Pocket Guide CPU 945

Index

	Page
Explanation of the Operations List	1
Explanation of the Operands	3
Basic Operations	
Boolean Logic Operations	8
Set/Reset Operations	8
Load Operations	10
Transfer Operations	16
Timer Operations	22
Counter Operations	22
Comparison Operations	24
Arithmetic Operations	28
Block Call Operations	30
Return Operations	34
”No” Operations	34
Stop Operation	34
Display Generation Operations	34
Supplementary Operations	
Boolean Logic Operations	38
Bit Test Operations	40
Bit Set Operations	42
Set/Reset Operations	42
Timer and Counter Operations	44
Load and Transfer Operations	46
Conversion Operations	46
Shift and Rotate Operations	48
Jump Operations	50
Disable and Enable Interrupts	54
Decrement/Increment	54
Other Operations	56
System Operations	
Set Operations	58
Load and Transfer Operations	58
Arithmetic Operations	64
Other Operations	66
Machine Code Listing	68
Alphabetical Index of Operations	80
List of Organization Blocks	84
List of Function Blocks	92
Data Block 1	96
Evaluation of CC 1 and CC 0	110

Explanation of the Operations List

Abbreviations	Explanations
ACCU 1	ACCUulator 1, 32 bits (when ACCU 1 is loaded, any prior existing contents are shifted into ACCU 2)*
ACCU 2	ACCUulator 2, 32 bits
ACCU x-H	High-word (bits 16 to 31) of ACCU x
ACCU x-L	Low-word (bits 0 to 15) of ACCU x
CC0/CC1	Condition code 0/Condition code 1
CSF	STEP 5 control system flowchart method of representation
Formal operand	Expression with a maximum of 4 characters. The first character must be a letter of the alphabet.
LAD	STEP 5 ladder diagram method of representation
OS	Stored overflow; This condition code is set if at least one arithmetic operation causes an overflow (used to detect arithmetic errors).
OV	Overflow. This condition code is set, e.g., if the number range is exceeded during arithmetic operations.
PII	Process image input
PIQ	Process image output
RLO	Result of logic operation
RLO reloaded Y N	The RLO does not change. The RLO cannot be combined any further. When the next binary operation takes place (but not assignment operation), the RLO is reloaded. Depending on whether the operation affects the RLO, the RLO is combined further or left unchanged according to the operation and the status of the bit that was scanned.

* not for LIR, LDI, LIM, MSA, MBA

Explanation of the Operations List

Abbreviations	Explanations
RLO dependent Y / N	The statement is executed only if the RLO is ”1”. The statement is executed only on positive/negative edge change of the RLO. The statement is always executed.
RLO affected Y/N	The RLO is affected/not affected by the operation.
STL	STEP 5 statement list method of representation

Explanation of the Operands

Abbr	Explanation	Permissible Value Range for Operands	Size in Bits
BN	Byte constant (fixed-point number)	- 128 to+127	8
C	Counter - for the bit test and set operations (system operations)	0 to 127 0.0 to 127.15	-
D	Data word	0.0 to 255.15	1
DB	Data block	0 to 255	-
DD	Data double word	0 to 254	32
DH	Constant (hexadecimal code as double word)	0 to FFFF FFFF	32
DL	Data word (left byte)	0 to 255	8
DR	Data word (right byte)	0 to 255	8
DW	Data word	0 to 255	16
DX	Data block (extension)	0 to 255	-
F	Flag	0.0 to 255.7	1
FB	Function block	0 to 255	-
FD	Flag double word	0 to 252	32
FW	Flag word	0 to 254	16
FX	Function block (extension)	0 to 255	-
FY	Flag byte	0 to 255	8
I	Input	0.0 to 127.7	1

Abbr	Explanation	Permissible Value Range for Operands	Size in Bits
IB	Input byte	0 to 127	8
ID	Input double word	0 to 124	32
IW	Input word	0 to 126	16
KB	Constant (1 byte)	0 to 255	8
KC	Constant (count)	0 to 999	16
KF	Constant (fixed-point number)	- 32768 to+32767	16
KG	Constant (floating-point number)	± 0.1469368 × 10^{-38} to ± 0.1701412 × 10^{39} and 0,0 × 10^{0}	32
KH	Constant (hexadecimal code)	0 to FFFF	16
KM	Constant (2 byte bit pattern)	arbitrary bit pattern (16 bit)	16
KS	Constant (2 characters)	any two alphanumeric characters	16
KT	Constant (time)	0.0 to 999.3	16
KY	Constant (2 bytes)	0 to 255 (per byte)	16
OB [1]	Organization block	0 to 255	-
OW	Word of the extended I/O area	0 to 254	16
PB	Program block (with block call and return operations)	0 to 255	-

1 For an overview of the organization blocks and their functions see page 84

Abbr	Explanation	Permissible Value Range for Operands	Size in Bits
PW	Peripheral word - Digital inputs - Analog inputs - Digital outputs - Analog outputs	 0 to 126 128 to 254 0 to 126 128 to 254	16
PY	Peripheral byte - Digital inputs - Analog inputs - Digital outputs - Analog outputs	 0 to 127 128 to 255 0 to 127 128 to 255	8
Q	Output	0.0 to 127.7	1
QB	Output byte	0 to 127	8
QD	Output double word	0 to 124	32
QW	Output word	0 to 126	16
QY	Byte of the extended I/O area	0 to 255	8
RS	System data area System data word - for load operations and transfer operations - for bit test and setting operations	 0 to 255 0.0 to 255.15	 16 1
RT	Extended system data area System data word - for load operations and transfer operations - for bit test and setting operations	 0 to 255 0.0 to 255.15	 16 1
S	Flag, additional	0.0 to 4095.7	1
SB	Sequence block	0 to 255	-
SD	Flag double word, additional	0 to 4092	32

Abbr	Explanation	Permissible Value Range for Operands	Size in Bits
SW	Flag word, additional	0 to 4094	16
SY	Flag byte, additional	0 to 4095	8
T	Timer - for the bit test and set operations (system operations)	0 to 127 0.0 to 127.15	-

Basic Operations

for organization blocks (OB) for function blocks (FB)

for program blocks (PB) for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Boolean Logic Operations							
A	I, Q, F, T, C S, D	N	Y	N	0.1 0.35		Scan operand for ”1” and combine with RLO through logic AND
AN	I, Q, F, T, C S, D	N	Y	N	0.1 0.35		Scan operand for ”0” and combine with RLO through logic AND
O	I, Q, F, T, C S, D	N	Y	N	0.1 0.35		Scan operand for ”1” and combine with RLO through logic OR
ON	I, Q, F, T, C S, D	N	Y	N	0.1 0.35		Scan operand for ”0” and combine with RLO through logic OR
O		N	Y	Y	0.1		Combine AND operations through logic OR
A(		N	Y	Y	0.1		Combine expressions enclosed in parentheses through logic AND (8 levels)
O(		N	Y	Y	0.1		Combine expressions enclosed in parentheses through logic OR (8 levels)
)		N	Y	N	0.1		Close parenthesis (conclusion of a parenthetical expression)
Set/Reset Operations							
S	I, Q, F S, D	Y	N	Y	0.1 0.45		Set operand to ”1”
R	I, Q, F S, D	Y	N	Y	0.1 0.45		Reset operand to ”0”

Basic Operations

for organization blocks (OB) for function blocks (FB)
for program blocks (PB) for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Set/Reset Operations (cont.)							
=	I, Q, F S, D	Y	N	Y	0.1 0.45		Assign value of RLO to operand.
Load Operations							
L	IB	N	N	N	0.1		Load an input byte from the PII into ACCU 1. Byte n bits 0 to 7
L	IW	N	N	N	0.1		Load an input word from the PII into ACCU 1. Byte n bits 8 to 15; byte n+1 bits 0 to 7
L	ID	N	N	N	0.2		Load an input double word into ACCU 1: Byte n bits 24 to 31; byte n+1 bits 16 to 23; Byte n+2 bits 8 to 15; byte n+3 bits 0 to 7;
L	QB	N	N	N	0.1		Load an output byte from the PIQ into ACCU 1. Byte n bits 0 to 7
L	QW	N	N	N	0.1		Load an output byte from the PIQ into ACCU 1: Byte n bits 8 to 15; byte n+1 bits 0 to 7
L	QD	N	N	N	0.2		Load an output double word into ACCU 1: Byte n bits 24 to 31; byte n+1 bits 16 to 23; Byte n+2 bits 8 to 15; byte n+3 bits 0 to 7;
L	PY	N	N	N	0.2*		Load a peripheral word from the digital/analog inputs into ACCU 1. Byte n bits 0 to 7

***** + ready delay time of the I/O modules addressed

Basic Operations

for organization blocks (OB) for function blocks (FB)
for program blocks (PB) for sequence blocks (SB)

Opera-tion (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Load operations (cont.)							
L	PW	N	N	N	0.2**		Load a peripheral word from the digital/analog inputs into ACCU 1: Byte n bits 8 to 15; byte n+1 bits 0 to 7
L	OY	N	N	N	0.2*		Load a byte of the extended I/O area into ACCU 1: Byte n bits 0 to 7
L	OW	N	N	N	0.2**		Load a word of the extended I/O area into ACCU 1: Byte n bits 8 to 15; byte n+1 bits 0 to 7
L	FY	N	N	N	0.1		Load a flag byte into ACCU 1. Byte n bits 0 to 7
L	FW	N	N	N	0.1		Load a flag word into ACCU 1: Byte n bits 8 to 15; byte n+1 bits 0 to 7
L	FD	N	N	N	0.2		Load a flag double word into ACCU 1: Byte n bits 24 to 31; byte n+1 bits 16 to 23; Byte n+2 bits 8 to 15; byte n+3 bits 0 to 7
L	SY	N	N	N	0.35		Load an extended flag byte into ACCU 1: Byte n bits 0 to 7
L	SW	N	N	N	0.35		Load an extended flag word into ACCU 1: Byte n bits 8 to 15; byte n+1 bits 0 to 7
L	SD	N	N	N	0.45		Load an extended flag double word into ACCU 1: Byte n bits 24 to 31; byte n+1 bits 16 to 23; Byte n+2 bits 8 to 15; byte n+3 bits 0 to 7

***** + ready delay time of the I/O modules addressed

****** +2× ready delay time of the I/O modules addressed

Basic Operations

for organization blocks (OB) for function blocks (FB)
for program blocks (PB) for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Load operations (cont.)							
L	DL	N	N	N	0.2		Load a data word (left-hand byte) of the current data block into ACCU 1: Byte n bits 0 to 7
L	DR	N	N	N	0.2		Load a data word (right-hand byte) of the current data block into ACCU 1: Byte n bits 0 to 7
L	DW	N	N	N	0.2		Load a data word of the current data block into ACCU 1: Byte n bits 8 to 15; byte n+1 bits 0 to 7
L	DD	N	N	N	0.3		Load a data double word of the current data block into ACCU 1: Byte n bits 24 to 31; byte n+1 bits 16 to 23; Byte n+2 bits 8 to 15; byte n+3 bits 0 to 7
L	KB	N	N	N	0.1		Load a constant (1-byte number) into ACCU 1
L	KC	N	N	N	0.2		Load a constant (2 characters in ASCII format) into ACCU 1
L	KF	N	N	N	0.2		Load a constant (fixed-point number) into ACCU 1
L	KH	N	N	N	0.2		Load a constant (hexadecimal code) into ACCU 1
L	DH	N	N	N	0.3		Load a data double word of the current data block into ACCU 1
L	KG	N	N	N	0.3		Load a constant (floating-point-number) into ACCU 1
L	KM	N	N	N	0.2		Load a constant (bit pattern) into ACCU 1

Basic Operations

for organization blocks (OB) for function blocks (FB)
for program blocks (PB) for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Load Operations (cont.)							
L	KY	N	N	N	0.2		Load a constant (bit pattern) into ACCU 1
L	KT	N	N	N	0.2		Load a constant (count in BCD) into ACCU 1
L	KC	N	N	N	0.2		Load a constant (count in BCD) into ACCU 1
L	T, C	N	N	N	0.1		Load a time or count (in binary code) into ACCU 1
LC	T, C	N	N	N	0.15		Load times or counts (in BCD) into ACCU 1
Transfer Operations							
T	IB	N	N	N	0.1		Transfer the contents of ACCU 1-L (bits 0 to 7) to an input byte (into the PII)
T	IW	N	N	N	0.1		Transfer the contents of ACCU 1-L to an input word (into the PIQ): Bits 8 to 15 byte n; bits 0 to 7 byte n+1
T	ID	N	N	N	0.2		Transfer the contents of ACCU 1 to an input double word (into the PII): Bits 24 to 31 byte n; bits 16 to 23 byte n+1; Bits 8 to 15 byte n+2; bits 0 to 7 byte n+3
T	QB	N	N	N	0.1		Transfer the contents of ACCU 1-L (bits 0 to 7) to an output word (into the PIQ)
T	QW	N	N	N	0.1		Transfer the contents of ACCU 1 to an output word (into the PIQ): Bits 8 to 15 byte n; bits 0 to 7 Byte n+1

Basic Operations

for organization blocks (OB) for function blocks (FB)
for program blocks (PB) for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Transfer Operations (cont.)							
T	QD	N	N	N	0.2		Transfer the contents of ACCU 1 to an output double word (into the PIQ): Bits 24 to 31 byte n; bits 16 to 23 byte n+1; Bits 8 to 15 byte n+2; bits 0 to 7 byte n+3
T	PY	N	N	N	0.2*		Transfer the contents of ACCU 1-L (bits 0 to 7) to an I/O byte of the digital output modules with updating of the PIQ or analog output modules
T	PW	N	N	N	0.2**		Transfer the contents of ACCU 1-L to an I/O byte of the digital output modules with updating of the PIQ or analog output modules. Bits 8 to 15 byte n; bits 0 to 7 byte n+1
T	OY	N	N	N	0.2*		Transfer the contents of ACCU 1-L (Bits 0 to 7) to a byte of the extended peripherals of the digital or analog outputs
T	OW	N	N	N	0.2**		Transfer the contents of ACCU 1-L to a word of the extended peripherals of the digital or analog outputs: Bits 8 to 15 byte n; bits 0 to 7 byte n+1
T	FY	N	N	N	0.1		Transfer the contents of ACCU 1-L (bits 0 to 7) to a flag byte
T	FW	N	N	N	0.1		Transfer the contents of ACCU 1-L to a flag word: Bits 8 to 15 byte n; bits 0 to 7 byte n+1

***** +ready delay time of the I/O modules addressed

****** +2× ready delay time of the I/O modules addressed

Basic Operations

for organization blocks (OB) for function blocks (FB)
for program blocks (PB) for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Transfer Operations (cont.)							
T	FD	N	N	N	0.2		Transfer the contents of ACCU 1 to a flag double word: Bits 24 to 31 byte n; bits 16 to 23 byte n+1; Bits 8 to 15 byte n+2; bits 0 to 7 byte n+3
T	SY	N	N	N	0.35		Transfer the contents of ACCU 1-L (bits 0 to 7) to an extended flag byte
T	SW	N	N	N	0.35		Transfer the contents of ACCU 1-L to an extended flag word: Bits 8 to 15 byte n; bits 0 to 7 byte n+1
T	SD	N	N	N	0.45		Transfer the contents of ACCU 1 to an extended flag double word: Bits 24 to 31 byte n; bits 16 to 23 byte n+1; Bits 8 to 15 byte n+2; bits 0 to 7 byte n+3
T	DL	N	N	N	0.2		Transfer the contents of ACCU 1-L (bits 0 to 7) to a data word (left-hand byte) of the current data block (DB/DX)
T	DR	N	N	N	0.2		Transfer the contents of ACCU 1-L (bits 0 to 7) to a data word (right-hand byte) of the current data block(DB/DX)
T	DW	N	N	N	0.2		Transfer the contents of ACCU 1-L to a data word of the current data block (DB/DX)
T	DD	N	N	N	0.3		Transfer the contents of ACCU 1 to a data double word of the current data block (DB/DX): Bits 24 to 31 byte n; bits 16 to 23 byte n+1; Bits 8 to 15 byte n+2; bits 0 to 7 byte n+3

Basic Operations

for organization blocks (OB)
for program blocks (PB)
for function blocks (FB)
for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Timer Operations							
SP	T	Y	N	Y	0.1		Start a timer (stored in ACCU 1-L) as signal-contracting pulse
SE	T	Y	N	Y	0.1		Start a timer (stored in ACCU 1-L) as extended pulse (signal-contracting and stretching)
SD	T	Y	N	Y	0.1		Start an on-delay timer (stored in ACCU 1) as extended pulse
SS	T	Y	N	Y	0.1		Start a stored on-delay timer (stored in ACCU 1)
SF	T	Y	N	Y	0.1		Start an off-delay timer (stored in ACCU 1)
R	T	Y	N	Y	0.1		Reset a timer
Counter Operations							
CU	C	Y	N	Y	0.1		Counter counts up 1
CD	C	Y	N	Y	0.1		Counter counts down 1
S	C	Y	N	Y	0.1		Set counter with the value stored in ACCU 1-L (BCD number from 0 to 999)
R	C	Y	N	Y	0.1		Reset a timer

Basic Operations

for organization blocks (OB) for function blocks (FB)

for program blocks (PB) for sequence blocks (SB)

Opera-tion (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Comparison Operations							
!=F		N	Y	N	0.1		Compare two fixed-point numbers for ”equal to”. If ACCU 2-L=ACCU 1-L, the RLO is ”1”. CC 1/CC 0 are affected.
><F		N	Y	N	0.1		Compare two fixed-point numbers for ”not equal to”. If ACCU 2-L ACCU 1-L, the RLO is ”1”. CC 1/CC 0 are affected.
>F		N	Y	N	0.1		Compare two fixed-point numbers for ”greater than”. If ACCU 2-L>ACCU 1-L, the RLO is ”1”. CC 1/CC 0 are affected.
>=F		N	Y	N	0.1		Compare two fixed-point numbers for ”greater than or equal to”. If ACCU 2-L ACCU 1-L, the RLO is ”1”. CC 1/CC 0 are affected.
<F		N	Y	N	0.1		Compare two fixed-point numbers for ”less than”. If ACCU 2-L<ACCU 1-L, the RLO is ”1”. CC 1/CC 0 are affected.
<=F		N	Y	N	0.1		Compare two fixed-point numbers for ”less than or equal to”. If ACCU 2-L ACCU 1-L, the RLO is ”1”. CC 1/CC 0 are affected.
!=D		N	Y	N	0.1		Compare two fixed-point double words for ”equal to”: if ACCU 2=ACCU 1, the RLO is ”1”. CC 1/CC 0 are affected.
><D		N	Y	N	0.1		Compare two fixed-point double words for ”not equal to”: if ACCU 2 ACCU 1, the RLO is ”1”. CC 1/CC 0 are affected.

Basic Operations

for organization blocks (OB) for function blocks (FB)
for program blocks (PB) for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Comparison Operations (cont.)							
>D		N	Y	N	0.1		Compare two fixed-point double words for ”greater than”: if ACCU 2 > ACCU 1, the RLO is ”1”. CC 1/CC 0 are affected.
>=D		N	Y	N	0.1		Compare two fixed-point double words for ”not equal to”: if ACCU 2 ACCU 1, the RLO is ”1” CC 1/CC 0 are affected.
<D		N	Y	N	0.1		Compare two fixed-point double words for ”less than”: if ACCU 2 < ACCU 1, the RLO is ”1”. CC 1/CC 0 are affected.
<=D		N	Y	N	0.1		Compare two fixed-point double words for ”less than or equal to”: if ACCU 2 ACCU 1, the RLO is ”1” CC 1/CC 0 are affected.
!=G		N	Y	N	0.55		Compare two floating-point numbers for ”equal to”: if ACCU 2 > ACCU 1, the RLO is ”1”. CC 1/CC 0 are affected.
><G		N	Y	N	0.55		Compare two floating-point numbers for ”not equal to”: if ACCU 2 ACCU 1, the RLO is ”1”. CC 1/CC 0 are affected.
>G		N	Y	N	0.55		Compare two floating-point numbers for ”greater than”: if ACCU 2 > ACCU 1, the RLO is ”1”. CC 1/CC 0 are affected.
>=G		N	Y	N	0.55		Compare two floating-point numbers for ”less than or equal to”: if ACCU 2 ACCU 1, the RLO is ”1”. CC 1/CC 0 are affected.

Basic Operations

for organization blocks (OB) for function blocks (FB)
for program blocks (PB) for sequence blocks (SB)

Opera-tion (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Comparison Operations (cont.)							
<G		N	Y	N	0.55		Compare two floating-point numbers for ”less than”: if ACCU 2 < ACCU 1, the RLO is ”1”. CC 1/CC 0 are affected.
<=G		N	Y	N	0.55		Compare two floating-point numbers for ”less than or equal to”: if ACCU 2 ACCU 1, the RLO is ”1”. CC 1/CC 0 are affected.
Arithmetic Operations							
+F		N	N	N	0.1		Add two fixed-point numbers: ACCU 1-L=(ACCU 2-L)+(ACCU 1-L). CC 1/CC 0/OV/OS can be evaluated.
- F		N	N	N	0.1		Subtract one fixed-point number from another: ACCU 1-L=(ACCU 2-L)-(ACCU 1-L). CC 1/CC 0/OV/OS can be evaluated.
×F		N	N	N	0.35		Multiply one fixed-point number by another: ACCU 1=(ACCU 2-L)×(ACCU 1-L). CC 1/CC 0/OV/OS can be evaluated.
: F		N	N	N	0.4		Divide one fixed-point number by another: (ACCU 2-L) : (ACCU 1-L). In ACCU 1-L : result; in ACCU 1-H : remainder. CC 1/CC 0/OV/OS can be evaluated.
+D		N	N	N	0.1		Subtract two fixed-point numbers (32 bits): ACCU 1=ACCU 2+ACCU 1. CC 1/CC 0/OV/OS can be evaluated.

Basic Operations

for organization blocks (OB) for function blocks (FB)
for program blocks (PB) for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Arithmetic Operations (cont.)							
- D		N	N	N	0.1		Subtract two fixed-point numbers (32 bits): ACCU 1=ACCU 2 - ACCU 1. CC 1/CC 0/OV/OS can be evaluated.
+G		N	N	N	0.75		Add two floating-point numbers: ACCU 1=ACCU 2+ACCU 1. CC 1/CC 0/OV/OS can be evaluated.
- G		N	N	N	0.75		Subtract one floating-point number from another: ACCU 1=ACCU 2 - ACCU 1. CC 1/CC 0/OV/OS can be evaluated.
×G		N	N	N	0.75		Multiply one floating-point number by another: ACCU 1=ACCU 2×ACCU 1. CC 1/CC 0/OV/OS can be evaluated.
: G		N	N	N	1.35		Divide one floating-point number by another: ACCU 1=ACCU 2 : ACCU 1. CC 1/CC 0/OV/OS can be evaluated.
Block Call Operations							
JU	OB	N	N	Y	1		Jump unconditionally to an organization block.
JU	PB	N	N	Y	1		Jump unconditionally to a program block.
JU	FB	N	N	Y	1.5		Jump unconditionally to a function block.
DOU	FX	N	N	Y	25.1		Jump unconditionally to a function block (FX).

Basic Operations

for organization blocks (OB) for function blocks (FB)
for program blocks (PB) for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Block Call Operations (cont.)							
JU	SB	N	N	Y	1		Jump unconditionally to a sequence block.
JC	OB	Y	Y[1]	Y	1 0.1		Jump conditionallly to an organization block. Time applies for RLO=1/RLO=0
JC	PB	Y	Y[1]	Y	1 0.1		Jump conditionally to a program block. Time applies for RLO=1/RLO=0
JC	FB	Y	Y[1]	Y	1.5 0.35		Jump conditionally to a function block. Time applies for RLO=1/RLO=0
DOC	FX	Y	Y[1]	Y	25.1 0.45		Jump conditionally to a function block (FX). Time applies for RLO=1/RLO=0
JC	SB	Y	Y[1]	Y	1 0.1		Jump conditionally to a sequence block. Time applies for RLO=1/RLO=0
A	DB	N	N	N	0.5		Call a data block.
CX	DX	N	N	N	0.6		Call an extended data block.
E	DB	N	N	N	45 to 60		Generate a data block. The number of data words in the block must be stored in ACCU 1.
GX	DX	N	N	N	45 to 60		Generate an extended data block. The number of its data words must be stored in ACCU 1.

[1] RLO is set to 1

Basic Operations

for organization blocks (OB) for function blocks (FB)

for program blocks (PB) for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Return Operations							
BE		N	N	Y	0.8		Block end (termination of a block)
BEC		Y	Y[1]	Y	0.8 0.1		Block end, conditional Time applies for RLO=1/RLO=0
BEU		N	N	Y	0.8		Block end, unconditional
”No” Operations							
NOP 0		N	N	N	0.1		No operation
NOP 1		N	N	N	0.1		No operation
Stop Operation							
STP		N	N	N	32		Stop at the end of program execution: current program processing is brought to an end; the PIQ is output; afterwards, the PLC enters STOP
Display Generation Operations							
BLD 0 to 255		N	N	N	0.1		Is treated by the CPU like a no-operation.
BLD 130		N	N	N	0.1		Display generation operation for the programmer: carriage return generates blank line

1 RLO is set to 1

Basic Operations

for organization blocks (OB) for function blocks (FB)

for program blocks (PB) for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Display Generation Operations (cont.)							
BLD 131		N	N	N	0,1		Display generation operation for the programmer: switch over to statement list (STL).
BLD 132		N	N	N	0,1		Display generation operation for the programmer: switch over to control system flowchart (CSF).
BLD 133		N	N	N	0,1		Display generation operation for the programmer: switch over to ladder diagram (LAD).
BLD 255		N	N	N	0,1		Display generation operation for the programmer: segment termination.

Supplementary Operations

for organization blocks (OB) for function blocks (FB)

for program blocks (PB) for sequence blocks (SB)

Oper-ation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Boolean Logic Operations							
A=	Formal oper. I, Q, F, T, C	N	Y	N	0.5*		AND operation: scan formal operand for ”1”. (Data type: BI)
AN=	Formal oper. I, Q, F, T, C	N	Y	N	0.5*		AND operation: scan formal operand for ”0”. (Data type: BI)
O=	Formal oper. I, Q, F, T, C	N	Y	N	0.5*		OR operation: scan formal operand for ”1”. (Data type: BI)
ON=	Formal oper. I, Q, F, T, C	N	Y	N	0.5*		OR operation: scan formal operand for ”0”. (Data type: BI)
AW		N	N	N	0.1		And ACCU 2-L and ACCU 1-L (word operation); result in ACCU 1-L. Result can be evaluated via CC 1/CC 0. ACCU 1-H remains unchanged.
OW		N	N	N	0.1		Or ACCU 2-L and ACCU 1-L (word operaton); result in ACCU 1-L. Result can be evaluated via CC 1/CC 0. ACCU 1-H remains unchanged.
XOW		N	N	N	0.1		Exclusive-OR ACCU 2-L and ACCU 1-L (word operation); result in ACCU 1-L. Result can be evaluated via CC 1/CC 0. ACCU 1-H remains unchanged.

* plus execution time of the substituted operation

Supplementary Operations

for organization blocks (OB) for function blocks (FB)

for program blocks (PB) for sequence blocks (SB)

Oper-ation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Bit Test Operations							
TB	I, Q, F	N	Y	N	0.2		Test bit of an input, output or flag byte for ”1”
TB	T, C	N	Y	N	0.2		Test a timer or counter word bit for ”1”
TB	D	N	Y	N	0.35		Test a data word bit (in a DB/DX) for ”1”
TB	RS	N	Y	N	0.35		Test a data word bit in the system data range for ”1”
TB	RT	N	Y	N	0.35		Test bit of a data word in the extended area of the system data for ”1”
TBN	I, Q, F	N	Y	N	0.2		Test bit of an input, output or flag byte for ”0”
TBN	T, C	N	Y	N	0.2		Test a timer or counter word bit for ”0”
TBN	D	N	Y	N	0.35		Test a data word bit (in a DB/DX) for ”0”
TBN	RS	N	Y	N	0.35		Test a data word bit in the system data range for ”0”
TBN	RT	N	Y	N	0.35		Test bit of a data word in the extended area of the system data for ”0”

Supplementary Operations

for organization blocks (OB) for function blocks (FB)

for program blocks (PB) for sequence blocks (SB)

Oper-ation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Bit Set Operations							
SU	I, Q, F	N	N	Y	0.2		One input bit (in PII); set an output bit (in PIQ) or a flag bit unconditionally
SU	T, C	N	N	Y	0.2		Set a timer or counter word bit unconditionally
SU	D	N	N	Y	0.45		Set a data word bit (in a DB/DX) unconditionally
RU	I, Q, F	N	N	Y	0.2		One input bit (in PII); reset an output bit (in PIQ) or a flag bit unconditionally
RU	T, C	N	N	Y	0.2		Reset a timer or counter word bit unconditionally
RU	D	N	N	Y	0.45		Reset a data word bit unconditionally
Set/Reset Operations							
S=	Formal oper. I, Q, F	Y	N	Y	0.5*		Set a formal operand, (with RLO =1). (Data type: BI)
RB=	Formal oper. I, Q, F	Y	N	Y	0.5*		Reset a formal operand, (with RLO =1). (Data type: BI)
==	Formal oper. I, Q, F	Y	N	Y	0.5*		The value of the RLO is assigned to the status of the formal operand. (Data type: BI)

* plus execution time of the substituted operation

Supplementary Operations

for organization blocks (OB) for function blocks (FB)

for program blocks (PB) for sequence blocks (SB)

Oper-ation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Timer and Counter Operations							
FR	T, C	Y	N	Y	0.1		Enable a timer/counter for cold restart. If RLO="1", - 'FR T' restarts the timer - 'FR C' sets, decrements, or increments the counter
FR=	Formal oper. T, C	Y	N	Y	0.5*		Enable formal operand (timer/counter) for cold re-start (for detailed description, see "FR" operation)
RD=	Formal oper. T, C	Y	N	Y	0.5*		Reset a formal operand (digital), (with RLO =1)
SP=	Formal oper. T	Y	N	Y	0.5*		Start a timer (formal operand) as pulse with the value stored in ACCU 1
SD=	Formal oper. T	Y	N	Y	0.5*		Start an on-delay timer (formal operand) with the value stored in ACCU 1
SEC=	Formal oper. T, C	Y	N	Y	0.5*		Start a timer (formal operand) as extended pulse with the value stored in ACCU 1, or set a counter (formal operand) with the next indicated count value
SSU=	Formal oper. T, C	Y	N	Y	0.5*		Start a stored on-delay timer (formal operand) with the value stored in ACCU 1, or increment a counter (formal operand)
SFD=	Formal oper. T, C	Y Y	N	Y	0.5*		Start an off-delay timer ( ) (formal operand) with the value stored in ACCU 1, or decrement a counter ( ) (formal operand)

* plus execution time of the substituted operation

Supplementary Operations

for organization blocks (OB) for function blocks (FB)

for program blocks (PB) for sequence blocks (SB)

Oper-ation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Load and Transfer Operations							
L=	Formal oper. I, Q, F, T, C	N	N	N	0.5*		Load the value of the formal operand into ACCU 1 (Data type: BY, W, D; Actual operands: FD, FY, FW, IB, IW, ID, QB, QW, QD, PY, PW, DL, DR, DW, DD, RS, RT, C, T)
LC=	Formal oper. T, C	N	N	N	0.5*		Load the value of the formal operand in BCD code into ACCU 1.
LW=	Formal oper.	N	N	N	0.45		Load a formal operand bit pattern into ACCU 1 (Parameter type: D; data type: KF, KH, KM, KY, KS, KT, KC)
LDW=	Formal oper.	N	N	N	0.55		Load the bit pattern (32 bits) of a formal operand into ACCU 1 (parameter type: D; data type: KG)
T=	Formal oper. I, Q, F	N	N	N	0.5*		Transfer the contents of ACCU 1 to the formal operand (Data type: BY, W, D; Actual operands: FD, FY, FW, IB, IW, ID, QB, QW, QD, PY, PW, DL, DR, DW, DD, RS, RT)
Conversion Operations							
CFW		N	N	N	0.1		Form one's complement of ACCU 1-L (Bits 0 to 15).
CSW		N	N	N	0.1		Form two's complement of ACCU 1-L (Bits 0 to 15). CC 1/CC 0 and OV are affected
CSD		N	N	N	0.1		Form two's complement of ACCU 1 (Bits 0 to 31). CC1/CC0 and OV are affected.
DEF		N	N	N	0.55		Convert a 16-bit fixed point from BCD into binary.

* plus execution time of the substituted operation

Supplementary Operations

for organization blocks (OB) for function blocks (FB)

for program blocks (PB) for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Conversion Operations (cont.)							
DUF		N	N	N	0.65		Convert a 16-bit fixed point from binary into BCD
DED		N	N	N	0.55		Convert a 32-bit fixed point from BCD into binary
DUD		N	N	N	0.80		Convert a 32-bit fixed point from binary into BCD
FDG		N	N	N	0.45		Convert a 32-bit fixed-point number into a floating-point number
GFD		N	N	N	0.45		Convert a floating-point number into a 32-bit fixed-point number
Shift and Rotate Operations							
SLW	Parameter n=0 to 15	N	N	N	0.1		Shift the contents of ACCU 1-L (word) to the left by the value specified in the parameter. Positions becoming vacant are padded with zeros. CC 1/CC 0 are affected
SRW	Parameter n=0 to 15	N	N	N	0.1		Shift the contents of ACCU 1-L (word) to the right by the value specified in the parameter. Positions becoming vacant are padded with zeros. CC 1/CC 0 are affected
SLD	Parameter n=0 to 32	N	N	N	0.1		Shift the contents of ACCU 1 (double word) to the left by the value specified in the parameter. Positions becoming vacant are padded with zeros. CC 1/CC 0 are affected

Supplementary Operations

for organization blocks (OB) for function blocks (FB)

for program blocks (PB) for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Shift and Rotate Operations (cont.)							
SSW	Parameter n=0 to 15	N	N	N	0.1		Shift the contents of ACCU 1-L (word) to the right by the value specified in the parameter. Positions becoming vacant are padded with the sign (bit 15). CC 1/CC 0 are affected.
SSD	Parameter n=0 to 32	N	N	N	0.1		Shift the contents of ACCU 1 (double word) to the right by the value specified in the parameter. Positions becoming vacant are padded with the sign (bit 31). CC 1/CC 0 are affected.
RLD	Parameter n=0 to 32	N	N	N	0.1		Rotate the contents of ACCU 1 (double word) to the left. Positions becoming vacant are padded with the shifted bits. CC 1/CC 0 are affected.
RRD	Parameter n=0 to 32	N	N	N	0.1		Rotate the contents of ACCU 1 (double word) to the right. Positions becoming vacant are padded with the shifted bits. CC 1/CC 0 are affected.
Jump Operations							
JU=	Symb. address max. 4 charact.	N	N	N	0.25		Unconditional jump to a symbolic address.
JC=	Symb. address max. 4 charact.	Y	Y[1]	Y	0.25 0.1		Conditional jump to a symbolic address, exectued only if RLO=1 (if RLO="0", it is set to "1") Time applies for RLO=1/RLO=0

1 RLO is set to "1".

Supplementary Operations

for organization blocks (OB) for function blocks (FB)

for program blocks (PB) for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Jump Operations (cont.)							
JZ=	Symb. address max. 4 charact.	N	N	N	0.25 0.1		Jump if the result is zero. The jump is made only if CC 1=0 and CC 0=0. The RLO is not changed. Time applies for RLO=1/RLO=0
JN=	Symb. address max. 4 charact.	N	N	N	0.25 0.1		Jump if the result is not zero. The jump is made only if CC 1 CC 0 . The RLO is not changed. Time applies for RLO=1/RLO=0
JP=	Symb. address max. 4 charact.	N	N	N	0.25 0.1		Jump if the result>0. The jump is made only if CC 1=1 und CC 0=0. The RLO is not changed. Time applies for RLO=1/RLO=0
JM=	Symb. address max. 4 charact.	N	N	N	0.25 0.1		Jump if the result <0. The jump is made only if CC 1=0 and CC 0=1. The RLO is not changed. Time applies for RLO=1/RLO=0
JO=	Symb. address max. 4 charact.	N	N	N	0.25 0.1		Jump on overflow: the jump is only made if the OV bit is set to 1. The RLO is not changed. Time applies for RLO=1/RLO=0
JOS=	Symb. address max. 4 charact.	N	N	N	0.25 0.1		Jmp on overflow latched: the jump is only made if the OS bit is set to 1. The RLO is not changed. Time applies for RLO=1/RLO=0
JUR		N	N	N	0.25		Linear program execution is interrupted and continued at the position determined by the jump distance. (Jump distance in words: -32768 to+32767)

Supplementary Operations

for organization blocks (OB) for function blocks (FB)

for program blocks (PB) for sequence blocks (SB)

Oper-ation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Disable and Enable Interrupts							
IA		N	N	N	28		Disable interrupt: I/O interrupt, time OB execution and time interrupt OB execution are disabled (does not apply for OBs 26, 33, 35)
RA		N	N	N	30 to 35		Enable interrupt: Cancels the effect of the IA operation (does not apply for OBs 26, 33, 35)
SIM		N	N	N	33 to 38		Set interrupt mask (32 bits)
LIM		N	N	N	29		Read interrupt mask (32 bits)
Decrement/Increment							
D	Parameter n=0 to 255	N	N	N	0.1		Decrement ACCU 1-LL by the number specified in the parameter (no carry)
I	Parameter n=0 to 255	N	N	N	0.1		Increment ACCU 1-LL by the number specified in the parameter (no carry)

Supplementary Operations

for organization blocks (OB) for function blocks (FB)

for program blocks (PB) for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Other Operations							
DO=	Formal oper. DB, PB, FB, SB, OB	N	N	N	0.5*		Process a block. (Only C DB, JU PB, JU FB, JU SB and JU OB can be substituted)
DO	DW**	N	N	N	0.25		Process data word. The next operation is combined through logic OR with the parameter specified in the data word and executed**
DO	FW**	N	N	N	0.25		Process flag word. The next operation is combined through logic OR with the parameter specified in the flag word and executed**

* plus execution time of the substituted operation

** Permissible operations:
A, AN, O, ON;[1]
S, R, =;[2]
FR T, R T, SF T, SR T, SP T, SS T, SE T;
FR C, R C, S C, CR C, CU C;
L, T[3]
LC
JU=, JC=, JZ=, JN=, JP=, JM=, JO=, SLW, SRW;
SLD, RLD, SSD, RRD,
D, I; ADD BN;
QDB, TNB, TNW, JU, JC

1 only for I, Q, F, T, C
2 only for I, Q, F
3 only for IB, QB, IW, QW, ID, QD, FY, FW, FD, DL, DR, DW, DD, RS, RT, PY, PT, OY, OW

System Operations

for organization blocks (OB) for function blocks (FB)
for program blocks (PB) for sequence blocks (SB)

Oper-ation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Set Operations							
SU	RS	N	N	Y	0.45		Set a bit in the system data area (register) unconditionally
SU	RT	N	N	Y	0.45		Set a bit in the extended system data area (register) unconditionally
RU	RS	N	N	Y	0.45		Reset a bit in the system data area (register) unconditionally
RU	RT	N	N	Y	0.45		Reset a bit in the extended system data area (register) unconditionally
Load and Transfer Operations							
LIR	0 (ACCU 1-L) 1 (ACCU 1-H) 2 (ACCU 2-L) 3 (ACCU 2-H) 8 (DBL)	N	N	N	0.25*		Load the 16-bit register specified with the contents of a memory word (addressed by ACCU 1)
TIR	0 (ACCU 1-L) 1 (ACCU 1-H) 2 (ACCU 2-L) 3 (ACCU 2-H) 8 (DBL)	N	N	N	0.25*		The contents of the specified 16-bit register are transferred to a memory word (addressed by ACCU 1)
LDI	A1 (ACCU 1) A2 (ACCU 2) SA (SAZ) BR (BR register)	N	N	N	0.4**		Load the specified 32-bit register with the contents of a memory word n (addressed by ACCU 1) and the following n+1, n+2 and n+3

***** When accessing the I/O area, + 2×ready delay time of the referenced I/O modules

****** When accessing the I/O area, + 4×ready delay time of the referenced I/O modules

System Operations

for organization blocks (OB) for function blocks (FB)
for program blocks (PB) for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Load and Transfer Operations (cont.)							
TDI	A1 (ACCU 1) A2 (ACCU 2) SA (SAZ) BR (BR register)	N	N	N	0.3*		Transfer the contents of the specified 32-bit register to a memory word n (addressed by ACCU 1) and to the following n+1, n+2 and n+3
TNW	Parameter n=0 to 255	N	N	N	0.15 + n × 0.35**		Block transfer word by word End address of target area: ACCU 1 End address of source area: ACCU 2 Only even addresses may be used. After execution of the operation, the ACCUs are decremented 2 times the number of words.
TNB	Parameter n=0 to 255	N	N	N	0.15 + n × 0.35**		Block transfer byte by byte End address of target area: ACCU 1 End address of source area: ACCU 2 After execution of the operation, the ACCUs are decremented by the number of bytes
L	RS	N	N	N	0.2		Load a word from the system data range into ACCU 1
L	RT	N	N	N	0.2		Load a word from the extended system data range into ACCU 1
T	RS	N	N	N	0.2		Transfer a word to the system data area
T	RT	N	N	N	0.2		Transfer a word to the extended system data area

***** When accessing the I/O area, + 4×ready delay time of the referenced I/O modules

****** When accessing the I/O area, + 1×ready delay time of the referenced I/O modules

System Operations

for organization blocks (OB) for function blocks (FB)
for program blocks (PB) for sequence blocks (SB)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Load and Transfer Operations (cont.)							
MBR	Parameter 0 to $FFFFF_H$	N	N	N	0.25		Load a 20-bit constant into the BR register
ABR	Offset - 32768 to +32767	N	N	N	0.25		The offset is multiplied by 2 (word offset). Add the result to the contents of the BR register
LRW	Offset - 32768 to +32767	N	N	N	0.35		The offset is multiplied by 2 (word offset). Add the result to the contents of the BR register and load the word with this address into ACCU 1-L
LRD	Offset - 32768 to +32767	N	N	N	0.45		The offset is multiplied by 2 (word offset). Add the result to the contents of the BR register and load the double word with this address into ACCU 1-L
TRW	Offset - 32768 to +32767	N	N	N	0.35		The offset is multiplied by 2 (word offset). Add the result to the contents of the BR register and transfer the contents of ACCU 1-L to the word with this address
TRD	Offset - 32768 to +32767	N	N	N	0.45		The offset is multiplied by 2 (word offset). Add the result to the contents of the BR register and transfer the contents of ACCU 1 to the double word with this address
MAS		N	N	N	0.25		Transfer the contents of ACCU 1 (bits 31 to 20 remain unchanged) to the base address register
MAB		N	N	N	0.1		Transfer the contents of ACCU 1 (bits 31 to 20 remain unchanged) to the base address register

System Operations

for organization blocks (OB) for function blocks (FB)

for program blocks (PB) for sequence blocks (SB)

Oper-ation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Load and Transfer Operations (cont.)							
MSA		N	N	N	0.1		Transfer the contents of the step address counter to ACCU 1 (bits 31 to 20 are set to zero) (address of the next command)
MSB		N	N	N	0.1		Transfer the contents of the step address counter to the base address register (address of the next command)
MBA		N	N	N	0.1		Transfer the contents of the base address register to ACCU 1 (bits 31 to 20 are set to zero)
MBS		N	N	N	0.25		Transfer the contents of the base address register to the step address counter
Arithmetic Operations							
ADD	BF - 128 to+127	N	N	N	0.1		Add byte constant (fixed point) to ACCU 1-L.
ADD	KF - 32768 to +32767	N	N	N	0.25		Add fixed-point constant (word) to ACCU 1-L.
ADD	DH 0 to FFFF FFFF	N	N	N	0.35		Add constant (hexadecimal code as double word) to ACCU 1.

System Operations

for organization blocks (OB) for function blocks (FB)

for program blocks (PB) for sequence blocks (SB)

Oper-ation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µs		Function
		1	2	3	CPU 945		
Other Operations							
STS		N	N	N	44		Stop operation. Program processing is interrupted immediately after this operation
STP		N	N	N	32		Stop at the end of program execution: current program execution is brought to an end: the PIQ is output; then the programmable controller enters the STOP status
TAK		N	N	N	0.1		Swap the contents of ACCU 1 and ACCU 2

Machine Code Listing

Explanation of indices

a	+ byte address
b	+ bit address
c	+ parameter address
d	+ timer number
e	+ constant
f	+ block number
g	+ word address
h	+ number of shifts
i	+ relative jump address
k	+ register address
l	+ block length
m	+ jump displacement (16 bits)
n	+ value
o	+ counter number

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	
0	A	0_a	0_a					L	FY
0	B	0_a	0_a					T	FY
0	C	0_d	0_d					LD	T
0	D	0_i	0_i					JO=	

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
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_n	0_n					I	
1	2	0_a	0_a					L	FW
1	3	0_a	0_a					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	9	0_n	0_n					D	
1	A	0_a	0_a					L	FD
1	B	0_a	0_a					T	FD
1	C	0_d	0_d					SE	T
1	D	0_f	0_f					JC	FB
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	

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
2	1	6	0					><F	
2	1	8	0					!=F	
2	1	A	0					>=F	
2	1	C	0					<=F	
2	2	0_g	0_g					L	DL
2	3	0_g	0_g					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_g	0_g					L	DR
2	B	0_g	0_g					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
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

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
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_g	0_g					L	DW
3	3	0_g	0_g					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	
3	9	C	0					<=D	
3	A	0_g	0_g					L	DD
3	B	0_g	0_g					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=	

[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		
4	0	0	0_k					LIR	
4	1	0	0					AW	
4	2	0_o	0_o					L	C
4	3	0_l	0_l					TNW	
4	4	0_o	0_o					FR	C
4	5	0_i	0_i					JZ=	
4	6	0_c	0_c					L=	
4	8	0	0_k					TIR	
4	9	0	0					OW	
4	A	0_a	0_a					L	IB
4	A	8_a	0_a					L	QB
4	B	0_a	0_a					T	IB
4	B	8_a	0_a					T	QB
4	C	0_o	0_o					LD	C
4	D	0_f	0_f					JC	OB
4	E	0_g	0_g					DO	FW
4	F	0_g	0_g					L	RT
5	0	0_e	0_e					ADD	BN
5	1	0	0					XOW	
5	2	0_a	0_a					L	IW
5	2	8_a	0_a					L	QW
5	3	0_a	0_a					T	IW
5	3	8_a	0_a					T	QW
5	4	0_o	0_o					CD	C
5	5	0_f	0_f					JC	PB

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
5	6	0_c	0_c					LDW=	
5	7	0_a	0_a					L	OW
5	8	0	0	0_e	0_e	0_e	0_e	ADD	KF
5	9	0	0					-F	
5	A	0_a	0_a					L	ID
5	A	8_a	0_a					L	QD
5	B	0_a	0_a					T	ID
5	B	8_a	0_a					T	QD
5	C	0_o	0_o					S	C
5	D	0_f	0_f					JC	SB
5	F	0_a	0_a					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	
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_g	0_g					L	RS
6	3	0_g	0_g					T	RS
6	4	0_h	0_h					RLD	

[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		
6	5	0	0					BE	
6	5	0	1					BEU	
6	6	0_c	0_c					T=	
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
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	

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
6	8	8	9					MBA	
6	8	9	9					MBS	
6	8	A	B					LDI	BR
6	8	A	F					TDI	BR
6	9	0	0_h					SRW	
6	C	0_o	0_o					CU	C
6	D	0_f	0_f					JU	OB
6	E	0_g	0_g					DO	DW
6	F	0_g	0_g					T	RT
7	0	0	0					STS	
7	0	0	2					TAK	
7	0	0	3					STP	
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	BT
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	1	5	0	0_b	0_o	0_o	RU	C
7	0	1	5	4	0_b	0_o	0_o	SU	C
7	0	1	5	8	0_b	0_o	0_o	TBN	C
7	0	1	5	C	0_b	0_o	0_o	TB	C
7	0	2	5	0	0_b	0_d	0_d	RU	T
7	0	2	5	4	0_b	0_d	0_d	SU	T

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
7	0	2	5	8	0_b	0_d	0_d	TBN	T
7	0	2	5	C	0_b	0_d	0_d	TB	T
7	0	3	8	0	0_b	0_a	0_a	RU	I
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	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_a	0_a					L	PY
7	3	0_a	0_a					T	PY

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
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_a	0_a					T	QW
7	8	0	1	0	1	0_f	0_f	DOU	FX
7	8	0	2	0	9	0_f	0_f	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_e	9	0_e	0_e	0_e	0_e	MBR	
7	8	0	A	0_e	0_e	0_e	0_e	ABR	
7	8	0	B	0_b	0_a	0_a	0_a	A	S
7	8	1	B	0_b	0_a	0_a	0_a	O	S
7	8	2	B	0_b	0_a	0_a	0_a	S	S
7	8	3	B	0_b	0_a	0_a	0_a	=	S
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	6	B	0_b	0_a	0_a	0_a	R	S

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
7	8	A	B	0	0_a	0_a	0_a	L	SY
7	8	B	B	0	0_a	0_a	0_a	T	SY
7	8	C	B	0	0_a	0_a	0_a	L	SW
7	8	D	B	0	0_a	0_a	0_a	T	SW
7	8	E	B	0	0_a	0_a	0_a	L	SD
7	8	F	B	0	0_a	0_a	0_a	T	SD
7	9	0	0					+F	
7	A	0_a	0_a					L	PW
7	B	0_a	0_a					T	PW
7	C	0_o	0_o					R	C
7	D	0_f	0_f					JU	SB
7	F	0_a	0_a					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_o	0_o					A	C
B	9	0_o	0_o					O	C
B	A	0	0					A(	
B	B	0	0					O(	
B	C	0_o	0_o					AN	C
B	D	0_o	0_o					ON	C

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
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
F	D	0_d	0_d					ON	T
F	F	F	F					NOP1	

Alphabetical Index of Operations

Operation	Page	Operation	Page
! = D	24, **71**	**> G**	26, **71**
! G	26, **71**	**>< D**	24, **71**
!=F	24, **70**	**><F**	24, **70**
)	8,**79**	**><G**	26, **71**
+ D	28, **73**	**>=F**	24, **70**
+ G	30, **73**	**>F**	24, **69**
+F	28, **78**	**A**	8, **77-79**
- D	30, **73**	**A(**	8, **78**
- F	28, **73**	**A=**	38, **68**
- G	30, **73**	**ABR**	62, **77**
:F	28, **73**	**ADD**	64, **72, 73**
:G	30, **73**	**AN**	8, **77-79**
< = D	26, **71**	**AN=**	38, **70**
< D	26, **71**	**AW**	38, **72**
< =F	24, **70**	**BE**	34, **74**
< F	24, **69**	**BEC**	34, **68**
<=G	28, **71**	**BEU**	34, **74**
<G	28, **71**	**BLD 0 to 255**	34, **69**
=	10, **77-79**	**BLD 130**	34, **69**
==	42, **69**	**BLD 131**	36, **69**
> = D	26, **71**	**BLD 132**	36, **69**
> = G	26, **71**	**BLD 133**	36, **69**
> D	26, **71**	**BLD 255**	36, **69**

Operation	Page	Operation	Page
CD	22, **72**	**JM=**	52, **70**
CFW	46, **68**	**JN=**	52, **71**
CSD	46, **74**	**JO=**	52, **68**
CSW	46, **68**	**JOS =**	52, **73**
CU	22, **75**	**JP=**	52, **69**
CX	32, **77**	**JU**	30, 32, **71, 75-78**
D	54, **69**	**JU=**	50, **70**
DED	48, **74**	**JUR**	52, **75**
DEF	46, **74**	**JZ=**	52, **72**
DO	56, **72, 75**	**L**	12-16, 60, **68-78**
DO=	56, **77**	**L=**	46, **72**
DOC	32, **77**	**LD**	16, **68, 72**
DOU	30, **77**	**LD=**	46, **69**
DUD	48, **74**	**LDI**	58, **74, 75**
DUF	48, **74**	**LDW =**	46, **73**
FDG	48, **74**	**LIM**	54, **75**
FR	44, **68, 72**	**LIR**	58, **72**
FR=	44, **68**	**LRD**	62, **74**
GFD	48, **74**	**LRW**	62, **74**
GX	32, **77**	**LW=**	46, **71**
I	32, **77**	**MAB**	62, **74**
I	54, **69**	**MAS**	62, **74**
IA	54, **68**	**MBA**	64, **75**
JC	32, **69, 72, 73**	**MBR**	62, **77**
JC=	50, **79**	**MBS**	64, **75**

Operation	Page	Operation	Page
MSA	64, **74**	**SF**	22, **69**
MSB	64, **74**	**SFD=**	44, **69**
NOP 0	34, **68**	**SIM**	54, **75**
NOP 1	34, **79**	**SLD**	48, **70**
O	8, **77-79**	**SLW**	48, **73**
O(	8, **78**	**SP**	22, **71**
O=	38, **69**	**SP=**	44, **71**
ON	8, **77-79**	**SRW**	48, **75**
ON=	38, **70**	**SS**	22, **70**
OW	38, **72**	**SSD**	50, **76**
Q	32, **69**	**SSU=**	44, **70**
R	8, 22, **71, 77-79**	**SSW**	50, **74**
RA	54, **68**	**STP**	34, 66, **75**
RB=	42, **71**	**STS**	66, **75**
RD=	44, **71**	**SU**	42, 58, **75, 76**
RLD	50, **73**	**T**	16-20, **68-78**
RRD	50, **77**	**T=**	46, **74**
RU	42, 58, **75, 76**	**TAK**	66, **75**
S	8, 22, **73, 77-79**	**TB**	40, **75, 76**
S=	42, **69**	**TBN**	40, **75, 76**
SD	22, **70**	**TDI**	60, **74, 75**
SD=	44, **70**	**TIR**	58, **72**
SE	22, **69**	**TNB**	60, **68**
SEC=	44, **69**	**TNW**	60, **72**

Operation	Page	Operation	Page
TRD	62, **74**	**×F**	28, **73**
TRW	62, **74**	**×G**	30, **73**
XOW	38, **72**		

List of Organization Blocks

You must program the OB.
The operating system calls up the OB.

OB No.	Function
OB1	Cyclic program execution
Interrupt-, timed-interrupt- and time-driven program exec.	
OB2	Interrupt A: Digital input module -434-7, digital I/O module 485-7 and IP generate interrupt
OB3	Interrupt B: IP generates interrupt
OB4	Interrupt C: IP generates interrupt
OB5	Interrupt D: IP generates interrupt
OB6	Interrupt generated by internal timers
OB10	Time-controlled program execution (variable in each case: 1 ms to 65535 ms)
OB11	
OB12	
OB13	

You must program the OB.
The operating system calls up the OB.

OB No.	**Function**
Controlling restart characteristics	
OB21	Manuel switch on
OB22	Automatic switch on when power is restored
Handling of system errors	
OB26	Scan time exceeded
OB33	Time interrupt error
OB35	I/O error
Handling programming errors and PLC faults	
OB19	When a block is called which has not been loaded
OB23	Time-out during individual access to the S5 bus (e.g. L PW, L PY, T PW, T PY, LIR, TIR)
OB24	Time-out during update of process image and interproc. communication flags
OB27	Substitution error
OB32	Transfer error
OB34	Battery failure

The OB is already programmed.
You must call up the OB

OB No.	Function		Execution time (in µs)
OB31	Scan time triggering		9.1
OB125*	Generate STEP 5 blocks		41
OB160	Programmable time loop		3 to 65535
OB182*	Copy data area		35+n ×2.65 (n: number of data words)
OB183*	Duplicate DX		52+n×2.35 (n: number of data words in DX)
OB184*	Duplicate DB		52+n×2.35 (n: numbe of data words in DB)
OB190*	Transfer flag into DB byte by byte		13+n×1.85 (n: number of bytes)
OB191*	Transfer data from DB to flag byte by byte		13+n×1.85 (n: number of bytes)
OB192*	Transfer flag into DB word by word		12+n×1.3 (n: number of bytes)
OB193*	Transfer data from DB to flag word by word		12+n×1.3 (n: number of bytes)
OB220*	Sign expansion		3
OB251*	PID control algorithm		max. 110
OB254	Read in process I/O image		65+n×(1.5+Ready delay time of the module)
OB255	Output process I/O image		65+n×(1.5+Ready delay time of the module)

* From software version Z 02

((Seite 86a))

((Seite 87a))

The OB is already programmed.
You must call up the OB

OB 250, Operating system services

Service No.	Function		Execution time (typical)
1	Activation of OB6		21 µs
2 to 5	New interval for OB10 to OB13		20 µs
6	Modification of entries in BS128 to 143		33 µs
7	Reduction of PIQ transfer		18 µs
8	Generation of list of all I/O bytes addressable		1.7 ms to 86 ms [t=1700 µs+n×(8 µs+Ready delay time)+ m×(20 µs+Ready delay time)+(512-m-n)×165 µs] m=existing analog I/O bytes n=existing digital I/O bytes
10/11	Creation of a DB/DX without TRAF		30 µs to 45 µs
12	New generation of the block address list		50 µs per existing, valid code block 43 µs per existing, valid DB/DX 26 µs per existing invalid code block 25 µs per existing, invalid DB/DX
13	Read address (byte) from S5 bus		33 µs+Ready delay time (on TO: 193 µs)
14	Write address (byte) on S5 bus		35 µs+Ready delay time (on TO: 195 µs)
15	Read address (word) from S5 bus		34 µs+Ready delay time (on TO: 194 µs)
16	Write address (word) on S5 bus		36 µs+2×Ready delay time (on TO: 196 µs)
17/18	Read byte from page frame/ Write byte from page frame		43 µs+Ready delay time (on TO: 200 µs)
19	Set and reset command output inhibit		24 µs

((Seite 88a))

((Seite 89a))

OB 250, Operating system services (cont.)

Service No.	Function		Execution Time (typical)
20	Read DBA register		25 µs
21	Write DBA register		26 µs
22	Read DBL register		25 µs
23	Write DBL register		25 µs
24	Indexed accessing of DX		29 µs
25	Indexed calling fo FX		30 µs
26	Remove block from block address list		28 µs
27	Change block identifier to ”valid in EPROM”		4.8 ms+n×19 µs (n: number of blocks to be changed)
28	Change block identifier to ”valid in RAM”		6 ms+n×19 µs (n: number of blocks to be changed)

((Seite 90a))

((Seite 91a))

List of Function Blocks

Integral Function Blocks

FB No.	Function	Execution Time (typical)		Execution Time (Basis for calculation)
FB238	Compress PLC memory	20 µs*		
FB239	Delete block	19 µs		
FB241	Read analog value from 463 analog input module	60 µs		t 45µs+2×Ready delay time
FB242	Read analog value from 464 analog input module	60 to 115 µs		t 50µs+2×Ready delay time
FB243	Read analog value from 466 analog input module	120 µs		t 45µs+2×Ready delay time
FB244	Send data	2.9ms**		• Idle run VKE=0: t 54µs+4×Ready delay time • with data transmission: t 1070µs+30×Ready delay time+n×(2.5µs+ Ready delay time) with n=minimum frame length, block size (in bytes)
FB245	Receive data	4.4ms**		• Idle run VKE=0: t 54µs+4×Ready delay time • with data acceptance: t 1280µs+20×Ready delay time+n×(5µs+Ready delay time); with n=minimum frame length, block size (in bytes)
FB246	Fetch data	0.9ms**		• Idle run VKE=0: t 54µs+4×Ready delay time • with parameter transmission: t 820µs+44×NAK
FB247	Monitor job processing	58 µs***		t 54µs+4×Ready delay time
FB248	Delete job	1.2ms		• Idle run t 30µs+3×NAK • active: t 1200µs+65×NAK

* only triggering of compress function
** for ready delay time=1µs and block size 6 (512 bytes)
*** for ready delay time=1µs

Integral Function Blocks (cont.)

FB No.	Function	Execution Time (typical)		Execution Time (Basis for calculation)
FB249	Initialize interface	<10s		
FB250	Read analog value from 460/465 analog input module	40 to 100 μs		t 54μs+2×Ready delay time (for cyclic sampling) t 40 to 100μs (for individual sampling)
FB251	Output analog value	60 μs		t 45μs+2×Ready delay time

Data Block 1

Data Block 1

Parameter	Argument	Permissible Range		Meaning
Block Identifier: SL1:				**SINEC L1**
SLN	p	p=0, 1 to 30, 255		"***SL***ave ***N***umber" "0" = Master function with point-to-point link (at 2nd interface); These parameters apply for the 2nd interface if no computer link or ASCII driver has been activated.
SM	DBxDWy or	x=0 to 255		Position of the ***S***end ***M***ailbox (start of SF)
RM	DXaDWy or	y=0 to 255		Position of the ***R***eceive ***M***ailbox (start of EF)
CBR	FBy	a=0 to 255		Position of the ***C***oordination ***B***yte ***R***eceive
CBS	SYz	z=0 to 4095		Position of the ***C***oordination ***B***yte ***S***end
PGN	p	p=1 to 30		***PG*** bus ***N***umber Note: CBS and CBR are in a flag byte or in the high-order byte of the specified data word (DL)!
Block Identifier: SDP:				**System-Dependent Parameters**
WD	p	p=0 to 2550		"***W***atch-***D***og" (scan time monitoring) can be set in milliseconds but only in steps of 10 ms
RDLY	r	r=0 to 65535		"***R***un ***D***e***L***a***Y***" restart delay after POWER ON in milliseconds
RT	Y/N	-		"***R***esident ***T***imers"; if "Y" all timers are retentive, if "N" only T 0 to 63*.
RC	Y/N	-		"***R***esident ***C***ounters"; if "Y" all counters are retentive, if "N" only C 0 to 63*.
RF	Y/N	-		"***R***esident ***F***lags"; if "Y" all flags, S flags are retentive, if "N" only the first half are retentive*.
PROT	Y/N	-		"***PROT***ection" activate software protection? (input/output of program no longer possible)
PIO	Y/N	-		"***P***rocess ***I***mage ***O***utput" disable output of process image?

* Additionally, set switch for Default/Overall Reset on the control panel of the CPU to "RE"

Data Block DB1 (cont.)

Parameter	Argument	Permissible Range		Meaning
Block Identifier: SDP: (cont.)				**System-Dependent Parameters (cont.)**
PII	Y/N	-		"***P***rocess ***I***mage ***I***nput" disable read in of process image?
RPIC	s	s=0 to 255		"***R***educed ***P***rocess ***I***mage Output ***C***ounter" counter for cycles with reduced PIQ transfer
PPIT	Y/N	-		"***P***arallel ***P***rocess ***I***mage ***T***ransfer"; if "N", the sequential PI transfer and if "Y" the parallel PI transfer is set
Block Identifier: TFB:				**Timer-Function Block**
OB10 OB11 OB12 OB13	p	p=0 to 65535		Interval during which OB 10 to 13 is called and processed Time in milliseconds
Block Identifier: ASC:				**ASCII Driver at 2nd Interface**
				Note: Driver number "1" for the 2nd interface is set automatically.
SM	DBxDWy or	x=0 to 255		Position of the ***S***end ***M***ailbox (start of SF)
RM	DXaDWy or	y=0 to 255		Position of the ***R***eceive ***M***ailbox (start of EF)
CBR	FBy or	a=0 to 255		Position of the ***C***oordination ***B***yte ***R***eceive
CBS	SYz	z=0 to 4095		Position of the ***C***oordination ***B***yte ***S***end
PAR				Position of the ***PAR***ameter set
MOD	n	n=1 to 8		"***MOD***e number" mode number of ASCII driver
BDR	m	m= 150, 200, 300, 600, 1200, 2400, 4800, 9600, 19200		"***B***au***D R***ate" baud rate

Data Block DB1 (cont.)

Parameter	Argument	Permissible Range		Meaning
Block Identifier: ASC: (cont.)				**ASCII Driver at 2nd Interface (cont.)**
PRTY	p	p=		"***P***a***R***i***TY***" Parity
		E		***E***ven
		O		***O***dd
		M		***M***ark ("1")
		S		***S***pace ("0")
		N		***N***one =No check
DF	q	q=0 to 5, 7, 8		"***D***ata ***F***ormat"
WCR	r	r=0 to 2550		"***W***ait time after ***C***arriage ***R***eturn" Wait time after CR in milliseconds
WLF	r	r=0 to 2550		"***W***ait time after ***L***ine ***F***eed" Wait time after LF in milliseconds
WFF	r	r=0 to 2550		"***W***ait time after ***F***orm ***F***eed" Wait time after FF in milliseconds
DT	s	s=10 to 655350		"***D***elay ***T***ime" character delay time in milliseconds
ML	t	t=0 to 1024		"***M***ail ***L***ength" frame length in bytes for receive frame; for mode 1, 8
ME	u v	u, v=0 to 255		"***M***ail ***E***nd" end identifier; for mode 2, 3, 4, 5, 6 RUB OUT = $7F_H$ = 127_D XON = 11_H = 17_D XOFF = 13_H = 19_D CR = $0D_H$ = 13_D LF = $0A_H$ = 10_D FF = $0C_H$ = 12_D EOT = 04_H = 4_D ETX = 03_H = 3_D
SLF	Y/N	-		"***S***kip ***LF***"
LPP	w	w=1 to 255		"***L***ines ***P***er ***P***age"
LM	w	w=0 to 255		"***L***eft ***M***argin"
PN	o/u	-		"***P***age ***N***umber"; o=top u=bottom

Data Block DB1 (cont.)

Parameter	Argument	Permissible Range		Meaning
Block Identifier: ASC: (cont.)				**ASCII Driver at 2nd Interface (cont.)**
HD1 HD2 FT1 FT2	"<string>" "<string>" "<string>" "<string>"	max. 119 alpha-numeric characters		"***H***ea***D***er 1" 1* "***H***ea***D***er 2" 2* "***F***oo***T***er 1" 1* "***F***oo***T***er 2" 2* Print control characters are entered as hexadecimal characters as follows: Example: "TEXT $1B$38 FURTHER TEXT" (1B38=double-width print ON)
Block Identifier: RKT:				**Computer Link at 2nd Interface**
				Note: Driver number "2" for the 2nd interface is set automatically.
SM	DBxDWy or	x=0 to 255		Position of the ***S***end ***M***ailbox (start of SF)
RM	DXaDWy or	N=0 to 255		Position of the ***R***eceive ***M***ailbox (start of EF)
CBS	FBy or	a=0 to 255		Position of the ***C***oordination ***B***yte ***R***eceive
CBR	SYz	z=0 to 4095		Position of the ***C***oordination ***B***yte ***S***end
PAR				Position of the ***PAR***ameter set
MOD	n	n=1, 2		"***MOD***e number" mode number 1=No block check character 2=Sending with block check character
BDR	m	m= 150, 200, 300, 600, 1200, 2400, 4800, 9600, 19200		"***B***au***D*** ***R***ate" baud rate

***** - CR (Carriage Return) is automatically generated in the parameter DB at the end of the ASCII string (separation of lines). If HD 1/2, FT 1/2 is missing, then only CR is stored (Section 12.5)

- For sending of the $ character $$ must be entered.
- For sending of the "character $" must be entered.

Data Block DB1 (cont.)

Parameter	Argument	Permissible Range		Meaning
Block Identifier: RKT: (cont.)				**Computer Link at 2nd Interface (cont.)**
PRTY	p	p=		"***P***a***R***i***TY***"
		E		***E***ven
		O		***O***dd
		M		***M***ark ("1")
		S		***S***pace ("0")
		N		***N***one = No check
DF	q	q=0 to 5, 7, 8		"***D***ata ***F***ormat"
PRI	l/h			"***PRI***ority"
				l=low
				h=high
DT*	s	s=10 to 655350		"***D***elay ***T***ime" character delay time in milliseconds
TIO*	s	s=10 to 655350		"***TI***me ***O***ut" time out delay in milliseconds
BWT*	s	s=10 to 655350		"***B***lock ***W***ait ***T***ime" in milliseconds
TTE	r	r =1 to 255		"***T***ries ***T***o ***E***rect"
TTS	r	r =1 to 255		"***T***ries ***T***o ***S***end"
Block Identifier: S2T:				**Deactivate Driver**
DEAC	Y/N	-		Deactivation of ASCII driver or computer link. If DEAC="Y" driver number "0" is entered (in BS46) and previously defined parameters for RKT and ASC are not accepted.

* DT<TIO<BWT

Data Block DB1 (cont.)

Parameter	Argument	Permissible Range		Meaning
Block Identifier: CLP:				**Clock Parameters**
CLK	DBxDWy or	x=0 to 255		"***CL***oc***K*** Data" start of clock data area
	DXxDWy or	N=0 to 255		
	MWb or	b=0 to 254		
STW	SWz	z=0 to 4094		"***ST***atus ***W***ord" position of the status word
SET				Set clock time, date
	wd dd.mm.yy	wd	=1 to 7	Week day=So to Sa
	hh:mm:ss	dd	=01 to 31	Day
	AM/PM[1]	mm	=01 to 12	Month
		yy	=00 to 99	Year
		hh	=1 to 12	AM/PM
			00 to 23	24-hour mode
		ss	=00 to 59	Seconds
TIS	wd dd.mm.			"***T***imer ***I***nterrupt ***S***et"
	hh:mm:ss			
	AM/PM[1]			
OHS	hhhhhh:mm:ss[2]			"***O***peration ***H***our counter ***S***et"
		hhhhhh	=000000 to 999999	Hours
		mm	=00 to 59	Minutes
		ss	=00 to 59	Seconds

[1] If an argument (e.g. week day) is not to be transferred, enter XX! The clock will then continue with the current value. If you specify AM or PM after the clock time, the clock will operate in the relevant 12-hour mode. If you omit this argument, the clock will operate in 24-hour mode.

[2] If an argument (e.g. minute) is not to be transferred, enter XX! The clock will then continue with the current value.

Data Block DB1 (cont.)

Parameter	Argument	Permissible Range		Meaning
Block Identifier: CLP: (cont.)				**Clock Parameters (cont.)**
OHE	Y/N	-		"***O***peration ***H***our counter ***E***nable"
STP	Y/N	-		"***ST***o***P***" Update clock in STOP state
SAV	Y/N	-		"***SAV***e" Save clock time after last RUN STOP or Power OFF
CF	p	-400 to+400		"***C***orrection ***F***actor" Enter correction factor
Block Identifier: ERT:				**Error Return**
ERR	DBxDWy or DXaDWy or MBz or SYp	x=0, 2 to 255 N=0 to 255 a=0 to 255 z=0 to 236 p=0 to 4076		"***ERR***ors" Position of the error code

Evaluation of CC 1 and CC 0

CC 1	CC 0	Arith-metic Opera-tions	Digital Logic Opera-tions	Com-parison Operations	Shift Opera-tions	Con-version Opera-tions	Jump Opera-tions executed
0	0	Result =0	Result =0	ACCU 2 = ACCU 1	shifted Bit =0	-	IZ
0	1	Result <0	-	ACCU 2 < ACCU 1	-	Result <0	IM IN
1	0	Result >0	Result 0	ACCU 2 > ACCU 1	shifted Bit =0	Result >0	IP IN

Siemens AG
AUT 125 Doku
Postfach 1963

D-92209 Amberg
Fed. Republic of
Germany

Should you come across any printing errors when reading this publication, we would ask you to inform us using this form. We would also welcome any suggestions you may have for improvement.

From *(please fill in)*
Name
Company/Department
Address

Telephone

Publication:
Programmable Controller
SIMATIC S5-115U (CPU 945)
Pocket Guide

Order No.: 6ES5 997-7LB21

Suggestions/Corrections: