

SIEMENS

SIMATIC S5

S5-135U/155U
CPU 922/CPU928/CPU 928B/CPU 948

List of Operations

Order No.
6ES5 997-3UA22, Release 03

This publication is protected by copyright. Transmission and reproduction of this document as well as use and notification of its contents are not permitted without express authority. This also applies to translation into other languages. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

Technical data subject to alteration.

Copyright © Siemens AG 1993 All Rights Reserved

Order No.: 6ES5 997-3UA22

Order from: Elektronikwerk Karlsruhe

Printed in the Federal Republic of Germany

Contents

	Page
Explanatory Notes on the List of Operations	1
Explanatory Notes on the Operands	3
Explanatory Notes on the Formal Operands (Block Parameters)	7
Basic Operations	10
Boolean Logic Operations	10
Set/Reset Operations/Binary	16
Load Operations	20
Transfer Operations	28
Timer Operations	34
Counter Operations	36
Arithmetic Operations	38
Comparison Operations	42
Block Call Operations	48
Block End Operations	52
Null Operations	54
Stop Operation	54
Display Construction Operations	54
Supplementary Operations	56
Logic Operations	56
Digital Operations	56
Bit Test Operations	58
Set/Reset Operations	62
Timer and Counter Operations	66
Load and Transfer Operations	70
Conversion Operations	74
Shift and Rotate Operations	76
Jump Operations	78
Other Operations	80

	Page
System Operations	86
Load and Transfer Operations	86
Arithmetic Operations	94
Jump Operations	96
Other Operations	96
Set Operations	100
Register to Register Transfer Operations	102
Load, Transfer and Arithmetic Operations with the Base Address Register	102
Access to local, word-oriented memory	106
Test/set Busy location (global area)	106
Access to global, byte-oriented memory	108
Access to global, word-oriented memory	110
Open page	110
Test/set Busy location (page area)	110
Access to byte-oriented pages	112
Access to word-oriented pages	114
Machine Code Listing	116
Alphabetical Index of Operations (with Machine Code)	131
Explanatory Notes on the Condition Codes	144
List of Organization Blocks	146
OBs for Program Processing	146
OBs for Start-up Procedures	148
OBs for Handling Controller Errors in the CPU 922/928/928B	150
OBs for Handling Controller Errors in the CPU 948	154
OBs with Special Functions	156

Explanatory Notes on the List of Operations

Abbreviations	Explanations
ACCU 1 ACCU 2 ACCU 3 ACCU 4	The four 32-bit accumulators
ACCU 1-H ACCU 2-H ACCU 3-H ACCU 4-H	The high word of the four 32-bit accumulators
ACCU 1-L ACCU 2-L ACCU 3-L ACCU 4-L	The low word of the four 32-bit accumulators
Condition codes CC0/CC1	Condition codes 0/1 (see pages 119, 120)
OV	Overflow; this condition code is set e.g. if the number range is exceeded during arithmetic operations.
OS	Stored overflow; this condition code is set if at least one arithmetic operation causes an overflow (for detection of arithmetic errors).
Y	The condition code is set/reset depending on the statement.
1	Condition code is set
0	Condition code is reset
N	Condition code is not affected (see Explanatory Notes on the Condition Codes)
Formal operand	Symbolic label with up to 4 characters. The first character must be a letter (see page 7ff).

Abbreviations	Explanations
PI	Process Image → memory areas for data that are read from the I/Os and/or transferred to the I/Os. The I/O image remains in these memory areas during one program cycle and is updated prior to the next. The binary logic and set/reset operations always use the PI.
PII/PIQ	Process Image of Inputs/Outputs
RLO	Binary Result of Logic Operation (1 bit)
RLO-dependent command flow?	Command execution depends on the RLO
Y	The statement is executed only if RLO = "1".
Y ↑	The statement is executed only on the leading edge of the RLO (RLO changes from "0" to "1").
Y ↓	The statement is executed only after the RLO changes from "1" to "0" (falling edge).
N	The statement is always executed.
RLO reset?	Command affects the RLO
Y	RLO is set to "1" or "0". Please refer to the function description of the corresponding statement for explanation on how the new RLO is formed.
1	RLO is set to "1".
N	RLO does not change.
RLO reloaded?	Y The RLO does not change. The RLO cannot be combined any further. If a command which reloads the RLO is followed by a binary logic operation, the scan result is reloaded and a new RLO is started.
N	The RLO can be combined further.
STL	Statement List method of representation in STEP 5.

Explanatory Notes on the Operands

Abbr	Description	Permissible Value Range for Operands		Size in Bits
		CPU	Range	
BN	Byte constant (fixed-point no.)	all	-128 to +127	8
C	Counter	all	0 to 255	-
D	Data bit	all	0.0 to 255.15	1
DB	Data block	922/ 928/ 928B 948	3 to 255 2 to 255	-
DD	Data double word	all	0 to 254	32
DH	Double word constant (hexadecimal)	all ¹⁾	0 to FFFF FFFF	32
DL	Data word (left-hand byte)	all	0 to 255	8
DR	Data word (right-hand byte)	all	0 to 255	8
DW	Data word (in a DB or DX)	all	0 to 255	16
DX	Data block (extension)	922/ 928 928B/ 948	1 to 255 3 to 255	-
F	Flag	all	0.0 to 255.7	1
FB	Function block	all	0 to 255	-
FD	Flag double word	all	0 to 252	32
FW	Flag word	all	0 to 254	16
FX	Function block (extension)	all	0 to 255	-

1) CPU 922 from version 09

Abbr	Description	Permissible Value Range for Operands		Size in Bits
		CPU	Range	
FY	Flag byte	all	0 to 255	8
I	Input (in PII)	all	0.0 to 127.0	1
IB	Input byte (in PII)	all	0 to 127	8
ID	Input double word (in PII)	all	0 to 124	32
IW	Input word (in PII)	all	0 to 126	16
KB	Constant (1 byte)	all	0 to 255	8
KC	Constant (count)	all	0 to 999	16
KF	Constant (fixed-point number)	all	-32768 to +32767	16
KG	Constant (floating-point number)	all	$\pm 0,1701412 \cdot 10^{39}$ to $\pm 0,1469368 \cdot 10^{-38}$	32
KH	Constant (hexadecimal code)	all	0 to FFFF	16
KM	Constant (2-byte bit pattern)	all	Arbitrary bit pattern	16
KS	Constant (2 characters)	all	ASCII characters	16
KT	Constant (time)	all	0.0 to 999.3	16
KY	Constant (2 bytes)	all	0 to 255 (per byte)	16
OB	Organization block	all	1 to 39	-
OB	Operating system special function	922/ 928/ 928B 948	110 to 255 121 to 255	-
OW	Word of the extended I/O area (without PII/PIQ update)	all	0 to 254	16
OY	Byte of the extended I/O area (without PII/PIQ update)	all	0 to 255	8

Abbr	Description	Permissible Value Range for Operands		Size in Bits
		CPU	Range	
PB	Program block	all	0 to 255	-
PW	Peripheral word of - digital inputs (direct reading of the PII) - analog inputs/digital inputs (without PII update) - digital outputs (with PIQ update) - analog outputs/digital outputs (without PIQ update)	all	0 to 126 128 to 254 0 to 126 128 to 254	16
PY	Peripheral byte of - digital inputs (direct reading of the PII) - analog inputs/digital inputs (without PII update) - digital outputs (with PIQ update) - analog outputs/digital outputs (without PIQ update)	all	0 to 127 128 to 255 0 to 127 128 to 255	8
Q	Output (with PIQ update)	all	0.0 to 127.0	1
QB	Output byte (with PIQ update)	all	0 to 127	8
QD	Output double word (with PIQ update)	all	0 to 124	32
QW	Output word (with PIQ update)	all	0 to 126	16
RI	Interface data area	all	0 to 255	16
RJ	Extended interface data area	all	0 to 255	16
RS	System data area	all	0 to 255	16
RT	Extended system data area	all	0 to 255	16

Abbr	Description	Permissible Value Range for Operands		Size in Bits
		CPU	Range	
S	Flag, additional (S flag)	922	n/a	1
		928	n/a	
		928B	0.0 to 1023.7	
		948	0.0 to 4095.7	
SB	Sequence block	all	0 to 255	-
SD	Flag double word, additional (S flag double word)	922	n/a	32
		928	n/a	
		928B	0 to 1020	
SW	Flag word, additional (S flag word)	922	n/a	16
		928	n/a	
		928B	0 to 1022	
SY	Flag byte, additional (S flag byte)	922	n/a	8
		928	n/a	
		928B	0 to 1023	
T	Timer	922	n/a	-
		928	n/a	
		928B	0 to 1023	
T	Timer	948	0 to 4095	-
		928B	0 to 1023	
		948	0 to 4095	
T	Timer	all	0 to 255	-

Explanatory Notes on the Formal Operands (Block Parameters)

A maximum of 126 different formal operands (nos. 1 to 126) can be programmed per FB/FX.

Parameter Type	Data Type	Actual Operands Permitted
I, Q	BI for an operand with bit address	I, Q, F
	BY for an operand with byte address	IB, QB, FY, DL, DR, PY, OY
	W for an operand with word address	IW, QW, FW, DW, PW, OW
	D for an operand with double word address	ID, QD, FD, DD
D	KM for a binary pattern (16 bits)	Constants
	KY for 2-byte serial absolute value numbers from 0 to 255	
	KH for a 4 digit hexadecimal number	
	KS for a character (max. 2 alphanum. characters)	
	KT for a time in BCD with time base 1.0 to 999.3	
	KC for a count value in BCD from 0 to 999	
	KF for a fixed-point number from -32768 to +32767	
	KG for a floating-point number from $\pm 0,1701412 \cdot 10^{39}$ to $\pm 0,1469368 \cdot 10^{-38}$	

Parameter Type	Data Type	Actual Operands Permitted
B	Type specification not permitted	DB Data blocks: statement C DB is executed
		FB Function blocks (permitted without parameters only) are called unconditionally: JU FB
		OB Organization blocks are called unconditionally: JU OB
		PB Program blocks are called unconditionally : JU PB
		SB Sequence blocks are called unconditionally: JU SB
T	Type specification not permitted	T
C	Type specification not permitted	C

Intentionally blank!

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function
		C	C	O	O	1 dep.	2 affect.	3 reload	☒ = Operation with this		not possible CPU		
		1	0	V	S	1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	

Boolean Logic Operations

All logic operations generate a result (RLO).
 The first RLO in a string of logic operations generates the new RLO from the signal status scanned. All subsequent logic operations generate the new RLO from the signal status scanned, and gate it with the old RLO. The string of logic operations is terminated by an operation that reloads the RLO (e.g., set/reset operation).

A	I	0.0 to 127.7	N	N	N	N	N	Y	N	20-25	0.9	0.57	0.18	Scan input for "1" and combine with RLO through logic AND
A	Q	0.0 to 127.7	N	N	N	N	N	Y	N	20-25	0.9	0.57	0.18	Scan output for "1" and combine with RLO through logic AND
A	F	0.0 to 255.7	N	N	N	N	N	Y	N	20-25	0.9	0.57	0.18	Scan flag for "1" and combine with RLO through logic AND
A	S	0.0 to 1023.7	N	N	N	N	N	Y	N	☒	☒	3.7	☒	Scan S flag for "1" and combine with RLO through logic AND
	S	0.0 to 4095.7	N	N	N	N	N	Y	N	☒	☒	☒	0.39	
A	D	0.0 to 255.15	N	N	N	N	N	Y	N	37-38	29	3.4	0.77	Scan a bit in the data block (DB/DX) for "1" and combine with RLO through logic AND
A	T	0 to 255	N	N	N	N	N	Y	N	20-25	0.9	0.57	0.18	Scan a time for "1" and combine with RLO through logic AND
A	C	0 to 255	N	N	N	N	N	Y	N	19-23	0.9	0.57	0.18	Scan a counter for "1" and combine with RLO with RLO through logic AND
AN	I	0.0 to 127.7	N	N	N	N	N	Y	N	20-25	0.9	0.57	0.18	Scan input for "0" and combine with RLO through logic AND
AN	Q	0.0 to 127.7	N	N	N	N	N	Y	N	20-25	0.9	0.57	0.18	Scan output for "0" and combine with RLO through logic AND

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function		
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU				
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948			
Boolean Logic Operations (continued)															
AN	F 0.0 to 255.7	N	N	N	N	N	Y	N	20-25	0.9			0.57	0.18	Scan flag for "0" and combine with RLO through logic AND
AN	S 0.0 to 1023.7	N	N	N	N	N	Y	N	☒	☒			3.7	☒	Scan S flag for "0" and combine with RLO through logic AND
	S 0.0 to 4095.7	N	N	N	N	N	Y	N	☒	☒			☒	0.39	
AN	D 0.0 to 255.15	N	N	N	N	N	Y	N	36-41	29			3.4	0.77	Scan a bit in the data block (DB/DX) for "0" and combine with RLO through logic AND
AN	T 0 to 255	N	N	N	N	N	Y	N	22	0.9			0.57	0.18	Scan a time for "0" and combine with RLO through logic AND
AN	C 0 to 255	N	N	N	N	N	Y	N	18-20	0.9			0.57	0.18	Scan a counter for "0" and combine with RLO through logic AND
O	I 0.0 to 127.7	N	N	N	N	N	Y	N	19-21	0.9			0.57	0.18	Scan input for "1" and combine with RLO through logic OR
O	Q 0.0 to 127.7	N	N	N	N	N	Y	N	19-21	0.9			0.57	0.18	Scan output for "1" and combine with RLO through logic OR
O	F 0.0 to 257.7	N	N	N	N	N	Y	N	19-21	0.9			0.57	0.18	Scan flag for "1" and combine with RLO through logic OR
O	S 0.0 to 1023.7	N	N	N	N	N	Y	N	☒	☒			3.7	☒	Scan S flag for "1" and combine with RLO through logic OR
	S 0.0 to 4095.7	N	N	N	N	N	Y	N	☒	☒			☒	0.39	
O	D 0.0 to 255.15	N	N	N	N	N	Y	N	35-38	28			3.4	0.77	Scan a bit in the data block (DB/DX) for "1" and combine with RLO through logic OR

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Boolean Logic Operations (continued)														
O	T 0 to 255	N	N	N	N	N	Y	N	18-21	0.9		0.57	0.18	Scan a time for "1" and combine with RLO through logic OR
O	C 0 to 255	N	N	N	N	N	Y	N	17-20	0.9		0.57	0.18	Scan a counter for "1" and combine with RLO through logic OR
ON	I 0.0 to 127.7	N	N	N	N	N	Y	N	18-21	0.9		0.57	0.18	Scan input for "0" and combine with RLO through logic OR
ON	Q 0.0 to 127.7	N	N	N	N	N	Y	N	18-21	0.9		0.57	0.18	Scan output for "0" and combine with RLO through logic OR
ON	F 0.0 to 255.7	N	N	N	N	N	Y	N	18-21	0.9		0.57	0.18	Scan flag for "0" and combine with RLO through logic OR
ON	S 0.0 to 1023.7	N	N	N	N	N	Y	N	☒	☒		3.7	☒	Scan S flag for "0" and combine with RLO through logic OR
	S 0.0 to 4095.7	N	N	N	N	N	Y	N	☒	☒		☒	0.39	
ON	D 0.0 to 255.15	N	N	N	N	N	Y	N	35-37	28		3.4	0.77	Scan a bit in the data block (DB/DX) for "0" and combine with RLO through logic OR
ON	T 0 to 255	N	N	N	N	N	Y	N	19-21	0.9		0.57	0.18	Scan a time for "0" and combine with RLO through logic OR
ON	C 0 to 255	N	N	N	N	N	Y	N	16-20	0.9		0.57	0.18	Scan a counter for "0" and combine with RLO through logic OR
O	-	N	N	N	N	N	Y	Y	12-14	0.8		0.57	0.18	Combine AND operations through logic OR

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this not possible CPU				
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	
Boolean Logic Operations (continued)													
A(-	N	N	N	N	N	Y	Y	20	1.2	0.57	0.18	Combine expressions enclosed with parentheses (8 levels) through logic AND
O(-	N	N	N	N	N	Y	Y	21	1.2	0.57	0.18	Combine expressions enclosed with parentheses (8 levels) through logic OR
)	-	N	N	N	N	N	Y	N	20-22	1.2	0.57	0.18	Right parenthesis (End of operation in parentheses)
Set/Reset Operations, Binary													
S	I 0.0 to 127.7	N	N	N	N	Y	N	Y	19-21	1.2	0.63	0.32	The input of the process image is set to "1" if the RLO is "1"
S	Q 0.0 to 127.7	N	N	N	N	Y	N	Y	19-21	1.2	0.63	0.32	The output of the process image is set to "1" if the RLO is "1"
S	F 0.0 to 255.7	N	N	N	N	Y	N	Y	19-21	1.2	0.63	0.32	The flag is set to "1" if the RLO is "1"
S	S 0.0 to 1023.7	N	N	N	N	Y	N	Y	☒	☒	3.9	☒	The S flag is set to "1" if the RLO is "1"
	S 0.0 to 4095.7	N	N	N	N	Y	N	Y	☒	☒	☒	0.48	
S	D 0.0 to 255.15	N	N	N	N	Y	N	Y	35-37	28	0.63	0.77	The bit in the data block (DB/DX) is set to "1" if the RLO is "1"
R	I 0.0 to 127.7	N	N	N	N	Y	N	Y	18-22	1.2	0.63	0.32	The input of the process image is reset to "0" if the RLO is "1"
R	Q 0.0 to 127.7	N	N	N	N	Y	N	Y	18-22	1.2	0.63	0.32	The output of the process image is reset to "0" if the RLO is "1"

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function		
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this						
						1	2	3	CPU 922	CPU 928		CPU 928B		CPU 948	
Set/Reset Operations, Binary (continued)															
R	F 0.0 to 127.7	N	N	N	N	Y	N	Y	18-22	1.2			0.63	0.32	The flag is reset to "0" if the RLO is "1"
R	S 0.0 to 1023.7	N	N	N	N	Y	N	Y	☒	☒			3.9	☒	The S flag is reset to "0" if the RLO is "1"
	S 0.0 to 4095.7	N	N	N	N	Y	N	Y	☒	☒			☒	0.48	
R	D 0.0 to 255.15	N	N	N	N	Y	N	Y	33-38	27			3.4	0.77	The bit in the data block (DB/DX) is reset to "0" if the RLO is "1"
=	I 0.0 to 127.7	N	N	N	N	N	N	Y	20	1.2			0.63	0.32	The value of the RLO is assigned to the input in the process image
=	Q 0.0 to 127.7	N	N	N	N	N	N	Y	20	1.2			0.63	0.32	The value of the RLO is assigned to the output in the process image
=	F 0.0 to 255.7	N	N	N	N	N	N	Y	20	1.2			0.63	0.32	The value of the RLO is assigned to the flag
=	S 0.0 to 1023.7	N	N	N	N	N	N	Y	☒	☒			3.9	☒	The value of the RLO is assigned to the S flag
	S 0.0 to 4095.7	N	N	N	N	N	N	Y	☒	☒			☒	0.48	
=	D 0.0 to 255.15	N	N	N	N	N	N	Y	37	29			3.4	0.77	The value of the RLO is assigned to the bit in the data block (DB/DX)

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function
		C1	C0	O	S	1 dep.	2 affect.	3 reload	☒ = Operation with this		not possible CPU		
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	

Load Operations

The original contents of ACCU 1 are passed on to ACCU 2 before the byte, word or double word addressed is loaded into ACCU 1. During byte and word operations, the high bits (not loaded) of ACCU 1 are deleted (bits 8 to 31 for byte operations, bits 16 to 31 for word operations). If you use ACCU 3 and ACCU 4, you must insert the "ENT" operation from the supplementary operation set to restore the accumulator contents.

Operation	Operands	C1	C0	O	S	RLO 1	RLO 2	RLO 3	CPU 922	CPU 928	CPU 928B	CPU 948	Function
L	IB 0 to 127	N	N	N	N	N	N	N	22	12	0.81	0.18	Load an input byte from the PII into ACCU 1-L
L	IW 0 to 126	N	N	N	N	N	N	N	22	13	0.94	0.50	Load an input word from the PII into ACCU 1-L: byte n → bits 8-15, byte n+1 → bits 0-7
L	ID 0 to 244	N	N	N	N	N	N	N	24	16	1.6	0.71	Load an input double word from the PII into ACCU 1: byte n → bits 24-31, byte n+1 → bits 16-23, byte n+2 → bits 8-15, byte n+3 → bits 0-7
L	QB 0 to 127	N	N	N	N	N	N	N	21	12	0.81	0.18	Load an output byte from the PIQ into ACCU 1-L
L	QW 0 to 126	N	N	N	N	N	N	N	22	12	0.94	0.50	Load an output word from the PIQ into ACCU 1-L: byte n → bits 8-15, byte n+1 → bits 0-7
L	QD 0 to 124	N	N	N	N	N	N	N	24	16	1.6	0.71	Load an output double word from the PIQ into ACCU 1: byte n → bits 24-31, byte n+1 → bits 16-23, byte n+2 → bits 8-15, byte n+3 → bits 0-7
L	FY 0 to 255	N	N	N	N	N	N	N	22	12	0.81	0.18	Load a flag byte into ACCU 1-L
L	FW 0 to 254	N	N	N	N	N	N	N	22	12	0.94	0.50	Load a flag word into ACCU 1-L: byte n → bits 8-15, byte n+1 → bits 0-7

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this					
						1	2	3	CPU 922	CPU 928		CPU 928B		CPU 948
Load Operations (continued)														
L	FD 0 to 252	N	N	N	N	N	N	N	24	16		1.6	0.71	Load a flag double word into ACCU 1: byte n → bits 24-31, byte n+1 → bits 16-23, byte n+2 → bits 8-15, byte n+3 → bits 0-7
L	SY 0 to 1023	N	N	N	N	N	N	N	☒	☒		2.4	☒	Load an S flag byte into ACCU 1-L
	SY 0 to 4095	N	N	N	N	N	N	N	☒	☒		☒	0.39	
L	SW 0 to 1022	N	N	N	N	N	N	N	☒	☒		2.5	☒	Load an S flag word into ACCU 1-L: Byte n → bits 8-15, byte n+1 → bits 0-7
	SW 0 to 4094	N	N	N	N	N	N	N	☒	☒		☒	0.59	
L	SD 0 to 1020	N	N	N	N	N	N	N	☒	☒		3.1	☒	Load an S flag double word into ACCU 1: byte n → bits 24-31, byte n+1 → bits 16-23, byte n+2 → bits 8-15, byte n+3 → bits 0-7
	SD 0 to 4092	N	N	N	N	N	N	N	☒	☒		☒	0.77	
L	DH0 to FFFF FFFF	N	N	N	N	N	N	N	20.1 ¹⁾	15		1.7	0.57	Load a constant (hexadecimal code as double word) into ACCU 1
L	DL 0 to 255	N	N	N	N	N	N	N	24	14		1.7	0.50	Load the left byte of a data word of the current data block into ACCU 1-L

¹⁾ Operation possible from version 09 onwards.

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution time in μs				Function
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU		
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	
Load Operations (continued)													
L	DR 0 to 255	N	N	N	N	N	N	N	24	14	1.7	0.50	Load the right byte of a data word of the current data block into ACCU 1-L
L	DW 0 to 255	N	N	N	N	N	N	N	24	14	1.5	0.50	Load a data word of the current data block into ACCU 1-L
L	DD 0 to 254	N	N	N	N	N	N	N	25	17	2.0	0.68	Load a flag double word into ACCU 1: word n → bits 16-31, word n+1 → bits 0-7
L	KB 0 to 255	N	N	N	N	N	N	N	19	8	0.63	0.18	Load a constant (1-byte number) into ACCU 1-L
L	KC 0 to 999	N	N	N	N	N	N	N	20	14	1.2	0.39	Load a constant (count in BCD) into ACCU 1-L
L	KF -32768 to +32767	N	N	N	N	N	N	N	20	14	1.2	0.39	Load a constant (fixed-point number) into ACCU 1-L
L	KG (see page 4)	N	N	N	N	N	N	N	21	15	1.7	0.57	Load a constant (floating point number) into ACCU 1-L
L	KH 0 to FFFF	N	N	N	N	N	N	N	20	14	1.2	0.39	Load a constant (hexadecimal code) into ACCU 1-L
L	KM bit pattern, 16 bit	N	N	N	N	N	N	N	20	14	1.2	0.39	Load a constant (bit pattern) into ACCU 1-L
L	KS (2 ASCII characters)	N	N	N	N	N	N	N	20	14	1.2	0.39	Load a constant (2 characters in ASCII format) into ACCU 1-L
L	KT 0.0 to 999.3	N	N	N	N	N	N	N	20	14	1.2	0.39	Load a constant time (time in BCD) into ACCU 1-L
L	KY 2 bytes 0 to 255 each	N	N	N	N	N	N	N	20	14	1.2	0.39	Load a constant (2-byte number) into ACCU 1-L

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Load Operations (continued)														
L	PY 0 to 255	N	N	N	N	N	N	N	30 ¹⁾	16 ¹⁾		1.4 ¹⁾	1.7 ¹⁾	Load a peripheral byte from the digital/analog inputs into ACCU 1-L
L	PW 0 to 254	N	N	N	N	N	N	N	31 ¹⁾	211)		2.1 ¹⁾	2.69 ¹⁾	Load a peripheral word from the digital/analog inputs into ACCU 1-L: byte n → bits 8-15, byte n+1 → bits 0-7
L	OY 0 to 255	N	N	N	N	N	N	N	30 ¹⁾	161)		1.4 ¹⁾	1.7 ¹⁾	Load a byte of the extended I/O area into ACCU 1-L
L	OW 0 to 254	N	N	N	N	N	N	N	31 ¹⁾	211)		2.1 ¹⁾	2.7 ¹⁾	Load a word of the extended I/O area into ACCU 1-L: byte n → bits 8-15, byte n+1 → bits 0-7
L	T 0 to 255	N	N	N	N	N	N	N	25	14		0.81	0.30	Load a time in binary code into ACCU 1-L
L	C 0 to 255	N	N	N	N	N	N	N	24	14		0.81	0.30	Load a count in binary code into ACCU 1-L
LC	T 0 to 255	N	N	N	N	N	N	N	38	16		3.7	0.39	Load a time in BCD into ACCU 1-L (including binary-BCD conversion)
LC	C 0 to 255	N	N	N	N	N	N	N	35	15		3.7	0.39	Load a count in BCD into ACCU 1-L (including binary-BCD conversion)

¹⁾ Execution time for single processing operation and for immediate bus access in multiprocessing operations. I/Os acknowledge within 0.1 μs or proportionally longer execution time for longer acknowledgement time.

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU		
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	

Transfer Operations

The contents of ACCU 1 are transferred to the operand specified.

T	IB 0 to 127	N	N	N	0	N	N	N	15	11	0.75	0.18	Transfer the contents of ACCU 1-L (bits 0-7) to an input byte (into the PII)
T	IW 0 to 126	N	N	N	0	N	N	N	15	11	0.88	0.41	Transfer the contents of ACCU 1-L (bits 0-7) to an input word (into PII): bits 8-15 → byte n, bits 0-7 → byte n+1
T	ID 0 to 124	N	N	N	0	N	N	N	20	15	1.9	0.59	Transfer the contents of ACCU 1 to an input double word (into the PII): bits 24-31 → byte n, bits 16-23 → byte n+1, bits 8-15 → byte n+2, bits 0-7 → byte n+3
T	QB 0 to 127	N	N	N	0	N	N	N	15	11	0.75	0.18	Transfer the contents of ACCU 1-L (bits 0-7) to an output byte (into the PIQ)
T	QW 0 to 126	N	N	N	0	N	N	N	15	11	0.88	0.41	Transfer the contents of ACCU 1-L (bits 0-7) to an output word (into the PIQ): bits 8-15 → byte n, bits 0-7 → byte n+1
T	QD 0 to 124	N	N	N	0	N	N	N	20	15	1.9	0.59	Transfer the contents of ACCU 1 to an output double word (into the PIQ): bits 24-31 → byte n, bits 16-23 → byte n+1, bits 8-15 → byte n+2, bits 0-7 → byte n+3
T	FY 0 to 255	N	N	N	0	N	N	N	15	11	0.75	0.18	Transfer the contents of ACCU 1-L to a flag byte (bits 0-7)
T	FW 0 to 254	N	N	N	0	N	N	N	15	11	0.88	0.41	Transfer the contents of ACCU 1-L to a flag word: bits 8-15 → byte n, bits 0-7 → byte n+1
T	FD 0 to 252	N	N	N	0	N	N	N	20	15	1.9	0.59	Transfer the contents of ACCU 1 to a flag double word: bits 24-31 → byte n, bits 16-23 → byte n+1, bits 8-15 → byte n+2, bits 0-7 → byte n+3

Basic Operations

Permissible for all blocks

Operation	Operands	Condition codes affected				RLO			Execution times in μs				Function
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU		
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	
Transfer Operations (continued)													
T	SY 0 to 1023	N	N	N	0	N	N	N	☒	☒	2.3	☒	Transfer the contents of ACCU 1-L to an S flag byte (bits 0-7)
	SY 0 to 4095	N	N	N	0	N	N	N	☒	☒	☒	0.39	
T	SW 0 to 1022	N	N	N	0	N	N	N	☒	☒	2.3	☒	Transfer the contents of ACCU 1-L to an S flag word: bits 8-15 → byte n, bits 0-7 → byte n+1
	SW 0 to 4094	N	N	N	0	N	N	N	☒	☒	☒	0.41	
T	SD 0 to 1020	N	N	N	0	N	N	N	☒	☒	3.4	☒	Transfer the contents of ACCU 1 to an S flag double word: bits 24-31 → byte n, bits 16-23 → byte n+1, bits 8-15 → byte n+2, bits 0-7 → byte n+3
	SD 0 to 4092	N	N	N	0	N	N	N	☒	☒	☒	0.59	
T	DL 0 to 255	N	N	N	0	N	N	N	22	15	1.5	0.68	Transfer the contents of ACCU 1-L (bits 0-7) to a data word (left byte) in a DB/DX
T	DR 0 to 255	N	N	N	0	N	N	N	22	14	1.4	0.68	Transfer the contents of ACCU 1-L (bits 0-7) to a data word (right byte) in a DB/DX
T	DW 0 to 255	N	N	N	0	N	N	N	22	14	1.4	0.41	Transfer the contents of ACCU 1-L (bits 0-15) to a data word in a DB/DX
T	DD 0 to 254	N	N	N	0	N	N	N	25	18	1.9	0.59	Transfer the contents of ACCU 1 to a data double word in a DB/DX: bits 16-31 → word n, bits 0-15 → word n+1

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU		
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	
Transfer Operations (continued)													
T	PY 0 to 127	N	N	N	0	N	N	N	27 ¹⁾	15 ¹⁾	2.0 ¹⁾	1.6 ¹⁾	Transfer the contents of ACCU 1-L (bits 0-7) to a peripheral byte of the digital or analog outputs. The PIQ is also corrected.
	PY 128 to 255	N	N	N	0	N	N	N	27 ¹⁾	14 ¹⁾	1.2 ¹⁾	1.5 ¹⁾	Transfer the contents of ACCU 1-L (bits 0-7) to a peripheral byte of the digital or analog outputs.
T	PW 0 to 126	N	N	N	0	N	N	N	30 ¹⁾	21 ¹⁾	3.2 ¹⁾	2.6 ¹⁾	Transfer the contents of ACCU 1-L (bits 0-15) to a peripheral word of the digital or analog outputs: bits 8-15 → byte n; bits 0-7 → byte n+1 The PIQ is also corrected.
	PW 128 to 254	N	N	N	0	N	N	N	30 ¹⁾	18 ¹⁾	2.0 ¹⁾	2.4 ¹⁾	Transfer the contents of ACCU 1-L (bits 0-15) to a peripheral word of the digital or analog outputs: bits 8-15 → byte n; bits 0-7 → byte n+1
T	OY 0 to 255	N	N	N	0	N	N	N	23 ¹⁾	14 ¹⁾	1.2 ¹⁾	1.5 ¹⁾	Transfer the contents of ACCU 1-L (bits 0-7) to a byte of the extended periphery of the digital or analog outputs (no process image).
T	OW 0 to 254	N	N	N	0	N	N	N	26 ¹⁾	18 ¹⁾	2.0 ¹⁾	2.4 ¹⁾	Transfer the contents of ACCU 1-L to a word of the extended periphery of the digital or analog outputs (no process image): bits 8-15 → byte n; bits 0-7 → byte n+1

¹⁾ Execution time for single processing operation and for immediate bus access in multiprocessing operation. I/Os acknowledge within 0.1 μs or proportionally longer execution time for longer acknowledgement time.

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Timer Operations														
SP	T 0 to 255	N	N	N	N	Y	↑	N	Y	20/ 42 ¹⁾	1.2/ 18 ¹⁾	3.6	0.18	Start timer (stored in ACCU 1-L) as pulse (start timer with continuous enable)
SE	T 0 to 255	N	N	N	N	Y	↑	N	Y	21/ 43 ¹⁾	1.2/ 18 ¹⁾	3.6	0.18	Start timer (stored in ACCU 1-L) as extended pulse (start timer with one-shot enable)
SD	T 0 to 255	N	N	N	N	Y	↑	N	Y	20/ 45 ¹⁾	1.2/ 18 ¹⁾	3.6	0.18	Start timer (stored in ACCU 1-L) as ON delay
SS	T 0 to 255	N	N	N	N	Y	↑	N	Y	21/ 46 ¹⁾	1.2/ 18 ¹⁾	3.6	0.18	Start timer (stored in ACCU 1-L) as stored ON delay
SF	T 0 to 255	N	N	N	N	Y	↓	N	Y	22/ 44 ¹⁾	1.2/ 18 ¹⁾	3.6	0.18	Start timer (stored in ACCU 1-L) as OFF delay
R	T 0 to 255	N	N	N	N	Y		N	Y	14/ 18 ²⁾	12/ 15 ²⁾	1.4	0.18	Reset timer

1) Time applies when "S5 time is not started/S5 time is started".

2) Time applies when RLO = 0 / RLO = 1.

Basic Operations

Permissible for all blocks

Operation	Operands	Condition codes affected				RLO			Execution times in μ s				Function		
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this	not possible CPU					
						1	2	3		CPU 922	CPU 928	CPU 928B		CPU 948	
Counter Operations															
CU	C 0 to 255	N	N	N	N	Y	↑	N	Y	18/ 23 ¹⁾	1.2/ 14 ¹⁾		2.1	0.18	Counter counts up 1
CD	C 0 to 255	N	N	N	N	Y	↑	N	Y	18/ 24 ¹⁾	1.2/ 14 ¹⁾		2.0	0.18	Counter counts down 1
S	C 0 to 255	N	N	N	N	Y	↑	N	Y	17/ 38 ¹⁾	16/ 23 ¹⁾		3.8	0.18	Set counter with the value stored in ACCU 1-L (BCD number from 0 to 999)
R	C 0 to 255	N	N	N	N	Y		N	Y	12/ 16 ²⁾	12/ 14 ²⁾		1.4	0.18	Reset counter

1) Time applies when "S5 time is not started/S5 time is started".

2) Time applies when RLO = 0 / RLO = 1.

Basic Operations

Permissible for all blocks

Operation	Operands	Condition codes affected				RLO			Execution times in μ s				Function
		C	C	O	O	1 dep. 2 affect. 3 reload			<input type="checkbox"/> = Operation with this		not possible CPU		
		1	0	V	S	1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	

Arithmetic Operations

The result (numerical value) of an arithmetic operation is stored in ACCU 1. All other accumulator contents change as follows:

For +F, -F, xF, :F: For +G, -G, xG, :G, +D, -D:
 ACCU-2-L: = ACCU-3-L ACCU 2: = ACCU 3
 ACCU-3-L: = ACCU-4-L ACCU 3: = ACCU 4
 ACCU-4-L: = ACCU-4-L ACCU 4: = ACCU 4

The original contents of ACCU 2-L or ACCU 2 are lost. Whether the result is <0, >0 or =0 can be evaluated via CC0 and CC1 (see Explanatory Notes on the Condition Codes).

Fixed-point numbers, 16 bits

Operation	Operands	CC1	CC0	OV	OS	RLO 1	RLO 2	RLO 3	CPU 922	CPU 928	CPU 928B	CPU 948	Function
+F	-	Y	Y	Y	Y	N	N	N	26-32	15-18	0.94	0.55	Add two fixed-point numbers: (ACCU 1-L) + (ACCU 2-L)
-F	-	Y	Y	Y	Y	N	N	N	26-33	15-20	0.94	0.55	Subtract one fixed-point number from another: (ACCU 2-L) - (ACCU 1-L)
xF	-	Y	Y	Y	Y	N	N	N	36-39	20-21	7.9	3.2	Multiply one fixed-point number by another: (ACCU 1-L) x (ACCU 2-L)
:F	-	Y	Y	Y	Y	N	N	N	27-50	13-24	10.4	3.8	Divide one fixed-point number by another: (ACCU 2-L) : (ACCU 1-L). ACCU 1-L: result; ACCU 2-H: remainder

Basic Operations

Permissible for all blocks

Operation	Operands	Condition codes affected				RLO			Execution times in μ s				Function
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU		
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	
Arithmetic Operations (continued)													
Floating-point numbers, 32 bits													
When performing arithmetic operations with a 16-bit mantissa (default), the eight low bits are set to "0".													
+G	-	Y	Y	Y	Y	N	N	N	56-86	30-33 ¹⁾ 31-56 ²⁾	9.1	3.3	Add two floating-point numbers: ACCU 1 + ACCU 2
-G	-	Y	Y	Y	Y	N	N	N	56-86	30-42 ¹⁾ 31-47 ²⁾	9.1	3.5	Subtract one floating-point number from another: ACCU 1 - ACCU 2
xG	-	Y	Y	Y	Y	N	N	N	52-74	29-39 ¹⁾ 30-66 ²⁾	12.1 1) 20.5 2)	5.2	Multiply one floating-point number by another: ACCU 1 x ACCU 2
:G	-	Y	Y	Y	Y	N	N	N	51-81	28-42 ¹⁾ 31-69 ²⁾	15.6 1) 23.3 2)	6.3	Divide one floating-point number by another: ACCU 2: ACCU 1; Result: ACCU 1-L: mantissa low ACCU 1-H: mantissa high and exponent

1) Time in the case of 16-bit mantissa (default)

2) Time in the case of 24-bit mantissa

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this				
						1	2	3	CPU 922	CPU 928		CPU 928B	

Comparison Operations

The contents of ACCU 2 (operand) are compared with the contents of ACCU 1 (operand 2). The RLO is set to "1" if the comparison condition is fulfilled or to "0" if it is not fulfilled. Whether the contents of ACCU 2 are <, > or = those in ACCU 1, can be evaluated via CC0 and CC1 (see Explanatory Notes on the Condition Codes).

Fixed-point numbers, 16 bits

Operation	Operands	CC0	CC1	OV	OS	RLO	CPU 922	CPU 928	CPU 928B	CPU 948	Function
!=F	-	Y	Y	0	0	N Y N	18	14-15	0.88	0.30	Compare two fixed-point numbers for equal to: if ACCU 2-L = ACCU 1-L, the RLO is "1"
>>F	-	Y	Y	0	0	N Y N	18	14-15	0.88	0.30	Compare two fixed-point numbers for not equal to: if ACCU 2-L \neq ACCU 1-L, the RLO is "1"
>F	-	Y	Y	0	0	N Y N	18	13-14	0.88	0.30	Compare two fixed-point numbers for greater than: if ACCU 2-L > ACCU 1-L, the RLO is "1"
>=F	-	Y	Y	0	0	N Y N	18	14-15	0.88	0.30	Compare two fixed-point numbers for greater than or equal to: if ACCU 2-L \geq ACCU 1-L, the RLO is "1"
<F	-	Y	Y	0	0	N Y N	18	14-15	0.88	0.30	Compare two fixed-point numbers for less than: if ACCU 2-L < ACCU 1-L, the RLO is "1"
<=F	-	Y	Y	0	0	N Y N	18	14-15	0.88	0.30	Compare two fixed-point numbers for less than or equal to: if ACCU 2-L \leq ACCU 1-L, the RLO is "1"

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU		
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	
Comparison Operations (continued)													
Floating-point numbers													
!=G	-	Y	Y	0	0	N	Y	N	19-27	17-19	1.9	1.4	Compare two floating-point numbers for equal to: if ACCU 2 = ACCU 1, the RLO is "1"
><G	-	Y	Y	0	0	N	Y	N	19-27	17-18	1.9	1.4	Compare two floating-point numbers for not equal to: if ACCU 2 \neq ACCU 1, the RLO is "1"
>G	-	Y	Y	0	0	N	Y	N	19-27	16-18	1.9	1.4	Compare two floating-point numbers for greater than: if ACCU 2 > ACCU 1, the RLO is "1"
>=G	-	Y	Y	0	0	N	Y	N	19-27	17-19	1.9	1.4	Compare two floating-point numbers for greater than or equal to: if ACCU 2 \geq ACCU 1, the RLO is "1"
<G	-	Y	Y	0	0	N	Y	N	19-27	17-19	1.9	1.4	Compare two floating-point numbers for less than: if ACCU 2 < ACCU 1, the RLO is "1"
<=G	-	Y	Y	0	0	N	Y	N	19-27	17-19	1.9	1.4	Compare two floating-point numbers for less than or equal to: if ACCU 2 \leq ACCU 1, the RLO is "1"

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU		
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	
Comparison Operations (continued)													
Fixed-point double words, 32 bits													
!=D	-	Y	Y	0	0	N	Y	N	19-24	15-16	1.6	0.52	Compare two fixed-point double words for equal to: if ACCU 2 = ACCU 1, the RLO is "1"
><D	-	Y	Y	0	0	N	Y	N	19-24	15-16	1.6	0.52	Compare two fixed-point double words for not equal to: if ACCU 2 \neq ACCU 1, the RLO is "1"
>D	-	Y	Y	0	0	N	Y	N	19-24	14-16	1.6	0.52	Compare two fixed-point double words for greater than: if ACCU 2 > ACCU 1, the RLO is "1"
>=D	-	Y	Y	0	0	N	Y	N	19-24	15-17	1.6	0.52	Compare two fixed-point double words for greater than or equal to: if ACCU 2 \geq ACCU 1, the RLO is "1"
<D	-	Y	Y	0	0	N	Y	N	19-24	15-16	1.6	0.52	Compare two fixed-point double words for less than: if ACCU 2 < ACCU 1, the RLO is "1"
<=D	-	Y	Y	0	0	N	Y	N	19-24	15-17	1.6	0.52	Compare two fixed-point double words for less than or equal to: if ACCU 2 \leq ACCU 1, the RLO is "1"

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function	
		A	N	Z	0	O	V	S	1 dep.	2 affect.	3 reload	☒ = Operation with this		not possible CPU
		1	0	0	0	0	0	0	1	2	3			
Block Call Operations														
JU	PB 0 to 255	N	N	N	0	N	N	Y	35	28		3.7	1.5/ 6.0 ⁵⁾	Unconditional program block call
JU	FB 0 to 255	N	N	N	0	N	N	Y	34	28		3.7	1.5/ 6.0 ⁵⁾	Unconditional function block call
DOU	FX 0 to 255	N	N	N	0	N	N	Y	41	32		4.8	1.5/ 6.0 ⁵⁾	Unconditional extended function call
JU	SB 0 to 255	N	N	N	0	N	N	Y	35	28		3.7	1.5/ 6.0 ⁵⁾	Unconditional sequence block call
JU	OB 1 to 39	N	N	N	0	N	N	Y	39	28		4.1	1.5/ 6.0 ⁵⁾	Unconditional organization block call
JU	OB 40 to 255	1)	1)	1)	1)	N	1)	Y	2)	2)		2)	2)	Unconditional call of a special function organization block of the operating system
JC	PB 0 to 255	N	N	N	0 ³⁾	Y	1	Y	12/ 35 ⁴⁾	0.8/ 27 ⁴⁾		3.7	1.6/ 6.1 ⁵⁾	Conditional program block call (if RLO is "1")
JC	FB 0 to 255	N	N	N	0 ³⁾	Y	1	Y	12/ 35 ⁴⁾	2.8/ 28 ⁴⁾		3.7	1.6/ 6.1 ⁵⁾	Conditional function block call (if RLO is "1")

1) The condition codes are set or not set according to the special function executed (see Programming Guide - Special Function OBs)

2) For execution times see List of Special Functions, page 130ff.

3) The Os bit remains unchanged if RLO = 0 (not for CPU 948).

4) Time applies when RLO = 0 / RLO = 1.

5) Time applies when "interruption at block limits".

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Block Call Operations (continued)														
DOC	FX 0 to 255	N	N	N	0 ¹⁾	Y	1	Y	19/ 43 2)	14/ 41 2)		4.8	1.6/ 6.1 6)	Conditional extended function block call (if RLO is "1")
JC	SB 0 to 255	N	N	N	0 ¹⁾	Y	1	Y	12/ 35 2)	0.8/ 27 2)		3.7	1.6/ 6.1 6)	Conditional sequence block call (if RLO is "1")
JC	OB 1 to 39	N	N	N	0 ¹⁾	Y	1	Y	12/ 41 2)	0.8/ 28 2)		4.1	1.6/ 6.1 6)	Conditional organization block call
JC	OB 110 to 255	3)	3)	3)	3)	Y	1 ⁴⁾	Y	5)	5)		5)	5)	Conditional call of special function organization block of the operating system
C	DB 2 to 255	N	N	N	N	N	N	N	☒	☒		☒	0.91	Call a data block
	DB 3 to 255	N	N	N	N	N	N	N	22	19		1.9	☒	
CX	DX 1 to 255	N	N	N	N	N	N	N	28	☒		☒	☒	Call an extended data block
	DX 2 to 255	N	N	N	N	N	N	N	☒	22		☒	☒	
	DX 3 to 255	N	N	N	N	N	N	N	☒	☒		2.7	1.0	

1) The OS bit remains unchanged if RLO = 0 (not for CPU 948).

2) Time applies when RLO = 0 / RLO = 1.

3) The condition codes are set or not set according to the special function executed (see Programming Guide - Special Function OBs).

4) Only if the RLO = 0 before the OB is called, otherwise the RLO can be influenced according to the special function executed (see Programming Guide - Special Function OBs).

5) For execution times see List of Special Functions, page 156ff.

6) Time applies when "interruption at block limits".

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Block Call Operations (continued)														
G	DB 2 to 255	N	N	N	N	N	N	N	☒	☒			498	Generate a data block. The number of its data words must be stored in ACCU 1 (max. 4091 DW)
	DB 3 to 255	N	N	N	N	N	N	N	67	48		76	☒	
GX	DX 1 to 255	N	N	N	N	N	N	N	67	☒		☒	☒	Generate an extended data block. The number of its data words must be stored in ACCU 1 (max. 4091 DW)
	DX 2 to 255	N	N	N	N	N	N	N	☒	48		☒	☒	
	DX 3 to 255	N	N	N	N	N	N	N	☒	☒		76	493	
Block End Operations														
BE	-	N	N	N	0	N	N	Y	26	11		3.8	2.0	Block end (termination of a block)
BEC	-	N	N	N	0 ¹⁾	Y	1	Y	12/ 27 ²⁾	0.8/ 19 ²⁾		3.8	2.1	Block end, conditional (if RLO is "1")
BEU	-	N	N	N	0	N	N	Y	26	11		3.8	2.0	Block end, unconditional

1) The OS bit remains unchanged if RLO = 0 (not for CPU 948).

2) Time applies when RLO = 0 / RLO = 1.

Basic Operations

Permissible for all blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Null Operations														
NOP 0	-	N	N	N	N	N	N	N	9	1.0		0.57	0.18	No operation (all bits set to 0)
NOP 1	-	N	N	N	N	N	N	N	9	0.8		0.57	0.18	No operation (all bits set to 1)
Stop Operation														
STP	-	N	N	N	N	N	N	N	-	-		-	-	Direct transition to "STOP" mode CPU 948: transition to communication stop (operating mode SMOOTH STOP), program processing aborted at cycle end or by the system program
Display Construction Operations														
BLD	0 - 255	N	N	N	N	N	N	N	9	0.8		0.57	0.18	Display construction statement/NOP for the programmable controller
BLD	130	N	N	N	N	N	N	N	9	0.8		0.57	0.18	Display construction operation for the programmer: generate blank line by carriage return
BLD	131	N	N	N	N	N	N	N	9	0.8		0.57	0.18	Display construction operation for the programmer: switch over to statement list (STL)
BLD	132	N	N	N	N	N	N	N	9	0.8		0.57	0.18	Display construction operation for the programmer: switch over to control system flowchart CSF)
BLD	133	N	N	N	N	N	N	N	9	0.8		0.57	0.18	Display construction operation for the programmer: switch over to ladder diagram (LAD)
BLD	255	N	N	N	N	N	N	N	9	0.8		0.57	0.18	Display construction operation for the programmer: terminate segment

Supplementary Operations

Permissible only in function blocks

Opera- tion STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			\boxtimes = Operation with this	not possible CPU				
						1	2	3		CPU 922	CPU 928	CPU 928B		CPU 948
Binary Logic Operations														
A=	Formal operand	N	N	N	N	N	Y	N	24 ¹⁾	24 ¹⁾		2.4 ¹⁾	0.91 ¹⁾	AND operation: scan a formal operand for "1" (parameter type: I, Q, T, C; data type: BI)
AN=	Formal operand	N	N	N	N	N	Y	N	24 ¹⁾	24 ¹⁾		2.4 ¹⁾	0.91 ¹⁾	AND operation: scan a formal operand for "0" (parameter type: I, Q, T, C; data type: BI)
O=	Formal operand	N	N	N	N	N	Y	N	24 ¹⁾	24 ¹⁾		2.4 ¹⁾	0.91 ¹⁾	OR operation: scan a formal operand for "1" (parameter type: I, Q, T, C; data type: BI)
ON=	Formal operand	N	N	N	N	N	Y	N	24 ¹⁾	24 ¹⁾		2.4 ¹⁾	0.91 ¹⁾	OR operation: scan a formal operand for "0" (parameter type: I, Q, T, C; data type: BI)
Digital Operations														
The result (= "0" or \neq "0") can be evaluated via CC0 and CC1 (see Explanatory Notes on the Condition Codes)														
AW	-	Y	0	0	N	N	N	N	16	11		0.57	0.18	Combine contents of ACCU 2 and ACCU 1 (word operation) through logic AND: result is stored in ACCU 1
OW	-	Y	0	0	N	N	N	N	16	11		0.57	0.18	Combine contents of ACCU 2 and ACCU 1 (word operation) through logic OR: result is stored in ACCU 1
XOW	-	Y	0	0	N	N	N	N	16	11		0.57	0.18	Combine contents of ACCU 2 and ACCU 1 (word operation) through logic EXOR: result is stored in ACCU 1

¹⁾ The execution time of the substituted operation must be added.

Supplementary Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Bit Test Operations														
These operations scan the status of a bit and update it in the RLO.														
TB	I 0.0 to 127.7	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan an input bit for signal status "1"
TB	Q 0.0 to 127.7	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan an output bit for signal status "1"
TB	F 0.0 to 255.7	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan a flag bit for signal status "1"
TB	T 0.0 to 255.15	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan a bit of a timer word for signal status "1"
TB	C 0.0 to 255.15	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan a bit of a counter word for signal status "1"
TB	D 0.0 to 255.15	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.77	Scan a bit of a data word (DB/DX) for signal status "1"
TB	RI 0.0 to 255.15	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan a bit in the RI area for signal status "1"
TB	RJ 0.0 to 255.15	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan a bit in the RJ area for signal status "1"
TB	RS 0.0 to 255.15	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan a bit in the RS area for signal status "1"
TB	RT 0.0 to 255.15	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan a bit in the RT area for signal status "1"

Supplementary Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Bit Test Operations (continued)														
These operations scan the status of a bit and update it in the RLO.														
TBN	I 0.0 to 127.7	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan an input bit for signal status "0"
TBN	Q 0.0 to 127.7	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan an output bit for signal status "0"
TBN	F 0.0 to 255.7	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan a flag bit for signal status "0"
TBN	T 0.0 to 255.15	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan a bit of a timer word for signal status "0"
TBN	C 0.0 to 255.15	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan a bit of a counter word for signal status "0"
TBN	D 0.0 to 255.15	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.77	Scan a bit of a data word (DB/DX) for signal status "0"
TBN	RI 0.0 to 255.15	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan a bit in the RI area for signal status "0"
TBN	RJ 0.0 to 255.15	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan a bit in the RJ area for signal status "0"
TBN	RS 0.0 to 255.15	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan a bit in the RS area for signal status "0"
TBN	RT 0.0 to 255.15	N	N	N	N	N	Y	N	☒	☒	☒	☒	0.48	Scan a bit in the RT area for signal status "0"

Supplementary Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function		
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU				
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948			
Set/Reset Operations															
S=	Formal operand	N	N	N	N	Y	N	Y	25 ¹⁾	23 ¹⁾			1.9 ¹⁾	0.64 ¹⁾	Binary setting of a formal operand (parameter type: I, Q; data type: BI)
RB=	Formal operand	N	N	N	N	Y	N	Y	25 ¹⁾	22 ¹⁾			1.9 ¹⁾	0.64 ¹⁾	Binary resetting of a formal operand (parameter type: I, Q; data type: BI)
RD=	Formal operand	N	N	N	N	Y	N	Y	14 ¹⁾	13 ¹⁾			1.9 ¹⁾	0.64 ¹⁾	Digital resetting of a formal operand for timers and counters (parameter type: T, C)
==	Formal operand	N	N	N	N	N	N	Y	25 ¹⁾	23 ¹⁾			1.9 ¹⁾	0.64 ¹⁾	Assignment of the RLO to a formal operand (parameter type: I, Q; data type: BI)
SU	I 0.0 to 127.7	N	N	N	N	N	N	Y	☒	☒			☒	0.48	Set an input bit (in the PII) unconditionally
SU	Q 0.0 to 127.7	N	N	N	N	N	N	Y	☒	☒			☒	0.48	Set an output bit (in the PIO) unconditionally
SU	F 0.0 to 255.7	N	N	N	N	N	N	Y	☒	☒			☒	0.48	Set a flag bit unconditionally
SU	T 0.0 to 255.15	N	N	N	N	N	N	Y	☒	☒			☒	0.48	Set a bit of a timer word unconditionally
SU	C 0.0 to 255.15	N	N	N	N	N	N	Y	☒	☒			☒	0.48	Set a bit of a counter word unconditionally

¹⁾ The execution time of the substituted operation must be added.

Supplementary Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU		
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	
Set/Reset Operations (continued)													
SU	D 0.0 to 255.15	N	N	N	N	N	N	Y	☒	☒		0.77	Set a bit of a data word (DB/DX) unconditionally
SU	RI 0.0 to 255.15	N	N	N	N	N	N	Y	☒	☒		0.48	Set a bit in the RI area unconditionally
SU	RJ 0.0 to 255.15	N	N	N	N	N	N	Y	☒	☒		0.48	Set a bit in the RJ area unconditionally
RU	I 0.0 to 127.7	N	N	N	N	N	N	Y	☒	☒		0.48	Reset an input bit (in the PII) unconditionally
RU	Q 0.0 to 127.7	N	N	N	N	N	N	Y	☒	☒		0.48	Reset an output bit (in the PIO) unconditionally
RU	F 0.0 to 255.7	N	N	N	N	N	N	Y	☒	☒		0.48	Reset a flag bit unconditionally
RU	T 0.0 to 255.15	N	N	N	N	N	N	Y	☒	☒		0.48	Reset a bit of a timer word unconditionally
RU	C 0.0 to 255.15	N	N	N	N	N	N	Y	☒	☒		0.48	Reset a bit of a counter word unconditionally
RU	D 0.0 to 255.15	N	N	N	N	N	N	Y	☒	☒		0.77	Reset a bit of a data word (DB/DX) unconditionally
RU	RI 0.0 to 255.15	N	N	N	N	N	N	Y	☒	☒		0.48	Reset a bit in the RI area unconditionally
RU	RJ 0.0 to 255.15	N	N	N	N	N	N	Y	☒	☒		0.48	Reset a bit in the RJ area unconditionally

Supplementary Operations

Permissible only in function blocks

Opera- tion STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function		
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU				
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948			
Timer and Counter Operations															
SP=	Formal operand	N	N	N	N	Y	↑	N	Y	14 ²⁾	16 ²⁾		1.9 ²⁾	0.64 ²⁾	Start timer specified as formal operand as pulse with the value stored in ACCU 1-L (parameter type: T)
SD=	Formal operand	N	N	N	N	Y	↑	N	Y	14 ²⁾	16 ²⁾		1.9 ²⁾	0.64 ²⁾	Start timer specified as formal operand as ON delay with the value stored in ACCU 1-L (parameter type: T)
SEC=	Formal operand	N	N	N	N	Y	↑	N	Y	14 ²⁾	15 ²⁾		1.9 ²⁾	0.64 ²⁾	Start timer specified as formal operand as extended pulse with the value stored in ACCU 1-L or set counter specified as formal operand with the count stored in ACCU 1-L (parameter type: T, C)
SSU=	Formal operand	N	N	N	N	Y	↑	N	Y	14 ²⁾	16 ²⁾		1.9 ²⁾	0.64 ²⁾	Start timer specified as formal operand as stored ON delay with the value stored in ACCU 1-L or increment a counter specified as formal operand (parameter type: T, C)
SFD=	Formal operand	N	N	N	N	¹⁾ N		N	Y	14 ²⁾	16 ²⁾		1.9 ²⁾	0.64 ²⁾	Start timer specified as formal operand as stored OFF delay with the value stored in ACCU 1-L or decrement a counter specified as formal operand (parameter type: T, C)
FR=	Formal operand	N	N	N	N	Y	↑	N	Y	12 ²⁾	13 ²⁾		1.9 ²⁾	0.64 ²⁾	Enable formal operand (timer/counter) for cold restart (for description see FR T or FR C); (parameter type: T, C)

¹⁾ The RLO is evaluated according to the executed operation.

²⁾ The execution time of the substituted operation must be added.

Supplementary Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function		
		C	C	O	O	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU				
		1	0	V	S	1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948			
Timer and Counter Operations (continued)															
FR	T 0 to 255	N	N	N	N	Y	↑	N	Y	¹⁾ 19/22	¹⁾ 15/16		1.6	0.18	Enable timer for cold restart. The operation is executed <u>only</u> on the leading edge of the RLO (change from "0" to "1"). The timer is restarted if the RLO is "1" at the time of the start operation.
FR	C 0 to 255	N	N	N	N	Y	↑	N	Y	¹⁾ 17/22	¹⁾ 16/18		1.6	0.18	Enable a counter for setting or counting up or down. This operation is executed <u>only</u> on the leading edge of the RLO (change from "0" to "1"). The counter is restarted if the RLO = "1" at the time of the set operation. The counter is counted up or down if the RLO = "1" at the time of the "counting up" (CU) or "counting down" (CD) operation.

¹⁾ Time applies when RLO = "0"/RLO = "1".

Supplementary Operations

Permissible only in function blocks

Opera- tion STL	Operands	Condition codes affected				RLO			Execution times in μs				Function	
		C	C	O	O	1 dep.	2 affect.	3 reload	not possible CPU					
		C 1	C 0	O V	O S	1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Load and Transfer Operations														
Load operations: the value in ACCU 1 is shifted and stored in ACCU 2. Zeros are supplied for unused bits in ACCU 1.														
L=	Formal operand	N	N	N	N	N	N	N	19 ¹⁾	16 ¹⁾		2.1 ¹⁾	0.64 ¹⁾	The value of the formal operand is loaded into ACCU 1 (parameter type: I, Q, T, C; data type: BY, D, W)
LD=	Formal operand	N	N	N	N	N	N	N	14 ¹⁾	14 ¹⁾		1.9 ¹⁾	0.64 ¹⁾	Load formal operand in BCD into ACCU 1 (parameter type: T, C)
LW=	Formal operand	N	N	N	N	N	N	N	5 ¹⁾	15 ¹⁾		0.5 ¹⁾	0.50 ¹⁾	Load the bit pattern of a formal operand into ACCU 1 (parameter type: D; data type: KF, KH, KM, KY, KS, KT, KC)
LDW=	Formal operand	N	N	N	N	N	N	N	5 ¹⁾	18 ¹⁾		0.5 ¹⁾	0.68 ¹⁾	The value of the formal operand is loaded into ACCU 1 (parameter type: D; data type: KG)
T=	Formal operand	N	N	N	0	N	N	N	18 ¹⁾	15 ¹⁾		2.1 ¹⁾	0.64 ¹⁾	The contents of ACCU 1 are transferred to the formal operand (parameter type: I, Q; data type: BY, D, W)
L	RI 0 to 255	N	N	N	N	N	N	N	23	13		0.68	0.18	Load a word from the interface data range (RI) into ACCU 1-L
L	RJ 0 to 255	N	N	N	N	N	N	N	X	13		0.68	0.18	Load a word from the extended interface data area range into ACCU 1-L
L	RS 0 to 255	N	N	N	N	N	N	N	23	13		0.68	0.18	Load a word from the system data area into ACCU 1-L
L	RT 0 to 255	N	N	N	N	N	N	N	X	13		0.68	0.18	Load a word from the extended system data area into ACCU 1-L

¹⁾ The execution time of the substituted operation must be added.

Supplementary Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Load and Transfer Operations (continued)														
Load operations: the value in ACCU 1 is shifted and stored in ACCU 2. Zeros are supplied for unused bits in ACCU 1.														
T	RI 0 to 255	N	N	N	0	N	N	N	15	11		0.57	0.18	Transfer the contents of ACCU 1-L to a word in the interface data area
T	RJ 0 to 255	N	N	N	0	N	N	N	☒	11		0.57	0.18	Transfer the contents of ACCU 1-L to a word of the extended interface data area
T	RS 60 to 63	N	N	N	0	N	N	N	15	11		0.57	0.18	Transfer the contents of ACCU 1-L to a word in the system data area
T	RT 0 to 255	N	N	N	0	N	N	N	☒	11		0.57	0.18	Transfer the contents of ACCU 1-L to a word of the extended system data area

Supplementary Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Conversion Operations														
The data in ACCU 1 is converted.														
CFW	-	N	N	N	N	N	N	N	12	9		0.57	0.18	Form one's complement of ACCU 1-L (bits 0-15)
CSW	-	Y	Y	Y	Y	N	N	N	15	11		0.57	0.18	Form two's complement of ACCU 1-L (bits 0 - 15). Result can be evaluated via CC0/CC1 and OV
CSD	-	Y	Y	Y	Y	N	N	N	25-34 ₁₎	13-19 ₁₎		0.94	0.43	Form two's complement of ACCU 1-L (bits 0 - 31). Result can be evaluated via CC0/CC1 and OV
DEF	-	N	N	N	N	N	N	N	34	14		1.9	0.30	Convert a 16-bit fixed point from BCD into binary
DUF	-	N	N	N	Y	N	N	N	37	15		3.3	0.43	Convert a 16-bit fixed point from binary into BCD
DED	-	N	N	N	N	N	N	N	68-100 ₁₎	31		7.7	0.48	Convert a 32-bit fixed point from BCD into binary
DUD	-	N	N	N	Y	N	N	N	60-79 ₁₎	19-35 ₁₎		9.8	0.62	Convert a 32-bit fixed point from binary into BCD
FDG	-	N	N	N	N	N	N	N	25-54 ₁₎	18-36 ₁₎		5.2	2.6	Convert a fixed-point number (32 bits) into a floating-point number
GFD	-	N	N	N	Y	N	N	N	33-64 ₁₎	15-20 ₁₎		4.4	1.5	Convert a floating-point number into a fixed-point number (32 bits)

¹⁾ The time is dependent on the data in ACCU 1 (non-linear).

Supplementary Operations

Permissible only in function blocks

Opera- tion STL	Operands	Condition codes affected				RLO			Execution times in μs				Function
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			<input type="checkbox"/> = Operation with this		not possible CPU		
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	
Shift and Rotate Operations													
The data in ACCU 1 is shifted or rotated. The bit shifted or rotated last can be evaluated via CC0 and CC1.													
SLW	0-15 ¹⁾	Y	0	0	0	N	N	N	12-19 ₂₎	6-12 ₂₎	1.9	0.32	Shift the contents of ACCU 1-L (word) to the left by the value n specified in the parameter (n = 0 to 15). Positions becoming vacant are padded with zeros.
SRW	0-15 ¹⁾	Y	0	0	0	N	N	N	11-18 ₂₎	6-12 ₂₎	2.0	0.32	Shift the contents of ACCU 1-L (word) to the right by the value n specified in the parameter (n = 0 to 15). Positions becoming vacant are padded with zeros.
SLD	0-32 ¹⁾	Y	0	0	0	N	N	N	23-36 ₂₎	7-23 ₂₎	2.6	0.48	Shift the contents of ACCU 1 (double word) to the left by the value specified in the parameter (n = 0 to 32). Positions becoming vacant are padded with zeros.
SSW	0-15 ¹⁾	Y	0	0	0	N	N	N	21-24 ₂₎	7-13 ₂₎	2.1	0.32	Shift the contents of ACCU 1-L (word) including its sign to the right by the value n specified in the parameter (n = 0 to 15). Positions becoming vacant are padded with the sign (bit 15)
SSD	0-32 ¹⁾	Y	0	0	0	N	N	N	26-38 ₂₎	7-25 ₂₎	3.5	0.48	Shift the contents of ACCU 1 (double word) to the right by the value n specified in the parameter (n = 0 to 32). Positions becoming vacant are padded with the sign (bit 32)
RLD	0-32 ¹⁾	Y	0	0	0	N	N	N	27-44 ₂₎	6-26 ₂₎	2.6	0.48	Rotate ACCU 1 to the left (32 bits wide) from position 0 to 32
RRD	0-32 ¹⁾	Y	0	0	0	N	N	N	26-44 ₂₎	7-26 ₂₎	2.7	0.48	Rotate ACCU 1 to the right (32 bits wide) from position 0 to 32

1) With the operand = "0" an NOP operation is executed; the condition codes are not affected.

2) The time is dependent on the size of the (non-linear) operand.

Supplementary Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function
		CC1	CC0	OV	OS	1 dep. 2 affect. 3 reload			<input type="checkbox"/> = Operation with this not possible CPU				
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	

Jump Operations

The jump operations are executed depending on the RLO (only operation JC) or CC0/CC1 and the OV and OS bits (see Evaluation of CC0 and CC1, page 120)

Operation	Operands	CC1	CC0	OV	OS	RLO 1	RLO 2	RLO 3	CPU 922	CPU 928	CPU 928B	CPU 948	Function
JU=	Symbolic address max. 4 characters	N	N	N	N	N	N	N	10	1.5	1.0	0.59	Unconditional jump to a symbolic address
JC=	Symbolic address max. 4 characters	N	N	N	N	Y	1	Y	13	1/2 ¹⁾	1.0	0.4/0.8 ₁₎	Conditional jump to a symbolic address, executed only if RLO = 1; if RLO = "0", it is set to "1"
JZ=	Symbolic address max. 4 characters	N	N	N	N	N	N	N	12	11/12 ₁₎	1.4	0.4/0.8 ₁₎	Jump if result is "0": the jump is only made if CC1 = 0 and CC0 = 0
JN=	Symbolic address max. 4 characters	N	N	N	N	N	N	N	12	10/13 ₁₎	1.4	0.4/0.8 ₁₎	Jump if result \neq "0": the jump is only made if ²⁾ CC1 = 0 and CC0 = 1 or CC1 = 1 and CC0 = 0 or CC1 = 1 and CC0 = 0
JP=	Symbolic address max. 4 characters	N	N	N	N	N	N	N	13	10/13 ₁₎	1.4	0.4/0.8 ₁₎	Jump if result > "0": the jump is only made if CC1 = 1 and CC0 = 0
JM=	Symbolic address max. 4 characters	N	N	N	N	N	N	N	13	12/14 ₁₎	1.4	0.4/0.8 ₁₎	Jump if result < "0": the jump is only made if CC1 = 0 and CC0 = 1
JO=	Symbolic address max. 4 characters	N	N	N	N	N	N	N	12	11/14 ₁₎	1.4	0.4/0.8 ₁₎	Jump on "overflow": the jump is only made if the OV bit is set.
JOS=	Symbolic address max. 4 characters	N	N	N	0	N	N	N	17	13/14 ₁₎	1.3	0.7/0.9 ₁₎	Jump on "stored overflow": the jump is only made if the OS bit is set

¹⁾ Jump condition: fulfilled/not fulfilled

²⁾ If CC 1 = "1" and CC 0 = "1", not executed for CPU 948

Supplementary Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Other Operations														
IA	-	N	N	N	N	N	N	N	12	25		52	0.30	Disable interrupt: process interrupts are no longer serviced
RA	-	N	N	N	N	N	N	N	12	25		52	0.30	Enable interrupt: cancels the effect of IA
IAE	-	N	N	N	N	N	N	N	☒	☒		☒	0.32	Disable addressing error
RAE	-	N	N	N	N	N	N	N	☒	☒		☒	0.32	Enable addressing error: cancels the effect of IAE
BAS	-	N	N	N	N	Y	N	Y	☒	☒		☒	0.50	Disable output command: PIQ is no longer affected, i.e., the outputs are no longer changed by the S Q, R Q, =Q, T PY, T PW operations.
BAF	-	N	N	N	N	Y	N	Y	☒	☒		☒	0.50	Enable output command: cancels the effect of BAS

Supplementary Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this	not possible CPU				
						1	2	3		CPU 922	CPU 928	CPU 928B		CPU 948
Other Operations (continued)														
D	0-255	N	N	N	N	N	N	N	11	9		0.57	0.18	Decrement the low byte (bits 0 to 7) of ACCU 1 by the value n (n=0 to 255) (without carry)
I	0-255	N	N	N	N	N	N	N	11	9		0.57	0.18	Increment the low byte (bits 0 to 7) of ACCU 1 by the value n (n=0 to 255) (without carry)
ENT	-	N	N	N	N	N	N	N	26	8		0.75	0.39	The contents of the accumulators are restored ¹⁾ .
SED	0-31 ²⁾	Y	0	N	N	N	N	N	32-37	23		4.1 ³⁾	3.0 ³⁾	Set semaphore with the number specified (operation applicable exclusively in multiprocessor mode)
SEE	0-31 ²⁾	Y	0	N	N	N	N	N	32-36	23		4.1 ³⁾	3.1 ³⁾	Enable semaphore with the number specified (operation applicable exclusively in multiprocessor mode)

1) New value of ACCU 1 := Old value of ACCU 1
 ACCU 2 := ACCU 2
 ACCU 3 := ACCU 2
 ACCU 4 := ACCU 3
 The original contents of ACCU 4 are lost.

2) Semaphore locations on the coordinator module

3) Add the waiting time for the bus allocation

Supplementary Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function	
		C	C	O	O	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
		1	0	V	S	1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Other Operations (continued)														
DO=	Formal operand	1)	1)	1)	1)	1)	1)	1)	12-15 ²⁾	15 ²⁾		1.7 ²⁾	0.82 ²⁾	Call block as formal operand (only C DB, JU PB/FB/SB/OB can be substituted)
DO	DW 0 to 255	N	N	N	N	N	N	N	20-26	12-26		3.3	0.84 ²⁾	Process data word: the following operation is executed with the parameter specified in the data word ³⁾
DO	FW 0 to 254	N	N	N	N	N	N	N	19-25	23-26		3.2	0.75 ²⁾	Process flag word: the following operation is executed with the parameter specified in the flag word ³⁾

1) The condition codes are evaluated and changed according to the operation executed.

2) The execution time of the substituted operation must be added.

3) The following operations are possible:

- A., AN., O., ON., S., R.,=.. with the areas I, Q, F and S,
- FR T, R T, SF T, SR T, SP T, SS T, SE T, FR C, R C, S C, CD C, CU C,
- L., T., with the areas P, O, I, Q, F, S, D, RI, RJ, RS and RT,
- L T, L C,
- LC T, LC C,
- JU=, JC=, JZ=, JN=, JP=, JM=, JO=,
- SLW, SRW,
- D, I, SED, SEE,
- C DB, JU., JC., G DB, GX DX, CX DX, DOC FX, DOU FX

System Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function ³⁾	
		C	C	O	O	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
		1	0	V	S	1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Load and Transfer Operations														
LIR	Register no. 0 to 15	N	N	N	N	N	N	N	26-35	2-23		45-51 ₂₎	0.9-2.1 ₂₎	Load register with the contents of a memory word addressed by ACCU 1 ¹⁾
TIR	Register no. 0 to 15	N	N	N	N	N	N	N	26-38	5-19		36-50 ₂₎	0.7-1.9 ₂₎	Transfer register contents into the memory word addressed by ACCU 1 ¹⁾

¹⁾Registers for LIR and TIR (register width = 16 bits)

Reg.-No.	Register designation	
0	ACCU 1-H	high word ACCU 1
1	ACCU 1-L	low word ACCU 1
2	ACCU 2-H	high word ACCU 2
3	ACCU 2-L	low word ACCU 2
5	BSP (only on CPU 922/948)	Block Stack Pointer
6	DBA	Start address of the current data block (address of the first DW)
8	DBL (not on CPU 922)	Length of the current data block (number of data words)
9	ACCU 3-H	high word ACCU 3
10	ACCU 3-L	low word ACCU 3
11	ACCU 4-H	high word ACCU 4
12	ACCU 4-L	low word ACCU 4
15	SAC (not on CPU 948)	Step Address Counter

- Access to the 8-bit memory:
 LIR: the high byte of the register is loaded with FFH (except on CPU 948, S flag and I/Os)
 TIR: the high byte of the register is lost

²⁾Execution time for single processing operation and for immediate bus access in multiprocessing operation. I/Os acknowledge within 0.1 μs or proportionally longer execution time for longer acknowledgement time.

³⁾**Differences in the CPU 948:**
 The operations LIR/TIR operate with 20 bit absolute addresses.

Specifying the address in ACCU 1:

ACCU-1-H: Bit no. 15 to 4= 0
 Bit no. 3 to 0= address bits nos. 19 to 16

ACCU-1-L: Bit no. 15 to 0= address bits nos. 15 to 0

System Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function ³⁾	
		C	C	O	O	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
		1	0	V	S	1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Load and Transfer Operations (continued)														
LDI	Register name ¹⁾	N	N	N	N	N	N	N	☒	☒	☒	☒	1.1-3.2 ₂₎	Load the specified 32 bit register with the contents of a memory word n addressed by ACCU 1-H/L and the following word n+1 ³⁾ : register HIGH = memory word n register LOW = memory word n+1
TDI	Register name ¹⁾	N	N	N	0	N	N	N	☒	☒	☒	☒	1.0-2.4 ₂₎	Transfer the contents of the specified 32 bit register into the memory word n addressed by ACCU 1-H/L and the following word n+1 ³⁾ : memory word n = register HIGH memory word n+1 = register LOW

¹⁾Registers for LDI and TDI (register width = 32 bits)

Reg.-No.	Register designation
A1	ACCU 1
A2	ACCU 2
SA	SAC = STEP address counter
BA	BA register (block start address, bit no. 0 to 19)
BR	BR register (block address register, bit no. 0 to 19)

²⁾Execution time for single processing operation and for immediate bus access in multiprocessing operation. I/Os acknowledge within 0.1 μs or proportionally longer execution time for longer acknowledgement time.

³⁾Specifying the address in ACCU 1:

ACCU-1-H: Bit no. 15 to 4= 0
Bit no. 3 to 0= address bits nos. 19 to 16

ACCU-1-L: Bit no. 15 to 0= address bits nos. 15 to 0

- Access to the 8-bit memory:

LDI: the HIGH byte of the register is loaded with FFH (except on CPU 948, S flag and I/Os)

TDI: the high byte of the register is lost

System Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function (only CPU 922/928/928B)	
		C	C	O	O	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
		1	0	V	S	1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Load and Transfer Operations (continued)														
TNB	Length of area 0 to 255	N	N	N	0 ¹⁾	N	N	N	54 - 1530	66 - 1226		100-1258 ²⁾	☒	Block transfer 0 to 255 bytes ³⁾ : End address of target area in ACCU 1-L End address of source area in ACCU 2-L
TNW	Length of area 0 to 255	N	N	N	0 ¹⁾	N	N	N	55 - 3010	65 - 2340		95-2400 ²⁾	☒	Block transfer 0 to 255 words ³⁾ : End address of target area in ACCU 1-L End address of source area in ACCU 2-L

¹⁾ With CPU 928/928B the OS bit is not influenced by TNB 0/TNW 0.

²⁾ Execution time for single processing operation and for bus access in multiprocessing operation. I/Os acknowledge within 0.1 μs or proportionally longer execution time for longer acknowledgement time.

³⁾ Block transfer operations function decrementally, i.e., the number of words/bytes specified is transferred starting with the end address. Source area and target area must be located completely within one of the following memory areas:

Address area	CPU 922	CPU 928	CPU 928B
0000 - 7FFF (16 bit)	■	■	■
8000 - DD7F (16 bit)		■	■
DD80 - E3FF (16 bit)		■	■
E400 - E7FF (8 bit)			■
E400 - E7FF (16 bit)		■	
E800 - EDFF (16 bit)		■	■
AC00 - EDFF (16 bit)	■		
EE00 - EFFF (8 bit)	■	■	■
F000 - FFFF (8 bit)	■	■	■

A conversion takes place in case of block transfers between 8 and 16 bit memory areas. Two bytes are converted into a word and vice versa.

System Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function (only CPU 948)
		C	C	O	O	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU		
		1	0	V	S	1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	

Load and Transfer Operations (continued)

The block transfer operations of the CPU 948 listed below function with 20 bit absolute addresses. Only these operations can be interrupted by timeout (QVZ) and power failure (NAU).

Operation	Operands	C	C	O	O	RLO 1	RLO 2	RLO 3	CPU 922	CPU 928	CPU 928B	CPU 948	Function
TNW	Length of area 0 to 255	N	N	N	0	N	N	N	☒	☒	☒	2-250 1) 3-560	Block transfer in words in the 16 bit memory area ²⁾
TXB	-	N	N	N	0	N	N	N	☒	☒	☒	3-180 1) 5-480	Block transfer from the 8 bit to the 16 bit memory area ²⁾ . The byte from address n is transferred into the high byte, the byte from address n+1 is transferred into the low byte of the target date.
TXW	-	N	N	N	0	N	N	N	☒	☒	☒	3-180 1) 5-480	Block transfer from the 16 bit to the 8 bit memory area ²⁾ . The high byte of the source date is transferred into the byte with address n, the low byte of the source date is transferred into the byte with address n+1.

1) Execution time for single processing operation and for immediate bus access in multiprocessing operation. I/Os acknowledge within 0.1 μs or proportionally longer execution time for longer acknowledgement time.

Address area of the CPU 948
0 0000 to E FBFF (16 bit)
E A000 to E AFFF (8 bit - S flag)
E FC00 to E FFFF (8 bit)
F 0000 to F FFFF (8/16 bit)

2) Block transfer operations function decrementally, i.e., the number of words specified is transferred starting with the end address. The end address of the target area (20 bit) must be located in ACCU 1, the end address of the source area (20 bit) must be located in ACCU 2. Both the source and the target area must be completely within a memory area listed in the table.

For TXB and TXW ACCU 3 must contain the block length (number of words, 0 to 127).

A conversion takes place in case of block transfers between 8 and 16 bit memory areas. Two bytes are converted into a word and vice versa.

System Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Arithmetic Operations														
ADD	BN -128 to + 127	N	N	N	N	N	N	N	12	11		0.57	0.18	Add byte constant (fixed-point number) to contents of ACCU 1-L (operation includes sign change); ACCUs 2 to 4 remain unchanged
ADD	KF - 32768 to + 32767	N	N	N	N	N	N	N	13	12		1.2	0.39	Add fixed-point constant (word) to contents of ACCU 1-L; ACCUs 2 to 4 remain unchanged
ADD	DH0 to FFFF FFFF	N	N	N	N	N	N	N	20.2 ¹⁾	14		1.7	0.57	Add fixed-point constant (double word) to contents of ACCU 1 ¹⁾ ; ACCUs 2 to 4 remain unchanged
+D	-	Y	Y	Y	Y	N	N	N	38-41 ¹⁾	17-19		1.6	0.64	Add two double word fixed-point numbers ²⁾ ; ACCU 1 + ACCU 2; result can be evaluated via CC0/CC1
-D	-	Y	Y	Y	Y	N	N	N	42-46 ¹⁾	20-23		1.6	0.62	Subtract two double word fixed-point numbers ²⁾ ; ACCU 2 - ACCU 1; result can be evaluated via CC0/CC1

¹⁾ Operation possible from version 09

²⁾ For changes to ACCU 2 and ACCU 3 see Arithmetic Operations, page 38

System Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Jump Operation														
JUR	- 32768 to + 32767	N	N	N	N	N	N	N	15	11		1.2	0.68	Any jump within a function block
Other Operations														
DI	-	1) ¹⁾	1) ¹⁾	1) ¹⁾	1) ¹⁾	1) ¹⁾	1) ¹⁾	1) ¹⁾	15-19 ²⁾	18 ²⁾		1.7 ²⁾	1.1 ²⁾	Execute an operation ³⁾ whose operation code is stored in a formal operand. The number of the formal operand must be stored in ACCU 1.
DO	RS 60 to 63	1) ¹⁾	1) ¹⁾	1) ¹⁾	1) ¹⁾	1) ¹⁾	1) ¹⁾	1) ¹⁾	16 ²⁾	17 ²⁾		0.93 ²⁾	0.71 ²⁾	Execute an operation ³⁾ whose operation code is stored in the system data
TAK	-	N	N	N	N	N	N	N	24	9		0.88	0.18 ²⁾	Swap the contents of ACCU 1 and ACCU 2.

¹⁾ The codes are evaluated and changed according to the operation executed.

²⁾ The execution time of the operation must be added.

³⁾ The following operations are possible:

- A., AN., O., ON., S., R., =.. with the areas I, Q, F, and S,
- FR T, R T, SF T, SR T, SP T, SS T, SE T, FR C, R C, S C, CD C, CU C,
- L., T.. with the areas P, O, I, Q, F, S, D, RI, RJ, RS and RT,
- L T, L C,
- LC T, LC C,
- JU=, JC=, JZ=, JN=, JP=, JM=, JO=,
- SLW, SRW,
- D, I, SED, SEE,
- C DB, JU., JC., G DB, GX DX, CX DX, DOC FX, DOU FX

System Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU		
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	
Other Operations (continued)													
STS	-	N	N	N	N	N	N	N	☒	☒		2.0 ¹⁾	Stop operation with direct transition to SMOOTH STOP mode (communication capability with CPs)
STW	-	N	N	N	N	N	N	N	☒	☒		-	Stop operation resulting in HARD STOP mode (can only be reset by POWER OFF/ POWER ON)
SIM	-	N	N	N	N	N	N	N	☒	☒		0.48	Set interrupt mask (bit pattern in ACCU 1 - 32 bit)
LIM	-	N	N	N	N	N	N	N	☒	☒		0.18	Load interrupt mask (32 bit) into ACCU 1

¹⁾ Add the time for the ISTACK operation (approx. 6.5 μs)

System Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU		
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	
Set Operations													
SU	RS 60.0 to 63.15	N	N	N	N	N	N	Y	☒	☒		0.48	Set a bit in the RS area unconditionally
SU	RT 0.0 to RT 255.15	N	N	N	N	N	N	Y	☒	☒		0.48	Set a bit in the RT area unconditionally
RU	RS 60.0 to 63.15	N	N	N	N	N	N	Y	☒	☒		0.48	Reset a bit in the RS area unconditionally
RU	RT 0.0 to RT 255.15	N	N	N	N	N	N	Y	☒	☒		0.48	Reset a bit in the RT area unconditionally

System Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function (only for CPU 928/928B)
		C1	C0	O	S	1 dep.	2 affect.	3 reload	☒ = Operation with this		not possible CPU		
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	

Register to Register Transfer Operations

These operations transfer the contents of one register into another register.

MAS	-	N	N	N	N	N	N	N	☒	9	0.88	☒	Transfer the contents of ACCU 1 (bits 2^0 to 2^{14}) into the step address counter (SAC)
MAB	-	N	N	N	N	N	N	N	☒	12	0.62	☒	Transfer the contents of ACCU 1 (bits 2^0 to 2^{31}) into the base address register (BR)
MSA	-	N	N	N	N	N	N	N	☒	11	0.69	☒	Transfer the contents of the step address counter (SAC) into ACCU 1
MSB	-	N	N	N	N	N	N	N	☒	11	0.69	☒	Transfer the contents of the step address counter (SAC) into the base address register (BR) ¹⁾
MBA	-	N	N	N	N	N	N	N	☒	12	0.62	☒	Transfer the contents of the base address register (BR) into ACCU 1
MBS	-	N	N	N	N	N	N	N	☒	10	0.88	☒	Transfer the contents of the base address register (BR) (bits 2^0 to 2^{14}) into the step address counter (SAC)

Load, Transfer and Arithmetic Operations with the Base Address Register

The base address register (32 bits) allows address arithmetic and indirect load and transfer operations without using the accumulators for addressing. The following applies:
Absolute address = contents of base address register + constant

MBR	0 to FF FFF	N	N	N	N	N	N	N	☒	15	1.1	☒	Load a 20-bit constant into the base address register ²⁾
ABR	- 32768 to + 32767	N	N	N	N	N	N	N	☒	14	1.1	☒	Add a 16-bit constant to the contents of the base address register

¹⁾ The bits 2^{15} to 2^{31} are set to "0".

²⁾ The bits 2^{20} to 2^{31} of the BR register are set to "0".

System Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function (only for CPU 948)
		C1	C0	O	S	1 dep.	2 affect.	3 reload	☒ = Operation with this		not possible CPU		
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	

Register to Register Transfer Operations

These operations transfer the contents of one register into another register.

MAS	-	N	N	N	N	N	N	N	☒	☒	☒	0.66	Transfer the contents of ACCU 1 (bits 2^0 to 2^{19}) into the step address counter (SAC)
MAB	-	N	N	N	N	N	N	N	☒	☒	☒	0.30	Transfer the contents of ACCU 1 (bits 2^0 to 2^{19}) into the base address register (BR)
MSA	-	N	N	N	N	N	N	N	☒	☒	☒	0.30	Transfer the contents of the step address counter (SAC) into ACCU 1
MSB	-	N	N	N	N	N	N	N	☒	☒	☒	0.18	Transfer the contents of the step address counter (SAC) into the base address register (BR) ¹⁾
MBA	-	N	N	N	N	N	N	N	☒	☒	☒	0.30	Transfer the contents of the base address register (BR) into ACCU 1
MBS	-	N	N	N	N	N	N	N	☒	☒	☒	0.48	Transfer the contents of the base address register (BR) into the step address counter (SAC)

Load, Transfer and Arithmetic Operations with the Base Address Register

The base address register (20 bits) allows address arithmetic and indirect load and transfer operations without using the accumulators for addressing. The following applies:
Absolute address = contents of base address register + constant

MBR	0 to F FFFF	N	N	N	N	N	N	N	☒	☒	☒	0.48	Load a 20-bit constant into the base address register
ABR	- 32768 to + 32767	N	N	N	N	N	N	N	☒	☒	☒	0.39	Add a 16-bit constant to the contents of the base address register

¹⁾ The bits 2^{20} to 2^{31} are set to "0".

System Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Access to local, word-oriented memory: ¹⁾														
LRW	- 32768 to + 32767	N	N	N	N	N	N	N	☒	22		3.6	0.59	Add the constant specified to the contents of the BR register and load the address of the word specified into ACCU 1-L ¹⁾ .
LRD	- 32768 to + 32767	N	N	N	N	N	N	N	☒	30		5.0	0.77	Add the constant specified to the contents of the BR register and load the address of the double word specified into ACCU 1 ¹⁾ .
TRW	- 32768 to + 32767	N	N	N	0	N	N	N	☒	21		3.4	0.59	Add the constant specified to the contents of the BR register and transfer the contents of ACCU 1-L to the address of the word specified ¹⁾ .
TRD	- 32768 to + 32767	N	N	N	0	N	N	N	☒	28		5.0	0.77	Add the constant specified to the contents of the BR register and transfer the contents of ACCU 1 to the address of the double word specified ¹⁾ .
Test/set Busy location (global area): ¹⁾														
TSG	- 32768 to + 32767	Y	Y	0	N	N	N	N	☒	26 ²⁾		4.7 ²⁾	2.9 ²⁾	Add the specified constant to the contents of the BR register, and test and set the Busy location ¹⁾ addressed.

¹⁾ Possible absolute addresses:

	CPU 928/928B	CPU 948
LRW/TRW	0000 to E3FF and E800 to EDFF	0 0000 to E FBFF
LRD/TRD	0000 to E3FE and E800 to EDFE	0 0000 to E FBFE
TSG	0000 to EFFF	F 0000 to F FFFF

²⁾ Execution time for single processor operation and for bus access in multiprocessor operation. I/Os acknowledge within 0.1 μs or proportionally longer execution times for longer acknowledgement time.

System Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function
		C 1	C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU		
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	
Access to global, byte-oriented memory:													
LY GB	- 32768 to + 32767	N	N	N	N	N	N	N	☒	22 ¹⁾	3.0 ¹⁾	1.8 ¹⁾	Add the specified constant to the contents of the BR register and load the byte addressed into ACCU 1-LL ²⁾ .
LY GW	- 32768 to + 32767	N	N	N	N	N	N	N	☒	26 ¹⁾	3.9 ¹⁾	2.4 ¹⁾	Add the specified constant to the contents of the BR register and load the word addressed into ACCU 1-L ²⁾ .
LY GD	- 32768 to + 32767	N	N	N	N	N	N	N	☒	31 ¹⁾	5.5 ¹⁾	4.4 ¹⁾	Add the specified constant to the contents of the BR register and load the double word addressed into ACCU 1 ²⁾ .
TY GB	- 32768 to + 32767	N	N	N	0	N	N	N	☒	21 ¹⁾	2.9 ¹⁾	1.8 ¹⁾	Add the specified constant to the contents of the BR register and transfer the contents of ACCU 1-LL to the byte addressed ²⁾ .
TY GW	- 32768 to + 32767	N	N	N	0	N	N	N	☒	25 ¹⁾	3.7 ¹⁾	2.5 ¹⁾	Add the specified constant to the contents of the BR register and transfer the contents of ACCU 1-L to the word addressed ²⁾ .
TY GD	- 32768 to + 32767	N	N	N	0	N	N	N	☒	30 ¹⁾	5.3 ¹⁾	4.0 ¹⁾	Add the specified constant to the contents of the BR register and transfer the contents of ACCU 1 to the double word addressed ²⁾ .

¹⁾ Execution time for single processor operation and for bus access in multiprocessor operation. I/Os acknowledge within 0.1 μ s or proportionally longer execution times for longer acknowledgement time.

²⁾ Possible absolute addresses:

	CPU 928/928B	CPU 948
LY GB/TY GW	0000 to EFFF	F 0000 to F FFFF
LY GW/TY GW	0000 to EFFE	F 0000 to F FFFE
LY GD/TY GD	0000 to EFFC	F 0000 to F FFFC

System Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Access to global, word-oriented memory:														
LW GW	- 32768 to + 32767	N	N	N	N	N	N	N	☒	27 ¹⁾		4.3 ¹⁾	1.8 ¹⁾	Add the specified constant to the contents of the BR register and load the word addressed into ACCU 1-L ²⁾ .
LW GD	- 32768 to + 32767	N	N	N	N	N	N	N	☒	33 ¹⁾		5.7 ¹⁾	2.4 ¹⁾	Add the specified constant to the contents of the BR register and load the double word addressed into ACCU 1-L ²⁾ .
TW GW	- 32768 to + 32767	N	N	N	0	N	N	N	☒	26 ¹⁾		4.0 ¹⁾	1.8 ¹⁾	Add the specified constant to the contents of the BR register and load the word addressed into ACCU 1-L ²⁾ .
TW GD	- 32768 to + 32767	N	N	N	0	N	N	N	☒	32 ¹⁾		5.4 ¹⁾	2.5 ¹⁾	Add the specified constant to the contents of the BR register and transfer the contents of ACCU 1 to the double word addressed ²⁾ .
Open page:														
ACR	-	N	N	N	N	N	N	N	☒	11 ¹⁾		0.57 ¹⁾	0.32 ¹⁾	Open the page whose number is in ACCU 1-L ³⁾ .
Test/set Busy location (page area):														
TSC	- 32768 to + 32767	Y	Y	0	N	N	N	N	☒	29 ¹⁾		5.3 ¹⁾	3.6 ¹⁾	Add the specified constant to the contents of the BR register and test/set the Busy location ²⁾ addressed on the page opened.

¹⁾ Execution time for single processor operation and for bus access in multiprocessor operation. I/Os acknowledge within 0.1 μ s or proportionally longer execution times for longer acknowledgement time.

²⁾ Possible absolute addresses:

	CPU 928/928B	CPU 948
LW GB/TW GW	0000 to EFFF	F 0000 to F FFFF
LW GW/ TW GW	0000 to EFFE	F 0000 to F FFEE
TSC	F400 to FBFF	F F400 to F FBFF

³⁾ Possible values: 0 to 255

System Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μs				Function
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU		
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948	
Access to byte-oriented pages:													
LY CB	- 32768 to + 32767	N	N	N	N	N	N	N	☒	29 ¹⁾	3.6 ¹⁾	2.6 ¹⁾	Add the specified constant to the contents of the BR register and load the byte addressed from the page opened into ACCU 1-LL ²⁾ .
LY CW	- 32768 to + 32767	N	N	N	N	N	N	N	☒	30 ¹⁾	4.5 ¹⁾	3.4 ¹⁾	Add the specified constant to the contents of the BR register and load the word addressed from the page opened into ACCU 1-L ²⁾ .
LY CD	- 32768 to + 32767	N	N	N	N	N	N	N	☒	34 ¹⁾	6.1 ¹⁾	5.2 ¹⁾	Add the specified constant to the contents of the BR register and load the double word addressed from the page opened into ACCU 1 ²⁾ .
TY CB	- 32768 to + 32767	N	N	N	0	N	N	N	☒	28 ¹⁾	3.5 ¹⁾	2.5 ¹⁾	Add the specified constant to the contents of the BR register and transfer the contents of ACCU 1-LL to the byte addressed on the page opened ²⁾ .
TY CW	- 32768 to + 32767	N	N	N	0	N	N	N	☒	29 ¹⁾	4.2 ¹⁾	3.3 ¹⁾	Add the specified constant to the contents of the BR register and transfer the contents of ACCU 1-L to the word addressed on the page opened ²⁾ .
TY CD	- 32768 to + 32767	N	N	N	0	N	N	N	☒	34 ¹⁾	5.9 ¹⁾	4.8 ¹⁾	Add the specified constant to the contents of the BR register and transfer the contents of ACCU 1 to the double word addressed on the page opened ²⁾ .

¹⁾ Execution time for single processor operation and for bus access in multiprocessor operation. I/Os acknowledge within 0.1 μs or proportionally longer execution times for longer acknowledgement time.

²⁾ Possible absolute addresses:

	CPU 928/928B	CPU 948
LY CB/TY CB	F400 to FBFF	F F400 to F FBFF
LY CW/TY CW	F400 to FBFE	F F400 to F FBFE
LY CD/TY CD	F400 to FBFC	F F400 to F FBFC

System Operations

Permissible only in function blocks

Operation STL	Operands	Condition codes affected				RLO			Execution times in μ s				Function	
		C C 1	C C 0	O V	O S	1 dep. 2 affect. 3 reload			☒ = Operation with this		not possible CPU			
						1	2	3	CPU 922	CPU 928	CPU 928B	CPU 948		
Access to word-oriented pages: ¹⁾														
LW CW	- 32768 to + 32767	N	N	N	N	N	N	N	☒	34 ¹⁾		4.9 ¹⁾	2.6 ¹⁾	Add the specified constant to the contents of the BR register and load the word addressed with the contents of the BR register from the page opened into ACCU 1-L ²⁾ .
LW CD	- 32768 to + 32767	N	N	N	N	N	N	N	☒	38 ¹⁾		6.3 ¹⁾	3.4 ¹⁾	Add the specified constant to the contents of the BR register and load the double word addressed with the contents of the BR register from the page opened into ACCU 1 ²⁾ .
TW CW	- 32768 to + 32767	N	N	N	0	N	N	N	☒	33 ¹⁾		4.7 ¹⁾	2.5 ¹⁾	Add the specified constant to the contents of the BR register and transfer the contents of ACCU 1-L to the word addressed with the contents of the BR register on the page opened ²⁾ .
TW CD	- 32768 to + 32767	N	N	N	0	N	N	N	☒	37 ¹⁾		6.0 ¹⁾	3.3 ¹⁾	Add the specified constant to the contents of the BR register and transfer the contents of ACCU 1 to the double word addressed with the contents of the BR register on the page opened ²⁾ .

¹⁾ Execution time for single processor operation and for bus access in multiprocessor operation. I/Os acknowledge within 0.1 μ s or proportionally longer execution times for longer acknowledgement time.

²⁾ Possible absolute addresses:

	CPU 928/928B	CPU 948
LW CW/TW CW	F400 to FBFF	F F400 to F FBFF
LW CD/TW CD	F400 to FBFE	F F400 to F FBFE

Machine Code Listing

Explanation of subscripts

- a + byte address
- b + bit address
- c + formal operand address
- d + operand value
- e + constant
- f + block number
- g + word address
- h + number of shifts
- i + relative jump destination address
- k + register number
- l + block length in bytes
- m + jump displacement (16 bits)
- n + semaphore number
- o + block length in words

B0 to B5: 1st to 6th machine code byte

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
0	0	0	0					NOP 0	
0	1	0	0					CFW	
0	2	0 _d	0 _d					L	T
0	3	0 _l	0 _l					TNB	
0	4	0 _d	0 _d					FR	T
0	5	0	0					BEC	
0	6	0 _c	0 _c					FR=	
0	7	0 _c	0 _c					A=	
0	8	0	0					IA	
0	8	8	0					RA	
0	9	0	0					CSW	

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
0	A	0 _d	0 _d					L	FY
0	B	0 _d	0 _d					T	FY
0	C	0 _d	0 _d					LD	T
0	D	0 _i	0 _i					JO=	
0	E	0 _c	0 _c					LD=	
0	F	0 _c	0 _c					O=	
1	0	0 _e	0 _e					BLD	0-255
1	0	8	2					BLD	130
1	0	8	3					BLD	131
1	0	8	4					BLD	132
1	0	8	5					BLD	133
1	0	F	F					BLD	255
1	1	0 _e	0 _e					I	
1	2	0 _d	0 _d					L	FW
1	3	0 _d	0 _d					T	FW
1	4	0 _d	0 _d					SF	T
1	5	0 _i	0 _i					JP=	
1	6	0 _c	0 _c					SFD=	
1	7	0 _c	0 _c					S=	
1	8	0 _d	0 _d					DO	RS
1	9	0 _e	0 _e					D	
1	A	0 _d	0 _d					L	FD
1	B	0 _d	0 _d					T	FD
1	C	0 _d	0 _d					SE	T
1	D	0 _f	0 _f					JC	FB

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
1	E	0 _c	0 _c					SEC=	
1	F	0 _c	0 _c					= =	
2	0	0 _f	0 _f					C	DB
2	1	2	0					>F	
2	1	4	0					<F	
2	1	6	0					><F	
2	1	8	0					!=F	
2	1	A	0					>=F	
2	1	C	0					<=F	
2	2	0 _d	0 _d					L	DL
2	3	0 _d	0 _d					T	DL
2	4	0 _d	0 _d					SD	T
2	5	0 _i	0 _i					JM=	
2	6	0 _c	0 _c					SD=	
2	7	0 _c	0 _c					AN=	
2	8	0 _e	0 _e					L	KB
2	9	0 _h	0 _h					SLD	
2	A	0 _d	0 _d					L	DR
2	B	0 _d	0 _d					T	DR
2	C	0 _d	0 _d					SS	T
2	D	0 _i	0 _i					JU=	
2	E	0 _c	0 _c					SSU=	
2	F	0 _c	0 _c					ON=	
3	0	0	1	0 _e	0 _e	0 _e	0 _e	L	KC
3	0	0	2	0 _e	0 _e	0 _e	0 _e	L	KT

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
3	0	0	4	0 _e	0 _e	0 _e	0 _e	L	KF
3	0	1	0	0 _e	0 _e	0 _e	0 _e	L	KS
3	0	2	0	0 _e	0 _e	0 _e	0 _e	L	KY
3	0	4	0	0 _e	0 _e	0 _e	0 _e	L	KH
3	0	8	0	0 _e	0 _e	0 _e	0 _e	L	KM
3	1	2	0					>G	
3	1	4	0					<G	
3	1	6	0					><G	
3	1	8	0					!=G	
3	1	A	0					>=G	
3	1	C	0					<=G	
3	2	0 _d	0 _d					L	DW
3	3	0 _d	0 _d					T	DW
3	4	0 _d	0 _d					SP	T
3	5	0 _i	0 _i					JN=	
3	6	0 _c	0 _c					SP=	
3	7	0 _c	0 _c					RB=	
3	8	0	0	0 _e	0 _e	0 _e	0 _e	L	KG ¹⁾
3	8	4	0	0 _e	0 _e	0 _e	0 _e	L	DH ¹⁾
3	9	2	0					>D	
3	9	4	0					<D	
3	9	6	0					><D	
3	9	8	0					!=D	
3	9	A	0					>=D	

¹⁾ 3-word command with B4 and B5, filled with 0_e

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
3	9	C	0					<=D	
3	A	0 _d	0 _d					L	DD
3	B	0 _d	0 _d					T	DD
3	C	0 _d	0 _d					R	T
3	D	0 _f	0 _f					JU	FB
3	E	0 _c	0 _c					RD=	
3	F	0 _c	0 _c					LW=	
4	0	0	0 _k					LIR	
4	1	0	0					AW	
4	2	0 _d	0 _d					L	C
4	3	0 _o	0 _o					TNW	
4	4	0 _d	0 _d					FR	C
4	5	0 _i	0 _i					JZ=	
4	6	0 _c	0 _c					L=	
4	7	0 _d	0 _d					L	RJ
4	8	0	0 _k					TIR	
4	9	0	0					OW	
4	A	0 _d	0 _d					L	IB
4	A	8 _d	0 _d					L	QB
4	B	0 _d	0 _d					T	IB
4	B	8 _d	0 _d					T	QB
4	C	0 _d	0 _d					LD	C
4	D	0 _f	0 _f					JC	OB
4	E	0 _d	0 _d					DO	FW
4	F	0 _d	0 _d					L	RT

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
5	0	0 _e	0 _e					ADD	BN
5	1	0	0					XOW	
5	2	0 _d	0 _d					L	IW
5	2	8 _d	0 _d					L	QW
5	3	0 _d	0 _d					T	IW
5	3	8 _d	0 _d					T	QW
5	4	0 _d	0 _d					CD	C
5	5	0 _f	0 _f					JC	PB
5	6	0 _c	0 _c					LDW=	
5	7	0 _d	0 _d					L	OW
5	8	0	0	0 _e	0 _e	0 _e	0 _e	ADD	KF
5	9	0	0					-F	
5	A	0 _d	0 _d					L	ID
5	A	8 _d	0 _d					L	QD
5	B	0 _d	0 _d					T	ID
5	B	8 _d	0 _d					T	QD
5	C	0 _d	0 _d					S	C
5	D	0 _f	0 _f					JC	SB
5	F	0 _d	0 _d					L	OY
6	0	0	0					:F	
6	0	0	3					:G	
6	0	0	4					xF	
6	0	0	5	0 _e	0 _e	0 _e	0 _e	ADD	DH ¹⁾
6	0	0	7					xG	

¹⁾ 3-word command with B4 und B5, filled with 0_e

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
6	0	0	8					ENT	
6	0	0	9					-D	
6	0	0	B					-G	
6	0	0	C	0	0	0 _i	0 _i	JOS=	
6	0	0	D					+D	
6	0	0	F					+G	
6	1	0	0 _h					SLW	
6	2	0 _d	0 _d					L	RS
6	3	0 _d	0 _d					T	RS
6	4	0 _h	0 _h					RLD	
6	5	0	0					BE	
6	5	0	1					BEU	
6	6	0 _c	0 _c					T=	
6	7	0 _d	0 _d					T	RJ
6	8	0	0	0 _e	0 _e	0 _e	0 _e	LRW	
6	8	0 _h	1					SSW	
6	8	0	2					GFD	
6	8	0	3	0 _e	0 _e	0 _e	0 _e	TRW	
6	8	0	4	0 _e	0 _e	0 _e	0 _e	LRD	
6	8	0	5	0 _e	0 _e	0 _e	0 _e	TRD	
6	8	0	6					FDG	
6	8	0	7					CSD	
6	8	0	8					DUF	
6	8	0	A					DUD	
6	8	0	B					LDI	A1

Machine Code								Operation	Operand
B0		B2		B3		B4			
L	R	L	R	L	R	L	R		
6	8	0	C					DEF	
6	8	0	E					DED	
6	8	0	F					TDI	A1
6	8	1	9					MAS	
6	8	2	9					MAB	
6	8	2	B					LDI	A2
6	8	2	F					TDI	A2
6	8	4	9					MSA	
6	8	4	B					LDI	SA
6	8	4	F					TDI	SA
6	8	6	9					MSB	
6	8	8	9					MBA	
6	8	9	9					MBS	
6	8	9	B					LDI	BA
6	8	9	F					TDI	BA
6	8	A	B					LDI	BR
6	8	A	F					TDI	BR
6	9	0	0 _h					SRW	
6	A	0 _d	0 _d					L	RI
6	B	0 _d	0 _d					T	RI
6	C	0 _d	0 _d					CU	C
6	D	0 _f	0 _f					JU	OB
6	E	0 _d	0 _d					DO	DW
6	F	0 _d	0 _d					T	RT
7	0	0	0					STS	

Machine Code								Operation	Operand
B0		B2		B3		B4			
L	R	L	R	L	R	L	R		
7	0	0	2					TAK	
7	0	0	3					STP	
7	0	0	4					STW	
7	0	0	B	0 _m	0 _m	0 _m	0 _m	JUR	
7	0	0	C					LIM	
7	0	0	D					SIM	
7	0	0	E	0	0 _b	0 _g	0 _g	RU	RT
7	0	0	E	4	0 _b	0 _g	0 _g	SU	RT
7	0	0	E	8	0 _b	0 _g	0 _g	TBN	RT
7	0	0	E	C	0 _b	0 _g	0 _g	TB	RT
7	0	0	F					TXW	
7	0	1	5	0	0 _b	0 _g	0 _g	RU	C
7	0	1	5	4	0 _b	0 _g	0 _g	SU	C
7	0	1	5	8	0 _b	0 _g	0 _g	TBN	C
7	0	1	5	C	0 _b	0 _g	0 _g	TB	C
7	0	1	E	0	0 _b	0 _g	0 _g	RU	RJ
7	0	1	E	4	0 _b	0 _g	0 _g	SU	RJ
7	0	1	E	8	0 _b	0 _g	0 _g	TBN	RJ
7	0	1	E	C	0 _b	0 _g	0 _g	TB	RJ
7	0	1	F					TXB	
7	0	2	5	0	0 _b	0 _g	0 _g	RU	T
7	0	2	5	4	0 _b	0 _g	0 _g	SU	T
7	0	2	5	8	0 _b	0 _g	0 _g	TBN	T
7	0	2	5	C	0 _b	0 _g	0 _g	TB	T
7	0	3	8	0	0 _b	0 _a	0 _a	RU	I

Machine Code								Operation	Operand
B0		B2		B3		B4			
L	R	L	R	L	R	L	R		
7	0	3	8	0	0 _b	8 _a	0 _a	RU	Q
7	0	3	8	4	0 _b	0 _a	0 _a	SU	I
7	0	3	8	4	0 _b	8 _a	0 _a	SU	Q
7	0	3	8	8	0 _b	0 _a	0 _a	TBN	I
7	0	3	8	8	0 _b	8 _a	0 _a	TBN	Q
7	0	3	8	C	0 _b	0 _a	0 _a	TB	I
7	0	3	8	C	0 _b	8 _a	0 _a	TB	Q
7	0	4	6	0	0 _b	0 _g	0 _g	RU	D
7	0	4	6	4	0 _b	0 _g	0 _g	SU	D
7	0	4	6	8	0 _b	0 _g	0 _g	TBN	D
7	0	4	6	C	0 _b	0 _g	0 _g	TB	D
7	0	4	7	0	0 _b	0 _g	0 _g	RU	RI
7	0	4	7	4	0 _b	0 _g	0 _g	SU	RI
7	0	4	7	8	0 _b	0 _g	0 _g	TBN	RI
7	0	4	7	C	0 _b	0 _g	0 _g	TB	RI
7	0	4	9	0	0 _b	0 _a	0 _a	RU	F
7	0	4	9	4	0 _b	0 _a	0 _a	SU	F
7	0	4	9	8	0 _b	0 _a	0 _a	TBN	F
7	0	4	9	C	0 _b	0 _a	0 _a	TB	F
7	0	5	7	0	0 _b	0 _g	0 _g	RU	RS
7	0	5	7	4	0 _b	0 _g	0 _g	SU	RS
7	0	5	7	8	0 _b	0 _g	0 _g	TBN	RS
7	0	5	7	C	0 _b	0 _g	0 _g	TB	RS
7	1	0 _h	0 _h					SSD	
7	2	0 _d	0 _d					L	PY

Machine Code								Operation	Operand
B0		B2		B3		B4			
L	R	L	R	L	R	L	R		
7	3	0 _d	0 _d					T	PY
7	4	0 _h	0 _h					RRD	
7	5	0 _f	0 _f					JU	PB
7	6	0 _c	0 _c					DO=	
7	7	0 _d	0 _d					T	OW
7	8	0	0					IAE	
7	8	0	1	0	1	0 _f	0 _c	DOU	FX
7	8	0	2	0	9	0 _f	0 _c	DOC	FX
7	8	0	3	1	1	0 _f	0 _f	CX	DX
7	8	0	4	0	0	0 _f	0 _f	GX	DX
7	8	0	5	0	0	0 _f	0 _f	G	DB
7	8	0	6	0	0	0 _n	0 _n	SED	
7	8	0	7	0	0	0 _n	0 _n	SEE	
7	8	0 _e	9	0 _e	0 _e	0 _e	0 _e	MBR	
7	8	0	A	0 _o	0 _o	0 _o	0 _o	ABR	
7	8	0	B	0 _b	0 _a	0 _a	0 _a	A	S
7	8	0	D	0 _e	0 _e	0 _e	0 _e	LYCB	
7	8	0	E	0 _e	0 _e	0 _e	0 _e	LYGB	
7	8	1	0					RAE	
7	8	1	B	0 _b	0 _a	0 _a	0 _a	O	S
7	8	1	D	0 _e	0 _e	0 _e	0 _e	LYCW	
7	8	1	E	0 _e	0 _e	0 _e	0 _e	LYGW	
7	8	2	B	0 _b	0 _a	0 _a	0 _a	S	S
7	8	2	D	0 _e	0 _e	0 _e	0 _e	LYCD	
7	8	2	E	0 _e	0 _e	0 _e	0 _e	LYGD	

Machine Code								Operation	Operand
B0		B2		B3		B4			
L	R	L	R	L	R	L	R		
7	8	3	B	0 _b	0 _a	0 _a	0 _a	=	S
7	8	3	D					ACR	
7	8	3	F	0	0 _b	0 _g	0 _g	A	D
7	8	3	F	1	0 _b	0 _g	0 _g	O	D
7	8	3	F	2	0 _b	0 _g	0 _g	AN	D
7	8	3	F	3	0 _b	0 _g	0 _g	ON	D
7	8	3	F	4	0 _b	0 _g	0 _g	S	D
7	8	3	F	5	0 _b	0 _g	0 _g	R	D
7	8	3	F	6	0 _b	0 _g	0 _g	=	D
7	8	4	B	0 _b	0 _a	0 _a	0 _a	AN	S
7	8	5	B	0 _b	0 _a	0 _a	0 _a	ON	S
7	8	5	D	0 _e	0 _e	0 _e	0 _e	LWCW	
7	8	5	E	0 _e	0 _e	0 _e	0 _e	LWGW	
7	8	6	B	0 _b	0 _a	0 _a	0 _a	R	S
7	8	6	D	0 _e	0 _e	0 _e	0 _e	LWCD	
7	8	6	E	0 _e	0 _e	0 _e	0 _e	LWGD	
7	8	8	D	0 _e	0 _e	0 _e	0 _e	TYCB	
7	8	8	E	0 _e	0 _e	0 _e	0 _e	TYGB	
7	8	9	D	0 _e	0 _e	0 _e	0 _e	TYCW	
7	8	9	E	0 _e	0 _e	0 _e	0 _e	TYGW	
7	8	A	B	0	0 _d	0 _d	0 _d	L	SY
7	8	A	D	0 _e	0 _e	0 _e	0 _e	TYCD	
7	8	A	E	0 _e	0 _e	0 _e	0 _e	TYGD	
7	8	B	B	0	0 _d	0 _d	0 _d	T	SY
7	8	C	B	0	0 _d	0 _d	0 _d	L	SW

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
7	8	C	D	0 _e	0 _e	0 _e	0 _e	TSC	
7	8	C	E	0 _e	0 _e	0 _e	0 _e	TSG	
7	8	D	B	0	0 _d	0 _d	0 _d	T	SW
7	8	D	D	0 _e	0 _e	0 _e	0 _e	TWCW	
7	8	D	E	0 _e	0 _e	0 _e	0 _e	TWGW	
7	8	E	B	0	0 _d	0 _d	0 _d	L	SD
7	8	E	D	0 _e	0 _e	0 _e	0 _e	TWCD	
7	8	E	E	0 _e	0 _e	0 _e	0 _e	TWGD	
7	8	F	B	0	0 _d	0 _d	0 _d	T	SD
7	9	0	0					+F	
7	A	0 _d	0 _d					L	PW
7	B	0 _d	0 _d					T	PW
7	C	0 _d	0 _d					R	C
7	D	0 _f	0 _f					JU	SB
7	E	0	0					DI	
7	F	0 _d	0 _d					T	OY
8	0 _b	0 _a	0 _a					A	F
8	8 _b	0 _a	0 _a					O	F
9	0 _b	0 _a	0 _a					S	F
9	8 _b	0 _a	0 _a					=	F
A	0 _b	0 _a	0 _a					AN	F
A	8 _b	0 _a	0 _a					ON	F
B	0 _b	0 _a	0 _a					R	F
B	8	0 _d	0 _d					A	C
B	9	0 _d	0 _d					O	C

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
B	A	0	0					A(
B	B	0	0					O(
B	C	0 _d	0 _d					AN	C
B	D	0 _d	0 _d					ON	C
B	E	0	0					BAS	
B	F	0	0)	
C	0 _b	0 _a	0 _a					A	I
C	0 _b	8 _a	0 _a					A	Q
C	8 _b	0 _a	0 _a					O	I
C	8 _b	8 _a	0 _a					O	Q
D	0 _b	0 _a	0 _a					S	I
D	0 _b	8 _a	0 _a					S	Q
D	8 _b	0 _a	0 _a					=	I
D	8 _b	8 _a	0 _a					=	Q
E	0 _b	0 _a	0 _a					AN	I
E	0 _b	8 _a	0 _a					AN	Q
E	8 _b	0 _a	0 _a					ON	I
E	8 _b	8 _a	0 _a					ON	Q
F	0 _b	0 _a	0 _a					R	I
F	0 _b	8 _a	0 _a					R	Q
F	8	0 _d	0 _d					A	T
F	9	0 _d	0 _d					O	T
F	A	0 _i	0 _i					JC=	
F	B	0	0					O	
F	C	0 _d	0 _d					AN	T

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
F	D	0 _d	0 _d					ON	T
F	E	0	0					BAF	
F	F	F	F					NOP 1	

Alphabetical Index of Operations

(with Machine Code)

For explanation of subscripts see page 116.

Operation	Operand	Page	Machine Code
A	C	10	B 8 0 _d 0 _d
	D	10	7 8 3 F 0 0 _b 0 _g 0 _g
	F	10	8 0 _b 0 _a 0 _a
	I	10	C 0 _b 0 _a 0 _a
	Q	10	C 0 _b 8 _a 0 _a
	S	10	7 8 0 B 0 _b 0 _a 0 _a 0 _a
	T	10	F 8 0 _d 0 _d
A(---	16	B A 0 0
A =	Formal oper.	56	0 7 0 _c 0 _c
ABR	Constant	102	7 8 0 A 0 _o 0 _o 0 _o 0 _o
ACR	---	110	7 8 3 D
ADD	BN	94	5 0 0 _e 0 _e
	DH	94	6 0 0 5 0 _e 0 _e 0 _e 0 _e 0 _e 0 _e 0 _e 0 _e
	KF	94	5 8 0 0 0 _e 0 _e 0 _e 0 _e
AN	C	12	B C 0 _d 0 _d
	D	12	7 8 3 F 4 0 _b 0 _g 0 _g
	F	12	A 0 _b 0 _a 0 _a

Operation	Operand	Page	Machine Code
AN	I	10	E 0 _b 0 _a 0 _a
	Q	10	E 0 _b 8 _a 0 _a
	S	12	7 8 4 B 0 _b 0 _a 0 _a 0 _a
	T	12	F C 0 _d 0 _d
AN=	Formal oper.	56	2 7 0 _c 0 _c
AW	---	56	4 1 0 0
BAF	---	80	F E 0 0
BAS	---	80	B E 0 0
BE	---	52	6 5 0 0
BEC	---	52	0 5 0 0
BEU	---	52	6 5 0 1
BLD	0 - 255	54	1 0 0 _e 0 _e
	130	54	1 0 8 2
	131	54	1 0 8 3
	132	54	1 0 8 4
	133	54	1 0 8 5
	255	54	1 0 F F
C	DB	50	2 0 0 _f 0 _f
CD	C	36	5 4 0 _d 0 _d
CFW	---	74	0 1 0 0
CSD	---	74	6 8 0 7
CSW	---	74	0 9 0 0
CU	C	36	6 C 0 _d 0 _d
CX	DX	50	7 8 0 3 1 1 0 _f 0 _f
D	0 - 255	82	1 9 0 _e 0 _e
DED	---	74	6 8 0 E
DEF	---	74	6 8 0 C

Operation	Operand	Page	Machine Code
DI	---	96	7 E 0 0
DO	DW	84	6 E 0 _d 0 _d
	FW	84	4 E 0 _d 0 _d
	RS	96	1 8 0 _d 0 _d
DO =	Formal oper.	84	7 6 0 _c 0 _c
DOC	FX	50	7 8 0 2 0 9 0 _f 0 _f
DOU	FX	48	7 8 0 1 0 1 0 _f 0 _f
DUD	---	74	6 8 0 A
DUF	---	74	6 8 0 8
ENT	---	82	6 0 0 8
FDG	---	74	6 8 0 6
FR	C	68	4 4 0 _d 0 _d
	T	68	0 4 0 _d 0 _d
FR =	Formal oper.	66	0 6 0 _c 0 _c
G	DB	52	7 8 0 5 0 0 0 _f 0 _f
GFD	---	74	6 8 0 2
GX	DX	52	7 8 0 4 0 0 0 _f 0 _f
I	0 - 255	82	1 1 0 _e 0 _e
IA	---	80	0 8 0 0
IAE	---	80	7 8 0 0
JC	FB	48	1 D 0 _f 0 _f
	OB	50	4 D 0 _f 0 _f
	PB	48	5 5 0 _f 0 _f
	SB	50	5 D 0 _f 0 _f
JC =	Symb. addr.	78	F A 0 _i 0 _i
JM =	Symb. addr.	78	2 5 0 _i 0 _i
JN =	Symb. addr.	78	3 5 0 _i 0 _i
JO =	Symb. addr.	78	0 D 0 _i 0 _i

Operation	Operand	Page	Machine Code
JOS =	Symb. addr.	78	6 0 0 C 0 0 0 _i 0 _i
JP =	Symb. addr.	78	1 5 0 _i 0 _i
JU	FB	48	3 D 0 _f 0 _f
	OB	48	6 D 0 _f 0 _f
	PB	48	7 5 0 _f 0 _f
	SB	48	7 D 0 _f 0 _f
JU =	Symb. addr.	78	2 D 0 _i 0 _i
JUR	Constant	96	7 0 0 B 0 _m 0 _m 0 _m 0 _m
JZ =	Symb. addr.	78	4 5 0 _i 0 _i
L	C	26	4 2 0 _d 0 _d
	DD	24	3 A 0 _d 0 _d
	DH	22	3 8 4 0 0 _e 0 _e 0 _e 0 _e 0 _e 0 _e 0 _e 0 _e
	DL	22	2 2 0 _d 0 _d
	DR	24	2 A 0 _d 0 _d
	DW	24	3 2 0 _d 0 _d
	FD	22	1 A 0 _d 0 _d
	FW	20	1 2 0 _d 0 _d
	FY	20	0 A 0 _d 0 _d
	IB	20	4 A 0 _d 0 _d
	ID	20	5 A 0 _d 0 _d
	IW	20	5 2 0 _d 0 _d
	KB	24	2 8 0 _e 0 _e
	KC	24	3 0 0 1 0 _e 0 _e 0 _e 0 _e
	KF	24	3 0 0 4 0 _e 0 _e 0 _e 0 _e
	KG	24	3 8 0 0 0 _e 0 _e 0 _e 0 _e 0 _e 0 _e 0 _e 0 _e

Operation	Operand	Page	Machine Code
L	KH	24	3 0 4 0 0 _e 0 _e 0 _e 0 _e
	KM	24	3 0 8 0 0 _e 0 _e 0 _e 0 _e
	KS	24	3 0 1 0 0 _e 0 _e 0 _e 0 _e
	KT	24	3 0 0 2 0 _e 0 _e 0 _e 0 _e
	KY	24	3 0 2 0 0 _e 0 _e 0 _e 0 _e
	OW	26	5 7 0 _d 0 _d
	OY	26	5 F 0 _d 0 _d
	PW	26	7 A 0 _d 0 _d
	PY	26	7 2 0 _d 0 _d
	QB	20	4 A 8 _d 0 _d
	QD	20	5 A 8 _d 0 _d
	QW	20	5 2 8 _d 0 _d
	RI	70	6 A 0 _d 0 _d
	RJ	70	4 7 0 _d 0 _d
	RS	70	6 2 0 _d 0 _d
	RT	70	4 F 0 _d 0 _d
	SD	22	7 8 E B 0 0 _d 0 _d 0 _d
	SW	22	7 8 C B 0 0 _d 0 _d 0 _d
	SY	22	7 8 A B 0 0 _d 0 _d 0 _d
	T	26	0 2 0 _d 0 _d
L =	Formal oper.	70	4 6 0 _c 0 _c
LC	C	26	4 C 0 _d 0 _d
	T	26	0 C 0 _d 0 _d
LDI	A1	88	6 8 0 B
	A2	88	6 8 2 B
	BA	88	6 8 9 B

Operation	Operand	Page	Machine Code
LDI	BR	88	6 8 A B
	SA	88	6 8 4 B
LD =	Formal oper.	70	0 E 0 _c 0 _c
LDW =	Formal oper.	70	5 6 0 _c 0 _c
LIM	---	98	7 0 0 C
LIR	Register no.	86	4 0 0 0 _k
LRD	Constant	106	6 8 0 4 0 _e 0 _e 0 _e 0 _e
LRW	Constant	106	6 8 0 0 0 _e 0 _e 0 _e 0 _e
LW =	Formal oper.	70	3 F 0 _c 0 _c
LW CD	Constant	114	7 8 6 D 0 _e 0 _e 0 _e 0 _e
LW CW	Constant	114	7 8 5 D 0 _e 0 _e 0 _e 0 _e
LW GD	Constant	110	7 8 6 E 0 _e 0 _e 0 _e 0 _e
LW GW	Constant	110	7 8 5 E 0 _e 0 _e 0 _e 0 _e
LY CB	Constant	112	7 8 0 D 0 _e 0 _e 0 _e 0 _e
LY CD	Constant	112	7 8 2 D 0 _e 0 _e 0 _e 0 _e
LY CW	Constant	112	7 8 1 D 0 _e 0 _e 0 _e 0 _e
LY GB	Constant	108	7 8 0 E 0 _e 0 _e 0 _e 0 _e
LY GD	Constant	108	7 8 2 E 0 _e 0 _e 0 _e 0 _e
LY GW	Constant	108	7 8 1 E 0 _e 0 _e 0 _e 0 _e
MAB	---	104	6 8 2 9
MAS	---	104	6 8 1 9
MBA	---	104	6 8 8 9
MBR	Constant	104	7 8 0 _e 9 0 _e 0 _e 0 _e 0 _e
MBS	---	104	6 8 9 9
MSA	---	104	6 8 4 9
MSB	---	104	6 8 6 9
NOP 0	---	54	0 0 0 0
NOP 1	---	54	F F F F

Operation	Operand	Page	Machine Code
O	C	14	B 9 0 _d 0 _d
	D	12	7 8 3 F 1 0 _b 0 _g 0 _g
	F	12	8 8 _b 0 _a 0 _a
	I	12	C 8 _b 0 _a 0 _a
	Q	12	C 8 _b 8 _a 0 _a
	S	12	7 8 1 B 0 _b 0 _a 0 _a 0 _a
	T	14	F 9 0 _d 0 _d
	---	14	F B 0 0
O(---	16	B B 0 0
O =	Formal oper.	56	0 F 0 _c 0 _c
ON	C	14	B D 0 _d 0 _d
	D	14	7 8 3 F 3 0 _b 0 _g 0 _g
	F	14	A 8 _b 0 _a 0 _a
	I	14	E 8 _b 0 _a 0 _a
	Q	14	E 8 _b 8 _a 0 _a
	S	14	7 8 5 B 0 _b 0 _a 0 _a 0 _a
	T	14	F D 0 _d 0 _d
ON =	Formal oper.	56	2 F 0 _c 0 _c
OW	---	56	4 9 0 0
R	C	36	7 C 0 _d 0 _d
	D	18	7 8 3 F 5 0 _b 0 _g 0 _g
	F	18	B 0 _b 0 _a 0 _a
	I	16	F 0 _b 0 _a 0 _a
	Q	16	F 0 _b 8 _a 0 _a
	S	18	7 8 6 B 0 _b 0 _a 0 _a 0 _a
	T	34	3 C 0 _d 0 _d
RA	---	80	0 8 8 0

Operation	Operand	Page	Machine Code
RAE	---	80	7 8 1 0
RB =	Formal oper.	62	3 7 0 _c 0 _c
RD =	Formal oper.	62	3 E 0 _c 0 _c
RLD	Constant	76	6 4 0 _h 0 _h
RRD	Constant	76	7 4 0 _h 0 _h
RU	C	64	7 0 1 5 0 0 _b 0 _g 0 _g
	D	64	7 0 4 6 0 0 _b 0 _g 0 _g
	F	64	7 0 4 9 0 0 _b 0 _g 0 _g
	I	64	7 0 3 8 0 0 _b 0 _a 0 _a
	Q	64	7 0 3 8 0 0 _b 8 _a 0 _a
	RI	64	7 0 4 7 0 0 _b 0 _g 0 _g
	RJ	64	7 0 1 E 0 0 _b 0 _g 0 _g
	RS	100	7 0 5 7 0 0 _b 0 _g 0 _g
	RT	100	7 0 0 E 0 0 _b 0 _g 0 _g
T	64	7 0 2 5 0 0 _b 0 _g 0 _g	
S	C	36	5 C 0 _d 0 _d
	D	16	7 8 3 F 4 0 _b 0 _g 0 _g
	F	16	9 0 _b 0 _a 0 _a
	I	16	D 0 _b 0 _a 0 _a
	Q	16	D 0 _b 8 _a 0 _a
	S	16	7 8 2 B 0 _b 0 _a 0 _a 0 _a
S =	Formal oper.	62	1 7 0 _c 0 _c
SD	T	34	2 4 0 _d 0 _d
SD =	Formal oper.	66	2 6 0 _c 0 _c
SE	T	34	1 C 0 _d 0 _d
SEC =	Formal oper.	66	1 E 0 _c 0 _c
SED	Constant	82	7 8 0 6 0 0 0 _n 0 _n

Operation	Operand	Page	Machine Code
SEE	Constant	82	7 8 0 7 0 0 0 _n 0 _n
SF	T	34	1 4 0 _d 0 _d
SFD =	Formal oper.	66	1 6 0 _c 0 _c
SIM	---	98	7 0 0 D
SLD	Constant	76	2 9 0 _h 0 _h
SLW	Constant	76	6 1 0 0 _h
SP	T	34	3 4 0 _d 0 _d
SP =	Formal oper.	66	3 6 0 _c 0 _c
SRW	Constant	76	6 9 0 0 _h
SS	T	34	2 C 0 _d 0 _d
SSD	Constant	76	7 1 0 _h 0 _h
SSU =	Formal oper.	66	2 E 0 _c 0 _c
SSW	Constant	76	6 8 0 _h 1
STP	---	54	7 0 0 3
STS	---	98	7 0 0 0
STW	---	98	7 0 0 4
SU	C	62	7 0 1 5 4 0 _b 0 _g 0 _g
	D	64	7 0 4 6 4 0 _b 0 _g 0 _g
	F	62	7 0 4 9 4 0 _b 0 _g 0 _g
	I	62	7 0 3 8 4 0 _b 0 _a 0 _a
	Q	62	7 0 3 8 4 0 _b 8 _a 0 _a
	RI	64	7 0 4 7 4 0 _b 0 _g 0 _g
	RJ	64	7 0 1 E 4 0 _b 0 _g 0 _g
	RS	100	7 0 5 7 4 0 _b 0 _g 0 _g
	RT	100	7 0 0 E 4 0 _b 0 _g 0 _g
T	62	7 0 2 5 4 0 _b 0 _g 0 _g	

Operation	Operand	Page	Machine Code	
T	DD	30	3 B 0 _d 0 _d	
	DL	30	2 3 0 _d 0 _d	
	DR	30	2 B 0 _d 0 _d	
	DW	30	3 3 0 _d 0 _d	
	FD	28	1 B 0 _d 0 _d	
	FW	28	1 3 0 _d 0 _d	
	FY	28	0 B 0 _d 0 _d	
	IB	28	4 B 0 _d 0 _d	
	ID	28	5 B 0 _d 0 _d	
	IW	28	5 3 0 _d 0 _d	
	OW	32	7 7 0 _d 0 _d	
	OY	32	7 F 0 _d 0 _d	
	PW	32	7 B 0 _d 0 _d	
	PY	32	7 3 0 _d 0 _d	
	QB	28	4 B 8 _d 0 _d	
	QD	28	5 B 8 _d 0 _d	
	QW	28	5 3 8 _d 0 _d	
	RI	72	6 B 0 _d 0 _d	
	RJ	72	6 7 0 _d 0 _d	
	RS	72	6 3 0 _d 0 _d	
	RT	72	6 F 0 _d 0 _d	
	SD	30	7 8 F B 0 0 _d 0 _d 0 _d	
	SW	30	7 8 D B 0 0 _d 0 _d 0 _d	
	SY	30	7 8 B B 0 0 _d 0 _d 0 _d	
	T =	Formal oper.	70	6 6 0 _c 0 _c
	TB	C	58	7 0 1 5 C 0 _b 0 _g 0 _g
D		58	7 0 4 6 C 0 _b 0 _g 0 _g	

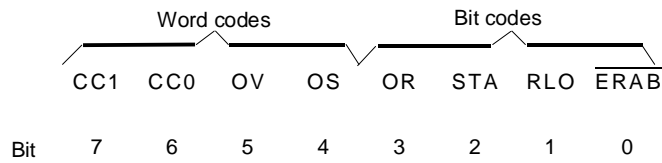
Operation	Operand	Page	Machine Code
TB	F	58	7 0 4 9 C 0b 0a 0a
	I	58	7 0 3 8 C 0b 0a 0a
	Q	58	7 0 3 8 C 0b 8a 0a
	RI	58	7 0 4 7 C 0b 0g 0g
	RJ	58	7 0 1 E C 0b 0g 0g
	RS	58	7 0 5 7 C 0b 0g 0g
	RT	58	7 0 0 E C 0b 0g 0g
	T	58	7 0 2 5 C 0b 0g 0g
TBN	C	60	7 0 1 5 8 0b 0g 0g
	D	60	7 0 4 6 8 0b 0g 0g
	F	60	7 0 4 9 8 0b 0a 0a
	I	60	7 0 3 8 8 0b 0a 0a
	Q	60	7 0 3 8 8 0b 8a 0a
	RI	60	7 0 4 7 8 0b 0g 0g
	RJ	60	7 0 1 E 8 0b 0g 0g
	RS	60	7 0 5 7 8 0b 0g 0g
	RT	60	7 0 0 E 8 0b 0g 0g
	T	60	7 0 2 5 8 0b 0g 0g
TAK	---	96	7 0 0 2
TDI	A1	88	6 8 0 F
	A2	88	6 8 2 F
	BA	88	6 8 9 F
	BR	88	6 8 A F
	SA	88	6 8 4 F
TIR	Register no.	86	4 8 0 0 _k
TNB	Constant	90	0 3 0 _l 0 _l
TNW	Constant	90	4 3 0 _o 0 _o

Operation	Operand	Page	Machine Code
TRD	Constant	106	6 8 0 5 0 _e 0 _e 0 _e 0 _e
TRW	Constant	106	6 8 0 3 0 _e 0 _e 0 _e 0 _e
TSC	Constant	110	7 8 C D 0 _e 0 _e 0 _e 0 _e
TSG	Constant	106	7 8 C E 0 _e 0 _e 0 _e 0 _e
TW CD	Constant	114	7 8 E D 0 _e 0 _e 0 _e 0 _e
TW CW	Constant	114	7 8 D D 0 _e 0 _e 0 _e 0 _e
TW GD	Constant	110	7 8 E E 0 _e 0 _e 0 _e 0 _e
TW GW	Constant	110	7 8 D E 0 _e 0 _e 0 _e 0 _e
TXB	---	92	7 0 1 F
TXW	---	92	7 0 0 F
TY CB	Constant	112	7 8 8 D 0 _e 0 _e 0 _e 0 _e
TY CD	Constant	112	7 8 A D 0 _e 0 _e 0 _e 0 _e
TY CW	Constant	112	7 8 9 D 0 _e 0 _e 0 _e 0 _e
TY GB	Constant	108	7 8 8 E 0 _e 0 _e 0 _e 0 _e
TY GD	Constant	108	7 8 A E 0 _e 0 _e 0 _e 0 _e
TY GW	Constant	108	7 8 9 E 0 _e 0 _e 0 _e 0 _e
XOW	---	56	5 1 0 0
)		16	B F 0 0
=	D	18	7 8 3 F 6 0 _b 0 _g 0 _g
	F	18	9 8 _b 0 _a 0 _a
	I	18	D 8 _b 0 _a 0 _a
	Q	18	D 8 _b 8 _a 0 _a
	S	18	7 8 3 B 0 _b 0 _a 0 _a 0 _a
==	Formal oper.	62	1 F 0 _c 0 _c
>D	---	46	3 9 2 0
<D	---	46	3 9 4 0
><D	---	46	3 9 6 0
!=D	---	46	3 9 8 0

Operation	Operand	Page	Machine Code
>=D	---	46	3 9 A 0
<=D	---	46	3 9 C 0
+D	---	94	6 0 0 D
-D	---	94	6 0 0 9
:F	---	38	6 0 0 0
xF	---	38	6 0 0 4
+F	---	38	7 9 0 0
-F	---	38	5 9 0 0
!=F	---	42	2 1 8 0
>F	---	42	2 1 2 0
<F	---	42	2 1 4 0
><F	---	42	2 1 6 0
>=F	---	42	2 1 A 0
<=F	---	42	2 1 C 0
>G	---	44	3 1 2 0
<G	---	44	3 1 4 0
><G	---	44	3 1 6 0
!=G	---	44	3 1 8 0
>=G	---	44	3 1 A 0
<=G	---	44	3 1 C 0
:G	---	40	6 0 0 3
xG	---	40	6 0 0 7
+G	---	40	6 0 0 F
-G	---	40	6 0 0 B

Explanatory Notes on the Condition Codes

Structure of the Condition Code Byte



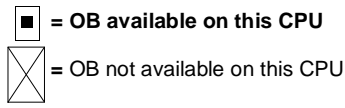
Abbreviations	Description
CC 0 / CC 1	Condition codes 0/1 (see Evaluation of CC 0 and CC 1)
OV	Overflow. This condition code is set if the maximum number range is exceeded during arithmetic operations.
OS	Stored overflow. The overflow bit is stored. This is an indication of whether and when an overflow error has occurred in the course of arithmetic operations.
OR	Internal condition code of the processor relating to AND and OR operations.
STA	STATUS; Signal status of the bit scanned.
RLO	Result of Logic Operation. Contains the result of individual bit operations and comparison operations.
ERAB	First bit scanned. $\overline{\text{ERAB}} = 0$ identifies the beginning and the end of a string of logic operations. The first operation of the string sets the $\overline{\text{ERAB}}$ bit to "1". Only at the end of the string is the ERAB bit reset (e.g. by a set/reset operation).

Evaluation of CC0 and CC1

C C 1	C C 0	Arith- metic Opera- tions	Digital Logic Opera- tions	Com- pari- son Opera- tions	Shift Opera- tions	For SED, SEE	Jump Opera- tions Exe- cuted
0	0	Result = 0	Result = 0	ACCU 2 = ACCU 1	shifted bit = 0	Sema- phore has been set	JZ
0	1	Result < 0	-	ACCU 2 < ACCU 1	-	-	JM JN
1	0	Result > 0	Result ≠ 0	ACCU 2 > ACCU 1	shifted bit = 1	Sema- phore is set now	JP JN
1	1	Divide by 0	-	-	-	-	JN ¹⁾

¹⁾ not executed with CPU 948

List of Organization Blocks

Organization Block					Function
	CPU 922	CPU 928	CPU 928B	CPU 948	
OBs for Program Processing					
OB 1	■ ¹⁾	■ ¹⁾	■ ¹⁾	■	OB for cyclic program processing
OB 2	■	■	■	■ ³⁾	Interrupt-driven program processing
OB 3 to OB 8	☒	☒	☒	■ ³⁾	Interrupt-driven program processing
OB 6	☒	☒	■	■ ³⁾	Delay interrupt
OB 9	☒	☒	■	■ ³⁾	Time-driven program processing
OB 10	☒	10 ms	10 ms	0.1 s ^{2) 3)}	Time interrupts with set time grid
OB 11	☒	20 ms	20 ms	0.2 s ^{2) 3)}	
OB 12	☒	50 ms	50 ms	0.5 s ^{2) 3)}	
OB 13	100 ms	100 ms	100 ms	1.0 s ^{2) 3)}	
OB 14	☒	200 ms	200 ms	2.0 s ^{2) 3)}	
OB 15	☒	500 ms	500 ms	5.0 s ^{2) 3)}	
OB 16	☒	1 s	1 s	10.0 s ^{2) 3)}	
OB 17	☒	2 s	2 s	20.0 s ^{2) 3)}	
OB 18	☒	5 s	5 s	50.0 s ^{2) 3)}	

¹⁾ alternative FB 0

²⁾ Default setting, can be changed via DX 0

³⁾ Details about the functions of these OBs of the CPU 948 can be found in the "CPU 948 Programming Guide".

List of Organization Blocks

Organization Block	<div style="display: flex; align-items: center; gap: 10px;"> ■ = OB available on this CPU ☒ = OB not available on this CPU </div>					Function
	CPU 922	CPU 928	CPU 928B		CPU 948	
OBs for Program Processing (continued)						
OB 31	☒	☒	☒		■ ¹⁾	Set cycle monitoring time
OB 39	☒	☒	☒		■	Organization of the cyclic program for communication in SMOOTH STOP
OBs for Start-up Procedures						
OB 20	■	■	■		■	Manual or automatic cold restart (can be set in DX 0)
OB 21	■	■	■		■	Manual warm restart
OB 22	■	■	■		■	Automatic warm restart after power failure
OB 38	☒	☒	☒		■	Organization of the restart behavior for communication in SMOOTH STOP

¹⁾ The setting of the cycle monitoring time via OB 31 has a higher priority than the setting via DX 0 (CPU 948).

List of Organization Blocks

Organization Block	<div style="display: flex; align-items: center; gap: 10px;"> ■ = OB not available on this CPU ☒ = OB not available on this CPU </div>			Cause of error	Reaction without OB
	CPU 922	CPU 928	CPU 928B		
OBs for Handling Controller Errors in the CPU 922/928/928B					
OB 19	■	■	■	Call of a block not programmed (LZF)	Stop
OB 23	■	■	■	Timeout in the case of direct access to the I/O module (QVZ)	none
OB 24	■	■	■	Timeout when updating the process image and transferring interprocessor communication flags	none
OB 25	■	■	■	Addressing error (ADF)	Stop
OB 26	■	■	■	Scan time exceeded (ZYK-FE)	Stop
OB 27	■	■	■	Substitution error (BCF)	Stop
OB 28	■	■	■	Stop by PG function/Stop switch/S5-BUS (ABBR)	Stop ¹⁾
OB 29	■	■	■	Operation error (BCF)	Stop
OB 30	■	■	■	Parameter assignment error (BCF)	Stop
OB 31	■	■	■	Other execution time errors (LZF)	Stop

¹⁾ Switchover to the STOP state always occurs independently of whether OB 28 is programmed and how it is programmed.

List of Organization Blocks

Organization Block	<div style="display: flex; align-items: center; gap: 10px;"> ■ = OB available on this CPU ☒ = OB not available on this CPU </div>			Cause of error	Reaction without OB
	CPU 922	CPU 928	CPU 928B		
OBs for Handling Controller Errors in the CPU 922/928/928B (continued)					
OB 32	■	■	■ ¹⁾	Transfer errors in the case of data blocks (LZF) ¹⁾	Stop
OB 33	■	■	■	Collision of two timed interrupts (WECK-FE)	Stop
OB 34	■	■	■	Error in PID controller processing	Stop
OB 35	☒	☒	■	Interface error	none

¹⁾ On CPU 928B also loading error

List of Organization Blocks

Organization Block	Cause of error	Reaction without OB
OBs for Handling Controller Errors in the CPU 948		
OB 19	Call a block that is not loaded (KB)	none
	Open a data block that is not loaded (KDB)	Stop
OB 23	Timeout during direct access (user program) to CP, IP, COR or I/O modules via the S5 bus (QVZ)	none
OB 24	Timeout while updating the process image or transferring the IPC flags	none
OB 25	Addressing error (ADF) ¹⁾	Stop
OB 26	Cycle time exceeded (ZYG)	Stop
OB 27	Substitution error (SUF)	Stop
OB 28	Timeout in input byte IB 0 (QVZ)	Stop

¹⁾ if not inhibited by IAE


Organization Block	Cause of error	Reaction without OB
OBs for Handling Controller Errors in the CPU 948 (continued)		
OB 29	Timeout for distributed peripherals for the address areas: - F 0000H to F EFFFH, F F200H to F FFFFH	none
OB 30	Parity error and QVZ in the user memory (PARE)	Stop
OB 32	Load/transfer error (TLAF)	Stop
OB 33	Collision of time interrupts: - Queue overflow (WEFES)	Stop
	- The time interrupt pulse has been masked for too long (WEFEH)	none
OB 34	Error while generating a data block with G DB or GX DX (FEDBX)	Stop

List of Organization Blocks



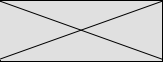
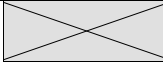
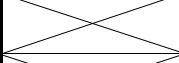
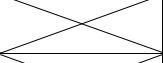
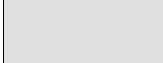
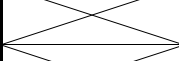
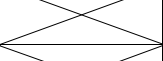

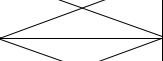
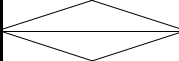
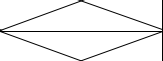
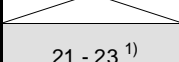
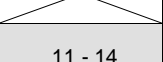

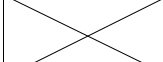

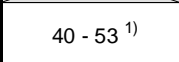
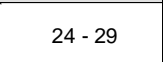
Organization Block	Execution times in μs				Function
	CPU 922	CPU 928	CPU 928B	CPU 948	
Special Function OBs					
OB 110	31 - 34 ¹⁾	18 - 20	1.7		Access to the condition-code byte
OB 111		12	1.2		Reset accumulators
OB 112		14	2.0		Roll up accumulator
OB 113		14	2.0		Roll down accumulator
OB 120	70 - 76 ¹⁾	36 - 128	68		Activate/deactivate "Disable all interrupts"
OB 121		36 - 136	68		Activate/deactivate "Disable cyclic time interrupts individually "
				58 - 78	Set/read system time (compatible to CPU 946/947)
OB 122	47 - 52 ¹⁾	37 - 39	73		Activate/deactivate "Delay all interrupts"
				26	Activate/deactivate "Disable all interrupts"
OB 123		39 - 54	73		Ativate/deactivate "Delay cyclic time interrupts individually "
OB 124				1327	Delete STEP 5 blocks
OB 125				1477	Generate STEP 5 blocks
OB 126				93	Define and transfer process images

¹⁾ OB only available from version 09

List of Organization Blocks

Organization Block	Execution times in μs				Function	
	CPU 922	CPU 928	CPU 928B	CPU 948		
 = OB not available on this CPU						
Special Functions OBs (continued)						
OB 129				15	Determine battery status	
OB 131				1.8	Delete Accu 1 to 4	
OB 132				2.2	Accu roll up	
OB 133				2.4	Accu roll down	
OB 141				47	Activate/deactivate "Disable cyclic time interrupts individually "	
OB 142				49	Activate/deactivate "Delay all interrupts"	
OB 143				47	Ativate/deactivate "Delay cyclic time interrupts individually "	
OB 150			265	266	Set system time	
			132	153	Read system time	
OB 151					Clock-controlled time interrupt	
			125	262	0	set
			max. 267	max. 284	1 to 7	
			123	166	0	read
			max. 152	202	1 to 7	

List of Organization Blocks

Organization Block	Execution times in μs				Function		
	CPU 922	CPU 928	CPU 928B	CPU 948			
 = OB not available on this CPU							
Special Function OBs (continued)							
OB 152			90		Cycle scan statistic		
OB 153			66		Delay interrupt		
					Function no.	Function	
					110	1	Define and start delay time
					60	2	Stop delay time
			70	80	3	Read current remaining time	
OB 160 - 163	21 - 23 ¹⁾	11 - 14	1.1		Repeat loops		
OB 170	$55 + n \cdot 27.5$ ¹⁾	$34 + n \cdot 15.8$	$60 + n \cdot 13.3$		Read block stack (BSTACK); n = number of BSTACK elements		
OB 180		16	1.0	76	Random data block access		
OB 181	40 - 53 ¹⁾	24 - 29	3.6	38	Test data blocks (DB/DX)		
OB 182			$171 + n \cdot 0.65$	$170+n \cdot 1$ ²⁾ $170+n \cdot 10.5$ ³⁾	Copy data area; ⁴⁾ n = number of data words		

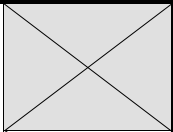
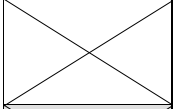
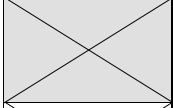
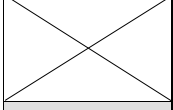
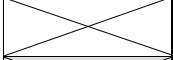
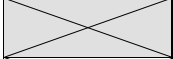

1) OB only available from version 09.

2) For copy direction decrementing

3) For copy direction incrementing

4) CPU 948: The copy direction "decrementing" is standard. The direction "incrementing" is only selected if the data areas overlap each other. This includes that the start address of the source area is smaller than the end address of the source area.

List of Organization Blocks

Organization Block	Execution times in μs				Function
	CPU 922	CPU 928	CPU 928B	CPU 948	
Special Function OBs (continued)					
OB 190	$^{1)} 47 + n \cdot 0.75$ $48.5 + n \cdot 0.5$ ₃₎	$25 + n \cdot 0.5$ ²⁾ $25 + n \cdot 0.3$ ³⁾	$54 + n \cdot 0.5$ ²⁾ $55 + n \cdot 0.3$ ³⁾		Transfer flag byte by byte into data block; n = number of flag bytes
OB 191	$^{1)} 47 + n \cdot 0.75$ $48 + n \cdot 0.5$ ³⁾	$25 + n \cdot 0.5$ ²⁾ $25 + n \cdot 0.3$ ³⁾	$54 + n \cdot 0.5$ ²⁾ $55 + n \cdot 0.3$ ³⁾		Transfer data field byte by byte into flag area; n = number of flag bytes
OB 192	$^{1)} 46 + n \cdot 2.8$ ²⁾ $46 + n \cdot 2.55$ ₃₎	$25 + n \cdot 1.8$ ²⁾ $40 + n \cdot 0.57$ ³⁾	$51 + n \cdot 1.8$ ²⁾ $53 + n \cdot 0.57$ ³⁾		Transfer flag word by word into a data block; n = number of flag bytes
OB 193	$^{1)} 46 + n \cdot 2.8$ ²⁾ $46 + n \cdot 2.55$ ₃₎	$25 + n \cdot 1.8$ ²⁾ $40 + n \cdot 0.57$ ³⁾	$51 + n \cdot 1.8$ ²⁾ $53 + n \cdot 0.57$ ³⁾		Transfer data field word by word into flag area; n = number of flag bytes
OB 200, 202 - 205	4)	4)	4)	4)	Interprocessor communication in multiprocessor mode
OB 216	31.8 - 42.5	28 - 35	58 - 65		Access to page frames
OB 217	32.6 - 43.1	30 - 35	60 - 66		Access to page frames
OB 218	26.3	21	54		Access to page frames

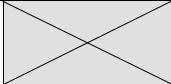
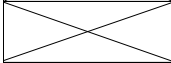
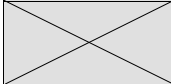
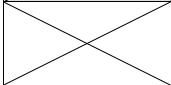
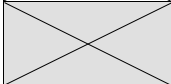
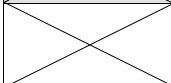
1) OB only available from version 09.

2) If number of first flag byte is uneven.

3) If number of first flag byte is even.

4) See manual of relevant CPU.


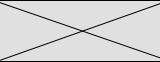
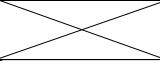
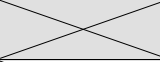
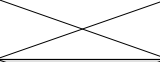
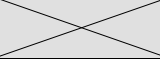
List of Organization Blocks

Organization Block	Execution times in μs					Function
	CPU 922	CPU 928	CPU 928B		CPU 948	
Special Function OBs (continued)						
OB 220	25	14	0.57			Convert the contents of ACCU 1 from a 16-bit fixed-point number to a 32-bit fixed-point number
OB 221	44.5	36	62			Set and trigger a new scan time monitor
OB 222	21.5	18	48		35	Retrigger the scan time monitor
OB 223	39	18	48		13	Change to stop status in case of non-uniform restart modes in multiprocessor mode
OB 224	23	11	41			Block transfer of the interprocessor communication flags in multiprocessor mode
OB 226	29	19	53			Read the contents of a system program memory location
OB 227	31	14	48			Read the check sum of the system program memory
OB 228	34.5 ¹⁾	21	56			Read status information of a program processing level
OB 230 - 237	2)	2)	2)		2)	Functions for handling blocks

¹⁾ OB only available from version 09

²⁾ See Manual "SIMATIC S5 - Standard Function Blocks Handling Blocks CPU 922, CPU 928, CPU 928B S5-135U, S5-155U Programmable Controllers"

List of Organization Blocks

Organization Block	Execution times in μs				Function
	CPU 922	CPU 928	CPU 928B	CPU 948	
 = OB not available on this CPU					
Special Function OBs (continued)					
OB 240	120 - 805	$73 + n * 12$	$105 + n * 12$		Initialize a shift register; n = number of pointers
OB 241	42 - 110	$28 + n * 12$	$60 + n * 9$		Call a shift register; n = number of pointers
OB 242	12	17	46		Delete a shift register
OB 250	158	114	144		Initialize a PID controller
OB 251	< 730	425	455		Call a PID controller
OB 254	$85 + n * 1$	$80 + n * 0.2$	$112 + n * 0.7$	1472 - 2869	Copy a DX data block (extension); n = number of data words to be transferred
OB 255	$85 + n * 1$	$80 + n * 0.2$	$112 + n * 0.7$	1472 - 2869	Copy a DB data block; n = number of data words to be transferred

Intentionally blank!

Siemens AG
AUT 1282

D-76181 Karlsruhe
Federal Republic of Germany

From:

Your Name: _____
Your Title: _____
Your Company: _____
Street: _____
City, Zip Code: _____
Country: _____
Phone: _____

Please check any industry that applies to you:

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



Remarks Form

Remarks Form

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

Title of Your Documentation: _____ _____
Order No. of Your Documentation: _____ _____

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

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

Additional comments:

