SIMATIC S5

S5-115U
Programmable Controller
CPU 941/942/943/944

Reference Guide

Order No.
6ES5 997-7LA21

Index

Explanation of the Operations List

Abbreviation	Explanation
ACCUM 1	Accumulator 1 (When accumulator 1 is loaded, any existing contents are shifted into accumulator 2.)
ACCUM 2	Accumulator 2
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 represent.
OV	Overflow. This condition code bit is set if, e.g., a numerical 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.
RLO dependent? Y 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 represent.

Explanation of the Operands

Abb.	Explanation	Permissible operand value range for CPU 941 942 943 944	
BN	Byte constant (fixed-point no.)	- 128 to+127	
C	Counter - for the bit test and set operations (system operations)	0 to 127 0.0 to 127.15	
D	Data word (1 bit) - for load operations (supplementary operations) and transfer operations (system operations) - for bit test and set operations (system operations)	0.0 to 255.15	
DB	Data block	2 to 255	
DL	Data word (left byte)	0 to 255	
DR	Data word (right byte)	0 to 255	
DW	Data word	0 to 255	
F	Flag	0.0 to 255.7	
FB	Function block	0 to 255	
FW	Flag word	0 to 254	
FY	Flag byte	0 to 255	
I	Input	0.0 to 63.7	0.0 to 127.7

Abb.	Explanation	Permissible operand value range for CPU 941 942	943 944
IB	Input byte	0 to 63	0 to 127
IW	Input word	0 to 62	0 to 126
KB	Constant (1 byte)	0 to 255	
KC	Constant (count)	0 to 999	
KF	Constant (fixed-point number)	- 32768 to +32767	
KH	Constant (hexadecimal code)	0 to FFFF	
KM	Constant (2-byte bit pattern)	arbitrary bit pattern (16 bit)	
KS	Constant (2 characters)	any two alphanumeric characters	
KT	Constant (time)	0.0 to 999.3	
KY	Constant (2 bytes)	0 to 255 (per byte)	
OB [1]	Organization block	0 to 255	
PB	Program block (with block call and return operations)	0 to 255	
PB/ PY [2]	Peripheral byte - Digital inputs - Analog inputs - Digital outputs - Analog outputs	 0 to 63 128 to 255 0 to 63 128 to 255	 0 to 127 128 to 255 0 to 127 128 to 255

[1] See page 52 for an overview of the organization blocks and their function

[2] PY in the case of S5-DOS programmers

Abb.	Explanation	Permissible operand value range for CPU 941 942 943 944	
PW	Peripheral word - Digital inputs - Analog inputs - Digital outputs - Analog outputs	0 to 63 128 to 254 0 to 63 128 to 254	0 to 126 128 to 254 0 to 126 128 to 254
Q	Output	0.0 to 63.7	0.0 to 127.7
QB	Output byte	0 to 63	0 to 127
QW	Output word	0 to 62	0 to 126
RS	System data range - for load operations (supplementary operations) and transfer operations (system operations) - for bit test and set operations (system operations)	0 to 255 0.0 to 255.15	
SB	Sequence block	0 to 255	
T	Timer - for the bit test and set operations (system operations)	0 to 127 0.0 to 127.15	

Note regarding execution times

Please note that, on account of the processor architecture, the execution times quoted for the following list of operations should be treated as approximate values. Depending on the type of CPU installed, the operations are executed in the standard processor **or** in the STEP 5 coprocessor.
In the case of a switch from direct execution in the coprocessor to execution in the standard processor, the pure processing time is incremented by the time required for switching. These switchover times are included in the specified execution times.

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 µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
Boolean Logic Operations										
A	I, Q, F, T, C	N	Y	N	1,6	1,6	0,8		0,8	Scan operand for ”1” and combine with RLO through logic AND.
AN	I, Q, F, T, C	N	Y	N	1,6	1,6	0,8		0,8	Scan operand for ”0” and combine with RLO through logic AND.
O	I, Q, F, T, C	N	Y	N	1,6	1,6	0,8		0,8	Scan operand for ”1” and combine with RLO through logic OR.
ON	I, Q, F, T, C	N	Y	N	1,6	1,6	0,8		0,8	Scan operand for ”0” and combine with RLO through logic OR.
O		N	Y	Y	1,6	1,6	0,8		0,8	Combine AND operations through logic OR.
A(		N	Y	Y	1,6	1,6	0,8		0,8	Combine expressions enclosed in parentheses through logic AND (6 levels).
O(		N	Y	Y	1,6	1,6	0,8		0,8	Combine expressions enclosed in parentheses through logic OR (6 levels).
)		N	Y	N	1,6	1,6	0,8		0,8	Close parenthesis (conclusion of a parenthetical expression).
Set/Reset Operations										
S	I, Q, F	Y	N	Y	1,6	1,6	0,8		0,8	Set operand to ”1”.
R	I, Q, F	Y	N	Y	1,6	1,6	0,8		0,8	Reset operand to ”0”.
=	I, Q, F	Y	N	Y	1,6	1,6	0,8		0,8	Assign value of RLO to operand.

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 µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
Load Operations										
L	IB	N	N	N	1,6	1,6	0,8		0,8	Load an input byte from the PII into ACCUM 1.
L	QB	N	N	N	1,6	1,6	0,8		0,8	Load an output byte from the PIQ into ACCUM 1.
L	IW	N	N	N	1,6	1,6	0,8		0,8	Load input word from the PII into ACCUM 1: byte n ACCUM 1 (bits 8-15); byte n+1 ACCUM 1 (bits 0-7).
L	QW	N	N	N	1,6	1,6	0,8		0,8	Load an output word from the PIQ into ACCUM 1: byte n ACCUM 1 (bits 8 - 15); byte n+1 ACCUM 1 (bits 0 - 7).
L	PB/PY[1]	N	N	N	93*	93*	93*		4*	Load an input byte from the digital/analog input modules into ACCUM 1.
L	PW	N	N	N	107*	107*	107*		4,8**	Load a peripheral word from the digital/analog inputs into ACCUM 1: byte n ACCUM 1 (bits 8 - 15); byte n+1 ACCUM 1 (bits 0 - 7).
L	FY	N	N	N	1,6	1,6	0,8		0,8	Load a flag byte into ACCUM 1.
L	FW	N	N	N	1,6	1,6	0,8		0,8	Load a flag word into ACCUM 1: byte n ACCUM 1 (bits 8 - 15); byte n+1 ACCUM 1 (bits 0 - 7).
L	DL	N	N	N	3,4	3,4	1,7		1,7	Load a data word (left-hand byte) of the current data block into ACCUM 1.

1 PY in the case of S5-DOS programmers

* + ready delay time of the referenced I/O modules (digital I/O: 2 µs/byte, analog I/O: 16 µs/byte)

** + 2 x ready delay time of the referenced I/O modules

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 µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
Load Operations (cont.)										
L	DR	N	N	N	3,4	3,4	1,7		1,7	Load a data word (right-hand byte) of the current data block into ACCUM 1.
L	DW	N	N	N	3,9	3,9	2		2	Load a data word of the current data block into ACCUM 1: byte n ACCUM 1 (bits 8 - 15); byte n+1 ACCUM 1 (bits 0 - 7).
L	KB	N	N	N	2,8	2,8	1,4		1,4	Load a constant (1-byte number) into ACCUM 1.
L	KS	N	N	N	1,6	1,6	0,8		0,8	Load a constant (2 characters in ASCII format) into ACCUM 1.
L	KF	N	N	N	1,6	1,6	0,8		0,8	Load a constant (fixed-point number) into ACCUM 1.
L	KH	N	N	N	1,6	1,6	0,8		0,8	Load a constant (hexadecimal code) into ACCUM 1.
L	KM	N	N	N	1,6	1,6	0,8		0,8	Load a constant (bit pattern) into ACCUM 1.
L	KY	N	N	N	1,6	1,6	0,8		0,8	Load a constant (bit pattern) into ACCUM 1.
L	KT	N	N	N	1,6	1,6	0,8		0,8	Load a constant (count in BCD) into ACCUM 1.
L	KC	N	N	N	1,6	1,6	0,8		0,8	Load a constant (count in BCD) into ACCUM 1.
L	T, C	N	N	N	1,6	1,6	0,8		0,8	Load a time or count (in binary code) into ACCUM 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 µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
Load Operations (cont.)										
LC	T, C	N	N	N	3,5	3,5	1,8		1,8	Load times or counts (in BCD) into ACCUM 1.
Transfer Operations										
T	IB	N	N	N	1,6	1,6	0,8		0,8	Transfer the contents of ACCUM 1 to an input byte (into the PII).
T	QB	N	N	N	1,6	1,6	0,8		0,8	Transfer the contents of ACCUM 1 to an output byte (into the PIQ).
T	IW	N	N	N	1,6	1,6	0,8		0,8	Transfer the contents of ACCUM 1 to an input word (into the PII): ACCUM 1 (bits 8 - 15) byte n; ACCUM 1 (bits 0 - 7) byte n+1.
T	QW	N	N	N	1,6	1,6	0,8		0,8	Transfer the contents of ACCUM 1 to an output word (into the PIQ): ACCUM 1 (bits 8 - 15) byte n; ACCUM 1 (bits 0 - 7) byte n+1.
T	PB/PY[1]	N	N	N	67*	67*	67*		3,9*	Transfer the contents of ACCUM 1 to an I/O byte of the digital output modules with updating of the PIQ or analog output modules.
T	PW	N	N	N	85*	85*	85*		4,7**	Transfer the contents of ACCUM 1 to an I/O byte of the digital output modules with updating of the PIQ or the analog output modules.

1 PY in the case of S5-DOS programmers

* + ready delay time of the referenced I/O modules (digital I/O: 2 µs/byte, analog I/O: 16 µs/byte)

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

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 µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
Transfer Operations (cont.)										
T	FY	N	N	N	1,6	1,6	0,8		0,8	Transfer the contents of ACCUM 1 to a flag byte.
T	FW	N	N	N	1,6	1,6	0,8		0,8	Transfer the contents of ACCUM 1 to a flag word (into the PIQ): ACCUM 1 (bits 8 - 15) byte n; ACCUM 1 (bits 0 - 7) byte n+1.
T	DL	N	N	N	2,2	2,2	1,1		1,1	Transfer the contents of ACCUM 1 to a data word (left-hand byte).
T	DR	N	N	N	2,2	2,2	1,1		1,1	Transfer the contents of ACCUM 1 to a data word (right-hand byte).
T	DW	N	N	N	2,7	2,7	1,4		1,4	Transfer the contents of ACCUM 1 to a data word.
Timer Operations										
SP	T	Y	N	Y	3,7	3,7	1,9		1,9	Start a timer (stored in ACCUM 1) as signal-contracting pulse on the leading edge of the RLO.
SE	T	Y	N	Y	3,7	3,7	1,9		1,9	Start a timer (stored in ACCUM 1) as extended pulse (signal contracting and stretching) on the leading edge of the RLO.
SR	T	Y	N	Y	3,7	3,7	1,9		1,9	Start an on-delay timer (stored in ACCUM 1) on the leading edge of the RLO.
SS	T	Y	N	Y	3,7	3,7	1,9		1,9	Start a stored on-delay timer (stored in ACCUM 1) on the leading edge of the RLO.

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 µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
Timer Operations (cont.)										
SA	T	Y	N	Y	3,7	3,7	1,9		1,9	Start an off-delay timer (stored in ACCUM 1) on the trailing edge of the RLO.
R	T	Y	N	Y	3,7	3,7	1,9		1,9	Reset a timer if RLO=”1”.
Counter Operations										
CU	C	Y	N	Y	3,7	3,7	1,9		1,9	Counter counts up 1 on the leading edge of the RLO.
CD	C	Y	N	Y	3,7	3,7	1,9		1,9	Counter counts down 1 on leading edge of the RLO.
S	C	Y	N	Y	3,7	3,7	1,9		1,9	Set counter if RLO=”1”.
R	C	Y	N	Y	3,7	3,7	1,9		1,9	Reset counter if RLO=”1”.
Arithmetic Operations										
+F		N	N	N	1,6	1,6	0,8		0,8	Add two fixed-point numbers: ACCUM 1 + ACCUM 2. Result evaluation via CC 1/CC 0/OV
- F		N	N	N	1,6	1,6	0,8		0,8	Subtract two fixed-point numbers: ACCUM 1 - ACCUM 2. Result evaluation via CC 1/CC 0/OV
Comparison Operations										
!=F		N	Y	N	1,6	1,6	0,8		0,8	Compare two fixed-point numbers for ”equal to”. If ACCUM 2=ACCUM 1, the RLO is ”1”. CC 1/CC 0 are affected.

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 µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
Comparison Operations (cont.)										
><F		N	Y	N	1,6	1,6	0,8		0,8	Compare two fixed-point numbers for ”not equal to”. If ACCUM 2 ACCUM 1, the RLO is ”1”. CC 1/CC 0 are affected.
>F		N	Y	N	1,6	1,6	0,8		0,8	Compare two fixed-point numbers for ”greater than”. If ACCUM 2>ACCUM 1, the RLO is ”1”. CC 1/CC 0 are affected.
>=F		N	Y	N	1,6	1,6	0,8		0,8	Compare two fixed-point numbers for ”greater than or equal to”. If ACCUM 2 ACCUM 1, the RLO is ”1”. CC 1/CC 0 are affected.
<F		N	Y	N	1,6	1,6	0,8		0,8	Compare two fixed-point numbers for ”less than”. If ACCUM 2<ACCUM 1, the RLO is ”1”. CC 1/CC 0 are affected.
<=F		N	Y	N	1,6	1,6	0,8		0,8	Compare two fixed-point numbers for ”less than or equal to”. If ACCUM 2 ACCUM 1, the RLO is ”1”. CC 1/CC 0 are affected.
Block Call Operations										
JU	OB	N	N	Y	6,7	6,7	3,4		3,4	Jump unconditionally to an organization block.
JU	PB	N	N	Y	6,7	6,7	3,4		3,4	Jump unconditionally to a program block.
JU	FB	N	N	Y	6,7	6,7	3,4		3,4	Jump unconditionally to a function block.

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 µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
Block Call Operations (cont.)										
JU	SB	N	N	Y	6,7	6,7	3,4		3,4	Jump unconditionally to a sequence block.
JC	OB	Y	Y[1]	Y	6,7 1,7	6,7 1,7	3,4 0,9		3,4 0,9	Jump conditionallly to an organization block. Time applies for RLO=1/RLO=0
JC	PB	Y	Y[1]	Y	6,7 1,7	6,7 1,7	3,4 0,9		3,4 0,9	Jump conditionally to a program block. Time applies for RLO=1/RLO=0
JC	FB	Y	Y[1]	Y	6,7 1,7	6,7 1,7	3,4 0,9		3,4 0,9	Jump conditionally to a function block. Time applies for RLO=1/RLO=0
JC	SB	Y	Y[1]	Y	6,7 1,7	6,7 1,7	3,4 0,9		3,4 0,9	Jump conditionally to a sequence block. Time applies for RLO=1/RLO=0
A	DB	N	N	N	3,6	3,6	1,8		1,8	Call a data block.
E	DB	N	N	N	270	270	270		270	Generate a data block. The number of data words in the block must be stored in ACCUM 1.
Return Operations										
BE		N	N	Y	5	5	2,5		2,5	Block end (termination of a block)
BEC		Y	Y[1]	Y	5 1,7	5 1,7	2,5 0,9		2,5 0,9	Block end, conditional Time applies for RLO=1/RLO=0
BEU		N	N	Y	5	5	2,5		2,5	Block end, unconditional

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 µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
”No” Operations										
NOP 0		N	N	N	1,6	1,6	0,8		0,8	No operation (all bits reset)
NOP 1		N	N	N	1,6	1,6	0,8		0,8	No operation (all bits set)
Stop Operation										
STP		N	N	N	50	50	50		50	Stop: scanning cycle is still completed. Error ID ”STS” is set in the ISTACK.
Display Generation Operations										
BLD 130		N	N	N	1,6	1,6	0,8		0,8	Display generation operation for the programmer: carriage return generates blank line.
BLD 131		N	N	N	1,6	1,6	0,8		0,8	Display generation operation for the programmer: switch over to statement list (STL).
BLD 132		N	N	N	1,6	1,6	0,8		0,8	Display generation operation for the programmer: switch over to control system flowchart (CSF).
BLD 133		N	N	N	1,6	1,6	0,8		0,8	Display generation operation for the programmer: switch over to ladder diagram (LAD).
BLD 255		N	N	N	1,6	1,6	0,8		0,8	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)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
Boolean Logic Operations										
A=	Formal oper. I, Q, F, T, C	N	Y	N	160*	160*	160*		3,6*	AND operation: scan formal operand for ”1”. (Data type: BI)
AN=	Formal oper. I, Q, F, T, C	N	Y	N	163*	163*	163*		3,6*	AND operation: scan formal operand for ”0”. (Data type: BI)
O=	Formal oper. I, Q, F, T, C	N	Y	N	164*	164*	164*		3,6*	OR operation: scan formal operand for ”1”. (Data type: BI)
ON=	Formal oper. I, Q, F, T, C	N	Y	N	165*	165*	165*		3,6*	OR operation: scan formal operand for ”0”. (Data type: BI)
AW		N	N	N	1,6	1,6	0,8		0,8	Combine contents of ACCUM 2 and ACCUM 1 through logic AND (word operation). Result is stored in ACCUM 1. CC 1/CC 0 are affected.
OW		N	N	N	1,6	1,6	0,8		0,8	Combine contents of ACCUM 2 and ACCUM 1 through logic OR (word operation). Result is stored in ACCUM 1. CC 1/CC 0 are affected.
XOW		N	N	N	1,6	1,6	0,8		0,8	Combine contents of ACCUM 2 and ACCUM 1 through EXCLUSIVE OR (word operation). Result is stored in ACCUM 1. CC 1/CC 0 are affected.
Bit Operations										
TB	T, C	N	Y	N	143	143	143		143	Test a timer or counter word bit for ”1”.

***** 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 Operanden	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
Bit Operations (cont.)										
TB	D	N	Y	N	155	155	155		155	Test a data word bit for ”1”.
TB	RS	N	Y	N	141	141	141		141	Test a data word bit in the system data range for ”1”.
TBN	T, C	N	Y	N	143	143	143		143	Test a timer or counter word bit for ”0”.
TBN	D	N	Y	N	159	159	159		159	Test a data word bit for ”0”.
TBN	RS	N	Y	N	139	139	139		139	Test a data word bit in the system data range for ”0”.
SU	T, C	N	N	Y	143	143	143		143	Set a timer or counter word bit unconditionally.
SU	D	N	N	Y	159	159	159		159	Set a data word bit unconditionally.
RU	T, C	N	N	Y	143	143	143		143	Reset a timer or counter word bit unconditionally.
RU	D	N	N	Y	158	158	158		158	Reset a data word bit unconditionally.
Set/Reset Operations										
S=	Formal oper. E, A, M	Y	N	Y	150*****	150*****	150*****		3,6*****	Set a formal operand, (with RLO =1). (Data type: BI)
RB=	Formal oper. E, A, M	Y	N	Y	150*****	150*****	150*****		3,6*****	Reset a formal operand, (with RLO =1). (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)

Operation (STL)	Permissible Operands	1 RLO depend. 2 RLO affected 3 RLO reloaded			Typical Execution Time in µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
Set/Reset Operations (cont.)										
RD=	Formal oper. T, C	Y	N	Y	146*	146*	146*		3,6*	Reset a formal operand (digital), (with RLO =1).
==	Formal oper. I, Q, F	Y	N	Y	150*	150*	150*		3,6*	Assign the value of the RLO is assigned to the status of the formal operand. (Data type: BI)
Timer and Counter Operations										
FR	T, C	Y	N	Y	3,7	3,7	3,7		1,9	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	144*	144*	144*		3,6*	Enable formal operand (timer/counter) for cold re-start (for detailed description, see ”FR” operation).
SP=	Formal oper. T	Y	N	Y	144*	144*	144*		3,6*	Start a timer (formal operand) as pulse with the value stored in ACCUM 1.
SR=	Formal oper. T	Y	N	Y	144*	144*	144*		3,6*	Start an on-delay timer (formal operand) with the value stored in ACCUM 1.
SEC=	Formal oper. T, C	Y	N	Y	144*	144*	144*		3,6*	Start a timer (formal operand) as extended pulse with the value stored in ACCUM 1, or set a counter (formal operand) with the next indicated count value.

* 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 µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
Timer and Counter Operations (cont.)										
SSU=	Formal oper. T, C	Y	N	Y	144*	144*	144*		3,6*	Start a stored on-delay timer (formal operand) with the value stored in ACCUM 1, or increment a counter (formal operand).
SFD=	Formal oper. T, C	Y Y	N	Y	144*	144*	144*		3,6*	Start an off-delay timer ( ) (formal operand) with the value stored in ACCUM 1, or decrement a counter ( ) (formal operand).
Load and Transfer Operations										
L=	Formal oper. I, Q, F, T, C	N	N	N	147*	147*	147*		3,6*	Load the value of the formal operand into ACCUM 1 (Data type: BY, W; additional actual operands: DL, DR, DW).
L	RS	N	N	N	89	89	89		89	Load a word from the system data range into ACCUM 1.
LC=	Formal oper. T, C	N	N	N	145*	145*	145*		3,6*	Load the value of the formal operand in BCD code into ACCUM 1.
LW=	Formal oper. I, Q, F, T, C	N	N	N	124*	124*	124*		3,6*	Load a formal operand bit pattern into ACCUM 1 (Parameter type: D; data type: KC, KF, KH, KM, KS, KT, KY).
T=	Formal oper. I, Q, F	N	N	N	148*	148*	148*		3,6*	Transfer the contents of ACCUM 1 to the formal operand (Data type: BY, W; additional actual operands: DL, DR, DW).

***** 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 µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
Conversion Operations										
CFW		N	N	N	1,6	1,6	0,8		0,8	Form the one's complement of ACCUM 1.
CSW		N	N	N	1,6	1,6	0,8		0,8	Form the two's complement of ACCUM 1. CC 1/CC 0 and OV are affected.
Shift Operations										
SLW	Parameter n=0 ... 15	N	N	N	1,6	1,6	0,8		0,8	Shift the contents of ACCUM 1 to the left by the value specified in the parameter. Unassigned positions are padded with zeros. CC 1/CC 0 are affected.
SRW	Parameter n=0 ... 15	N	N	N	1,6	1,6	0,8		0,8	Shift the contents of ACCUM 1 to the right by the value specified in the parameter. Unassigned positions are padded with zeros. CC 1/CC 0 are affected.
Jump Operations										
JU=	Symb. address max. 4 charact.	N	N	N	1,6	1,6	0,8		0,8	Jump unconditionally to the symbolic address.
JC=	Symb. address max. 4 charact.	Y	Y[1]	Y	1,6	1,6	0,8		0,8	Jump conditionally to the symbolic address. (If the RLO is ”0”, it is set to ”1”).
JZ=	Symb. address max. 4 charact.	N	N	N	1,6	1,6	0,8		0,8	Jump if the result is zero. The jump is made only if CC 1=0 and CC 0=0. The RLO is not changed.

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 µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
Jump Operations (cont.)										
JN=	Symb. address max. 4 charact.	N	N	N	1,6	1,6	0,8		0,8	Jump if the result is not zero. The jump is made only if CC 1 CC 0 . The RLO is not changed.
JP=	Symb. address max. 4 charact.	N	N	N	1,6	1,6	0,8		0,8	Jump if the result>0. The jump is made only if CC 1=1 und CC 0=0. The RLO is not changed.
JM=	Symb. address max. 4 charact.	N	N	N	1,6	1,6	0,8		0,8	Jump if the result <0. The jump is made only if CC 1=0 and CC 0=1. The RLO is not changed.
JO=	Symb. address max. 4 charact.	N	N	N	1,6	1,6	0,8		0,8	Jump on overflow. The jump is made only if the OVERFLOW bit is set. The RLO is not changed.
Other Operations										
IA		N	N	N	55	55	55		55	Disable interrupt. Input/output interrupt or timer OB processing is disabled.
RA		N	N	N	55	55	55		55	Enable interrupt. This operation cancels the effect of IA.
D		N	N	N	1,7	1,7	0,9		0,9	Decrement the low byte (bits 0 to 7) of ACCUM 1 by the value n (n=0 to 255).
I		N	N	N	1,7	1,7	0,9		0,9	Increment the low byte (bits 0 to 7) of ACCUM 1 by the value n (n=0 to 255).

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 µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
Other Operations (cont.)										
DO=	Formal oper. I, Q, F, T, C	N	N	N	170*	170*	170*		3,6*	Process a block. (Only C DB, JU PB, JU FB, JU SB and JU OB can be substituted).
DO	DW**	N	N	N	162*	162*	162*		3,6*	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	134*	134*	134*		2,6*	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;
S, R, =;
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, LC, T;
JU, JC, JZ, JN, JP, JM, JO, SLW, SRW;
D, I;
C DB; T RS, TNB

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 µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
Set Operations										
SU	RS	N	N	Y	142	142	142		142	Set bit in system data range unconditionally.
RU	RS	N	N	Y	142	142	142		142	Reset bit in system data range unconditionally.
Load and Transfer Oprations										
LIR	0 (ACCUM 1) 2 (ACCUM 2)	N	N	N	126*	126*	126*		4,5**	Load the contents of a memory word (addressed by ACCUM 1) indirectly into the register (0: ACCUM 1; 2: ACCUM 2) [1]
TIR	0 (ACCUM 1) 2 (ACCUM 2)	N	N	N	105*	105*	105*		4,5**	Transfer the contents of the accumulator indirectly to the memory word (addressed by ACCUM 1) (0: ACCUM 1; 2: ACCUM 2) [1]
LDI	A1 (ACCUM 1) A2 (ACCUM 2)	N	N	N	-	-	-		126	Load the contents of a memory word (addressed by ACCUM 1) indirectly into ACCUM 1 or ACCUM 2 (A1=ACCUM 1, A2=ACCUM 2) [2]
TDI	A1 (ACCUM 1) A2 (ACCUM 2)	N	N	N	-	-	-		105	Transfer the register contents indirectly into the memory word (addressed by ACCUM 1) (A1=ACCUMW1, A2=ACCUM 2) [2]
TNB	Parameter n=0 ... 255	N	N	N	* 68 + 34 · n	* 68 + 34 · n	* 68 + 34 · n		* 2,9+ n(1,7 +*)	Transfer a block byte by byte (number of bytes 0 to 255). End address source: ACCUM 2 End address target: ACCUM 2

1 In the case of CPU 944 access to memory bank 1
2 In the case of CPU 944 access to memory bank 2

* When accessing the I/O area, the relevant timeouts for each byte access must be added.
** +2× 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 µsec.					Function
		1	2	3	CPU 941	CPU 942	CPU 943		CPU 944	
Load and Transfer Oprations (cont.)										
T	RS	N	N	N	75	75	75		75	Transfer a word to the system data range.
Jump Operation										
JUR		N	N	N	105	105	105		105	Jump randomly within a function block (jump displacement: - 32768 to+32767).
Arithmetic Operations										
ADD	BF	N	N	N	57	57	57		57	Add byte constant (fixed point) to ACCUM 1.
ADD	KF	N	N	N	90	90	90		90	Add fixed-point constant (word) to ACCUM 1.
Other Operations										
DI	Formal oper. I, Q, F, T, C	N	N	N	174*	174*	174*		174*	Process via a formal operand (indirectly). The number of the formal operand is in ACCUM 1.
STS		N	N	N	50	50	50		50	Stop operation. Program processing is interrupted immediately after this operation.
TAK		N	N	N	80	80	80		80	Swap the contents of ACCUM 1 and ACCUM 2.

***** plus execution time of the substituted operation

Machine Code Listing

Explanation of the 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 in bytes
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					LC	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					LC=	
0	F	0_c	0_c					O=	
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	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	
2	1	6	0					><F	
2	1	8	0					!=F	
2	1	A	0					>=F	

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
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					SR	T
2	5	0_i	0_i					JM=	
2	6	0_c	0_c					SR=	
2	7	0_c	0_c					AN=	
2	8	0_e	0_e					L	KB
2	A	0_g	0_g					L	DR
2	B	0_g	0_g					T	DR
2	C	0_g	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
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=	

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
3	7	0_c	0_c					RB=	
3	C	0_d	0_d					R	T
3	D	0_f	0_f					JU	FB
3	E	0_c	0_c					RD=	
3	F	0_c	0_c					LW=	
4	0	0	0_k					LIR	
4	1	0	0					AW	
4	2	0_o	0_o					L	C
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					LC	C
4	D	0_f	0_f					JC	OB
4	E	0_g	0_g					DO	FW
5	0	0_e	0_e					ADD	BF
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

Maschinen-Code								Opera-tion	Ope-rand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
5	4	0_o	0_o					CD	C
5	5	0_f	0_f					JC	PB
5	8	0	0	0_e	0_e	0_e	0_e	ADD	KF
5	9	0	0					-F	
5	C	0_o	0_o					S	C
5	D	0_f	0_f					JC	SB
6	1	0_h	0_h					SLW	
6	2	0_g	0_g					L	RS
6	3	0_g	0_g					T	RS
6	5	0	0					BE	
6	5	0	1					BEU	
6	6	0_c	0_c					T=	
6	8	0	B					LDI	A1
6	8	0	F					TDI	A1
6	8	2	B					LDI	A2
6	8	2	F					TDI	A2
6	9	0_h	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
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	JRA	
7	0	1	5	C	0	0_o	0_o	TB	C
7	0	1	5	8	0	0_o	0_o	TBN	C

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
7	0	1	5	4	0	0_o	0_o	SU	C
7	0	1	5	0	0	0_o	0_o	RU	C
7	0	2	5	C	0	0_d	0_d	TB	T
7	0	2	5	8	0	0_d	0_d	TBN	T
7	0	2	5	4	0	0_d	0_d	SU	T
7	0	2	5	0	0	0_d	0_d	RU	T
7	0	4	6	C	0_b	0_g	0_g	TB	D
7	0	4	6	8	0_b	0_g	0_g	TBN	D
7	0	4	6	4	0_b	0_g	0_g	SU	D
7	0	4	6	0	0_b	0_g	0_g	RU	D
7	0	5	7	C	0_b	0_g	0_g	TB	RS
7	0	5	7	8	0_b	0_g	0_g	TBN	RS
7	0	5	7	4	0_b	0_g	0_g	SU	RS
7	0	5	7	0	0_b	0_g	0_g	RU	RS
7	2	0_a	0_a					L	PB/PY*
7	3	0_a	0_a					T	PB/PY*
7	5	0_f	0_f					JU	PB
7	6	0_c	0_c					DO=	
7	8	0	5	0	0	0_f	0_f	G	DB
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	E	0	0					DI	

* PY in the case of S5-DOS programmers

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
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
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

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
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					NOP 1	

Alphabetical Index of Operations

Integral Blocks

Intregral Organization Blocks

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

OB-No.	Function	OB integrated in CPU			
		941	942	943	944
OB1	Cyclic program scanning	▒	▒	▒	▒
Interrupt-driven program scanning with priorities A, B, C, D					
OB2	Interrupt A: Digital input module -434 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 scanning (variable in each case: 10 msec. to 10 min.)	▒	▒	▒	▒
OB11		▒	▒	▒	▒
OB12		▒	▒	▒	▒
OB13		▒	▒	▒	▒

▒ OB available

Integral Organization Blocks

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

OB-No.	Function	OB integrated in CPU			
		941	942	943	944
Controlling restart characteristics					
OB21	Manual switch on	▒	▒	▒	▒
OB22	Automatic switch on when power is restored	▒	▒	▒	▒
Handling programming errors and PLC faults					
OB19	When a bl. is called which has not been loaded.	▒	▒	▒	▒
OB23	Time-out during individual access to the S5 bus (e.g. LIR)	▒	▒	▒	▒
OB24	Time-out during update of process image and interproc. communic. flags	▒	▒	▒	▒
OB27	Substitution error	▒	▒	▒	▒
OB32	Transfer errors in DB or with GDB operation	▒	▒	▒	▒
OB34	Battery failure	▒	▒	▒	▒

▒ OB available

Integral Organization Blocks

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

OB-No.	Function	OB integrated in CPU			
		941	942	943	944
OBs which offer operating functions					
OB31	Scan time triggering	▒	▒	▒	▒
OB160	Programmable time loop	▒	▒	▒	▒
OB251	PID algorithm	▒	▒	▒	▒
OB254	Read in process I/O image				▒
OB255	Output process I/O image				▒

▒ OB available

Integral Function Blocks

FB-No.	Function	FB integrated in CPU			
		941	942	943	944
FB238	Compress PLC memory				
FB239	Delete block				
FB240	4-tetrad BCD code converter				
FB241	16-bit fixed-point converter				
FB242	16-bit binary multiplier				
FB243	16-bit binary divider				
FB244	Send data				
FB245	Receive data				
FB246	Fetch data				
FB247	Monitor job processing				
FB248	Delete job				
FB249	Initialize interface				
FB250	Read analog value				
FB251	Output analog value				

FB available

Integral Data Block 1

Parameter	Argument		Meaning
Block Identifier: SL1:			**SINEC L1**
SLN	p		"***SL***ave-***N***umber" (p=1 to 30; p=0 to 30 in the case of CPU 943/944 with 2 interfaces)
SM	DBxDWy or MBy		Position of the ***S***end ***M***ailbox (start of SM)
RM			Position of the ***R***eceive ***M***ailbox (start of RM)
CBR			Position of the ***C***oordination ***B***yte ***R***eceive
CBS			Position of the ***C***oordination ***B***yte ***S***end (x=2 to 255; y=0 to 255)
PGN	p		***PG*** bus ***N***umber (p=1 to 30) 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		"***W***atch***D***og" (scan time monitoring) can be set in milliseconds but only in steps of 10 msec. (p=0 to 2550)
RDLY	r		"***R***un***D***e***L***a***Y***" restart delay after POWER ON in msec. (r=0 to 65535) Note: RDLY only becomes effective in case of backup oper.: in case of EPROM oper., the setting is fixed at 1000 ms!
RT	Y/N		"***R***esident ***T***imers" (if "Y", all timers are retentive, if "N", only the first half are retentive)[1]
RC	Y/N		"***R***esident ***C***ounters" (if "Y", all counters are retentive, if "N", only the first half are retentive)[1]
RF	Y/N		"***R***esident ***F***lags" (if "Y", all flags are retentive, if "N", only the first half are retentive)[1]
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?
PII	Y/N		"***P***rocess ***I***mage ***I***nput" disable read in of process image?
PRIO	s		OB6 ***PRIO***rity (the following is the descending order of priority:) s=0 **OB 6**, OB 2 to 5, OB 13 to 10 s=1 OB 2 to 5, **OB 6**, OB 13 to 10) (OBs 2 to 6 cannot be interrupted!)
	N=no Y=yes		
Block Identifier: TFB:			**Timer-Function Block**
OB10	p		Interval (msec.) during which OB10 to 13 is called and processed (p=0 to 655350 (programmable in 10-msec. intervals)
OB11	p		
OB12	p		
OB13	p		

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

Integral Data Block 1

Parameter	Argument		Meaning
Block Identifier: PFB:			**Placement of FB**
SFB	p q		"***S***ubstitute ***FB***" Replace number p of the integral FB p (COMPR or DELETE) with the number q
p = 238, 239 q=0 to 239, 252 to 255			
Block Identifier: CLP:			**Clock Parameters (only in the case of CPU 943/944 with two interfaces)**
CLK	DBxDWy or MBy		"***CL***oc***K*** Data" start of clock data area
STW	DBxDWy or MBy		"***ST***atus ***W***ord" Position of the status word
SET	wd dd.mm.yy hh:mm:ss AM/PM[1]		Set clock time, date
TIS	wd dd.mm. hh:mm:ss AM/PM[1]		"***T***imer ***I***nterrupt ***S***et"
OHS	hhhhhh:mn:ss[2]		"***O***peration ***H***our counter ***S***et"
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		"***C***orrection ***F***actor" Enter correction factor
wd = 1 to 7 (Weekday=Su to Sa) dd = 01 to 31 (Day) mm = 01 to 12 (Month) yy = 0 to 99 (Year) hh = 1 to 12 (AM/PM) 00 ... 23 mn = 00 to 59 (Minutes) ss = 00 to 59 (Seconds) hhhhhh = 000000 to 999999 (Hours)			p=- 400 to +400 x=2 to 255 y=0 to 255 y/Y=yes n/N=no
Block Identifier: ERT:			**Error Return**
ERR	MBx or DByDWz		"***ERR***ors" Position of the error code (x=0 to 236 y=2 to 255 z=0 to 255)

[1] If an argument (e.g. weekday) 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-hr mode.

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

Evaluation of CC 1 and CC 0

CC 1	CC 0	Arith-metic Opera-tions	Digital Logic Opera-tions	Compa-rison Opera-tions	Shift Opera-tions	Con-version Opera-tions
0	0	Result =0	Result =0	ACCUM 2 = ACCUM 1	shifted Bit =0	-
0	1	Result <0	-	ACCUM 2 < ACCUM 1	-	Result <0
1	0	Result >0	Result 0	ACCUM 2 > ACCUM 1	shifted Bit =0	Result >0

Siemens AG
AUT E1114B
Postfach 1963
Werner-von-Siemens-Str. 50

D-92209 Amberg
Fed. Rep. 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:

Name
Company/Department
Address

Telephone

Publication:
Programmable Controller
SIMATIC S5-115U (CPU 941/942/943/944)
Reference Guide

Order No.: 6ES5 997-7LA21

Suggestions/Corrections:

Siemens AG
Automation Group
Industrial Automation Systems
Postfach 4848, 8500 Nürnberg 1

Siemens Aktiengesellschaft

Order No. 6ES5 997-7LA21
Printed in the Fed. Rep. of Germany