SIMATIC S5

S5-100U
Programmable Controller
CPU 100/102/103

Reference Guide

Order No.
6ES5 997-8MA21

Index

Explanation of the Operations List

Abbreviation	Explanation
ACCU 1	Accumulator 1 (When accumulator 1 is loaded, any existing contents are shifted into accumulator 2.)
ACCU 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 Y reloaded? 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 Y 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 Y/N affected?	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 100	**CPU 102**	**CPU 103**
BN	Byte constant (fixed-point number)	- 127 to +127		
C	Counter - remanent - not remanent - for the bit test and set operations (system operat.)	 0 to 7 8 to 15 0 to 15	 0 to 7 8 to 127 0 to 127	 0 to 7 8 to 127 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 63		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 - remanent - not remanent	 0.0 to 63.7 64.0 to 127.7		
FB	Function block	0 to 63	0 to 63*	0 to 255

* +integrated FBs such as CPU 103

Abb.	Explanation	Permissible operand value range for		
		CPU 100	CPU 102	CPU 103
FB	Flag byte			
	- remanent	0 to 62		0 to 255
	- not remanent	64 to 126		64 to 254
FW	Flag word			
	- remanent	0 to 62		0 to 255
	- not remanent	64 to 126		64 to 254
I	Input	0.0 to 127.7		
IB	Input byte	0 to 127		
IW	Input word	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 bits)		
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)		

Abb.	**Explanation**	**Permissible operand value range for**		
		CPU 100	**CPU 102**	**CPU 103**
OB	Organization block for special applications: 1, 2, 13, 21, 22, 31, 34, 251	0 to 63		0 to 255
PB	Program block (with block call and return operations)	0 to 63		0 to 255
PB/ PY*	Peripheral byte			0 to 127
PW	Peripheral word			0 to 126
Q	Output	0.0 to 127.7		
QB	Output byte	0 to 127		
QW	Output word	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 operat.)	0 to 15	0 to 31	0 to 31 0.0 to 127.15

* Depending on the type of programmer used

Basic 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 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
Boolean Logic Operations										
A	I, Q, F	N	Y	N	typ. 70	4	1,6		0,8	Scan operand for “1” and combine with RLO through logic AND.
	T, C					7				
AN	I, Q, F	N	Y	N	typ. 75	4	1,6		0,8	Scan operand for “0” and combine with RLO through logic AND.
	T, C					9				
O	I, Q, F	N	Y	N	typ. 75	4	1,6		0,8	Scan operand for “1” and combine with RLO through logic OR.
	T, C					7				
ON	I, Q, F	N	Y	N	typ. 80	4	1,6		0,8	Scan operand for “0” and combine with RLO through logic OR.
	T, C					9				
O		N	Y	Y	41	7	1,6		0,8	Combine AND operations through logic OR.
A(		N	Y	Y	61	6	1,6		0,8	Combine expressions enclosed in parentheses through logic AND (6 nesting levels).
O(		N	Y	Y	64	6	1,6		0,8	Combine expressions enclosed in parentheses through logic OR (6 nesting levels).
)		N	Y	N	51	13	1,6		0,8	Close parentheses (conclusion of a parenthetical expr.).
Set/Reset Operations										
S	I, Q, F	Y	N	Y	typ. 70	7	1,6		0,8	Set operand to “1”.

Basic Operations

For organization blocks (OB)

For program blocks (PB)

For function blocks (FB)

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 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
Set/Reset Operations (cont.)										
R	I, Q, F	Y	N	Y	typ. 70	7	1,6		0,8	Reset operand to “0”.
=	I, Q, F	N	N	Y	typ. 70	6	1,6		0,8	Assign value of RLO to operand.
Load Operations										
L	IB	N	N	N	59	14	1,6		0,8	Load an input byte from the PII into ACCU 1.
L	QB	N	N	N	63	14	1,6		0,8	Load an output byte from the PIQ into ACCU 1.
L	IW	N	N	N	59	17	1,6		0,8	Load an input word from the PII into ACCU 1: byte n ACCU 1 (bits 8-15); byte n+1 ACCU 1 (bits 0-7)
L	QW	N	N	N	63	17	1,6		0,8	Load an output word from the PIQ into ACCU 1: byte n ACCU 1 (bits 8-15); byte n+1 ACCU 1 (bits 0-7)
L	PB/PY (PG dep.)	--	--	N	--	--	91		68	Permissible in OB2 and OB13. Load an input byte of the digital/analog inputs from the interrupt PII into ACCU 1.
L	PW	--	--	N	--	--	92		69	Permissible in OB2 and OB13. Load an input byte of the digital/analog inputs from the interrupt PII into ACCU 1.
L	FB	N	N	N	64	14	1,6		0,8	Load a flag byte into ACCU 1.

Basic 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 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
Load Operations (cont.)										
L	FW	N	N	N	71	17	1,6		0,8	Load a flag word into ACCU 1: byte n ACCU 1 (bits 8-15); byte n+1 ACCU 1 (bits 0-7)
L	DL	N	N	N	65	39	82		1,7	Load a data word (left-hand byte) of the current data block into ACCU 1.
L	DR	N	N	N	65	41	83		1,7	Load a data word (right-hand byte) of the current data block into ACCU 1.
L	DW	N	N	N	66	43	85		2,0	Load a data word of the cur-rent data block into ACCU 1: byte n ACCU 1 (bits 8-15); byte n+1 ACCU 1 (bits 0-7)
L	KB	N	N	N	54	7	59		1,45	Load a constant (1-byte number) into ACCU 1.
L	KS	N	N	N	57	7	1,6		0,8	Load a constant (2 characters in ASCII format) into ACCU 1.
L	KF	N	N	N	57	7	1,6		0,8	Load a constant (fixed-point number) into ACCU 1.
L	KH	N	N	N	57	7	1,6		0,8	Load a constant (hexadecimal code) into ACCU 1.
L	KM	N	N	N	57	7	1,6		0,8	Load a constant (bit pattern) into ACCU 1.
L	KY	N	N	N	57	7	1,6		0,8	Load a constant (2-byte number) into ACCU 1.
L	KT	N	N	N	57	7	1,6		0,8	Load a constant (time in BCD) into ACCU 1.

Basic 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 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
Load Operations (cont.)										
L	KC	N	N	N	57	7	1,6		0,8	Load a constant (count in BCD) into ACCU 1.
L	T, C	N	N	N	typ. 70	19	1,6		0,8	Load a time or count (in binary code) into ACCU 1.
LC	T, C	N	N	N	125	69	154		1,8	Load times or counts (in BCD) into ACCU 1.
Transfer Operations										
T	IB	N	N	N	51	5	1,6		0,8	Transfer the contents of ACCU 1 to an input byte (into the PII).
T	QB	N	N	N	54	5	1,6		0,8	Transfer the contents of ACCU 1 to an output byte (into the PIQ).
T	IW	N	N	N	53	11	1,6		0,8	Transfer the contents of ACCU 1 to an input word (into the PII): ACCU 1 (bits 8-15) byte n; ACCU 1 (bits 0-7) byte n+1
T	QW	N	N	N	56	11	1,6		0,8	Transfer the contents of ACCU 1 to an output word (into the PIQ: ACCU 1 (bits 8-15) byte n; ACCU 1 (bits 0-7) byte n+1
T	PB/PY (PG dep.)	--	--	N	--	--	60		37	Permissible in OB2 and OB13. Transfer the contents of ACCU 1 to the interrupt PIQ with updating of the PIQ.

Basic 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 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
Transfer Operations (cont.)										
T	PW	--	--	N	--	--	67		51	Permissible in OB2 and OB13. Transfer the contents of ACCU 1 to the interrupt PIQ with updating of the PIQ.
T	FY	N	N	N	55	5	1,6		0,8	Transfer the contents of ACCU 1 to a flag byte.
T	FW	N	N	N	64	11	1,6		0,8	Transfer the contents of ACCU 1 to a flag word (into the PIQ): ACCU 1 (bits 8-15) byte n; ACCU 1 (bits 0-7) byte n+1
T	DL	N	N	N	53	31	75		1,15	Transfer the contents of ACCU 1 to a data word (left-hand byte)
T	DR	N	N	N	57	33	78		1,15	Transfer the contents of ACCU 1 to a data word (right-hand byte)
T	DW	N	N	N	59	36	81		1,4	Transfer the contents of ACCU 1 to a data word
Timer Operations										
SP	T	Y	N	Y	125	74	147		1,9	Start a timer (stored in ACCU 1) as a signal-contracting pulse.
SE	T	Y	N	Y	125	74	147		1,9	Start a timer (stored in ACCU 1) as extended pulse (signal contracting and stretching).

Basic 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 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
Timer Operations (cont.)										
SD	T	Y	N	Y	127	76	150		1,9	Start an on-delay timer (stored in ACCU 1).
SS	T	Y	N	Y	127	76	150		1,9	Start a stored on-delay timer (stored in ACCU 1).
SF	T	Y	N	Y	125	74	144		1,9	Start an off-delay timer (stored in ACCU 1).
R	T	Y	N	Y	126	75	96		1,9	Reset a timer.
Counter Operations										
CU	C	Y	N	Y	79	42	105		1,9	Counter counts up 1.
CD	C	Y	N	Y	92	31	117		1,9	Counter counts down 1.
S	C	Y	N	Y	118	67	141		1,9	Set counter.
R	C	Y	N	Y	69	12	96		1,9	Reset counter.
Arithmetic Operations										
+F		N	N	N	55	26	1,6		0,8	Add two fixed-point numbers: ACCU 1+ACCU 2. CC 1/CC 0/OV are affected.
-F		N	N	N	58	23	1,6		0,8	Subtract one fixed-point number from another: ACCU 2 - ACCU 1. CC 1/CC 0/OV are affected.

Basic 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 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
Comparison Operations										
!=F		N	Y	N	79	24	1,6		0,8	Compare two fixed-point numbers for “equal to”: If ACCU 2=ACCU 1, the RLO is “1”. CC 1/CC 0 are affected.
><F		N	Y	N	82	27	1,6		0,8	Compare two fixed-point numbers for “not equal to”: If ACCU 2 ACCU 1, the RLO is “1”. CC 1/CC 0 are affected.
>F		N	Y	N	79	24	1,6		0,8	Compare two fixed-point numbers for “greater than”: If ACCU 2 > ACCU 1, the RLO is “1”. CC 1/CC 0 are affected.
>=F		N	Y	N	79	24	1,6		0,8	Compare two fixed-point numbers for “greater than or equal to”: If ACCU 2 ACCU 1, the RLO is “1”. CC 1/CC 0 are affected.
<F		N	Y	N	82	27	1,6		0,8	Compare two fixed-point numbers for “less than”: If ACCU 2 < ACCU 1, the RLO is “1”. CC 1/CC 0 are affected.
<=F		N	Y	N	82	27	1,6		0,8	Compare two fixed-point numbers for “less than or equal to”: If ACCU 2 ACCU 1, the RLO is “1”. CC 1/CC 0 are affected.
Block Call Operations										
JU	PB	N	N	Y	125	49	185		3,35	Jump unconditionally to a program block.
JU	FB	N	N	Y	147	49	187		3,35	Jump unconditionally to a function block.

Basic Operations

For organization blocks (OB)

For program blocks (PB)

For function blocks (FB)

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 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
Block Call Operations (cont.)										
JU	SB	N	N	Y	--	--	185		3,35	Jump unconditionally to a sequence block.
JC	PB	Y	Y1)	Y	130	53	190		3,35	Jump conditionally to a program block.
JC	FB	Y	Y1)	Y	152	53	196		3,35	Jump conditionally to a function block.
JC	SB	Y	Y1)	Y	--	--	194		3,35	Jump conditionally to a sequence block.
C	DB	N	N	N	70	28	79		1,75	Call a data block.
G	DB	N	N	Y	--	--	233		182	Generate or delete a data block. The number of data words in the block must be stored in ACCU 1.
Return Operations										
BE		N	N	Y	88	36	119		2,5	Block end (termination of a block)
BEC		Y	Y1)	Y	90	38	121		2,5	Block end, conditional
BEU		N	N	Y	88	36	119		2,5	Block end, unconditional (BEU cannot be used in organization blocks.)
“No” Operations										
NOP 0		N	N	N	35	0	1,6		0,8	No operation (all bits reset)

1) RLO is set to “1”.

Basic 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 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
“No” Operations (cont.)										
NOP 1		N	N	N	35	0	1,6		0,8	No operation (all bits set)
Stop Operations										
STP		N	N	N	35	1	53		25	Stop: scanning is still completed before a stop. Error ID “STS” is set in the ISTACK.
Display Generation Operations										
BLD 130		N	N	N	35	0	1,6		0,8	Display generation operation for the programmer: carriage return generates blank line.
BLD 131		N	N	N	35	0	1,6		0,8	Display generation operation for the programmer: switch to statement list (STL).
BLD 132		N	N	N	35	0	1,6		0,8	Display generation operation for the programmer: switch to control system flowchart (CSF).
BLD 133		N	N	N	35	0	1,6		0,8	Display generation operation for the programmer: switch to ladder diagram (LAD).
BLD 255		N	N	N	35	0	1,6		0,8	Display generation operation for the programmer: terminate a segment.

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 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
Boolean Logic Operations										
A=	Formal operand L, Q, F, T, C	N	Y	N	--	--	202		151	AND operation: scan formal operand for “1”. (Data type: BI)
AN=	Formal operand L, Q, F, T, C	N	Y	N	--	--	202		151	AND operation: scan formal operand for “0”. (Data type: BI)
O=	Formal operand L, Q, F, T, C	N	Y	N	--	--	202		151	OR operation: scan formal operand for “1”. (Data type: BI)
ON=	Formal operand L, Q, F, T, C	N	Y	N	--	--	202		151	OR operation: scan formal operand for “0”. (Data type: BI)
AW		N	N	N	53	19	1,6		0,8	Combine contents of ACCU 2 and ACCU 1 through logic AND (word operation). Result is stored in ACCU 1. CC 1/CC 0 are affected.
OW		N	N	N	53	19	1,6		0,8	Combine contents of ACCU 2 and ACCU 1 through logic OR (word operation). Result is stored in ACCU 1. CC 1/CC 0 are affected.
XOW		N	N	N	51	19	1,6		0,8	Combine contents of ACCU 2 and ACCU 1 through logic EXCLUSIVE OR (word operation). Result is stored in ACCU 1. CC 1/CC 0 are affected.
Bit Operations										
TB	T, C	N	Y	N	--	--	187		123	Test a bit of a timer or counter word for “1”.

Supplementary Operations

For organization blocks (OB)
For program blocks (PB)

For function blocks (FB)
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 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
Bit Operations (cont.)										
TB	D	N	Y	N	--	--	187		144	Test a bit of a data word for “1”.
TB	RS	N	Y	N	--	--	185		121	Test a bit of a data word in the system data area for “1”.
TBN	T, C	N	Y	N	--	--	188		124	Test a bit of a timer or counter word for “0”.
TBN	D	N	Y	N	--	--	188		145	Test a bit of a data word for “0”.
TBN	RS	N	Y	N	--	--	186		122	Test a bit of a data word in the system data area for “0”.
SU	T, C	N	N	Y	--	--	180		125	Set a bit of a timer or counter word unconditionally.
SU	D	N	N	Y	--	--	183		146	Set a bit of a data word unconditionally.
RU	T, C	N	N	Y	--	--	189		124	Reset a bit of a timer or counter word unconditionally.
RU	D	N	N	Y	--	--	189		146	Reset a bit of a data word unconditionally.
Set/Reset Operations										
S=	Formal operand I, Q, F	Y	N	Y	--	--	202		151	Set a formal operand (when RLO=1). (Data type: BI)
RB=	Formal operand I, Q, F	Y	N	Y	--	--	203		152	Reset a formal operand (when RLO=1). (Data type: BI)

Supplementary Operations

For organization blocks (OB)
For program blocks (PB)

For function blocks (FB)
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 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
Set/Reset Operations (cont.)										
RD=	Formal operand T, C	Y	N	Y	--	--	197		147	Reset a formal operand (digital) (when RLO=1).
==	Formal operand I, Q, F	Y	N	Y	--	--	202		151	Assign the value of the RLO to the status of the formal operand. (Data type: BI)
Timer and Counter Operations										
FR	T, C	Y	N	Y	--	--	98		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 operand T, C	Y	N	Y	--	--	194*		145*	Enable formal operand (timer/ counter) for cold restart (for detailed description, see “FR” operation).
SP=	Formal operand T	Y	N	Y	--	--	194*		145*	Start a timer (formal operand) as pulse with the value stored in ACCU 1.
SD=	Formal operand T	Y	N	Y	--	--	194*		145*	Start an on-delay timer (formal operand) with the value stored in ACCU 1.
SEC =	Formal operand T, C	Y	N	Y	--	--	194*		145*	Start a timer (formal operand) as an extended pulse with the value stored in ACCU 1, or set a counter (formal operand) with the next count value indicated.

* + Processing time for the substituted command

Supplementary Operations

For organization blocks (OB)

For program blocks (PB)

For function blocks (FB)

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 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
Timer and Counter Operations (cont.)										
SSU =	Formal operand T, C	Y	N	Y	--	--	194*		145*	Start a stored on-delay timer (formal operand) with the value stored in ACCU 1, or increment a counter (formal operand).
SFD =	Formal operand T, C	Y	N	Y	--	--	194*		145*	Start an off-delay timer (formal operand) with the value stored in ACCU 1, or decrement a counter (formal operand).
Load and Transfer Operations										
L=	Formal operand I, Q, F, T, C	N	N	N	--	--	142*		148*	Load the value of the formal operand into ACCU 1. (Data type: BY, W Additional actual operands: DL, DR, DW)
L	RS	N	N	N	--	--	77		61	Load a word from the system data area into ACCU 1.
LC=	Formal operand T, C	N	N	N	--	--	194*		145*	Load the value of the formal operand in BCD code into ACCU 1.
LW=	Formal operand	N	N	N	--	--	152		76	Load a formal operand bit pattern into ACCU 1. (Data type: D Parameter type: KC, KF, KH, KM, KS, KT, KY)
T=	Formal operand I, Q, F	N	N	N	--	--	195*		149*	Transfer the contents of ACCU 1 to the formal operand. (Data type: BY, W Additional actual operands: DR, DL, DW)

* + Processing time for the substituted command

Supplementary Operations

For organization blocks (OB)
For program blocks (PB)

For function blocks (FB)
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 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
Conversion Operations										
CFW		N	N	N	42	4	1,6		0,8	Form the one's complement of ACCU 1.
CSW		N	N	N	60	23	1,6		0,8	Form the two's complement of ACCU 1. CC 1/CC 0 and OV are affected.
Shift Operations										
SLW	Parameter n=0 ... 15	N	N	N	47+ n·10	12+ n·10	1,6		0,8	Shift the contents of ACCU 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	47+ n·10	12+ n·10	1,6		0,8	Shift the contents of ACCU 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	62	2	1,6		0,8	Jump unconditionally to the symbolic address.
JC =	Symb. address max. 4 charact.	Y	Y[1]	Y	65	5	1,6		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	69	6	1,6		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)

Oper- ation (STL)	Permissible Operanden	1 RLO depend. 2 RLO affected 3 RLO reloaded			typical Execution Time in µs					Function
		1	2	3	CPU 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
Jump Operations (cont.)										
JN =	Symb. address max. 4 charact.	N	N	N	69	10	1,6		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	71	6	1,6		0,8	Jump if the sign of the result is “+”. The jump is made only if CC 1=1 and CC 0=0. The RLO is not changed.
JM =	Symb. address max. 4 charact.	N	N	N	71	6	1,6		0,8	Jump if the sign of the result is “-”. 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	65	4	1,6		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	--	--	58		24	Disable interrupt. Input/output interrupt or timer OB processing is disabled.
RA		N	N	N	--	--	58		26	Enable interrupt. This operation cancels the effect of IA.
D		N	N	N	--	--	49		0,9	Decrement the low byte (bits 0 to 7) of ACCU 1 by the value n (n=0 to 255).
I		N	N	N	--	--	49		0,9	Increment the low byte (bits 0 to 7) of ACCU 1 by the value n (n=0 to 255).

Supplementary Operations

For organization blocks (OB)
For program blocks (PB)

For function blocks (FB)
For sequence blocks (SB)

Oper-ation (AWL)	Permissible Operanden	1 RLO depend. 2 RLO affected 3 RLO reloaded			typical Execution Time in µs					Function
		1	2	3	CPU 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
Other Operations (cont.)										
DO=	Formal operand	N	N	Y	--	--	252*		188*	Process a block. (Only C DB, JU OB, J U PB, JU FB, JU SB can be substituted.) Actual operands: C DB, JU OB, JU PB, JU FB, JU SB
DO	DW **	N	N	N	--	--	229		171	Process data word. The next operation is combined with the parameter specified in the data word (OR operation) and then carried out.
DO	FW **	N	N	N	--	--	179		138	Process flag word. The next operation is combined with the parameter specified in the flag word (OR operation) and then carried out.

***** + Processing time for the substituted command
****** Permissible operations:
A, AN, O, ON;
S, R, =;
FR T, RT, SF T, SD T, SP T, SS T, SE T;
FR C, RC, SC, CU, CD C;
L, LC, T;
JU, JC, JZ, JN, JP, JM, JO, SLW, SRW;
D, I;
C DB, T RS, TNB

System Operations
(for CPU 102 and higher)

For organization blocks (OB)

For program blocks (PB)

For function blocks (FB)

For sequence blocks (SB)

Oper-ation (STL)	Permissible Operanden	1 RLO depend 2 RLO affected 3 RLO reloaded			typical Execution Time in µs					Function
		1	2	3	CPU 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
Set Operations										
SU	RS	N	N	Y	--	--	167		123	Set bit in system data area unconditionally.
RU	RS	N	N	Y	--	--	167		123	Reset bit in system data area unconditionally.
Load and Transfer Operations										
LIR	0 (ACCU 1) 2 (ACCU 2)	N	N	N	--	--	105		76	Load the contents of a memory word (addressed by ACCU 1) indirectly into the register (0: ACCU 1; 2: ACCU 2).
TIR	0 (ACCU 1) 2 (ACCU 2)	N	N	N	--	--	85		61	Transfer the register contents (0: ACCU 1; 2: ACCU 2) indirectly into the memory word (addressed by ACCU 1).
TNB	Parameter n=0 ... 255	N	N	N	--	13+ n·19 (48+ n·19)	97+ n·21		75+ n·16	Transfer a field byte by byte (number of bytes 0 to 255).
T	RS	N	N	N	--	--	71		59	Transfer a word to the system data area.
Block Call Operations and Return Operations										
JU	OB	N	N	Y	--	--	187		3,35	Call an organization block unconditionally.
JC	OB	Y	Y[1]	Y	--	--	194		3,35	Call an organization block conditionally.

[1] RLO is set to ”1”

System Operations
(for CPU 102 and Higher)

For organization blocks (OB)

For program blocks (PB)

For function blocks (FB)

For sequence blocks (SB)

Oper-ation (STL)	Permissible Operanden	1 RLO depend. 2 RLO affected 3 RLO reloaded			typical Execution Time in µs					Function
		1	2	3	CPU 100	CPU 102	CPU 103 MA02		CPU 103 MA03	
Jump Operation										
JUR		N	N	N	--	--	131		82	Jump at random within a function block (jump distance -32768 to + 32767)
Arithmetic Operations										
ADD	BF	N	N	N	--	--	58		35	Add byte constant (fixed point) to ACCU 1.
ADD	KF	N	N	N	--	--	104		68	Add fixed-point constant (word) to ACCU 1.
Other Operations										
STS		N	N	N	--	--				Stop operation. Program processing is interrupted immediately after this operation.
TAK		N	N	N	--	--	74		57	Swap the contents of ACCU 1 and ACCU 2.

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								Oper-ation	Oper-and
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								Oper-ation	Oper-and
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					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	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
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								Oper- ation	Oper- and
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	FY
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

Machine Code								Oper-ation	Oper-and
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	0_e	0_e	0_e	0_e	JC	PB
5	8	0	0					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	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					JUR	
7	0	1	5	C	0	0_o	0_o	TB	C
7	0	1	5	8	0	0_o	0_o	TBN	C
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

Machine Code								Operation	Operand
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
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_d	0_d					L	PB
7	3	0_d	0_d					T	PB
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
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

Machine Code								Oper-ation	Oper-and
B0		B1		B2		B3			
L	R	L	R	L	R	L	R		
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
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	

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

OB-No.	Function	OB integrated in CPU		
		100	102	103
You must program the OB. The operating system calls up the OB.				
OB1	Cyclic program processing	▨	▨	▨
Interrupt-driven program processing				
OB2	Interrupt-driven program processing			▨
OB13	Cyclic program processing			▨
Handling start-up procedures				
OB21	When starting manually (STOP to RUN)	▨	▨	▨
OB22	When power returns	▨	▨	▨
Handling programming errors and device errors				
OB34	Battery failure	▨	▨	▨
The OB is already programmed. You must call up the OB.				
OB31	Scan time triggering			▨
OB251	PID control algorithm			▨

▨ OB is ready or is supported by the operating system

Integral Function Blocks

FB-No.	Function	FB integrated in CPU		
		100	102	103
FB240	4-tetrad BCD code converter		▨	▨
FB241	16-bit fxed-point converter		▨	▨
FB242	16-bit binary multiplier		▨	▨
FB243	16-bit binary divider		▨	▨
FB250	Read analog value		▨	▨
FB251	Output analog value		▨	▨

▨ FB is ready

Integral Data Block 1

Parameter	Argument	Meaning
Block ID: SL1:		**SINEC L1**
SLN	p	Slave number
SF	DBx DWy	Location of Send Mailbox
EF	DBxDWy	Location of Receive Mailbox
KBE	MBy	Location of Coordination Byte “Receive”
KBS	MBy	Location of Coordination Byte “Send”
PGN	p	Programmer bus number
p=1 ... 30	x=2 ... 255	y=0 ... 255
Block ID: SDP:		**S**ystem-**D**ependent-**P**arameter
WD	D	Number of timers being processed (***W***atch-***D***og-***T***imer)
p=1 ... 2550		
Block ID: TFB:		**T**imer-**F**unktions-**B**austein
OB13	p	Intervals (ms) at which OB13 is called up and is processed
p=0 ... 655350 (State in 10 ms steps)		
Block ID: CLP:		**Cl**ock-**P**arameters
CF	p	Inputting the correction factor (***C***orrection ***F***actor)
CLK	DBxDWy,MWz,EWv or AWv	Location of the clock data (***CL***oc***K*** Data)
STW	DBxDWy,MWz,EWv or AWv	Location of the status word (***ST***atus ***W***ord)
STP	J/Y/N	Updating the clock during STOP (***ST***o***P)***
SAV	J/Y/N	Saving the clock time after the last change from RUN to STOP or Power OFF (***SAV***e)
OHE	J/Y/N	Enabling the operating hours counter (***O***peration ***H***our counter ***E***nable)
SET	wd dd.mm.yy hh:mn:ss[1] AM/PM[2]	Setting the clock time and date
TIS	wd dd.mm. hh:mn:ss[1] AM/PM[2]	Setting the prompting time (***T***imer ***I***nterrupt ***S***et)
OHS	hhhhhh:mn:ss[1]	Setting the operating hours counter (***O***peration ***H***our counter ***S***et)

wd=1 ... 7 (weekday=Sun ... Sat)
dd=01 ... 31 (day)
mm=01 ...12 (month)
yy=0 ... 99 (year)
hh=00 ... 23 (hours)
		mn=00 ... 59 (minutes)
		ss=00 ... 59 (seconds)
		hhhhhh=0 ... 999999 (hours)
		p=– 400 ... 400
		v=0 ... 126
		x=2 ... 255
		y=0 ... 255
		z=0 ... 254
		j/J=ja (yes)
		y/Y=yes
		n/N=no

[1] If an argument such as seconds, for example, is not to be entered, input XX. The clock continues to run with the updated data. The TIS parameter block does not acknowledge this argument.

[2] If you input AM or PM after the clock time, the clock runs in the 12-hour mode. If you omit this argument, the clock runs in the 24-hour mode. You must use the same time mode in the SET and TIS parameter blocks.

Evaluation of CC 1 and CC 0

CC 1	CC 0	Arith-metic Oper-ations	Digital Logic Oper-ations	Compa-rison Oper-ations	Shift Oper-ations	Conver-sion Oper-ations
0	0	Result =0	Result =0	ACCU 2 = ACCU 1	shifted bit =0	-
0	1	Result <0	-	ACCU 2 < ACCU 1	-	Result <0
1	0	Result >0	Result 0	ACCU 2 > ACCU 1	shifted bit =1	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 S5-100U (CPU 100/102/103)
Reference Guide

Order No.: 6ES5 997-8MA21

Suggestions / Corrections:

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

Siemens Aktiengesellschaft

Order No. 6ES5 997-8MA21
Printed in the Fed. Rep. of Germany