	3	3
		Internal DP	4	4
6	RD_SINFO	Read start information of current OB	3	3
9	EN_MSG	Enable block-related, symbol-related, and group status messages. First call, REQ = 1	11	29
		Last call	3	13
10	DIS_MSG	Disable block-related, symbol-related, and group status messages. First call, REQ = 1	11	29
		Last call	3	13
13	DPNRM_DG	Reading diagnostics data of a DP slave First call	20	31
		Intermediate call	7	8
		Last call (28 bytes)	9	13
14	DPRD_DAT	Read consistent user data via integrated DP interface, 3 bytes	8	19
		Via integrated DP interface, 32 bytes	8	24
		Via external DP interface, 3 bytes	16	25
		Via external DP interface, 32 bytes	36	42
		Via integrated PROFINET interface, 8 bytes	9	19
		Via integrated PROFINET interface, 32 bytes	9	24

SFC no.	SFC name	Description	Execution time in µs	
			CPU 410-5H Process Automation solo	CPU 410-5H Process Automation redundant
15	DPWR_DAT	Write consistent user data via integrated DP interface, 3 bytes	10	16
		Via integrated DP interface, 32 bytes	11	19
		Via external DP interface, 3 bytes	14	26
		Via external DP interface, 32 bytes	42	54
		Write consistent user data via integrated PROFINET interface, 8 bytes	12	18
		Via integrated PROFINET interface, 32 bytes	13	22
17	ALARM_SQ 1)	Generate blockrelated messages that can be acknowledged. First call, SIG = 0 -> 1	53 - 120	89 - 167
		Empty call	37 - 99	57 - 111
18	ALARM_S 1)	Generate blockrelated messages that cannot be acknowledged. First call, SIG = 0 -> 1	29 - 125	88 - 168
		Empty call	10 - 98	32 - 112
19	ALARM_SC 1)	Acknowledgment state of the last ALARM_SC incoming message	9 - 100	28 - 112
1) For 1 to 200 assigned system resources				
20	BLKMOV	Copy variable within work memory (n = number of bytes to be copied)	4 + n * 0.016	4 + n * 0.016
		Source = Load memory	5 + n * 0.016	5 + n * 0.016
21	FILL	Prefill field within work memory (n = length of target variable in bytes)	3 + n * 0.01	3 + n * 0.01
22	CREAT_DB	Create data block	13 + n * 0.02	51 + n * 0.02
		Save last free DB no. from field of 100 DBs	89	125
23	DEL_DB	Delete data block	17	117
24	TEST_DB	Test data block	6	44
25	COMPRESS	Compress user memory First call (trigger)	11	30
		Subsequent call	2	2
26	UPDAT_PI	Update process image input table (runtime entry for 1 DI 32 in central controller)	9	12
		AI 8 * 13 Bit	29	51
27	UPDAT_PO	Update outputs (runtime entry for 1 DO 32 in central controller)	9	13
		AO 8 * 13 Bit	26	49
28	SET_TINT	Set time-of-day interrupt	6	15
29	CAN_TINT	Cancel time-of-day interrupt	15	92
30	ACT_TINT	Activate time-of-day Interrupt	5	13

SFC no.	SFC name	Description	Execution time in µs	
			CPU 410-5H Process Automation solo	CPU 410-5H Process Automation redundant
31	QRY_TINT	Query time-of-day interrupt	1	1
32	SRT_DINT	Start time-delay interrupt	3	3
33	CAN_DINT	Cancel time-delay interrupt	3	3
34	QRY_DINT	Query time-delay interrupt	1	1
36	MSK_FLT	Mask synchronous error events	2	2
37	DMSK_FLT	Unmask synchronous error events	2	2
38	READ_ERR	Read event status register	2	2
39	DIS_IRT	Discard new events Block all events (MODE = 0)	16	16
		Block all events of an alarm class (MODE = 1)	4	4
		Block one event (MODE = 2)	2	2
40	EN_IRT	Stop discarding events Enable all events (MODE = 0)	14	14
		Enable all events of an alarm class (MODE = 1)	4	4
		Enable one event (MODE = 2)	2	2
41	DIS_AIRT	Delay interrupt events the first time the delay is activated [2]	12	12
		If the delay is already activated	1	1
42	EN_AIRT	If other delays are present	2	2
		Stop delaying interrupt events when last delay is canceled [3]	28	28
[2] When the delay is activated for the first time, the runtime of SFC 41 depends on the priority class in which SFC 41 is called. The specified runtime refers to the call in OB 1. It decreases as the priority class number increases. [3] When the delay is activated for the first time, the runtime of SFC 42 depends on the priority class in which SFC 42 is called. The specified runtime refers to the call in OB 1. It decreases as the priority class number increases.				
43	RE_TRIGR	Re-trigger cycle time monitoring	14	82
44	REPL_VAL	Transfer substitute value to ACCU 1	2	2
46	STP	Set CPU to STOP cannot be measured	-	-
47	WAIT	Delay program execution in addition to waiting time	2	2
48	SNC_RTCB	Synchronize slave clocks	2	12
49	LGC_GADR	Find slot belonging to a logical address (central and PROFIBUS DP)	4	4
50	RD_LGADR	Find all logical addresses of a module (runtime entry for 1 DI 32 in central controller)	8	8
51	RDSYSST	"Module identification" partial list Read one data record (0111)	9	44

SFC no.	SFC name	Description	Execution time in µs	
			CPU 410-5H Process Automation solo	CPU 410-5H Process Automation redundant
51	RDSYSST	"CPU characteristics" partial list Read all data records (0012)	16	16
		Read one data record (0112)	10	10
		Read header information (0F12)	7	7
51	RDSYSST	"Save" partial list Read one data record (0113)	9	9
51	RDSYSST	"System areas" partial list Read all data records (0014)	9	9
		Read header information (0F14)	7	7
51	RDSYSST	"Block types" partial list Read all data records (0015)	9	9
51	RDSYSST	"Status of module LEDs" partial list Read the status of all LEDs (0019)	23	-
		Read header information (0F19)	10	-
51	RDSYSST	"Component identification" partial list Read all components (001C)	18	107
		Read one component (011C)	12	102
		Read all components of a CPU of the H-system (021C)	18	107
		Read one component of all redundant CPUs of the H-system (031C)	11	101
		Read header information (0F1C)	9	98
51	RDSYSST	"Alarm status" partial list Read one data record (0222)	11	11
51	RDSYSST	"PIP/CPU assignment" partial list Assignment between all process image partitions and OBs (0025)	22	22
		Assignment between a process image partition and the corresponding OB (0125)	8	8
51	RDSYSST	"Communication "status information" partial list Read status information of a communication unit (0132)	11 - 17	18 - 19
		"Communication "status information" partial list Read status information of a communication unit (0232)	16	52
51	RDSYSST	"H-CPU group information" partial list Current status of the H system (0071)	-	16
		Read header information (0F71)	-	7
51	RDSYSST	"Module LEDs" partial list Status of an LED (0174)	11	19

SFC no.	SFC name	Description	Execution time in µs	
			CPU 410-5H Process Automation solo	CPU 410-5H Process Automation redundant
51	RDSYSST	"Connected DP slaves/IO devices in the H System" partial list Group information (0075)	-	602
		Communication status between the H system and connected DP slave/IO device (0C75)	-	51
		Read header information (0F75)	-	540
51	RDSYSST	"DP master system information" partial list All DP master systems known to the CPU (0090)	25	25
		One DP master system (0190)	9	9
		Read header information (0F90)	7	7
51	RDSYSST	"Module status information" partial list All submodules of the host module (0591)	13	91
		A module in a central configuration via the logical start address (0C91)	15	35
		A distributed module connected to an integrated DP interface via the logical start address (0C91)	17	39
		A distributed module connected to an PROFINET interface via the logical start address (0C91)	15	35
		All PROFINET IO systems (0A91)	21	39
51	RDSYSST	"Module status information" partial list of a distributed module connected to an external DP interface via the logical start address (4C91) First call	25	39
		Intermediate call	16	16
		Last call	17	17
51	RDSYSST	"Module status information" partial list of all modules in the specified rack (n=DR number) (0D91)	18 + n * 10	50 + n * 12
		All modules in the specified DP station (0D91)	15 - 19	46 - 53
		All modules in the specified PNIO station (0D91)	34	76
51	RDSYSST	"Rack/station status information" partial list Centralized configuration Read the expected status of rack 0 (0092)	9	17
		Distributed configuration Read the expected status of DP system 1 (0092)	39	50
51	RDSYSST	Read the expected status of DP system 1 (via external DP interface) (4092)	16	33
51	RDSYSST	Read the activation status of DP master system 1 (via integrated DP interface) (0192)	61	67

SFC no.	SFC name	Description	Execution time in µs	
			CPU 410-5H Process Automation solo	CPU 410-5H Process Automation redundant
51	RDSYSST	Centralized configuration Read the actual status of rack 0 (0292)	9	17
		Distributed configuration Read the actual status of DP system 1 (0292)	41	54
51	RDSYSST	Read the actual status of the stations of a DP master system (via external DP interface) (4292)	16	33
51	RDSYSST	Read the status of the battery buffer of rack 0 if at least one battery has failed (0392)	9	17
51	RDSYSST	Read the status of the entire battery buffer of a CPU (0492)	9	17
51	RDSYSST	Read the status of the 24 V supply of all racks of a CPU (0592)	9	17
51	RDSYSST	Centralized configuration Read the diagnostics status of the expansion devices (0692)	18	26
51	RDSYSST	Distributed configuration Read the diagnostics status of the stations of DP system 1 (via integrated DP interface) (0692)	48	58
51	RDSYSST	Diagnostics status of the stations of a DP master system that is connected via an external DP interface (4692) First call	16	33
		Intermediate call	9	9
		Last call	11	11

SFC no.	SFC name	Description	Execution time in µs	
			CPU 410-5H Process Automation solo	CPU 410-5H Process Automation redundant
51	RDSYSST	"Rack/station status information" partial list Expected status of the central rack (0094)	15	93
		Expected status of the stations of a DP station connected to an integrated interface (0094)	36	107
		Expected status of the stations of an IO controller system connected to an integrated interface (0094)	36	106
		Actual status of central rack (0294)	14	94
		Actual status of the stations of a DP station connected to an integrated interface (0294)	36 + n * 0.5	108 + n * 1.5
		Actual status of the stations of an IO controller system connected to an integrated interface (0294): (n = number of stations)	36 + n * 0.5	104 + n * 1.5
		Diagnostics status of the central rack (0694)	23	100
		Diagnostics status of the stations of a DP station connected to an integrated interface (0694)	47 + n * 1	119 + n * 3
		Diagnostics status of the stations of an IO controller system connected to an integrated interface (0694): (n = number of stations)	47 + n * 1	96 + n * 3
51	RDSYSST	Maintenance status of the central rack (0794)	38	107
		Maintenance status of the stations of a DP station connected to an integrated interface (0794)	47 + n * 1	119 + n * 3
		Maintenance status of the stations of an IO controller system connected to an integrated interface (0794) (n = number of stations)	47 + n * 1	96 + n * 3
		Read header information (0F94) (central, DP station, and PROFINET IO)	11	11

SFC no.	SFC name	Description	Execution time in µs	
			CPU 410-5H Process Automation solo	CPU 410-5H Process Automation redundant
51	RDSYSST	"Extended DP master system / PROFINET IO system information" partial list Read extended status information about a DP master system connected to an integrated or external interface and about a PROFINET IO system connected to an integrated interface (0195)	10	10
		Read header information (0F95)	7	7
51	RDSYSST	"Module status information of all submodules of a specified module" partial list for PROFINET IO connected to an integrated interface (0696)	16	36
		Module status information of a central module (0C96)	556	563
		PROFIBUS DP via an integrated interface (0C96)	19	39
		PROFINET IO via an integrated interface (0C96)	16	36
51	RDSYSST	"Diagnostics buffer" partial list Read all event information that can be supplied in the current operating mode (max. 21) (00A0)	22	22
		Read the n latest entries (n = 1-23) (01A0)	9 + n * 0.5	9 + n * 0.5
		Read header information (0FA0)	8	8
51	RDSYSST	"Diagnostics data DR* 0" partial list Read via logical start address (00B1) Centralized configuration	45	59
		Distributed configuration (00B1) First call	23	31
		Distributed configuration (00B1) Intermediate call, REQ = 0	12	12
		Distributed configuration (00B1) Last call	13	13
51	RDSYSST	"Diagnostics data DR 1" partial list Read via physical address (00B2) Read a 16-byte long DR 1	41	58

SFC no.	SFC name	Description	Execution time in µs	
			CPU 410-5H Process Automation solo	CPU 410-5H Process Automation redundant
51	RDSYSST	"Diagnostics data DR* 1" partial list Read via logical start address (00B3) Read a 16-byte long DR 1 Centralized configuration	80	95
		Distributed configuration, first call (00B3)	23	32
		Distributed configuration, intermediate call (00B3)	12	12
		Distributed configuration, last call (00B3)	14	14
51	RDSYSST	"Diagnostics data DP slave" partial list Read via configured diagnostics address (00B4) First call	23	31
		Intermediate call, REQ = 0 (00B4)	11	11
		Last call (6 - 240 bytes) (00B4)	19	19
52	WR_USMSG	Write user entry in diagnostics buffer with message	9	17
		Without message	9	19
54	RD_DPARM	Read dynamic parameters, centralized configuration, AI 8 * 13 bits	11	16
		Distributed configuration, AI 8 * 12 bits (DR1 = 14 bytes)	13	18
55	WR_PARM	Write dynamic parameters centralized configuration AI 8 * 13 bits	63	73
		Distributed configuration First call AI 8 * 12 bits (14 - 240 bytes)	22	31
		Distributed configuration Follow-up/last call, REQ = 0	10	10
56	WR_DPARM	Write predefined dynamic parameters, AI 8 * 13 bits centralized configuration	100	110
		Distributed configuration, first call, AI 8 * 12 bits (2 - 240 bytes)	19	27
		Follow-up/last call	9	9
57	PARM_MOD	Assign module parameters, centralized configuration Module/DR count/DR lengths in bytes AI 8 * 13 bits	175	192
		Distributed configuration, AO 8 * 12 bits First call (16 - 240 bytes)	19	27
		Distributed configuration, follow-up/last call	9	9

SFC no.	SFC name	Description	Execution time in µs	
			CPU 410-5H Process Automation solo	CPU 410-5H Process Automation redundant
58	WR_REC	Write parameter data record centralized configuration (n = number of bytes)	40 + n * 2.2	69 + n * 2.2
		First call, integrated DP interface (n = number of bytes)	20 + n * 0.04	32 + n * 0.04
		Intermediate call, REQ = 0, integrated DP interface	8	8
		Last call, integrated DP interface	8	8
		First call, external DP interface (n = number of bytes)	20 + n * 0.03	32 + n * 0.03
		Intermediate call, REQ = 0, external DP interface	8	8
		Last call, external DP interface	8	8
59	RD_REC	Read data record first call, centralized configuration (n = number of bytes)	40 + n * 2.3	72 + n * 2.4
		First call, integrated DP interface module	20	32
		Intermediate call, REQ = 0, integrated DP interface module	8	8
		Last call, integrated DP interface (n = number of bytes)	16 + n * 0.13	17 + n * 0.13
		First call, external DP interface	20	32
		Intermediate call, REQ = 0, external DP interface	8	8
		Last call, external DP interface (n = number of bytes)	16 + n * 0.02	17 + n * 0.02
62	CONTROL	Query the status of the connection belonging to a local communication SFB instance	8	22
64	TIME_TCK	Read out millisecond timer	2	9
70	GEO_LOG	Determine the start address of a module from its slot	4	4
71	LOG_GEO	Determine the slot that belongs to a logical address	4	4
78	OB_RT	Determine the OB program runtime	4	14
79	SET	Set bit array in the I/O area, n = number of bits to be set to 1	5 + n * 0.4	10 + n * 1.3
80	RSET	Delete bit array in the I/O area n = number of bits to be set to 0	5 + n * 0.4	10 + n * 1.3
81	UBLKMOV	Copy variable without interruption, n = number of bytes to be copied	3 + n * 0.02	3 + n * 0.02
87	C_DIAG	Determine current connection status MODE = 0	3	11
		Mode = 1, 2, 3	75	158

SFC no.	SFC name	Description	Execution time in µs	
			CPU 410-5H Process Automation solo	CPU 410-5H Process Automation redundant
90	H_CTRL	Influence sequences of H systems	2	2
100	SET_CLKS	Set time-of-day and TOD status MODE = 1	37	155
		MODE = 2	20	94
		MODE = 3	31	99
101	RTM	Handle operating hours counter, Mode = 0 Read	4	11
		Mode = 1, 2 Start/Stop	4	11
		Mode = 4, 5, 6 Set	4	12
103	DP_TOPOL	Determine the bus topology in a DP master system, first call, REQ = 1	18	48
		Intermediate call	3	3
		Last call, BUSY = 0	4	4
104	CIR	Control CiR, MODE = 0, Information	3	-
		MODE = 1, Enable CiR	3	-
		MODE = 2, Disable CiR completely	3	-
		MODE = 3, Disable CiR conditionally	3	-
105	READ_SI	Read dynamically assigned system resources, MODE = 0	10 - 260 1)	236 – 257 1)
		MODE = 1	12 - 413 2)	232 - 636 2)
		MODE = 2	12 - 1246 2)	233 - 1462 2)
		MODE = 3	12 - 559 3)	233 - 776 3)
106	DEL_SI	Enable dynamically assigned system resources, MODE = 1 2)	16 - 2999 1)	491 - 25256 1)
		MODE = 2 2)	16 - 890 1)	495 - 1381 1)
		MODE = 3 3)	16 - 2999 2)	492 - 25153 2)
1) Depending on the size of the SYS_INST target area and the number of system resources still to be read 2) Depending on the number of active messages (assigned system resources) 3) Depending on the number of active messages (assigned system resources) and the number of assigned instances with the searched CMP_ID.				
107	ALARM_DQ	Generate block-related messages that can be acknowledged, first call, SIG = 0 -> 1	26 - 125 1)	60 - 168 1)
		Empty call	9 - 99 1)	30 - 115 1)
108	ALARM_D	Generate block-related messages that cannot be acknowledged, first call, SIG = 0 -> 1	33 - 127 1)	69 - 165 1)
		Empty call	9 - 102 1)	25 - 112 1)

SFC no.	SFC name	Description	Execution time in µs	
			CPU 410-5H Process Automation solo	CPU 410-5H Process Automation redundant
109	PROTECT	Activate write protection	1	1
1) For 1 to 200 assigned system resources				

3.34 System Function Blocks (SFB)

The following table lists the system function blocks provided by the operating system of the CPU 410-5H PN/DP and the execution times on the respective CPU.

SFB no.	SFB name	Description	Execution time in µs	
			CPU 410-5H Process Automation solo	CPU 410-5H Process Automation redundant
0	CTU	Count up	1	1
1	CTD	Count down	1	1
2	CTUD	Count up and down	1	1
3	TP	Generate pulse	2	13
4	TON	Generate ON delay	2	13
5	TOF	Generate OFF delay	1	5
8	USEND	Send data without coordination (one send parameter supplied) JOB activation (1-440 bytes)	30	64
		JOB checked	9	24
		JOB finished, DONE = 1	9	25
9	URCV	Receive data without coordination (one receive parameter supplied) JOB activation	8	24
		JOB checked	8	23
		JOB finished (NDR = 1; 1-440 bytes)	20	36
12	BSEND	Send data block by block JOB activation (1-3000 bytes)	23	46
		JOB checked	10	25
		JOB finished, DONE = 1	10	25
13	BRCV	Receive data block by block JOB activation (1-3000 bytes)	12	27
		JOB checked	11	26
		JOB finished	10	26

SFB no.	SFB name	Description	Execution time in µs	
			CPU 410-5H Process Automation solo	CPU 410-5H Process Automation redundant
14	GET	Read data from remote CPU (one area specified) JOB activation	20	44
		JOB checked	9	25
		JOB finished (NDR = 1 (1-450 bytes)	20	36
15	PUT	Write data to remote CPU (one area specified) JOB activation (1-404 bytes)	30	65
		JOB checked	9	24
		JOB finished, DONE = 1	9	24
16	PRINT	Send data to a printer Job activation, REQ = 1	32	66
		JOB checked	9	24
		JOB finished, DONE = 1	9	24
19	START	Perform warm restart or cold restart in remote device Job activation, REQ = 1	27	53
		JOB checked	9	25
		JOB finished, DONE = 1	10	29
20	STOP	Set remote device to STOP Job activation, REQ = 1	26	48
		JOB checked	9	25
		JOB finished, DONE = 1	9	25
22	STATUS	Query device status of a remote partner, JOB activation, REQ = 1	16	41
		JOB checked	9	24
		JOB finished, NDR = 1	25	41
23	USTATUS	Receive status of remote device without coordination, JOB activation, NDR = 1	8	24
		JOB checked	8	23
		JOB finished	25	41
31	NOTIFY_8P	Generate block-related message without acknowledgment display First call or JOB activation, SIG = 0-> 1 (1-420 bytes)	37	84
		JOB checked	12	27
		JOB finished, DONE = 1	12	27
32	DRUM	Implement sequencer	2	18

SFB no.	SFB name	Description	Execution time in µs	
			CPU 410-5H Process Automation solo	CPU 410-5H Process Automation redundant
33	ALARM	Generate block-related message with acknowledgment display First call or JOB activation, SIG = 0 → 1 (1-420 bytes)	37	87
		JOB checked	12	28
		JOB finished, DONE = 1	13	28
34	ALARM_8	Generate block-related message without associated value for 8 signals, first call or JOB activated, SIG = 0-> 1 (1-420 bytes)	26	61
		JOB checked	12	28
		JOB finished, DONE = 1	12	29
35	ALARM_8P	Generate block-related message with associated value for 8 signals, first call or JOB activation, SIG = 0-> 1 (1-420 bytes)	36	86
		JOB checked	12	27
		JOB finished, DONE = 1	12	28
36	NOTIFY	Generate block-related message without acknowledgment display, first call or JOB activation, SIG = 0-> 1 (1-420 bytes)	37	82
		JOB checked	12	27
		JOB finished, DONE = 1	12	27
37	AR_SEND	Send archive data First call or JOB activation, REQ = 1 (1-3000 bytes)	26	60
		JOB checked	10	26
		JOB finished, DONE = 1	10	26
52	RDREC	Read data record from a central module	42	58
52	RDREC	Read data record from a DP slave integrated DP interface, first call (2 - 16 bytes)	24	36
		Intermediate call	8	10
		Last call	9	11
52	RDREC	Read data record from a DP slave external DP interface, first call (4 - 16 bytes)	18	20
		Intermediate call	8	9
		Last call	9	10

SFB no.	SFB name	Description	Execution time in µs	
			CPU 410-5H Process Automation solo	CPU 410-5H Process Automation redundant
52	RDREC	Read data record from an IO device, integrated PROFINET interface, First call (2-16 bytes)	19	28
		Intermediate call	8	9
		Last call	8	9
53	WRREC	Centralized configuration	38	50
53	WRREC	Write data record to a DP slave, integrated DP interface First call (1-10 bytes)	24	35
		Intermediate call	8	10
		Last call	9	10
53	WRREC	Write data record to a DP slave, external DP interface First call (2-14 bytes)	18	24
		Intermediate call	8	9
		Last call	9	10
53	WRREC	Write data record to an IO device, integrated PROFINET interface First call (1-10 bytes)	21	31
		Intermediate call	8	11
		Last call	8	11
54	RALRM	Receive interrupt, runtime measurement for non-I/O-dependent OBs, MODE = 1, OB 1	8	19
54	RALRM	Receive interrupt, runtime measurement for central I/O, MODE = 1, OB 40, OB 82, OB 83, OB 86	19	31
54	RALRM	Receive interrupt, runtime measurement for integrated DP interface, MODE = 1, OB 40, OB 83, OB 86	21	55
		OB 55 to OB 57, OB 82	23	58
		OB 70	22	56
54	RALRM	Receive interrupt, runtime measurement for external DP interface, MODE = 1, OB 40, OB 83, OB 86	68	100
		OB 55 to OB 57, OB 82	107	141
		OB 70	75	110
54	RALRM	Receive interrupt, runtime measurement for integrated PROFINET interface MODE = 1, OB 40, OB 83, OB 86	33	199
		OB 82	35	205
		OB 70	34	202

SFB no.	SFB name	Description	Execution time in µs	
			CPU 410-5H Process Automation solo	CPU 410-5H Process Automation redundant
81	RD_DPAR	Read predefined parameters, central	19	40
		Read predefined parameters, internal DP	19	35
		Read predefined parameters, external DP, First call	24	35
		Last call	24	38
		Read predefined parameters, internal PNIO, First call	26	48
		Intermediate call	28	51
		Last call	10	18

3.35 Function Blocks for Open Communication via Industrial Ethernet

The following tables list the function blocks for open communication via Industrial Ethernet provided by the operating system of the CPU 410-5H Process Automation and the execution times on the respective CPU.

SFB no.	SFB name	Description	Execution time in µs	
			CPU410-5H Process Automation solo	CPU 410-5H Process Automation redundant
63	TSEND	Send data via TCP and ISO on TCP (n bytes) First call	21 + n*0.008	65 + n*0.008
		Intermediate call	9	33
		Last call	9	33
64	TRCV	Receive data via TCP and ISO on TCP (n bytes) First call	19	51 + n*0.008
		Intermediate call	9	33
		Last call	15	51
65	TCON	Establish connection First call	21	49
		Intermediate call	6	39
		Last call	7	49

SFB no.	SFB name	Description	Execution time in µs	
			CPU410-5H Process Automation solo	CPU 410-5H Process Automation redundant
66	TDISCON	Terminate connection First call	13	64
		Intermediate call	6	41
		Last call	7	49
67	TUSEND	Send data via UDP (n bytes) First call	26 + n*0.008	69 + n*0.008
		Intermediate call	9	33
		Last call	9	33
68	TURCV	Receive data via UDP First call	19	51
		Intermediate call	9	33
		Last call	23	59

3.36 IEC Functions

You can use the following IEC functions in STEP 7.

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

FC no.	FC name	Description
DATE_AND_TIME		
3	D_TOD_DT	Combines the data formats DATE and TIME_OF_DAY (TOD) and converts to data format DATE_AND_TIME.
6	DT_DATE	Extracts the DATE data format from the DATE_AND_TIME data format.
7	DT_DAY	Extracts the day of the week from the DATE_AND_TIME data format.
8	DT_TOD	Extracts the TIME_OF_DAY data format from the DATE_AND_TIME data format.
Time formats		
33	S5TI_TIM	Converts S5 TIME data format to TIME data format.
40	TIM_S5TI	Converts TIME data format to S5 TIME data format.
Duration		
1	AD_DT_TM	Add a duration in TIME format to a point in time in DT format; the result is a new point in time in DT format.
35	SB_DT_TM	Subtract a duration in TIME format from a point in time in DT format; the result is a new point in time in DT format.
34	SB_DT_DT	Subtract two points in time in DT format; the result is a duration in TIME format.
Compare DATE_AND_TIME		
9	EQ_DT	Compares the contents of two variables in the DATE_AND_TIME format for equal to.
12	GE_DT	Compares the contents of two variables in the DATE_AND_TIME format for greater than or equal to.
14	GT_DT	Compares the contents of two variables in the DATE_AND_TIME format for greater than.
18	LE_DT	Compares the contents of two variables in the DATE_AND_TIME format for less than or equal to.

FC no.	FC name	Description
23	LT_DT	Compares the contents of two variables in the DATE_AND_TIME format for less than.
28	NE_DT	Compares the contents of two variables in the DATE_AND_TIME format for unequal to.
Compare STRING		
10	EQ_STRNG	Compares the contents of two variables in the STRING format for equal to.
13	GE_STRNG	Compares the contents of two variables in the STRING format for greater than or equal to.
15	GT_STRNG	Compares the contents of two variables in the STRING format for greater than.
19	LE_STRNG	Compares the contents of two variables in the STRING format for less than or equal to.
24	LT_STRNG	Compares the contents of two variables in the STRING format for less than.
29	NE_STRNG	Compares the contents of two variables in the STRING format for unequal to.
Processing STRING variables		
21	LEN	Reads out the actual length of a STRING variable.
20	LEFT	Reads the first L character of a STRING variable.
32	RIGHT	Reads the last L character of a STRING variable.
26	MID	Reads the center L character of a STRING variable. (starting from the defined character).
2	CONCAT	Combines two STRING variables in one STRING variable.
17	INSERT	Inserts a STRING variable into another STRING variable at a defined point.
4	DELETE	Deletes L characters of a STRING variable.
31	REPLACE	Replaces L characters of a STRING variable with a second STRING variable.
11	FIND	Finds the position of the second STRING variable in the first STRING variable.
Format conversions with STRING		
16	I_STRNG	Converts a variable from INTEGER format to STRING format.
5	DI_STRNG	Converts a variable from INTEGER (32bit) format to STRING format.
30	R_STRNG	Converts a variable from REAL format to STRING format.
38	STRNG_I	Converts a variable from STRING format to INTEGER format.
37	STRNG_DI	Converts a variable from STRING format to INTEGER (32bit) format.
39	STRNG_R	Converts a variable from STRING format to REAL format.
Processing of numbers		
22	LIMIT	Limits a number to a defined limit value.
25	MAX	Selects the largest of three numerical variables.
27	MIN	Selects the smallest of three numerical variables.
36	SEL	Selects one of two variables.

SSL partial list

4

SSL ID	Index	Message function
		Module identification
0111_H		Identification data record corresponding to the specified index
	0001_H	CPU type and version number
	0006_H	Identification of basic hardware
	0007_H	Identification of basic firmware
		CPU characteristics
0012_H	–	All characteristics
0112_H		Characteristics of a group
	0000_H	STEP 7 processing
	0100_H	Time system in the CPU
	0200_H	System behavior of the CPU
	0300_H	STEP 7 instruction supply
$0F12_H$	–	Header information only
		User memory areas
0013_H	–	All data records of available user memory areas
0113_H		One data record for the specified memory area
	0001_H	Working memory
		System areas
0014_H	–	Data records of all system areas
$0F14_H$	–	Header information only
		Block types
0015_H	–	Data records of all block types
		Status of the module LEDs
0019_H	–	Read the status of all LEDs
$0F19_H$	–	Header information only
		Component identification
$001C_H$	–	Read all data records
$011C_H$		Data record for specified index
	0001_H	Station name
	0002_H	Name of the module
	0003_H	Plant ID of the module
	0004_H	Copyright entry
	0005_H	Serial number of the module
	0007_H	Module type name
	0008_H	Serial number of the micro memory card
	0009_H	Manufacturer and profile of a CPU module

SSL ID	Index	Message function
	$000A_H$	OEM identifier
	$000B_H$	Location ID
$01FC_H$	–	Header information only
		Alarm status
0222_H		Data record for specified interrupt
	OB no.	Number of the OB (OB1 only)
		Assignment between process image partitions and CPUs (only for CPUs that support synchronous cycle)
0025_H	–	Assignment of all partial process images and OBs
0125_H	TPA no. (number of the partial process image)	Assignment of a partial process image to the corresponding OB
0225_H	OB no.	Assignment of an OB to the corresponding partial process images
$0F25_H$	–	Only SSL partial list header information
		Communication status data
0132_H		Communication status information on the specified communication unit (only one data record)
	0004_H	OMS/contactor
	0005_H	Diagnostics
	0008_H	Time system (TIME)
	$000B_H$	Runtime meter (32 bit) 0 to 7
	$000C_H$	Runtime meter (32 bit) 8 to 15
0232_H		Communication status information on specified communication unit
	0004_H	OMS/contactor
		Status of the module LEDs
0074_H	–	Read the status of all LEDs
0174_H		Read the status of individual LEDs
	0001_H	GE, group error
	0004_H	RUN, RUN LED
	0005_H	STOP, STOP LED
	0006_H	FRCE, Force LED
	$000B_H$	BF1 LED
	$000C_H$	BF2 LED
	0014_H	BF3 LED
	0015_H	MAINT LED
		DP master system information
0090_H	0000_H	Information DP master systems known to the CPU
0190_H	DP master system ID	Information about a DP Master system
$0F90_H$	0000_H	Only SSL partial list header information

SSL ID	Index	Message function
		Module status information
0591_H	–	Module status information for all submodules that a host recognizes
$0A91_H$	–	Module status information of all DP master systems known to CPU (only CPUs with DP interface)
$0C91_H$		Module status information of a module
	Any logical address of a module/submodule	Module status information of a module using logical address
$0D91_H$		Module status information of a rack or station
	Centralized configuration: 0000_H: Rack 0 0001_H: Rack 1 0002_H: Rack 2 0003_H: Rack 3 **PROFIBUS DP:** $xxyy_H$: DP subnet ID/station no. **PROFINET IO:** Slot address of PROFINET IO device: Bit 15: is always = 1 Bit 11-14: PN IO subsystem ID (value range 100-115; in which only 0 to 15 must be specified) Bit 0-10: Station number of the PROFINET IO device	Module status information of all modules in specified rack/station
		Rack/station status information
0092_H		Expected state of the rack in the central configuration or stations of a subnet
	0000_H	Information about the state of the rack in the central configuration
	DP master system ID	Information about the state of the stations in the subnet
0292_H		Actual state of the rack in the central configuration or stations of a subnet
	0000_H	Information about the state of the rack in the central configuration
	DP master system ID	Information about the state of the stations in the subnet
0292_H		Status of the backup batteries in a rack of a CPU after at least one battery has failed
0292_H		Status of the overall battery backup status of all racks/module racks of a CPU
0292_H		Status of the 24-V power supply to all racks/module racks of a CPU
$0F92_H$		

SSL ID	Index	Message function
0692_H		Diagnostic state of the rack in the central configuration or stations of a subnet
	0000_H	Information about the state of the rack in the central configuration
	DP master system ID	Information about the state of the stations in the subnet
		Rack/station status information
0094_H		Expected state of the rack in the central configuration or stations of a subnet
	0000_H	Information about the state of the rack in the central configuration
	DP master system ID or PN IO subsystem no.	Information about the state of the stations in the subnet
0294_H		Actual state of the rack in the central configuration or stations of a subnet
	0000_H	Information about the state of the rack in the central configuration
	DP master system ID or PN IO subsystem no.	Information about the state of the stations in the subnet
0694_H		Diagnostic state of the rack in the central configuration or stations of a subnet
	0000_H	Information about the state of the rack in the central configuration
	DP master system ID or PN IO subsystem no.	Information about the state of the stations in the subnet
0794_H		Faulty- and/or maintenance status of station
	0000_H	Information about the state of the rack in the central configuration
	DP master system ID or PN IO subsystem no.	Information about the state of the stations in the subnet
$0F94_H$	–	Header information only
		Extended DP master system information
0195_H	$xxyy_H$: DP master system ID/00_H	Extended DP master system information of a DP master system (only CPUs with DP interface)
$0F95_H$	–	Header information only (only CPUs with DP interface)
		Submodule status information
0696_H	Any logical address of a module/submodule	Status data of all submodules of a module
$0C96_H$	Any logical address of a module/submodule	Status data of a submodule
		ToolChanger information (only for CPUs with PN interface)
$009C_H$		Information about all tool changers and their tools in a PN IO subsystem
$019C_H$		Information about all tool changers
$029C_H$		Information about a tool changer and its tools
$039C_H$		Information about a tool and its IO device

SSL ID	Index	Message function
$0F9C_H$		only header information
		Diagnostic buffer
$00A0_H$		All input event information (in the RUN of CPU default mode outputs only 10 entries; the number of event information output in RUN can be parameterized from 10 - 499)
$01A0_H$	x	The "x" most recent input event information
$0FA0_H$	–	Header info SSL only
		Diagnostic data on modules
$00B1_H$	Any logical address of a module/submodule	The first four diagnostic bytes of a module (diagnostics data record DS0)
$00B2_H$	Rack and slot number	All diagnostics data of a module (diagnostics data record DS1-only for centrally mounted modules)
$00B3_H$	Any logical address of a module/submodule	All diagnostics data of a module (diagnostics data record DS1)
$00B4_H$	Logical basic address (diagnosis address of the slave)	Standard diagnostics data of a DP slave (only CPUs with DP interface)

Index