SINUMERIK 840/840C
SINUMERIK 880/880 GA2
PLC 135WB/WB2/WD

Quick Reference 03.96 Edition

Operation List

Contents

Explanatory notes on the operation list for the 135 WB Programmable Logic Controller (PLC)

WORD 0	B0	B1	Word length = 16 bits
WORD 1	B2	B3	
WORD 2	B4	B5	

PII		Process input image
PIQ		Process output image
RLO		Result of logic operation
RLO reloaded?	Y	If this operation is followed by a logic operation (conditional operation, e.g. A I X.X), the RLO is reloaded.
RLO-dependent?	J	The statement is executed only if RLO = "1" or the RLO is allocated
	N	The statement is always executed.
Condition code words		The inputs have the following meanings:
	X	The condition code is set/reset depending on the statement.
	1	Condition code is set.
	0	Condition code is reset.
	–	Condition code is not affected
CC 0, CC 1		See the table below for the meanings of these condition codes.
OV condition code		Overflow; this condition code is set if the number range is exceeded in arithmetic operations.
OS condition code		Stored overflow; this condition code is set if at least one arithmetic operation causes an overflow.
Execution Time		The right column applies only if the option ”ACOP” is used (**SINUMERIK 840** only)

Condition Code Table

Condition codes		Arithmetic, Fixed point	Digital logic operations	Comparison of contents of ACCUM 1 and	Shift operation Shifted
CC 1	CC 0	Result	Result	ACCUM 2	bit
0	0	Res. = 0	= 0	ACCUM 2 = ACCUM 1	0
0	1	Res. < 0	–	ACCUM 2 < ACCUM 1	–
1	0	Res. > 0	0	ACCUM 2 > ACCUM 1	1

Parameter type	I	Input parameter
	Q	Output parameter
	D	Data
	B	Block
	T	Timer
	C	Counter
Formal operand		Max. 4 characters, the first must be alphabetic

Basic operation set

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Operation	Parameters	Machine code (hexadecimal)			RLO-dependent?	RLO reloaded?	Condition codes affected				Execution time in µs		Function
		Word 0 B0 B1	Word 1 B2 B3	Word 2 B4 B5			CC1	CC0	OV	OS	COP		
Binary logic operations													**AND logic**
A I	0.0 to 127.7	C0 00			N	N	–	–	–	–	0.5	0.5	Scan input for "1"
A Q	0.0 to 127.7	C0 80			N	N	–	–	–	–	0.5	0.5	Scan output for "1"
A F	0.0 to 255.7	80 00			N	N	–	–	–	–	0.5	0.5	Scan flag for "1"
A D	0.0 to 255.15	78 3F	00 00		N	N	–	–	–	–	28.8	0.75	Scan data word for "1"
A N I	0.0 to 127.7	E0 00			N	N	–	–	–	–	0.5	0.5	Scan input for "0"
A N Q	0.0 to 127.7	E0 80			N	N	–	–	–	–	0.5	0.5	Scan output for "0"
A N F	0.0 to 255.7	A0 00			N	N	–	–	–	–	0.5	0.5	Scan flag for "0"
A N D	0.0 to 255.15	78 3F	20 00		N	N	–	–	–	–	28.8	0.75	Scan data word for "0"
													OR logic
O I	0.0 to 127.7	C8 00			N	N	–	–	–	–	0.5	0.5	Scan input for "1"
O Q	0.0 to 127.7	C8 80			N	N	–	–	–	–	0.5	0.5	Scan output for "1"
O F	0.0 to 255.7	88 00			N	N	–	–	–	–	0.5	0.5	Scan flag for "1"
O D	0.0 to 255.15	78 3F	10 00		N	N	–	–	–	–	28.8	0.75	Scan data word for "1"
O N I	0.0 to 127.7	E8 00			N	N	–	–	–	–	0.5	0.5	Scan input for "0"
O N Q	0.0 to 127.7	E8 80			N	N	–	–	–	–	0.5	0.5	Scan output for "0"
O N F	0.0 to 255.7	A8 00			N	N	–	–	–	–	0.5	0.5	Scan flag for "0"
O N D	0.0 to 255.15	78 3F	30 00		N	N	–	–	–	–	28.8	0.75	Scan data word for "0"
		+word address in double-word statements +bit address +byte address in single-word statements +bit address											**AND logic**
A T	0 to 127	F8 00			N	N	–	–	–	–	0.5	0.5	Timer for "1"
A N T	0 to 127	FC 00			N	N	–	–	–	–	0.5	0.5	Timer for "0". Results in RLO = "1" if time not yet started
A Z	0 to 127	B8 00			N	N	–	–	–	–	0.5	0.5	Counter for > "0"

Basic operation set

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs
for organization blocks (OBs)

Operation	Parameters	Machine code (hexadecimal)			RLO-de-pen-dent?	RLO reloa-ded?	Condition codes affected				Execution time in µs		Function
		Word 0 B0 B1	Word 1 B2 B3	Word 2 B4 B5			CC1	CC0	OV	OS	COP		
Logic operations													
A N C	0 to 127	BC 00			N	N	–	–	–	–	0.5	0.5	Counter for = "0" Results in RLO = "1" if counter not yet set
											0.5	0.5	**OR logic**
O T	0 to 127	F9 00			N	N	–	–	–	–	0.5	0.5	Scan timer for "1"
O N T	0 to 127	FD 00			N	N	–	–	–	–	0.5	0.5	Scan timer for "0". Results in RLO = "1" if timer not yet started.
O C	0 to 127	B9 00			N	N	–	–	–	–	0.5	0.5	Scan counter for > 0
O N C	0 to 127	BD 00			N	N	–	–	–	–	0.5	0.5	Scan counter for > 0. Results in RLO "1" if counter not yet started.
		+parameter address (hexadecimal)											
)		BF 00			N	N	–	–	–	–	0.25	0.25	Right parenthesis
A(		BA 00			N	Y	–	–	–	–	0.25	0.25	ANDing of bracketed expressions, eight nesting levels
O(		BB 00			N	Y	–	–	–	–	0.25	0.25	ORing of bracketed expressions, eight nesting levels
O		FB 00			N	Y	–	–	–	–	0.25	0.25	OR before AND
Setting/resetting operations													
													Set:
S I	0.0 to 127.7	D0 00			Y	Y	–	–	–	–	0.75	0.75	input in process image
S Q	0.0 to 127.7	D0 80			Y	Y	–	–	–	–	0.75	0.75	Output in process image
S F	0.0 to 255.7	90 00			Y	Y	–	–	–	–	0.75	0.75	Flag
S D	0.0 to 255.15	78 3F	40 00		Y	Y	–	–	–	–	33.4	1.0	Data
													Reset:
R I	0.0 to 127.7	F0 00			Y	Y	–	–	–	–	0.75	0.75	input in process image
R Q	0.0 to 127.7	F0 80			Y	Y	–	–	–	–	0.75	0.75	output in process image

Basic operation set

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Operation	Parameter	Maschine code (hexadecimal)			RLO-de-pen-dent?	RLO reloa-ded?	Condition codes affected				Execution time in µs		Function
		WordØ BØ B1	Word1 B2 B3	Word2 B4 B5			CC1	CCØ	OV	OS	Standard interpreter	ACOP	
Setting/resetting operations													
R F	0.0 to 255.7	BØ ØØ			Y	Y	–	–	–	–	0.75	0.75	flag
R D	0.0 to 255.15	78 3F	5Ø ØØ		Y	Y	–	–	–	–	34.0	1.0	data
													Allpcation of RLO to:
= I	0.0 to 127.7	D8 ØØ			N	Y	–	–	–	–	0.75	0.75	an input
= Q	0.0 to 127.7	D8 8Ø			N	Y	–	–	–	–	0.75	0.75	an output
= F	0.0 to 255.7	98 ØØ			N	Y	–	–	–	–	0.75	0.75	a flag
= D	0.0 to 255.15	78 3F	6Ø ØØ		N	Y	–	–	–	–	35.2	1.0	data

+wort address — in double-word statements
+bit address

+byte address — in single-word statements
+bit address

Operation	Parameter	WordØ BØ B1	Word1 B2 B3	Word2 B4 B5	RLO-de-pen-dent?	RLO reloa-ded?	CC1	CCØ	OV	OS	Standard interpreter	ACOP	Function
Timer and counter operations													
SI T	0 to 127	34 ØØ			Y	Y	–	–	–	–	16.4	16.4	Start time as pulse
SE T	0 to 127	1C ØØ			Y	Y	–	–	–	–	16.4	16.4	Start timer as extended pulse
SD T	0 to 127	24 ØØ			Y	Y	–	–	–	–	16.4	16.4	Start timer as ”on” delay
SS T	0 to 127	2C ØØ			Y	Y	–	–	–	–	16.4	16.4	Start timer as stored ”on” delay
SF T	0 to 127	14 ØØ			Y	Y	–	–	–	–	16.4	16.4	Start timer as ”off” delay
R T	0 to 127	3C ØØ			Y	Y	–	–	–	–	7.0	7.0	Reset timer
S Z	0 to 127	5C ØØ			Y	Y	–	–	–	–	16.4	16.4	Set counter
R C	0 to 127	7C ØØ			Y	Y	–	–	–	–	7.0	7.0	Reset counter
CU C	0 to 127	6C ØØ			Y	Y	–	–	–	–	11.2	11.2	Increment counter (count up)
CD C	0 to 127	54 ØØ			Y	Y	–	–	–	–	11.8	11.8	Decrement counter (count down)

+parameter address (hexadecimal)

Basic operation set

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Operation	Parameters	Machine code (hexadecimal)			RLO-dependent?	RLO reloaded?	Condition codes affected				Execution time in µs		Function
		Word 0 B0 B1	Word 1 B2 B3	Word 2 B4 B5			CC1	CC0	OV	OS	Standard interpreter	ACOP	
Load and transfer operations													**Load (into ACCUM)**
L IB	0 to 127	4A 00			N	N	–	–	–	–	0.5	0.5	an input byte (from PII)
L IW	0 to 126	52 00			N	N	–	–	–	–	0.75	0.75	an input word (from PII) byte n ACCUM 1 HIGH (bits 8 - 15) byte n + 1 ACCUM 1 LOW (bits 0 - 7)
L ID	0 to 124	5A 00			N	N	–	–	–	–	1.0	1.0	an input double-word (from PII)
L QB	0 to 127	4A 80			N	N	–	–	–	–	0.5	0.5	an output byte (from PIQ)
L QW	0 to 126	52 80			N	N	–	–	–	–	0.75	0.75	an output word (from PIQ) [as for LIW]
L QD	0 to 124	5A 80			N	N	–	–	–	–	1.0	1.0	an output double-word (from PIQ)
L FY	0 to 255	0A 00			N	N	–	–	–	–	0.5	0.5	a flag byte
L FW	0 to 254	12 00			N	N	–	–	–	–	0.75	0.75	a flag word [as for LIW]
L FD	0 to 252	1A 00			N	N	–	–	–	–	1.0	1.0	a flag double-word
L DR	0 to 255	2A 00			N	N	–	–	–	–	0.5	0.5	data (right-hand byte)
L DL	0 to 255	22 00			N	N	–	–	–	–	0.5	0.5	data (left-hand byte)
L DW	0 to 255	32 00			N	N	–	–	–	–	0.5	0.5	data (word)
L DD	0 to 254	3A 00			N	N	–	–	–	–	0.75	0.75	data (double-word)
L T	0 to 127	02 00			N	N	–	–	–	–	0.5	0.5	a time
L C	0 to 127	42 00			N	N	–	–	–	–	0.5	0.5	a count
L PY	0 to 127	72 00			N	N	–	–	–	–	10.8	1.0	a peripheral byte of the digital inputs (bypassing the PII)
L PW	0 to 126	7A 00			N	N	–	–	–	–	11.8	1.75	a peripheral word of the digital inputs (bypassing the PII)
LD T	0 to 127	0C 00			N	N	–	–	–	–	17.4-48.0	17.4-48.0	times (BCD)
LD C	0 to 127	4C 00			N	N	–	–	–	–	16.2-47.2	16.2-47.2	counts (BCD)
		+address (hexadecimal)											
L KB	0 to 255	28 00			N	N	–	–	–	–	8.6	0.25	a constant , 1 byte
		+constant (hexadecimal)											

Basic operation set

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Operation	Parameters	Machine code (hexadecimal)			RLO-dependent?	RLO reloaded?	Condition codes affected				Execution time in µs		Function
		Word 0 B0 B1	Word 1 B2 B3	Word 2 B4 B5			CC 1	CC 0	OV	OS	Standard interpreter	ACOP	
Load and transfer operations													
L KS	any 2 alphanumeric characters	30 10	00 00		N	N	–	–	–	–	14.4	0.5	a constant, two ASCII symbols
L KT	0.0 to 999.3	30 02	00 00		N	N	–	–	–	–	14.4	0.5	a time (constant)
L KC	0 to 999	30 01	00 00		N	N	–	–	–	–	14.4	0.5	a count (constant)
L KM	any bit pattern (16 bits)	30 80	00 00		N	N	–	–	–	–	14.4	0.5	a constant as bit pattern
L KH	0 to FFFF	30 40	00 00		N	N	–	–	–	–	14.4	0.5	a constant in hexadecimal code
L KF	– 32768 to +32767	30 04	00 00		N	N	–	–	–	–	14.4	0.5	a constant as fixed-point number
L KG	±0.1701412 $\cdot 10^{39}$ to ±0.1469368 $\cdot 10^{-38}$	38 00	00 00	00 00	N	N	–	–	–	–	15.8	0.75	a constant as floating-point number
L KY	0 to 255 for each byte	30 20	00 00		N	N	–	–	–	–	14.4	0.5	a constant, 2 bytes
		+constant (hexadecimal)											**Transfer (the contents of <ACCUM 1>) to**
T IB	0 to 127	4B 00			N	N	–	–	–	0	0.5	0.5	an input byte (into the PII)
T IW	0 to 126	53 00			N	N	–	–	–	0	0.75	0.75	an input word (PII) ACCUM 1 HIGH (bits 8 - 15) n ACCUM 1 LOW (bits 0 - 7) n + 1
T ID	0 to 124	5B 00			N	N	–	–	–	0	1.0	1.0	an input double-word (into the PII)
T QB	0 to 127	4B 80			N	N	–	–	–	0	0.5	0.5	an output byte (into the PIQ)
T QW	0 to 126	53 80			N	N	–	–	–	0	0.75	0.75	an output word (into the PIQ) [as for LIW]
T QD	0 to 124	5B 80			N	N	–	–	–	0	1.0	1.0	an output double-word (into the PIQ)
T FY	0 to 255	0B 00			N	N	–	–	–	0	0.5	0.5	a flag byte
T FW	0 to 254	13 00			N	N	–	–	–	0	0.75	0.75	a flag word [as for LIW]
T FD	0 to 252	1B 00			N	N	–	–	–	0	1.0	1.0	a flag double-word
T DR	0 to 255	2B 00			N	N	–	–	–	0	0.5	0.5	data (right-hand byte)
T DL	0 to 255	23 00			N	N	–	–	–	0	0.5	0.5	data (left-hand byte)
T DW	0 to 255	33 00			N	N	–	–	–	0	0.5	0.5	data (word)
T DD	0 to 254	3B 00			N	N	–	–	–	0	0.75	0.75	data (double-word)
T PY	0 to 127	73 00			N	N	–	–	–	0	14.6	1.25	a peripheral byte of the digital outputs (bypassing the PIQ)
T PW	0 to 126	7B 00			N	N	–	–	–	0	18.4	2.25	a peripheral word of the digital outputs (bypassing the PIQ)
		+address (hexadecimal)											

Basic operation set

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Operation	Parameters	Machine code (hexadecimal)			RLO-dependent?	RLO reloaded?	Condition codes affected				Execution time in µs		Function
		Word 0 B0 B1	Word 1 B2 B3	Word 2 B4 B5			CC1	CC0	OV	OS	Standard interpreter	ACOP	
Comparison operations													
													Compare fixed-point numbers in ACCUM 1 and ACCUM 2 (the first operand specified is compared with the operand following, according to the comparison function):
!=F		21 80			N	N	X	X	–	–	11.0-11.6	0.25	for equal to: <ACCUM 2> = <ACCUM 1>
><F		21 60			N	N	X	X	–	–	10.8-11.2	0.25	for not equal to: <ACCUM 2> >< <ACCUM 1>
>F		21 20			N	N	X	X	–	–	10.8-11.2	0.25	for greater than: <ACCUM 2> > <ACCUM 1>
>=F		21 A0			N	N	X	X	–	–	11.0-11.6	0.25	for greater than or equal to: >= <ACCUM 2> <ACCUM 1>
<F		21 40			N	N	X	X	–	–	10.8-11.2	0.25	for less than: <ACCUM 2> < <ACCUM 1>
<=F		21 C0			N	N	X	X	–	–	11.0-11.2	0.25	for less than or equal to: <ACCUM 2> <= <ACCUM 1>
Arithmetic operations													
+ F		79 00			N	N	X	X	X	X	12.0-12.8	0.25	Addition of fixed-point numbers: <ACCUM 2> + <ACCUM 1>
– F		59 00			N	N	X	X	X	X	12.0-12.8	0.25	Subtraction of fixed-point numbers: <ACCUM 2> - <ACCUM 1>
: F		60 00			N	N	X	X	X	X	11.0-23.2	11.0-23.2	Division of fixed-point numbers: <ACCUM 2> : <ACCUM 1> ACCUM 1 LOW: Result ACCUM 1 HIGH: Remainder
x F		60 04			N	N	X	X	X	X	14.2-16.8	14.2-16.8	Multiplication of fixed-point numbers: <ACCUM 2> x <ACCUM 1>

Basic operation set

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Operation	Parameters	Machine code (hexadecimal)			RLO-de-pen-dent?	RLO reloa-ded?	Condition codes affected				Execution time in µs		Function
		Word 0 B0 B1	Word 1 B2 B3	Word 2 B4 B5			CC1	CC0	OV	OS	Standard interpreter	ACOP	
Comparison operations													**Compare floating-point numbers**
!=G		31 80			N	N	X	X	–	0	15.8	15.8	for equal to: <ACCUM 2> = <ACCUM 1>
><G		31 60			N	N	X	X	–	0	15.6	15.6	for not equal to: <ACCUM 2> >< <ACCUM 1>
>G		31 20			N	N	X	X	–	0	15.4	15.4	for greater than: <ACCUM 2> > <ACCUM 1>
>=G		31 A0			N	N	X	X	–	0	15.4	15.4	for greater than or equal to: >= <ACCUM 2> <ACCUM 1>
<G		31 40			N	N	X	X	–	0	15.8	15.8	for less than: <ACCUM 2> < <ACCUM 1>
<=G		31 C0			N	N	X	X	–	0	15.6	15.6	for less than or equal to: <ACCUM 2> <= <ACCUM 1>
Arithmetic operations													
+ G		60 0F			N	N	X	X	X	X	17.8-168.2	17.8-168.2	Addition of floating-point numbers: <ACCUM 2> + <ACCUM 1>
– G		60 0B			N	N	X	X	X	X	16.2-169.0	16.2-169.0	Subtraction of floating-point numbers: <ACCUM 2> - <ACCUM 1>
: G		60 03			N	N	X	X	X	X	8.8-29.4	8.8-29.4	Division of floating-point numbers: <ACCUM 2> : <ACCUM 1>
x G		60 07			N	N	X	X	X	X	15.4-25.9	15.4-25.9	Multiplication of floating-point numbers: <ACCUM 2> x <ACCUM 1>

Basic operation set

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Operation	Parameters	Machine code (hexadecimal)			RLO-de-pen-dent?	RLO reloa-ded?	Condition codes affected				Execution time in µs		Function
		Word 0 B0 B1	Word 1 B2 B3	Word 2 B4 B5			CC1	CC0	OV	OS	Stan-dard inter preter	ACOP	
Comparison operations													**Comparison of fixed-point numbers (double-length)**
													in ACCUM 1 and ACCUM 2 (The operand first specified is compared with the subsequent operand in accordance with the comparison function):
! = D		39 80			N	N	X	X	–	0	12.8	0.25	for equal to: <ACCUM 2> = <ACCUM 1>
><D		39 60			N	N	X	X	–	0	13.0	0.25	for not equal to: <ACCUM 2> >< <ACCUM 1>
> D		39 20			N	N	X	X	–	0	12.0	0.25	for greater than: <ACCUM 2> > <ACCUM 1>
>=D		39 A0			N	N	X	X	–	0	12.2	0.25	for greater than or equal to: <ACCUM 2> >= <ACCUM 1>
< D		39 40			N	N	X	X	–	0	12.4	0.25	for less than: <ACCUM 2> < <ACCUM 1>
<=D		39 C0			N	N	X	X	–	0	12.4	0.25	for less than or equal to: <ACCUM 2> <= <ACCUM 1>
Block calls													**Unconditional jump**
JU PB	0 to 255	75 00			N	Y	–	–	–	0	31.0	2.0	to a program block
JU FB	0 to 255	3D 00			N	Y	–	–	–	0	42.6	2.0- 42.6	to a function block 26.6 (ASM - FB)
JU SB	0 to 255	7D 00			N	Y	–	–	–	0	31.0	2.0	to a sequence block
DO FX	0 to 255	78 01	00 00		N	Y	–	–	–	0	53.2	2.25- 43.6	to a function block (type FX) 32.0 (ASM - FX)

Basic operation set

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Execution time in µs

Operation	Parameters	Machine code (hexadecimal)			RLO-de-pen-dent?	RLO reloa-ded?	Condition codes affected				Execution time in µs		Function
		Word 0 B0 B1	Word 1 B2 B3	Word 2 B4 B5			CC1	CC0	OV	OS	Standard interpreter	ACOP	
Block calls													**Conditional jump** (if RLO = "1")
JC PB	0 to 255	55 00			Y	Y	–	–	–	0	33.0	2.0	to a program block
JC FB	0 to 255	1D 00			Y	Y	–	–	–	0	45.0	2.0- 42.6	to a function block 26.6 (ASM - FB)
JC SB	0 to 255	5D 00			Y	Y	–	–	–	0	33.0	2.0	to a sequence block
DOC FX	0 to 255	78 02	00 00		Y	Y	–	–	–	0	55.0	2.25- 43.6	to a function block (type FX) 32.0 (ASM - FX)
Time for RLO = 0 i.e jump is not executed											8.0	0.25	
Block calls													
C DB	0 to 255	20 00			N	N	–	–	–	–	14.8	1.0	Calling a data block (type DB)
		+block number				(hexadecimal)							
CX DX	0 bis 255	78 Ø3	ØØ ØØ		N	N					25.4	1.25	Calling a data block (type DX)
BE		65 ØØ			N	Y	–	–	–	Ø	2.25	2.25	Block end
BEC		Ø5 ØØ			Y	Y	–	–	–	Ø	2.25	2.25	Block end, conditional (if RLO=„1”)
BEU		65 Ø1			N	Y	–	–	–	Ø	2.25	2.25	Block end, unconditional
Other operations													
NOP 0		00 00			N	N	–	–	–	–	0.25	0.25	No operation (all bits reset)
NOP 1		FF FF			N	N	–	–	–	–	0.25	0.25	No operation (all bits set)
STP		70 03			N	N	–	–	–	–	9.2	9.2	Stop
BLD	0 to 255	10 00			N	N	–	–	–	–	0.25	0.25	Display construction statement
BLD 130		10 82			N	N	–	–	–	–	0.25	0.25	Display construct. statement for the generat. of a blank line by carriage return
BLD 131		10 83			N	N	–	–	–	–	0.25	0.25	Displ. construct. statement for changing to statement lists. Equivalent to: STL
BLD 255		10 FF			N	N	–	–	–	–	0.25	0.25	Displ. constr. statement for end of seg.

Supplementary operations

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Operation	Parameters	Machine code (hexadecimal)			RLO-dependent?	RLO reloaded?	Condition codes affected				Execution time in µs		Function
		Word 0 B0 B1	Word 1 B2 B3	Word 2 B4 B5			CC1	CC0	OV	OS	Standard interpreter	ACOP	
Binary operations													
A =	Formal operand	07 00			)[1]	)[1]	–	–	–	–	0.31)[2]	0.31)[2]	AND operation; scanning a formal operand for "1"
AN =	Formal operand	27 00			)[1]	)[1]	–	–	–	–	0.31)[2]	0.31)[2]	AND operation; scanning a formal operand for "0"
O =	Formal operand	0F 00			)[1]	)[1]	–	–	–	–	0.31)[2]	0.31)[2]	OR operation; scanning a formal operand for "1"
ON =	Formal operand	2F 00			)[1]	)[1]	–	–	–	–	0.31)[2]	0.31)[2]	OR operation; scanning a formal operand for "0"
		+parameter address			(hexadecimal)								
Digital operations													
AW		41 00			N	N	X	X	–	–	8.4	0.25	Digital ANDing of ACCUM 1 and ACCUM 2
OW		49 00			N	N	X	X	–	–	8.4	0.25	Digital ORing of ACCUM 1 and ACCUM 2
XOW		51 00			N	N	X	X	–	–	8.4	0.25	Digital EXORing of ACCUM 1 and ACCUM 2

)[1] See relevant operation description
)[2] Plus time for substituted operation

Supplementary operations

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Operation	Parameters	Machine code (hexadecimal)			RLO-de-pen-dent?	RLO reloa-ded?	Condition codes affected				Execution time in µs		Function
		Word 0 B0 B1	Word 1 B2 B3	Word 2 B4 B5			CC1	CC0	OV	OS	Standard interpreter	ACOP	
Bit test operations													**Test bit for "1":**
TB I	0.0 to 127.7	70 38	C0 00		N	N	–	–	–	–	22.8	0.75	in an input
TB Q	0.0 to 127.7	70 38	C0 80		N	N	–	–	–	–	22.8	0.75	in an output
TB F	0.0 to 255.7	70 49	C0 00		N	N	–	–	–	–	22.8	0.75	in a flag
TB C	0.0 to 255.15	70 15	C0 00		N	N	–	–	–	–	24.8	0.75	in a counter word
TB T	0.0 to 255.15	70 25	C0 00		N	N	–	–	–	–	24.8	0.75	in a timer word
TB D	0.0 to 255.15	70 46	C0 00		N	N	–	–	–	–	27.6	0.75	in a data word
													Test bit for "0":
TBN I	0.0 to 127.7	70 38	80 00		N	N	–	–	–	–	22.8	0.75	in an input
TBN Q	0.0 to 127.7	70 38	80 80		N	N	–	–	–	–	22.8	0.75	in an output
TBN F	0.0 to 255.7	70 49	80 00		N	N	–	–	–	–	22.8	0.75	in a flag
TBN C	0.0 to 255.15	70 15	80 00		N	N	–	–	–	–	24.8	0.75	in a counter word
TBN T	0.0 to 255.15	70 25	80 00		N	N	–	–	–	–	24.8	0.75	in a timer word
TBN D	0.0 to 255.15	70 46	80 00		N	N	–	–	–	–	27.6	0.75	in a data word

+word address
+bit address

Supplementary operations

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Operation	Parameters	Machine code (hexadecimal)			RLO-dependent?	RLO reloaded?	Condition codes affected				Execution time in µs		Function
		Word 0 B0 B1	Word 1 B2 B3	Word 2 B4 B5			CC1	CC0	OV	OS	Standard interpreter	ACOP	
Setting operations													
S =	Formal operand	17 00			)[1]	)[1]	–	–	–	–	0.31)[2]	0.31)[2]	Binary setting of a formal operand
RB =	Formal operand	37 00			)[1]	)[1]	–	–	–	–	0.31)[2]	0.31)[2]	Binary resetting of a formal operand
RD =	Formal operand	3E 00			)[1]	)[1]	–	–	–	–	7.31)[2]	7.31)[2]	Digital resetting of a formal operand for timers and counters
= =	Formal operand	1F 00			)[1]	)[1]	–	–	–	–	0.31)[2]	0.31)[2]	Assignment of the RLO to a formal operand
			+parameter address (hexadecimal)										
													Set bit unconditionally
SU I	0.0 to 127.7	70 38	40 00		N	Y	–	–	–	–	22.8	1.0	in an input
SU Q	0.0 to 127.7	70 38	40 80		N	Y	–	–	–	–	22.8	1.0	in an output
SU F	0.0 to 255.7	70 49	40 00		N	Y	–	–	–	–	22.8	1.0	in a flag
SU C	0.0 to 255.15	70 15	40 00		N	Y	–	–	–	–	23.6	1.0	in a counter word
SU T	0.0 to 255.15	70 25	40 00		N	Y	–	–	–	–	23.6	1.0	in a timer word
SU D	0.0 to 2042.15	70 46	40 00		N	Y	–	–	–	–	29.8	1.0	in a data word
													Reset bit unconditionally
RU I	0.0 to 127.7	70 38	00 00		N	Y	–	–	–	–	23.0	1.0	in an input
RU Q	0.0 to 127.7	70 38	00 80		N	Y	–	–	–	–	23.0	1.0	in an output
RU F	0.0 to 255.7	70 49	00 00		N	Y	–	–	–	–	23.0	1.0	in a flag
RU C	0.0 to 255.15	70 15	00 00		N	Y	–	–	–	–	24.0	1.0	in a counter word
RU T	0.0 to 255.15	70 25	00 00		N	Y	–	–	–	–	24.0	1.0	in a timer word
RU D	0.0 to 2042.15	70 46	00 00		N	Y	–	–	–	–	30.0	1.0	in a data word
			+word address										
		+bit address											

)[1] See relevant operation description

)[2] Plus time for substituted operation

Supplementary operations

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Operation	Parameters	Machine code (hexadecimal)			RLO-dependent?	RLO reloaded?	Condition codes affected				Execution time in µs		Function
		Word 0 B0 B1	Word 1 B2 B3	Word 2 B4 B5			CC1	CC0	OV	OS	Standard interpreter	ACOP	
Timer and counter operations													
FR =	Formal operand	06 00			)[1]	)[1]	–	–	–	–	0.31)[2]	0.31)[2]	Enable formal operand for cold restart (for description, see FT or FC; according to formal opearand; parameter type: T, C).
SP =	Formal operand	36 00			)[1]	)[1]	–	–	–	–	0.31)[2]	0.31)[2]	Start timer specified as formal operand with the values stored in the ACCUM as pulse (parameter type: T).
SR =	Formal operand	26 00			)[1]	)[1]	–	–	–	–	0.31)[2]	0.31)[2]	Start timer specified as formal operand with the value stored in the ACCUM as "on" delay (parameter type: T)
SEC =	Formal operand	1E 00			)[1]	)[1]	–	–	–	–	0.31)[2]	0.31)[2]	Start timer specified as formal operand with the value stored in the ACCUM as extended pulse or set counter specified as formal operand with the count specified (parameter type: T, C)
SSU =	Formal operand	2E 00			)[1]	)[1]	–	–	–	–	0.31)[2]	0.31)[2]	Start timer specified as formal operand with the value stored in the ACCUM as stored "on" delay or increment a counter specfied as formal operand (parameter type: T, C)
SFD =	Formal operand	16 00			)[1]	)[1]	–	–	–	–	0.31)[2]	0.31)[2]	Start timer specified as formal operand with the value stored in the ACCUM as stored "off" delay or decrement a counter specified as formal operand (parameter type: : T, C)
		+parameter address		(hexadecimal)									
FR T	0 to 255	04 00			Y	Y	–	–	–	–	0.75	0.75	Enable timer for cold restart. The operation is executed only on the leading edge of the RLO. The cold restart of the timer results if the RLO is "1" at the time of the start operation.
FR C	0 to 255	44 00			Y	Y	–	–	–	–	0.75	0.75	Enable counter for cold restart. The operation is executed only on the leading edge of the RLO. The setting, up- or down-counting of the counter results only if the RLO is "1" for the relevant operation.
		+word address											

[1] See relevant operation description

[2] Plus time for substituted operation

Supplementary operations

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Operation	Parameters	Machine code (hexadecimal)			RLO-dependent?	RLO reloaded?	Condition codes affected				Execution time in µs		Function
		Word 0 B0 B1	Word 1 B2 B3	Word 2 B4 B5			CC1	CC0	OV	OS	Standard interpreter	ACOP	
Load and transfer operations													
L =	Formal operand	46 00			)[1]	)[1]	–	–	–	–	0.31)[2]	0.31)[2]	Load formal operand. The value of the operand specified as formal operand is loaded into ACCUM 1 (parameter type: I, Q, T, Z; data type: BY, W).
LD =	Formal operand	0E 00			)[1]	)[1]	–	–	–	–	12.4)[2]	12.4)[2]	Load formal operand in BCD. The value of the timer or counter location specified as formal operand is loaded in BCD into ACCUM 1 (paramaeter type: T, C).
LW =	Formal operand	3F 00			)[1]	)[1]	–	–	–	–	21.2)[2]	0.31)[2]	Load the bit pattern of a formal operand into ACCUM 1 (parameter type: KF, KH, KM, KY, KC, KT, KZ).
LDW =	Formal operand	56 00			)[1]	)[1]	–	–	–	–	16.2)[2]	0.31)[2]	Load the bit pattern of a formal operand. (Parameter type: D; see LW = parameter type: KG).
T =	Formal operand	66 00			)[1]	)[1]	–	–	–	–	0.31)[2]	0.31)[2]	Transfer to a formal operand. The contents of ACCUM 1 are transferred to the operand specified as formal operand (parameter type: I, Q; data type: BY, W).
		+parameter address			(hexadecimal)								

)[1] See relevant operation description
)[2] Plus time for substituted operation

Supplementary operations

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Operation	Parameters	Machine code (hexadecimal)			RLO-dependent?	RLO reloaded?	Condition codes affected				Execution time in µs		Function
		Word 0 B0 B1	Word 1 B2 B3	Word 2 B4 B5			CC1	CC0	OV	OS	Standard interpreter	ACOP	
Conversion operations													
CFW		01 00			N	N	–	–	–	–	4.1	0.25	Formation of one's complement of ACCUM 1 (16 bits)
CSW		09 00			N	N	X	X	X	–	8.4-9.4	0.25	Formation of two's complement of ACCUM 1 (16 bits)
CSD		68 07			N	N	X	X	X	–	12.4-12.8	12.4-12.8	Formation of two's complement of ACCUM 1 (32 bits)
DEF		68 0C			N	N	–	–	–	X	13.0-20.2	13.0-20.2	16-bit fixed-point conversion from BCD to binary
DUF		68 08			N	N	–	–	–	X	12.6-49.6	12.6-49.6	16-bit fixed-point conversion from binary to BCD
DED		68 0E			N	N	–	–	–	X	12.8-40.8	12.8-40.8	32-bit fixed-point conversion from BCD to binary
DUD		68 0A			N	N	–	–	–	X	15.6-142.2	15.6-142.2	32-bit fixed-point conversion from binary to BCD
FDG		68 06			N	N	–	–	0	–	131.2-150.6	131.2-150.6	Conversion of a fixed-point number to a floating-point number (32 bits)
GFD		68 02			N	N	–	–	–	X	11.6-105.8	11.6-105.8	Conversion of a floating-point number to a fixed-point number

Supplementary operations

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Operation	Parameters	Machine code (hexadecimal)			RLO-de-pen-dent?	RLO reloa-ded?	Condition codes affected				Execution time in µs		Function
		Word 0 B0 B1	Word 1 B2 B3	Word 2 B4 B5			CC1	CC0	OV	OS	Stan-dard inter preter	ACOP	
Shift functions													
													Shift the contents of ACCUM 1 by the value specified in the parameter (positions becoming vacant are padded with zeros):
SLW	0 to 15	61 00			N	N	X	–	–	0	8.4-9.2	1.5	to the left
SRW	0 to 15	69 00			N	N	X	–	–	0	8.6-9.4	1.5	to the right
SLD	0 to 32	29 00			N	N	X	–	–	0	12.4-100.6	2.5	to the left (double word)
SSW	0 to 15	68 01			N	N	X	–	–	0	11.2-12.0	1.5	to the right with sign (word)
SSD	0 to 32	71 00			N	N	X	–	–	0	13.6-101.0	2.5	to the right with sign (double-word)
		+number of shifts (hexadecimal)											
Jump operations													
JU =	Symbolic address max. 4 characters	2D 00			N	N	–	–	–	–	7.8	0.31	Unconditional jump: The unconditional jump is made without regard to conditions
JC =	Symbolic address max. 4 characters	FA 00			Y	Y	–	–	–	–	7.6/ 11.6)*	0.31	Conditional jump: The conditional jump is made if the RLO = "1". If the RLO = "0", no jump is made and the RLO is set to "1".
JZ =	Symbolic address max. 4 characters	45 00			N	N	–	–	–	–	5.0/ 10.2)*	0.31	Jump if the result is 0: The jump is only made if CC1 = 0 and CC0 = 0. The RLO is not changed.
JN =	Symbolic address max. 4 characters	35 00			N	N	–	–	–	–	5.0/ 10.2)*	0.31	Jump if the result is not 0: The jump is only made if CC1 CC0. The RLO is not changed.
JP =	Symbolic address max. 4 characters	15 00			N	N	–	–	–	–	5.0/ 10.2)*	0.31	Jump if the result is greater than 0: The jump is only executed if CC1 = 1 and CC0 = 0. The RLO is not changed.
JM =	Symbolic address max. 4 characters	25 00			N	N	–	–	–	–	5.0 / 10.2)*	0.31	Jump if the result is less than 0: The jump is only made if CC1 = 0 and CC0 = 1. The RLO is not changed.
		+relative jump address max. ± 127											

)1 Jump not executed/jump executed

Supplementary operations

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Operation	Parameters	Machine code (hexadecimal)			RLO-de-pen-dent?	RLO reloa-ded?	Condition codes affected				Execution time in µs		Function
		Word 0 B0 B1	Word 1 B2 B3	Word 2 B4 B5			CC1	CC0	OV	OS	Stan-dard inter preter	ACOP	
Jump operations													
JO =	Symbolic address max. 4 characters	0D 00			N	N	–	–	–	–	5.0 / 10.2)[3]	0.31	Jump on overflow: The jump is only executed if the OVERFLOW bit is set. The RLO is not changed.
		+relative jump address max. ± 127											
JOS =	Symbolic address max. 4 characters	60 0C	00 00		N	N	–	–	–	0	14.6 / 14.8)[3]	0.56	Jump if the OVERFLOW bit is set for "stored" (OS).
Other operations													
DO =	Formal operand	76 00			)[1]	)[1]	–	–	–	–	0.81	0.81	Process formal operand (parameter type: B). Only CDB, JU PB, JU FB, JU SB can be substituted.
DO DW	0 to 255	6E 00			)[1]	)[1]	–	–	–	–	0.81- 25.0)[2]	0.81	Process data word: The following operation is ORed with the parameter given in the data word and executed. The following are permissible: A-, AN-, O-, ON-, S-, R-, =-, SU, RU, TB, FRT, RT, SFT, SRT, SPT, SST, SET, FRC, RC, SC, CDC, CUC, L-, LD-, T-, JU, JC, DO, DOC, JZ, JN, JP, JO, SLW, SRW, D, I CDB, JU-, JC-
DO FW	0 to 254	4E 00			)[1]	)[1]	–	–	–	–	0.81- 24.0)[2]	0.81	Process flag word: The following operation specified is ORed with the parameter specified in the flag word and executed. for permissible substitutions, see DO DW.
D	0 to 255	19 00			N	N	–	–	–	–	5.8	0.25	Decrement: the low-order byte of ACCUM 1 (without carry)
I	0 to 255	11 00			N	N	–	–	–	–	5.8	0.25	Increment: the low-order byte of ACCUM 1 (without carry)

)[1] See relevant operation description
)[2] Plus time of substituted operation
)[3] Jump not executed/jump executed

System operations

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Operation	Parameters	Machine code (hexadecimal)			RLO-dependent?	RLO reloaded?	Condition codes affected				Execution time in µs		Function
		Word 0 B0 B1	Word 1 B2 B3	Word 2 B4 B5			CC1	CC0	OV	OS	Standard interpreter	ACOP	
Load and transfer operations													
LIR	0	40 00			N	N	–	–	–	–	10.2	10.2	Load register (indirectly): with the contents of the memory word addressed by ACCUM 1.
TIR	2	48 00			N	N	–	–	–	–	10.2	10.2	Transfer register contents (indirectly) into the memory word addressed by ACCUM 1.
		+ register address (hexadecimal)											
TNB	0 to 255	03 00			N	N	–	–	–	–	21.2-148	21.2-148	Block transfer byte by byte, Source in ACCUM 2, destination in ACCUM 1
		+ block length in bytes (hexadecimal)											
TNW	0 to 255	43 00			N	N	–	–	–	–	22.0-148	22.0-148	Block transfer word by word, Source in ACCUM 2, destination in ACCUM 1
		+ block length in words (hexadecimal)											

System operations

for program blocks (PBs)
for function blocks (FBs) + (FXs)
for sequence blocks (SBs)
for organization blocks (OBs)

Operation	Parameter	Maschine code (hexadecimal)			RLO-dependent	RLO reloaded	Condition codes affected				Execution time in µs		Function
		WordØ BØ B1	Word1 B2 B3	Word2 B4 B5			CC1	CCØ	OV	OS	Standardinterpreter	ACOP	
Arithmetic operations													
ADD BF	–127 bis +127	5Ø ØØ			N	N	–	–	–	–	5.2	0.25	Add byte constant (fixed point) to ACCUM 1
			+ constant (16 bit)										
ADD KF	–32 768 to +32 767	58 ØØ	ØØ ØØ		N	N	–	–	–	–	11.0	0.5	Add fixed-point constant (word) to ACCUM 1
			+constant (16 bit)										
Other operations													
SED	0 – 31	78 Ø6	ØØ ØØ				–	–	–	–	24.2-27.4	24.2-27.4	Set user semaphore
SEE	0 – 31	78 Ø7	ØØ ØØ				–	–	–	–	33.6-66.4	33.6-66.4	Reset user semaphore
STS		7Ø ØØ					–	–	–	–	30.0	30.0	Stop
TAK		7Ø Ø2					–	–	–	–	14.8	0.25	Swap the contents of ACCUM 1 and ACCUM 2
L RS	0 – 255	62 ØØ			N	N	–	–	–	–	10.6	0.5	Load a word from the system data area
T RS	0 – 255	63 ØØ			N	N	–	–	–	–	8.8	0.5	Transfer a word to the system data area
D RS	0 – 255	18 ØØ			N	N	–	–	–	–	0.31)[1]	0.31)[1]	Execute a statement in the system data area
LIM		7Ø ØC			N	N	–	–	–	–	13.6	13.6	Read interrupt mask
SIM		7Ø ØD			N	N	–	–	–	–	15.4-150	15.4-150	Set interrupt mask
JU OB	0-255	6D ØØ			N	J	–	–	–	0	31.8	2.0-25.0	Unconditinal jump to OB
JUC OB	0-255	4D ØØ			J	J	–	–	–	0	33.8)[2]	2.0-25.0	conditional jump to OB, if result = ”1”
			+OB number										
AFS		7800			N	N	–	–	–	–	6.4	6.4	Addressing fault inhibit
AFF		7810			N	N	–	–	–	–	6.4	6.4	Addressing fault enable

)[1] plus time for substituted operations
)[2] if result = 1

Alphabetical list of operations

Operation	from page	Operation	from page
! = D	16	DO FW	34
! = F	12	DO	16
! = G	14	DOC	18
)	4	BE	18
+ F	12	BEU	18
+ G	14	BEC	18
- F	12	BLD	18
- G	14	D	34
: F	12	DED	30
: G	14	DEF	30
< = D	16	DUD	30
< = F	12	DUF	30
< = G	14	FDG	30
< D	16	FRT	26
< F	12	FRC	26
< G	14	GDF	30
=	6	I	34
= =	24	CFW	30
> < D	16	CSD	30
> < F	12	CSW	30
> < G	14	L	8,10
> = D	16	L =	28
> = F	12	L RS	38
> = G	14	LD	8
> D	16	LD =	28
> F	12	LDW =	28
> G	14	LIM	38
C	18	LIR	36
ADD BN	38	LW =	28
ADD KF	38	NOP	18
CX	18	O	2,4
DO =	34	O (	4
DO RS	38	O =	20
DO DW	34	ON	2

Operation	from page	Operation	from page
ON =	20	SSU	26
OW	20	STP	18
TB	22	STS	38
TBN	22	SU	24
R	4,6	SE	6
RB =	24	SSD	32
RD =	24	SSW	32
RR =	26	SEC =	26
RU	24	T	10
S	4	T =	28
S =	24	T RS	38
SQ	6	TAK	38
SFD=	26	TIR	36
SI	6	TNB	36
SR =	26	TNW	36
SEE	38	A	2
SED	38	A (	4
SI	6	A =	20
SIP=	26	AN	2
SIM	38	AN =	20
SLD	32	AW	20
SLW	32	x F	12
JU	16	x G	14
JU =	32	XOW	20
JC	18	CD	6
JC =	32	CU	6
JM =	32		
JN =	32		
JO =	34		
JP =	32		
JOS =	34		
JZ =	32		
SRW	32		
SS	6		