

Befehlsliste



SAIA® PCD Serie xx7

**Die SIMATIC® S7 kompatible
speicherprogrammierbare Steuerung**



SAIA[®] PCD

Process Control Devices

SAIA-Burgess Electronics



SWITCHES • MOTORS • CONTROLLERS

Serie xx7

Kompatibel zu SIMATIC S7-300/400

Befehlsliste

SIMATIC[®] ist eingetragenes Warenzeichen der Siemens AG

Notizen

Inhaltsverzeichnis

	Seite		Seite
Operanden und Parameterbereiche	2	Quadratwurzel, Quadrat (32 Bit)	47
Abkürzungen	5	Logarithmusfunktionen (32 Bit)	48
Register	6	Trigonometrische Funktionen (32 Bit)	49
Adressierungsbeispiele	8	Addition von Konstanten	50
Operationsliste	10	Addition über Adressregister	51
Verknüpfungsoperationen mit Bitoperanden	10	Vergleichsoperationen (16-Bit-Integerzahlen)	52
Verknüpfungsoperationen von Klammerausdrücken	13	Vergleichsoperationen (32-Bit-Integerzahlen)	53
ODER-Verknüpfung von UND-Funktionen	15	Vergleichsoperationen (32-Bit-Realzahlen)	54
Verknüpfungsoperationen mit Timer und Zählern	16	Schiebeoperationen	55
Verknüpfungsoperationen mit dem Inhalt von AKKU 1	19	Rotieroperationen	57
Verknüpfungsoperationen mit Anzeigenbits	20	AKKU-Transferoperationen, Inkrementieren, Dekrementieren	58
Flankenoperationen	22	Bildaufbauoperation, Nulloperation	59
Setzen/Rücksetzen von Bitoperanden	23	Datentyp-Umwandlungsoperationen	60
VKE direkt beeinflussende Operationen	25	Komplementbildung	63
Timeroperationen	27	Baustein-Aufrufoperationen	64
Zähloperationen	29	Baustein-Endeoperationen	66
Ladeoperationen	30	Tausche Datenbausteine	67
Ladeoperationen für Timer und Zähler	35	Sprungoperationen	68
Transferoperationen	36	Operationen für das Master Control Relay (MCR)	72
Lade- und Transferoperationen für Adressregister	39	Bausteine und Funktionen der CPUs	73
Lade- und Transferoperationen für das Statuswort	41	Funktionsbausteine, Funktionen und Datenbausteine	76
Ladeoperationen für DB-Nummer und DB-Länge	42	Systemfunktionen	77
Festpunktarithmetik (16 Bit)	43	Systemfunktionen SAIA® PCD-Serie xx7	78
Festpunktarithmetik (32 Bit)	44	Systemfunktionsbausteine	79
Gleitpunktarithmetik (32 Bit)	45	Alphabetisches Verzeichnis der Operationen	80

Operanden und Parameterbereiche

Operand	Parameterbereich			Beschreibung
	PCD1. M137	PCD2. M127	PCD2. M157/177 M257	
A	0.0 bis 255.7	0.0 bis 255.7	0.0 bis 255.7	Ausgang (im PAA)
AB	0 bis 255	0 bis 255	0 bis 255	Ausg.byte (im PAA)
AW	0 bis 254	0 bis 254	0 bis 254	Ausg.wort (im PAA)
AD	0 bis 252	0 bis 252	0 bis 252	Ausgangsdoppelwort (im PAA)
DBX	0.0 bis 65535.7	0.0 bis 65535.7	0.0 bis 65535.7	Datenbit im Daten- baustein
DB	1 bis 1023	1 bis 1023	1 bis 1023	Datenbaustein
DBB	0 bis 65535	0 bis 65535	0 bis 65535	Datenbyte im DB
DBW	0 bis 65534	0 bis 65534	0 bis 65534	Datenwort im DB
DBD	0 bis 65532	0 bis 65532	0 bis 65532	Datendoppelwort im DB
DIX	0.0 bis 65533.7	0.0 bis 65533.7	0.0 bis 65533.7	Datenbit im Instanz-DB
DI	1 bis 1023	1 bis 1023	1 bis 1023	Instanz-Datenbaustein
DIB	0 bis 65535	0 bis 65535	0 bis 65535	Datenbyte im Instanz-DB
DIW	0 bis 65534	0 bis 65534	0 bis 65534	Datenwort im Instanz-DB
DID	0 bis 65532	0 bis 65532	0 bis 65532	Datendoppelwort im Instanz-DB

Operand	Parameterbereich			Beschreibung
	PCD1. M137	PCD2. M127	PCD2. M157/177 M257	
E	0.0 bis 255.7	0.0 bis 255.7	0.0 bis 255.7	Eingang (im PAE)
EB	0 bis 255	0 bis 255	0 bis 255	Eing.byte (im PAA)
EW	0 bis 254	0 bis 254	0 bis 254	Eing.wort (im PAA)
ED	0 bis 252	0 bis 252	0 bis 252	Eingangsdoppelwort (im PAE)
L	0.0 bis 255.7	0.0 bis 511.7	0.0 bis 511.7	Lokaldaten
LB	0 bis 255	0 bis 511	0 bis 511	Lokaldatenbyte
LW	0 bis 254	0 bis 510	0 bis 510	Lokaldatenwort
LD	0 bis 252	0 bis 508	0 bis 508	Lokaldatendoppelwort
M	0.0 bis 2047.7	0.0 bis 2047.7	0.0 bis 2047.7	Merker
MB	0 bis 2047	0 bis 2047	0 bis 2047	Merkerbyte
MW	0 bis 2046	0 bis 2046	0 bis 2046	Merkerwort
MD	0 bis 2044	0 bis 2044	0 bis 2044	Merkerdoppelwort

*) Zusätzlich begrenzt durch die Grösse des Arbeitsspeichers

Operanden und Parameterbereiche, Fortsetzung

Operand	Parameterbereich			Beschreibung
	PCD1. M137	PCD2. M127	PCD2. M157/177 M257	
PAB	0 bis 65535	0 bis 65535	0 bis 65535	Peripherieausgangs- byte (direkter Peripheriezugriff)
PAW	0 bis 65534	0 bis 65534	0 bis 65534	Peripherieausgangs- wort (direkter Peripheriezugriff)
PAD	0 bis 65532	0 bis 65532	0 bis 65532	Peripherieausgangs- Doppelwort (direkter Peripheriezugriff)
PEB	0 bis 65535	0 bis 65535	0 bis 65535	Peripherieeingangs- byte (direkter Peripheriezugriff)
PEW	0 bis 65534	0 bis 65534	0 bis 65534	Peripherieeingangs- wort (direkter Peripheriezugriff)
PED	0 bis 65532	0 bis 65532	0 bis 65532	Peripherieeingangs- Doppelwort (direkter Peripheriezugriff)
T	0 bis 255	0 bis 255	0 bis 255	Timer (Zeiten)
Z	0 bis 255	0 bis 255	0 bis 255	Zähler

Konstante	Wertebereich	Beschreibung
B (b1, b2) B (b1, b2, b3, b4)		Konstante, 2 oder 4 Byte
D# Datum		IEC-Datumskonstante
L# Integer		32-Bit-Integer-Konstante
P# Bitpointer		Pointerkonstante
S5T# Zeitwert		S7-Zeitkonstante *)
T# Zeitwert		Zeitkonstante
TOD# Zeitwert		IEC-Zeitkonstante
C# Zählwert		Zählerkonstante (BCD-codiert)
2#n		Binärkonstante
W#16# DW#16#		Hexadezimalkonstante

*) dient zum Laden der S7-Timer

Notizen

Abkürzungen

Abkürzung	... steht für	Beispiel
k8	8-Bit-Konstante 0 bis 255	32
k16	16-Bit-Konstante 0 bis 65 535	62 531, 0010 0111 0010 1100
k 32	32-Bit-Konstante 0 bis 4 294 967 295	127 624
i8	8-Bit-Integer -128 bis +127	-113
i16	16-Bit-Integer -32768 bis +32767	+6523
i32	32-Bit-Integer - 2 147 483 648 bis +2 147 483 647	-2 222 222
m	Pointer-Konstante	P#240.3
n	Binärkonstante	1001 1100
p	Hexadezimalkonstante	EA12
MARKE	symbolische Sprung- adresse (max 4 Buchstaben)	ZIEL

Abkürzung	... steht für	Beispiel
a	Byteadresse	
b	Bitadresse	
c	Operandenbereich	E, A, M, L, DBX, DIX
d	Adresse steht im: MD, DBD, DID oder LD	
e	Nummer steht im: MW, DBW, DIW oder LW	
f	Timer-/Zähler-Nr.	
g	Operandenbereich	EB, AB, PEB, MB, LB, DBB, DIB
h	Operandenbereich	EW, AW, PEW, MW, LW, DBW, DIW
i	Operandenbereich	ED, AD, PED, MD, LD, DBD, DID
q	Baustein-Nr.	

Die aufgelisteten Abkürzungen werden in der Operationsliste verwendet.

Register

AKKU1 bis AKKU4 (32 Bit)

Die AKKUs sind Register für die Verarbeitung von Bytes, Worten oder Doppelworten. Dazu werden die Operanden in die AKKUs geladen und dort verknüpft. Das Ergebnis der Operation steht immer im AKKU1 und kann von dort in eine Speicherzelle transferiert werden.

Die AKKUs sind 32 Bit breit.

AKKU	Bit
AKKU _x (x = 1 bis 4)	Bit 0 bis 31
AKKU _x -L	Bit 0 bis 15
AKKU _x -H	Bit 16 bis 31
AKKU _x -LL	Bit 0 bis 7
AKKU _x -LH	Bit 8 bis 15
AKKU _x -HL	Bit 16 bis 23
AKKU _x -HH	Bit 24 bis 31

Statuswort (16 Bit)

Die Anzeigen werden durch die Operationen ausgewertet oder gesetzt bzw. rückgesetzt.

Das Statuswort ist 16 Bit breit.

Bit	Belegung	Bedeutung
0	/ER	Erstabfrage
1	VKE	Verknüpfungsergebnis
2	STA	Status
3	OR	Oder (und-vor-oder)
4	OS	Überlauf speichernd
5	OV	Überlauf
6	A0	Ergebnisanzeige 0
7	A1	Ergebnisanzeige 1
8	BIE	Binärerergebnis
9 bis 15	nicht belegt	-

Register, Fortsetzung

Adressregister AR1 und AR2 (32 Bit)

Die Adressregister enthalten die bereichsinternen oder bereichsübergreifenden Zeiger für die registerindirekt adressierenden Operationen. Die Adressregister sind 32 Bit breit

Die bereichsinternen bzw. bereichsübergreifenden Zeiger haben folgenden Aufbau:

- bereichsinterner Zeiger:

00000000 00000bbb bbbbbbbb bbbbxxxx

- bereichsübergreifender Zeiger:

yyyyyyyy 00000bbb bbbbbbbb bbbbxxxx

Legende: b Byteadresse
 x Bitnummer
 y Bereichskennung
 (siehe Kapitel Adressierungsbeispiele)

Adressierungsbeispiele

Adressierungsbeispiele	Beschreibung
Unmittelbare Adressierung	
L +27	Lade 16-Bit-Integer-Konstante "27" in AKKU1
L L#-1	Lade 32-Bit-Integer-Konstante "-1" in AKKU1
L 2#1010101010101010	Lade Binärkonstante in AKKU1
L DW#16#A0F0BCFD	Lade Hexadezimalkonstante in AKKU1
L 'ENDE'	Lade ASCII-Zeichen in AKKU1
L T#500 ms	Lade Zeitwert in AKKU1
L C#100	Lade Zeitwert in AKKU1
L B#(100,12)	Lade Konstante als 2 Byte
L B#(100,12,50,8)	Lade Konstante als 4 Byte
L P#10.0	Lade bereichsinternen Pointer in AKKU1
L P#E20.6	Lade bereichübergreifenden Pointer in AKKU1
L -2.5	Lade Realzahl in AKKU1
L D# 1997-09-02	Lade Datum
L TOD 13:20:33.125	Lade Uhrzeit

Adressierungsbeispiele	Beschreibung
Direkte Adressierung	
U E0.0	UND-Verkn. des Eing.bits 0.0
L EB 1	Lade Eingangsbyte 1 in AKKU1
L EW 0	Lade Eingangswort 0 in AKKU1
L ED 0	Lade Eing.doppelw. 0 in AKKU1
Indirekte Adressierung Timer/Zähler	
SI T [LW 8]	Starte Timer; die Timer-Nr. steht im Lokaldatenwort 8
ZV Z [LW 10]	Zähle vorwärts; Die Zähler-Nr. steht im Lokaldatenwort 10
Speicherindirekte, bereichsinterne Adressierung	
U E [LD 12] Beispiel: L P#22.2 T LD 12 U E [LD 12]	UND-Operation; die Adresse des Eingangs steht als Pointer im Lokaldaten-Doppelwort 12
U E [DBD 1]	UND-Op.; die Adr. des Eingangs steht als P. im Daten-Doppelw. 1 des aufgeschlagenen DB
U A [DID 1]	UND-Op. ; die Adr. des Ausg. steht als P. im Datendoppelw 12 des aufgeschlagenen Instanz-DB
U A [MD 12]	UND-Operation; die Adresse des Ausgangs steht als Pointer im Merkerdoppelwort 12

Adressierungsbeispiele, Fortsetzung

Adressierungsbeispiele Beschreibung			
Registerindirekte, bereichsinterne Adressierung			
U E [AR1,P#12.2]		UND-Operation; die Adresse des Eingangs errechnet sich zu "Pointerwert im Adressregister 1 + Pointer P#12.2"	
Registerindirekte, bereichsübergreifende Adressierung			
Für die bereichsübergreifende, registerindirekte Adressierung muss die Adresse zusätzlich eine Bereichskennung enthalten. Die Adresse steht im Adressregister. Es gibt folgende Bereichskennungen:			
Bereichskennung	Codierung		Bereich
	binär	hex.	
P	1000 0000	80	Peripheriebereich
E	1000 0001	81	Eingangsbereich
A	1000 0010	82	Ausgangsbereich
M	1000 0011	83	Merkerbereich
DB	1000 0100	84	Datenbereich
DI	1000 0101	85	Instanz-Datenbereich
L	1000 0110	86	Lokaldaten
VL	1000 0111	87	Vorgänger-Lokaldatenbereich. (Zugriff auf Lokaldaten des aufzufolgenden Bausteins)

Adressierungsbeispiele	Beschreibung
Registerindir., bereichsübergreifende Adressierung, Fortsetzung	
L B [AR1,P#8.0]	Lade Byte in AKKU1; die Adresse errechnet sich aus "Pointerwert im Adressregister 1 + Pointer P#8.0"
U [AR1,P#32.3]	UND-Operation; die Adresse des Operanden errechnet sich aus "Pointerwert im Adressregister 1 + Pointer P#32.3"
Adressierung über Parameter	
U Parameter	Der Operand wird über den Parameter adressiert

Beispiel zur Pointerberechnung

- Beispiel bei Summe der Bitadresse ≤ 7 :
LAR1 P#8.2
U E [AR1,P#10.2]
Ergebnis: Adressiert wird Eingang 18.4 (durch jeweilige Addition der Byte- und Bitadressen)
- Beispiel bei Summe der Bitadressen > 7 :
L P#10.5
LAR1
U E [AR1,P#10.7]
Ergebnis: Adressiert wird Eingang 21.4 (durch Addition der Byte- und Bitadressen mit Übertrag).

Verknüpfungsoperationen mit Bitoperanden

U/UN

Befehl	Operand	Erläuterung
U/UN		UND/UND-NICHT
	E/A a,b	Eingang/Ausgang
	M a,b	Merker
	L a,b	Lokaldatenbit
	DBX a,b	Datenbit
	DIX a,b	Instanz-Datenbit
	c [d]	speicherindirekt, bereichsintern
	c [AR1,m]	registerindirekt, bereichsintern (AR1)
	c [AR2,m]	registerindirekt, bereichsintern (AR2)
	[AR1,m]	bereichsübergreifend (AR1)
	[AR2,m]	bereichsübergreifend (AR2)
	Parameter	über Parameter

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	ja	-	ja	ja
Operation beeinflusst:	-	-	-	-	-	ja	ja	ja	1

Alle Verknüpfungsoperationen (VKO) bilden ein Verknüpfungsergebnis (VKE-neu). Die erste VKO einer Verknüpfungskette bildet das VKE-neu aus dem abgefragten Signalzustand. Die nun folgenden VKOs bilden das VKE-neu aus dem abgefragten Signalzustand und dem VKE-alt. Die Verknüpfungskette wird durch eine Operation abgeschlossen, die das VKE begrenzt (z.B. Speicheroperation).

Verknüpfungsoperationen mit Bitoperanden, Fortsetzung

O/ON

Befehl	Operand	Erläuterung
O/ON		ODER/ODER-NICHT
	E/A a,b	Eingang/Ausgang
	M a,b	Merker
	L a,b	Lokaldatenbit
	DBX a,b	Datenbit
	DIX a,b	Instanz-Datenbit
	c [d]	speicherindirekt, bereichsintern
	c [AR1,m]	registerindirekt, bereichsintern (AR1)
	c [AR2,m]	registerindirekt, bereichsintern (AR2)
	[AR1,m]	bereichsübergreifend (AR1)
	[AR2,m]	bereichsübergreifend (AR2)
	Parameter	über Parameter

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	ja	ja
Operation beeinflusst:	-	-	-	-	-	0	ja	ja	1

Verknüpfungsoperationen mit Bitoperanden, Fortsetzung

X/XN

Befehl	Operand	Erläuterung
X/XN		EXKLUSIV-ODER/ EXKLUSIV-ODER-NICHT
	E/A a, b	Eingang/Ausgang
	M a, b	Merker
	L a, b	Lokaldatenbit
	DBX a,b	Datenbit
	DIX a, b	Instanz-Datenbit
	c [d]	speicherindirekt, bereichsintern
	c [AR1,m]	registerindirekt, bereichsintern (AR1)
	c [AR2,m]	registerindirekt, bereichsintern (AR2)
	[AR1,m]	bereichsübergreifend (AR1)
	[AR2,m]	bereichsübergreifend (AR2)
	Parameter	über Parameter

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	ja	ja
Operation beeinflusst:	-	-	-	-	-	0	ja	ja	1

Verknüpfungsoperationen von Klammerausdrücken

U(, UN(, O(, ON(, X(, XN(

Befehl	Operand	Erläuterung
U(UND-Klammer-Auf
UN(UND-NICHT-Klammer-Auf
O(ODER-Klammer-Auf
ON(ODER-NICHT-Klammer-Auf
X(EXKLUSIV-ODER-Klammer-Auf
XN(EXKLUSIV-ODER-NICHT-Klammer-Auf

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	ja	-	ja	ja
Operation beeinflusst:	-	-	-	-	-	0	1	-	0

Retten der Bits VKE, OR und der entsprechenden Funktionskennung (U, UN, ...) auf den Klammerstack. 7 Klammerebenen sind pro Baustein möglich. Nach "Klammer zu" wird das gerettete VKE mit dem aktuellen VKE verknüpft, entsprechend der Funktionskennung; das aktuelle OR wird vom geretteten OR überschrieben.

Verknüpfungsoperationen von Klammersausdrücken, Fortsetzung

)

Befehl	Operand	Erläuterung
)		Klammer zu, Entfernen eines Eintrags vom Klammerstack.

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	ja	-
Operation beeinflusst:	-	-	-	-	-	ja	1	ja	1

ODER-Verknüpfung von UND-Funktionen

O

Befehl	Operand	Erläuterung
O		ODER-Verknüpfung von UND-Funktionen nach der Regel: UND-vor-ODER.

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	ja	ja
Operation beeinflusst:	-	-	-	-	-	ja	1	-	ja

Es erfolgt die ODER-Verknüpfung von UND-Funktionen nach der Regel: UND vor ODER.

Verknüpfungsoperationen mit Timern und Zählern

U/UN

Befehl	Operand	Erläuterung
U/UN		UND/UND-NICHT
	T f	Timer
	T [e]	Timer, speicherindirekt adressiert
	Z f	Zähler
	Z [e]	Zähler, speicherindirekt adressiert
	Timerparameter Zählerparameter	Timer/Zähler (über Parameter adressiert)

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	ja	-	ja	ja
Operation beeinflusst:	-	-	-	-	-	ja	ja	ja	1

Abfrage des adressierten Timer/Zähler auf den Zustand. Das Ergebnis der Abfrage wird nach der entsprechenden Funktion mit dem VKE verknüpft.

Verknüpfungsoperationen mit Timern und Zählern, Fortsetzung

O/ON

Befehl	Operand	Erläuterung
O/ON		ODER/ODER-NICHT
	T f	Timer
	T [e]	Timer, speicherindirekt adressiert
	Z f	Zähler
	Z [e]	Zähler, speicherindirekt adressiert
	Timerparameter Zählerparameter	Timer/Zähler (über Parameter adressiert)

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	ja	ja
Operation beeinflusst:	-	-	-	-	-	0	ja	ja	1

Abfrage des adressierten Timer/Zähler auf den Zustand. Das Ergebnis der Abfrage wird nach der entsprechenden Funktion mit dem VKE verknüpft.

Verknüpfungsoperationen mit Timern und Zählern, Fortsetzung

X/XN

Befehl	Operand	Erläuterung
X/XN		EXKLUSIV-ODER/ EXKLUSIV-ODER-NICHT
	T f	Timer
	T [e]	Timer, speicherindirekt adressiert
	Z f	Zähler
	Z [e]	Zähler, speicherindirekt adressiert
	Timerparameter	EXKLUSIV-ODER Timer/Zähler
	Zählerparameter	(über Parameter adressiert)

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	ja	ja
Operation beeinflusst:	-	-	-	-	-	0	ja	ja	1

Abfrage des adressierten Timer/Zähler auf den Zustand. Das Ergebnis der Abfrage wird nach der entsprechenden Funktion mit dem VKE verknüpft.

Verknüpfungsoperationen mit dem Inhalt von AKKU1

UW, OW, XOW, UD, OD, XOD

Befehl	Operand	Erläuterung
UW		UND AKKU2-L
UW	W#16#p	UND 16-Bit-Konstante
OW		ODER AKKU2-L
OW	W#16#p	ODER 16-Bit-Konstante
XOW		EXKLUSIV-ODER AKKU2-L
XOW	W#16#p	EXKLUSIV-ODER 16-Bit-Konstante
UD		UND AKKU2
UD	DW#16#p	UND 32-Bit-Konstante
OD		ODER AKKU2
OD	DW#16#p	ODER 32-Bit-Konstante
XOD		EXKLUSIV-ODER AKKU2
XOD	DW#16#p	EXKLUSIV-ODER 32-Bit-Konstante

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	ja	0	0	-	-	-	-	-

Verknüpfung des Inhalts von AKKU1 bzw. AKKU1-L mit einem Wort bzw. einem Doppelwort nach der entsprechenden Funktion. Das Wort bzw. Doppelwort wird entweder bei der Operation als Operand oder im AKKU2 mit angegeben. Das Ergebnis steht im AKKU1 bzw. AKKU1-L

Verknüpfungsoperationen mit Anzeigenbits

U/UN, O/ON, X/XN

Befehl	Operand	Erläuterung
U/UN O/ON X/XN		UND/UND-NICHT, ODER/ODER-NICHT, EXKLUSIV-ODER / EXKLUSIV-ODER-NICHT
	== 0	Ergebnis = 0 (A1 = 0 und A0 = 0)
	> 0	Ergebnis > 0 (A1 = 1 und A0 = 0)
	< 0	Ergebnis < 0 (A1 = 0 und A0 = 1)
	<> 0	Ergebnis ≠ 0 ((A1 = 0 und A0 = 1) oder (A1 = 1 und A0 = 0))
	<= 0	Ergebnis <= 0 ((A1 = 0 und A0 = 1) oder (A1 = 0 und A0 = 0))
	>= 0	Ergebnis >= 0 ((A1 = 1 und A0 = 0) oder (A1 = 0 und A0 = 0))

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	ja	ja	-	-	ja	-	ja	ja
Operation beeinflusst:	-	-	-	-	-	ja	ja	ja	1

Alle Verknüpfungsoperationen (VKO) bilden ein Verknüpfungsergebnis (VKE-neu). Die erste VKO einer Verknüpfungskette bildet das VKE-neu aus dem abgefragten Signalzustand. Die nun folgenden VKOs bilden das VKE-neu aus dem abgefragten Signalzustand und dem VKE-alt. Die Verknüpfungskette wird durch eine Operation abgeschlossen, die das VKE begrenzt (z.B. Speicheroperation).

Verknüpfungsoperationen mit Anzeigenbits, Fortsetzung

U/UN, O/ON, X/XN

Befehl	Operand	Erläuterung
U/UN O/ON X/XN	UO	UND/UND-NICHT, ODER/ODER-NICHT, EXKLUSIV-ODER / EXKLUSIV-ODER-NICHT unordered/unzulässige Arithmetikoperation (A1 = 1 und A0 = 1)
	OS	UND OS = 1
	BIE	UND BIE = 1
	OV	UND OV = 1

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	ja	ja	ja	ja	ja	ja	-	ja	ja
Operation beeinflusst:	-	-	-	-	-	ja	ja	ja	1

Flankenoperationen

FP/FN

Befehl	Operand	Erläuterung
FP/FN	E/A a,b M a,b L a,b *) DBX a,b DIX a,b c [d] c [AR1,m] c [AR2,m] [AR1,m] [AR2,m] Parameter	Anzeigen der steigenden/fallenden Flanke mit VKE = 1. Flankenhilfsmerker ist das in der Operation adressierte Bit.

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	ja	-
Operation beeinflusst:	-	-	-	-	-	0	ja	ja	1

Das aktuelle VKE wird verglichen mit dem Status des Operanden, dem "Flankenmerker". FP erkennt einen Flankenwechsel von "0" nach "1". FN erkennt einen Flankenwechsel von "1" nach "0".

*) nicht sinnvoll, falls zu überwachendes Bit im Prozessabbild liegt (Lokaldaten eines Bausteins sind nur zu dessen Laufzeit gültig.)

Setzen/Rücksetzen von Bitoperanden

S, R

Befehl	Operand	Erläuterung
S		Setze adressiertes Bit auf "1"
R		Setze adressiertes Bit auf "0"
	E/A a,b	Eingang/Ausgang
	M a,b	Merker
	L a,b	Lokaldatenbit
	DBX a,b	Datenbit
	DIX a,b	Instanz-Datenbit
	c [d]	speicherindirekt, bereichsintern
	c [AR1,m]	registerindirekt, bereichsintern (AR1)
	c [AR2,m]	registerindirekt, bereichsintern (AR2)
	[AR1,m]	bereichsübergreifend (AR1)
	[AR2,m]	bereichsübergreifend (AR2)
	Parameter	über Parameter

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	ja	-
Operation beeinflusst:	-	-	-	-	-	0	ja	-	0

Zuweisen des Wertes "1" bzw. "0" an den adressierten Operanden, wenn VKE = 1. MCR-Abhängigkeit beachten. (siehe Seite 72)

Setzen/Rücksetzen von Bitoperanden, Fortsetzung

=

Befehl	Operand	Erläuterung
=		Zuweisen des VKE
	E/A a,b	Eingang/Ausgang
	M a,b	Merker
	L a,b	Lokaldatenbit
	DBX a,b	Datenbit
	DIX a,b	Instanz-Datenbit
	c [d]	speicherindirekt, bereichsintern
	c [AR1,m]	registerindirekt, bereichsintern (AR1)
	c [AR2,m]	registerindirekt, bereichsintern (AR2)
	[AR1,m]	bereichsübergreifend (AR1)
	[AR2,m]	bereichsübergreifend (AR2)
	Parameter	über Parameter

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	ja	-
Operation beeinflusst:	-	-	-	-	-	0	ja	-	0

Der VKE-Wert wird in den adressierten Operanden geschrieben. MCR-Abhängigkeit beachten (siehe Seite 72).

VKE direkt beeinflussende Operationen

CLR, SET

Befehl	Operand	Erläuterung
CLR		Setze VKE auf "0"

Statuswort										
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	
Operation wertet aus:	-	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	0	0	0	0	0

Befehl	Operand	Erläuterung
SET		Setze VKE auf "1"

Statuswort										
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	
Operation wertet aus:	-	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	0	1	1	0	0

Diese Operationen bearbeiten direkt das VKE.

VKE direkt beeinflussende Operationen, Fortsetzung

NOT, SAVE

Befehl	Operand	Erläuterung
NOT		Negiere das VKE

Statuswort										
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	
Operation wertet aus:	-	-	-	-	-	ja	-	ja	-	
Operation beeinflusst:	-	-	-	-	-	-	1	ja	-	

Befehl	Operand	Erläuterung
SAVE		Rette das VKE in das BIE-Bit

Statuswort										
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	
Operation wertet aus:	-	-	-	-	-	-	-	ja	-	
Operation beeinflusst:	ja	-	-	-	-	-	-	-	-	

Diese Operationen bearbeiten direkt das VKE.

Timeroperationen

SI, SV, SE, SS

Befehl	Operand	Erläuterung
SI	T f T [e]	Starte Timer als Impuls bei Flankenwechsel von "0" nach "1".
	Timerparameter	
SV	T f T [e]	Starte Timer als verlängerten Impuls bei Flankenwechsel von "0" nach "1".
	Timerparameter	
SE	T f T [e]	Starte Timer als speichernde Einschaltverzögerung bei Flankenwechsel von "0" nach "1".
	Timerparameter	
SS	T f T [e]	Starte Timer als speichernde Einschaltverzögerung bei Flankenwechsel von "0" nach "1".
	Timerparameter	

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	ja	-
Operation beeinflusst:	-	-	-	-	-	0	-	-	0

Starten bzw. Rücksetzen eines Timers. Die Zeitdauer muss im AKKU1-L stehen. Die Operationen werden durch einen Flankenwechsel am VKE ausgelöst. Das heisst, wenn das VK zwischen zwei Aufrufen seinen Zustand geändert hat, wird die Operation ausgelöst.

Timeroperationen, Fortsetzung

SA, FR, R

Befehl	Operand	Erläuterung
SA	T f T [e]	Starte Timer als Ausschaltverzögerung bei Flankenwechsel von "1" nach "0".
	Timerparameter	
FR	T f T [e]	Freigabe eines Timers für das erneute Starten bei Flankenwechsel von "0" nach "1" (Löschen des Flankenmerkers für das Starten der Zeit).
	Timerparameter	
R	T f T [e]	Rücksetzen einer Zeit
	Timerparameter	

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	ja	-
Operation beeinflusst:	-	-	-	-	-	0	-	-	0

Starten bzw. Rücksetzen eines Timers. Die Zeitdauer muss im AKKU1-L stehen. Die Operationen werden durch einen Flankenwechsel am VKE ausgelöst. Das heisst, wenn das VK zwischen zwei Aufrufen seinen Zustand geändert hat, wird die Operation ausgelöst.

Zähloperationen

S, R, ZV, ZR, FR

Befehl	Operand	Erläuterung
S	Z f	Vorbelegen eines Zählers bei
	Z [e] Zählerparameter	Flankenwechsel von "0" nach "1".
R	Z f	Rücksetzen des Zählers auf "0"
	Z [e] Zählerparameter	bei VKE = "1"
ZV	Z f	Zähle um 1 vorwärts bei
	Z [e] Zählerparameter	Flankenwechsel von "0" nach "1".
ZR	Z f	Zähle um 1 rückwärts bei
	Z [e] Zählerparameter	Flankenwechsel von "0" nach "1".
FR	Z f	Freigabe eines Zählers bei Flankenwechsel
	Z [e] Zählerparameter	von "0" nach "1". (Löschen des Flankenmerkers für Vorwärts-, Rückwärtszählen und setzen eines Zählers)

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	ja	-
Operation beeinflusst:	-	-	-	-	-	0	-	-	0

Der Zählwert muss im AKKU1-L als BCD-Zahl (0 - 999) vorliegen.

Ladeoperationen

L

Befehl	Operand	Erläuterung
L		Lade ...
	EB a	Eingangsbyte
	AB a	Ausgangsbyte
	PEB a	Peripherie-Eingangsbyte
	MB a	Merkerbyte
	LB a	Lokaldatenbyte
	DBB a	Datenbyte
	DIB a	Instanz-Datenbyte ... in AKKU1
	g [d]	speicherindirekt, bereichsintern
	g [AR1,m]	registerindirekt, bereichsintern (AR1)
	g [AR2,m]	registerindirekt, bereichsintern (AR2)
	B[AR1,m]	bereichsübergreifend (AR1)
	B[AR2,m]	bereichsübergreifend (AR2)
Parameter	über Parameter	

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Laden der Operanden in AKKU1. Zuvor wird der Inhalt von AKKU1 in AKKU2 gerettet. Das Statuswort wird nicht beeinflusst.

Ladeoperationen, Fortsetzung

L

Befehl	Operand	Erläuterung
L		Lade ...
	EW a	Eingangswort
	AW a	Ausgangswort
	PEW a	Peripherie-Eingangswort
	MW a	Merkerwort
	LW a	Lokaldatenwort
	DBW a	Datenwort
	DIW a	Instanz-Datenwort ... in AKKU1-L
	h [d]	speicherindirekt, bereichsintern
	h [AR1,m]	registerindirekt, bereichsintern (AR1)
	h [AR2,m]	registerindirekt, bereichsintern (AR2)
	W[AR1,m]	bereichsübergreifend (AR1)
	W[AR2,m]	bereichsübergreifend (AR2)
	Parameter	über Parameter

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Ladeoperationen, Fortsetzung

L

Befehl	Operand	Erläuterung
L		Lade ...
	ED a	Eingangsdoppelwort
	AD a	Ausgangsdoppelwort
	PED a	Peripherie-Eingangsdoppelwort
	MD a	Merkerdoppelwort
	LD a	Lokaldatendoppelwort
	DBD a	Datendoppelwort
	DID a	Instanz-Datendoppelwort ... in AKKU1
	i [d]	speicherindirekt, bereichsintern
	i [AR1,m]	registerindirekt, bereichsintern (AR1)
	i [AR2,m]	registerindirekt, bereichsintern (AR2)
	D[AR1,m]	bereichsübergreifend (AR1)
	D[AR2,m]	bereichsübergreifend (AR2)
	Parameter	über Parameter

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Ladeoperationen, Fortsetzung

L

Befehl	Operand	Erläuterung
L		Lade ...
	k8	8-Bit-Konstante in AKKU1-LL
	k16	16-Bit-Konstante in AKKU1-L
	k32	32-Bit-Konstante in AKKU1
	Parameter	Lade Konstante in AKKU1 (über Parameter adressiert)
L	2#n	Lade 16-Bit-Binärkonstante in AKKU1-L Lade 32-Bit-Binärkonstante in AKKU1
	B#16#p	Lade 8-Bit-Hexadezimal konst. in AKKU1-L
L	W#16#p	Lade 16-Bit-Hexadezimal konst. in AKKU1-L
	DW#16#p	Lade 32-Bit-Hexadezimal konst. in AKKU1

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Ladeoperationen, Fortsetzung

L

Befehl	Operand	Erläuterung
L	'x'	Lade 1 Zeichen
	'xx'	Lade 2 Zeichen
	'xxx'	Lade 3 Zeichen
	'xxxx'	Lade 4 Zeichen
L	D# Zeitwert	Lade IEC-Datumskonstante
L	S5T# Zeitwert	Lade S7-Zeitkonstante (16-Bit)
L	TOD# Zeitwert	Lade IEC-Zeitkonstante
L	T# Zeitwert	Lade 16-Bit-Zeitkonstante
		Lade 32-Bit-Zeitkonstante
L	C# Zählwert	Lade Zählerkonstante (BCD-kodiert)
L	B# (b1, b2)	Lade Konstante als Byte (b1, b2)
	B# (b1,b2,b3,b4)	Lade Konstante als 4 Byte (b1, b2, b3, b4)
L	P# Bitpointer	Lade Bitpointer
L	L# Integerzahl	Lade 32-Bit-Integerkonstante
L	Realzahl	Lade Gleitpunktzahl

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Ladeoperationen für Timer und Zähler

L, LC

Befehl	Operand	Erläuterung
L	T f T (e)	Lade Zeitwert
	Timerparameter	Lade Zeitwert (über Parameter adressiert)
L	Z f Z (e)	Lade Zählwert
	Zählerparameter	Lade Zählwert (über Parameter adressiert)
LC	T f T (e)	Lade Zeitwert BCD-codiert
	Timerparameter	Lade Zeitwert BCD-codiert (über Parameter adressiert)
LC	Z f Z (e)	Lade Zählwert BCD-codiert
	Zählerparameter	Lade Zählwert BCD-codiert (über Parameter adressiert)

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Laden eines Zeitwertes oder Zählwertes in AKKU1. Zuvor wird der Inhalt von AKKU1 in AKKU2 gerettet. Das Statuswort wird nicht beeinflusst.

Transferoperationen

T

Befehl	Operand	Erläuterung
T		Transferiere Inhalt von AKKU1-LL zum ...
	EB a	Eingangsbyte
	AB a	Ausgangsbyte
	PAB a	Peripherie-Ausgangsbyte
	MB a	Merkerbyte
	LB a	Lokaldatenbyte
	DBB a	Datenbyte
	DIB a	Instanz-Datenbyte
	g [d]	speicherindirekt, bereichsintern
	g [AR1,m]	registerindirekt, bereichsintern (AR1)
	g [AR2,m]	registerindirekt, bereichsintern (AR2)
	B [AR1,m]	bereichübergreifend (AR1)
	B [AR2,m]	bereichübergreifend (AR2)
	Parameter	über Parameter

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Transferieren des Inhalts von AKKU1 in den adressierten Operanden. MCR-Abhängigkeit beachten (siehe Seite 72). Das Statuswort wird nicht beeinflusst.

Transferoperationen, Fortsetzung

T

Befehl	Operand	Erläuterung
T		Transferiere Inhalt von AKKU1-L zum ...
	EW a	Eingangswort
	AW a	Ausgangswort
	PAW a	Peripherie-Ausgangswort
	MW a	Merkerwort
	LW a	Lokaldatenwort
	DBW a	Datenwort
	DIW a	Instanz-Datenwort
	h [d]	speicherindirekt, bereichsintern
	h [AR1,m]	registerindirekt, bereichsintern (AR1)
	h [AR2,m]	registerindirekt, bereichsintern (AR2)
	W [AR1,m]	bereichübergreifend (AR1)
	W [AR2,m]	bereichübergreifend (AR2)
	Parameter	über Parameter

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Transferoperationen, Fortsetzung

T

Befehl	Operand	Erläuterung
T		Transferiere Inhalt von AKKU1 zum ...
	ED a	Eingangswort
	AD a	Ausgangswort
	PAD a	Peripherie-Ausgangswort
	MD a	Merkerwort
	LD a	Lokaldatenwort
	DBD a	Datenwort
	DID a	Instanz-Datenwort
	i [d]	speicherindirekt, bereichsintern
	i [AR1,m]	registerindirekt, bereichsintern (AR1)
	i [AR2,m]	registerindirekt, bereichsintern (AR2)
	D [AR1,m]	bereichübergreifend (AR1)
	D [AR2,m]	bereichübergreifend (AR2)
	Parameter	über Parameter

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Wenn der Zugriff auf ungerade Adressen erfolgt, MCR-Abhängigkeit beachten (siehe Seite 72).

Lade- und Transferoperationen für Adressregister

LAR1, LAR2

Befehl	Operand	Erläuterung
LAR1	-	Lade Inhalt aus ... AKKU1
	AR2	Adressregister 2
	DBD a	Datendoppelwort
	DID a	Instanz-Datendoppelwort
	m	32-Bit-Konstante aus Pointer
	LD a	Lokaldatendoppelwort
	MD a	Merkerdoppelwort ... in AR1
LAR2	-	Lade Inhalt aus ... AKKU1
	DBD a	Datendoppelwort
	DID a	Instanz-Datendoppelwort
	m	32-Bit-Konstante als Pointer
	LD a	Lokaldatendoppelwort
	MD a	Merkerdoppelwort ... in AR2

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Laden eines Doppelwortes aus einem Speicher oder einem Register in Adressregister 1 (AR1) oder Adressregister 2 (AR2). Das Statuswort wird nicht beeinflusst.

Lade- und Transferoperationen für Adressregister, Fortsetzung

TAR1, TAR2, TAR

Befehl	Operand	Erläuterung
TAR1	-	Lade Inhalt aus AR1 in ... AKKU1
	AR2	Adressregister 2
	DBD a	Datendoppelwort
	DID a	Instanz-Datendoppelwort
	m	32-Bit-Konstante aus Pointer
	LD a	Lokaldatendoppelwort
	MD a	Merkerdoppelwort
TAR2	-	Lade Inhalt aus AR2 in ... AKKU1
	DBD a	Datendoppelwort
	DID a	Instanz-Datendoppelwort
	m	32-Bit-Konstante als Pointer
	LD a	Lokaldatendoppelwort
	MD a	Merkerdoppelwort
TAR		Tausche die Inhalte von AR1 und AR2

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Transferieren eines Doppelwortes aus Adressregister 1 (AR1) oder Adressregister 2 (AR2) in einen Speicher oder ein Register. Zuvor wird der Inhalt von AKKU1 in AKKU2 gerettet. Das Statuswort wird nicht beeinflusst.

Lade- und Transferoperationen für das Statuswort

L STW, T STW

Befehl	Operand	Erläuterung
L	STW	Lade Statuswort in AKKU1

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	ja	ja	ja	ja	ja	ja	ja	ja	ja
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

T	STW	Transferiere AKKU1 (Bits 0 bis 8) in das Statuswort
---	-----	--

Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	ja	ja	ja	ja	ja	ja	ja	ja	ja

Ladeoperationen für DB-Nummer und DB-Länge

L

Befehl	Operand	Erläuterung
L	DBNO	Lade Nummer des Datenbausteins
L	DINO	Lade Nummer des Instanz-Datenbausteins
L	DBLG	Lade Länge des Datenbausteins in Byte
L	DILG	Lade Länge des Instanz-Datenbausteins in Byte

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Laden der Nummer/Länge eines Datenbausteins in AKKU1. Der alte Inhalt von AKKU1 wird in AKKU2 gerettet. Das Statuswort wird nicht beeinflusst.

Festpunktarithmetik (16 Bit)

+I, -I, *I, /I

Befehl	Operand	Erläuterung
+I		Addiere 2 Integerzahlen (16-Bit) (AKKU1-L) = (AKKU1-L) + (AKKU2-L)
-I		Subtrahiere 2 Integerzahlen (16-Bit) (AKKU1-L) = (AKKU2-L) - (AKKU1-L)
*I		Multipliziere 2 Integerzahlen (16-Bit) (AKKU1) = (AKKU2-L) * (AKKU1-L)
/I		Dividiere 2 Integerzahlen (16-Bit) (AKKU1-L) = (AKKU2-L) : (AKKU1-L) Im AKKU1-H steht der Rest der Division

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	ja	ja	ja	ja	-	-	-	-

Arithmetische Operationen zweier 16-Bit-Zahlen. Das Ergebnis wird in AKKU1 bzw. AKKU1-L geschrieben. Danach werden AKKU3 und AKKU4 nach AKKU2 und AKKU3 übertragen

Festpunktarithmetik (32 Bit)

+D, -D, *D, /D, MOD

Befehl	Operand	Erläuterung
+D		Addiere 2 Integerzahlen (32-Bit) (AKKU1) = (AKKU2) + (AKKU1)
-D		Subtrahiere 2 Integerzahlen (32-Bit) (AKKU1) = (AKKU2) - (AKKU1)
*D		Multipliziere 2 Integerzahlen (32-Bit) (AKKU1) = (AKKU2) * (AKKU1)
/D		Dividiere 2 Integerzahlen (32-Bit) (AKKU1) = (AKKU2) : (AKKU1)
MOD		Dividiere 2 Integerzahlen (32-Bit) und lade den Rest der Division in AKKU1: (AKKU1) = Rest von [(AKKU2) : (AKKU1)]

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	ja	ja	ja	ja	-	-	-	-

Arithmetische Operationen zweier 32-Bit-Zahlen. Das Ergebnis wird in AKKU1 geschrieben. Danach werden AKKU3 und AKKU4 nach AKKU2 und AKKU3 übertragen

Gleitpunktarithmetik (32 Bit)

+R, -R, *R, /R

Befehl	Operand	Erläuterung
+R		Addiere 2 Realzahlen (32-Bit) (AKKU1) = (AKKU2) + (AKKU1)
-R		Subtrahiere 2 Realzahlen (32-Bit) (AKKU1) = (AKKU2) - (AKKU1)
*R		Multipliziere 2 Realzahlen (32-Bit) (AKKU1) = (AKKU2) * (AKKU1)
/R		Dividiere 2 Realzahlen (32-Bit) (AKKU1) = (AKKU2) : (AKKU1)

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	ja	ja	ja	ja	-	-	-	-

Das Ergebnis der arithmetischen Operationen steht im AKKU1.

Gleitpunktarithmetik (32 Bit), Fortsetzung

NEGR, ABS

Befehl	Operand	Erläuterung
NEGR		Negiere Realzahl im AKKU1
ABS		Bilde Betrag der Realzahl im AKKU1

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Das Ergebnis der arithmetischen Operationen steht im AKKU1.

Quadratwurzel, Quadrat (32 Bit)

SQRT, SQR

Befehl	Operand	Erläuterung
SQRT		Berechne die Quadratwurzel einer Realzahl in AKKU1
SQR		Quadriere die Realzahl in AKKU1

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	ja	ja	ja	ja	-	-	-	-

Das Ergebnis der Operation steht im AKKU1. Die Operation SQRT ist durch Alarme unterbrechbar.

Logarithmusfunktionen (32 Bit)

LN, EXP

Befehl	Operand	Erläuterung
LN		Bilde den natürlichen Logarithmus einer Realzahl in AKKU1
EXP		Berechne den Exponentialwert einer Realzahl in AKKU1 zur Basis e (= 2,71828)

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	ja	ja	ja	ja	-	-	-	-

Das Ergebnis der Logarithmusfunktion steht im AKKU1. Die Operationen sind durch Alarme unterbrechbar.

Trigonometrische Funktionen (32 Bit)

SIN, ASIN, COS, ACOS, TAN, ATAN

Befehl	Operand	Erläuterung
SIN		Berechne den Sinus einer Realzahl
ASIN		Berechne den Arcussinus einer Realzahl
COS		Berechne den Cosinus einer Realzahl
ACOS		Berechne den Arcuscosinus einer Realzahl
TAN		Berechne den Tangens einer Realzahl
ATAN		Berechne den Arcustangens einer Realzahl

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	ja	ja	ja	ja	-	-	-	-

Das Ergebnis der Operation steht im AKKU1. Die Operationen sind durch Alarmer unterbrechbar.

Addition von Konstanten

+

Befehl	Operand	Erläuterung
+	i8	Addiere eine 8-Bit-Integer-Konstante
+	i16	Addiere eine 16-Bit-Integer-Konstante
+	i32	Addiere eine 32-Bit-Integer-Konstante

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Addition von Integer-Konstanten zum AKKU1. Das Statuswort wird nicht beeinflusst.

Addition über Adressregister

+AR1, +AR2

Befehl	Operand	Erläuterung
+AR1		Addiere Inhalt von AKKU1-L zum AR1
+AR1	m (0 bis 4095)	Addiere Pointer-Konstante zum AR1
+AR2		Addiere Inhalt von AKKU1-L zum AR2
+AR2	m (0 bis 4095)	Addiere Pointer-Konstante zum AR2

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Addition einer 16-Bit-Integerzahl zum Inhalt des Adreßregisters. Der Wert wird entweder als Operand bei der Operation angegeben oder steht im AKKU1-L. Das Statuswort wird nicht beeinflusst.

Vergleichsoperationen (16-Bit-Integerzahlen)

==I, <>I, <I, <=I, >I, >=I

Befehl	Operand	Erläuterung
==I		AKKU2-L = AKKU1-L
<>I		AKKU2-L ≠ AKKU1-L
<I		AKKU2-L < AKKU1-L
<=I		AKKU2-L ≤ AKKU1-L
>I		AKKU2-L > AKKU1-L
>=I		AKKU2-L ≥ AKKU1-L

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	ja	ja	0	-	0	ja	ja	1

Vergleich der 16-Bit-Integerzahlen in AKKU1-L und AKKU2-L.
VKE = 1, wenn Bedingung erfüllt.

Vergleichsoperationen (32-Bit-Integerzahlen)

==D, <>D, <D, <=D, >D, >=D

Befehl	Operand	Erläuterung
==D		AKKU2 = AKKU1
<>D		AKKU2 ≠ AKKU1
<D		AKKU2 < AKKU1
<=D		AKKU2 ≤ AKKU1
>D		AKKU2 > AKKU1
>=D		AKKU2 ≥ AKKU1

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	ja	ja	0	-	0	ja	ja	1

Vergleich der 32-Bit-Integerzahlen in AKKU1 und AKKU2.
VKE = 1, wenn Bedingung erfüllt.

Vergleichsoperationen (32-Bit-Realzahlen)

==R, <>R, <R, <=R, >R, >=R

Befehl	Operand	Erläuterung
==R		AKKU2 = AKKU1
<>R		AKKU2 ≠ AKKU1
<R		AKKU2 < AKKU1
<=R		AKKU2 ≤ AKKU1
>R		AKKU2 > AKKU1
>=R		AKKU2 ≥ AKKU1

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	ja	ja	ja	ja	0	ja	ja	1

Vergleich der 32-Bit-Realzahlen in AKKU1 und AKKU2.
VKE = 1, wenn Bedingung erfüllt.

Schiebeoperationen

SLW, SLD, SRW, SRD, SSI

Befehl	Operand	Erläuterung
SLW *)		Schiebe Inh. von AKKU1-L nach links. Freiwerdende Stellen werden mit Nullen aufgefüllt.
SLW	0 ... 15	
SLD		Schiebe Inhalt von AKKU1 nach links. Freiwerdende Stellen werden mit Nullen aufgefüllt.
SLD	0 ... 32	
SRW *)		Schiebe Inh. von AKKU1-L nach rechts. Freiwerdende Stellen werden mit Nullen aufgefüllt.
SRW	0 ... 15	
SRD		Schiebe Inhalt von AKKU1 nach rechts. Freiwerdende Stellen werden mit Nullen aufgefüllt.
SRD	0 ... 32	
SSI *)		Schiebe Inhalt von AKKU1-L mit Vorzeichen nach rechts. Freiwerdende Stellen werden mit dem Vorzeichen (Bit 15) aufgefüllt.
SSI	0 ... 15	

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	ja	0	0	-	-	-	-	-

*) Anzahl der geschobenen Stellen: 0 bis 16

Schiebe Inhalt von AKKU1 oder AKKU1-L um die angegebene Anzahl von Stellen nach links/rechts. Ist kein Operand angegeben, wird als Anzahl der Inhalt von AKKU2-LL genommen. Das zuletzt geschobene Bit wird ins Anzeigenbit A1 geladen.

Schiebeoperationen, Fortsetzung

SSD

Befehl	Operand	Erläuterung
SSD		Schiebe Inhalt von AKKU1 mit Vorzeichen nach rechts. Freiwerdende Stellen werden mit dem Vorzeichen-Bit (Bit 31) aufgefüllt.
SSD	0 ... 32	

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	ja	0	0	-	-	-	-	-

Rotieroperationen

RLD, RRD, RLDA, RRDA

Befehl	Operand	Erläuterung
RLD		Rotiere Inhalt von AKKU1 nach links
RLD	0 ... 32	
RRD		Rotiere Inhalt von AKKU1 nach rechts
RRD	0 ... 32	

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	ja	ja	ja	-	-	-	-	-

Befehl	Operand	Erläuterung
RLDA		Rotiere Inhalt von AKKU1 um 1 Bitposition nach links über Anzeigenbit A1.
RRDA		Rotiere Inhalt von AKKU1 um 1 Bitposition nach rechts über Anzeigenbit A1.

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	ja	0	0	-	-	-	-	-

Rotiere Inhalt von AKKU1 um die angegebene Anzahl von Stellen nach links/rechts. Ist kein Operand angegeben, wird als Anzahl der Inhalt von AKKU2-LL genommen. Das zuletzt geschobene Bit wird ins Anzeigenbit A1 geladen..

AKKU-Transferoperationen, Inkrementieren, Dekrementieren

TAW, TAD, TAK, ENT, LEAVE, PUSH, POP, INC, DEC

Befehl	Operand	Erläuterung
TAW		Umkehr der Reihenfolge der Bytes im AKKU1-L
TAD		Umkehr der Reihenfolge der Bytes in AKKU1
TAK		Tausche Inhalte von AKKU1 und AKKU2
ENT		Inhalt von AKKU2 und AKKU3 wird nach AKKU3 und AKKU4 übertragen.
LEAVE		Inhalt von AKKU3 und AKKU4 wird nach AKKU2 und AKKU 3 übertragen
PUSH		Inhalt von AKKU1, AKKU2 und AKKU3 wird nach AKKU2, AKKU3 u. AKKU4 übertragen
POP		Inhalt von AKKU2, AKKU3 und AKKU4 wird nach AKKU1, AKKU2 u. AKKU3 übertragen
INC	k8	Inkrementiere AKKU1-LL
DEC	k8	Dekrementiere AKKU1-LL

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Das Statuswort wird nicht beeinflusst.

Bildaufbauoperation, Nulloperation

BLD, NOP

Befehl	Operand	Erläuterung
BLD	k8	Bildaufbau-Operation: wird von der CPU wie eine Nulloperation behandelt.
NOP	0 1	Nulloperation

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Das Statuswort wird nicht beeinflusst.

Datentyp-Umwandlungsoperationen

BTI, BTD, DTR, ITD

Befehl	Operand	Erläuterung
BTI		Konvertiere AKKU1-L von BCD (0 bis +/- 999) in Integerzahl (16 Bit) (<u>B</u> CD <u>I</u> o <u>I</u> nt)
BTD		Konvertiere AKKU1 von BCD (0 bis +/- 9 999 999) in Doppel-Integerzahl (32 Bit) (<u>B</u> CD <u>I</u> o <u>D</u> oubleint)
DTR		Konvertiere AKKU1 von Double-Integerzahl (32 Bit) in Realzahl (32 Bit) (<u>D</u> oubleint <u>I</u> o <u>R</u> eal)
ITD		Konvertiere AKKU1 von Integerzahl (16 Bit) in Double-Integerzahl (32 Bit) (<u>I</u> nt <u>I</u> o <u>D</u> oubleint)

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Die Ergebnisse der Wandlung stehen im AKKU1.

Datentyp-Umwandlungsoperationen, Fortsetzung

ITB, DTB

Befehl	Operand	Erläuterung
ITB		Konvertiere AKKU1-L von Integerzahl (16 Bit) nach BCD 0 bis +/- 999. (<u>I</u> nt <u>I</u> o <u>B</u> CD)
DTB		Konvertiere AKKU1 von Double-Integerzahl (32 Bit) nach BCD 0 bis +/- 9 999 999. (<u>D</u> oubleint <u>I</u> o <u>B</u> CD)

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	ja	ja	-	-	-	-

Die Ergebnisse der Wandlung stehen im AKKU1.

Datentyp-Umwandlungsoperationen, Fortsetzung

RND, RND-, RND+, TRUNC

Befehl	Operand	Erläuterung
RND		Wandle Realzahl in 32-Bit-Integerzahl um.
RND-		Wandle Realzahl in 32-Bit-Integerzahl um. Es wird abgerundet zur nächsten ganzen Zahl.
RND+		Wandle Realzahl in 32-Bit-Integerzahl um. Es wird aufgerundet zur nächsten ganzen Zahl.
TRUNC		Wandle Realzahl in 32-Bit-Integerzahl um. Es werden die Nachkommastellen abgeschnitten.

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	ja	ja	-	-	-	-

Die zu wandelnde Realzahl steht in AKKU1.

Komplementbildung

INVI, INVD, NEGI, NEGD

Befehl	Operand	Erläuterung
INVI		Bilde 1er-Komplement von AKKU1-L
INVD		Bilde 1er-Komplement von AKKU1

NEGI		Bilde 2er-Komplement von AKKU1-L (Integerzahl)
NEGD		Bilde 2er-Komplement von AKKU1 (Double-Integerzahl)

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	ja	ja	ja	ja	-	-	-	-

Baustein-Aufrufoperationen

CALL, UC, CC

Befehl	Operand	Erläuterung
CALL	FB q, DB q	Unbedingter Aufruf eines FB mit Parameterübergabe
CALL	SFB q, DB q	Unbedingter Aufruf eines SFB, mit Parameterübergabe
CALL	FC q	Unbedingter Aufruf einer Funktion mit Parameterübergabe
CALL	SFC q	Unbedingter Aufruf einer SFC, mit Parameterübergabe
UC	FB q FC q SFC q FB [e] FC [e] Parameter	Unbedingter Aufruf von Bausteinen ohne Parameterübergabe. speicherdirekter FB-Aufruf speicherdirekter FC-Aufruf FB/FC-Aufruf über Parameter
CC	FB q FC q FB [e] FC [e] Parameter	Unbedingter Aufruf von Bausteinen ohne Parameterübergabe. speicherdirekter FB-Aufruf speicherdirekter FC-Aufruf FB/FC-Aufruf über Parameter

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	0	0	1	-	0

Die Aussagen zum Statuswort beziehen sich nur auf den Bausteinaufruf selbst und nicht auf die in diesem Baustein ausgeführten Befehle.

Statuswort für CC: Abhängig von VKE, setzt VKE=1

Baustein-Aufrufoperationen, Fortsetzung

AUF

Befehl	Operand	Erläuterung
AUF		Aufschlagen eines
	DB q	Datenbausteins
	DI q	Instanz-Datenbausteins
	DB [e]	Datenbausteins, speicherindirekt
	DI [e]	Instanz-Datenbausteins, speicherindirekt
	Parameter	Datenbaustein über Parameter

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Baustein-Endeoperationen

BE, BEA, BEB

Befehl	Operand	Erläuterung
BE		Beende Baustein
BEA		Beende Baustein absolut

BEB		Beende Baustein bedingt bei VKE = "1"
-----	--	---------------------------------------

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	0	0	1	-	0

Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	ja	-	-
Operation beeinflusst:	-	-	-	-	ja	0	1	1	0

Tausche Datenbausteine

TDB

Befehl	Operand	Erläuterung
TDB		Tausche Datenbausteine

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Tauschen der beiden aktuellen Datenbausteine. Der aktuelle Datenbaustein wird zum aktuellen Instanz-Datenbaustein und umgekehrt. Das Statuswort wird nicht beeinflusst.

Sprungoperationen

SPA, SPB, SPBN, SPBB, SPBNB

Befehl	Operand	Erläuterung
SPA	MARKE	Springe unbedingt

Sprung, abhängig von der Bedingung.

SPB	MARKE	Springe bei VKE = "1"
SPBN	MARKE	Springe bei VKE = "0"

SPBB	MARKE	Springe bei VKE = "1" Retten des VKE in das BIE-Bit
SPBNB	MARKE	Springe bei VKE = "0" Retten des VKE in das BIE-Bit

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	ja	-
Operation beeinflusst:	-	-	-	-	-	0	1	1	0

Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	ja	-
Operation beeinflusst:	ja	-	-	-	-	0	1	1	0

Sprungoperationen, Fortsetzung

SPBI, SPBIN, SPO, SPS

Befehl	Operand	Erläuterung
SPBI	MARKE	Springe bei BIE = "1"
SPBIN	MARKE	Springe bei BIE = "0"

SPO	MARKE	Springe bei Überlauf speichernd (OV = "1")
-----	-------	---

SPS	MARKE	Springe bei Überlauf speichernd (OS = "1")
-----	-------	---

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	ja	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	0	1	-	0

Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	ja	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	ja	-	-	-	-
Operation beeinflusst:	-	-	-	-	0	-	-	-	-

Sprungoperationen, Fortsetzung

SPU, SPZ, SPP, SPM, SPN, SPMZ, SPPZ

Befehl	Operand	Erläuterung
SPU	MARKE	Springe bei "Unzulässiger Arithmetik-operation" (A1 = 1 und A0 = 1)
SPZ	MARKE	Springe bei Ergebnis = 0 (A1 = 0 und A0 = 0)
SPP	MARKE	Springe bei Ergebnis > 0 (A1 = 1 und A0 = 0)
SPM	MARKE	Springe bei Ergebnis < 0 (A1 = 0 und A0 = 1)
SPN	MARKE	Springe bei Ergebnis ≠ 0 (A1 = 1 und A0 = 0) oder (A1 = 0 und A0 = 1)
SPMZ	MARKE	Springe bei Ergebnis ≤ 0 (A1 = 0 und A0 = 1) oder (A1 = 0 und A0 = 0)
SPPZ	MARKE	Springe bei Ergebnis ≥ 0 (A1 = 1 und A0 = 0) oder (A1 = 0 und A0 = 0)

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	ja	ja	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Sprungoperationen, Fortsetzung

SPL, LOOP

Befehl	Operand	Erläuterung
SPL	MARKE	Sprungverteiler. Der Operation folgt eine Liste von Sprungoperationen. Der Operand ist eine Sprungmarke auf die der Liste folgenden Operation. AKKU1-LL enthält die Nr. der Sprungoperation (max. 254), welche ausgeführt werden soll, wobei die erste Sprungoperationsnummer 0 ist.
LOOP	MARKE	Dekrementiere AKKU1-L und springe bei AKKU1-L ≠ 0 (Schleifenprogrammierung)

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

Operationen für das Master Control Relay (MCR)

MCR(,)MCR, MCRA, MCRD

Befehl	Operand	Erläuterung
MCR(Öffnen einer MCR-Zone. Retten des VEK auf den MCR-Stack.

Statuswort									
Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	ja	-
Operation beeinflusst:	-	-	-	-	-	0	1	-	0

)MCR		Schliessen einer MCR-Zone. Entfernen eines Eintrags vom MCR-Stack.
------	--	---

Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	0	1	-	0

MCRA		Aktiviere MCR
MCRD		Desaktiviere MCR

Statusbit	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Operation wertet aus:	-	-	-	-	-	-	-	-	-
Operation beeinflusst:	-	-	-	-	-	-	-	-	-

MCR = 1 → MCR ist deaktiviert

MCR = 0 → MCR ist aktiviert; "T"- und "="-Operationen schreiben bei VKE = "0" Nullen auf die entsprechenden Operanden; "S"- und "R"-Operationen lassen den Speicherinhalt unverändert.

Bausteine und Funktionen der CPUs

Organisations- bausteine	PCD1.M137	PCD2.M127	PCD2.M157 PCD2.M177 PCD2.M257
Freier Zyklus:			
OB 1	x	x	x
Uhrzeitalarme:			
OB 10	x	x	x
OB 11	x	x	x
OB 12		x	x
OB 13		x	x
OB 14			x
OB 15			x
OB 16			x
OB 17			x
Verzögerungsalarme:			
OB 20	x	x	x
OB 21		x	x
OB 22		x	x
OB 23		x	x

Startereignisse (Hexadezimalwert)
1101, 1102, 1103
1111
1112
1113
1114
1115
1116
1117
1118
1121
1122
1123
1124

Ein Anwenderprogramm für eine PCD1/2.xx7 besteht aus Bausteinen welche die Anweisungen, Parameter und Daten für die jeweilige CPU enthalten. Die einzelnen CPUs der PCD1/2.xx7 unterscheiden sich in der Menge der Bausteine, welche Sie für die jeweilige CPU anlegen können bzw. welche vom Betriebssystem der CPU bereitgestellt werden. Eine ausführliche Beschreibung der OBs und deren Anwendung finden Sie im *STEP 7 Programmierhandbuch*.

Bausteine und Funktionen der CPUs, Fortsetzung

Organisations- bausteine	PCD1.M137	PCD2.M127	PCD2.M157 PCD2.M177 PCD2.M257
Weckalarme:			
OB 30			x
OB 31			x
OB 32		x	x
OB 33		x	x
OB 34		x	x
OB 35	x	x	x
OB 36			x
OB 37			x
OB 38			x
Prozessalarme:			
OB 40	x	x	x
OB 41	x	x	x
OB 42		x	x
OB 43		x	x
OB 44		x	x
OB 45			x
OB 46			x
OB 47			x

Startereignisse (Hexadezimalwert)
1131
1132
1133
1134
1135
1136
1137
1138
1139
1141, 1142, 1143, 1144
1141, 1142, 1143, 1144
1141, 1142, 1143, 1144
1141, 1142, 1143, 1144
1141, 1142, 1143, 1144
1141, 1142, 1143, 1144
1141, 1142, 1143, 1144
1141, 1142, 1143, 1144

Bausteine und Funktionen der CPUs, Fortsetzung

Organisations- bausteine	PCD1.M137	PCD2.M127	PCD2.M157 PCD2.M177 PCD2.M257
Asynchrone Fehleralarme:			
OB 80	x	x	x
OB 81	x	x	x
OB 82	x	x	x
OB 83	x	x	x
OB 84	x	x	x
OB 85	x	x	x
OB 86	x	x	x
OB 87	x	x	x
Neustart:			
OB 100	x	x	x
Synchrone Fehleralarme:			
OB 121	x	x	x
OB 122	x	x	x

Startereignisse (Hexadezimalwert)
3501, 3502, 3505, 3506, 3507
3821, 3822, 3823, 3831, 3832, 3833
3921, 3922, 3923, 3931, 3932, 3933
3842, 3942
3861, 3863, 3864, 3961
3881, 3981
35A1, 35A3, 39B1, 39B2
38C1, 38C2, 39C1
35D2, 35D3, 35D4, 35D5, 35E1, 35E3, 35E4, 35E5, 35E6
1381, 1382
2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 253A, 253C, 253D, 253E, 253F
2942, 2943, 2944, 2945

Funktionsbausteine, Funktionen und Datenbausteine

Funktions- Bausteine FB	PCD1.M137	PCD2.M127	PCD2.M157 PCD2.M177 PCD2.M257
Anzahl	512	512	512
Zulässige Nummer	0 bis 511	0 bis 511	0 bis 511
Maximale Grösse eines Funktionsbau- steins (ablauf- relevanter Code)	64 KByte - 2 Byte *)	64 KByte - 2 Byte *)	64 KByte - 2 Byte *)

Daten-Bau- steine DB	PCD1.M137	PCD2.M127	PCD2.M157 PCD2.M177 PCD2.M257
Anzahl	1023	1023	1023
Zulässige Nummer	1 bis 1023	1 bis 1023	1 bis 1023
Maximale Grösse eines Datenbau- steins (Anzahl Datenbytes)	64 KByte - 2 Byte *)	64 KByte - 2 Byte *)	64 KByte - 2 Byte *)

Funktionen FC	PCD1.M137	PCD2.M127	PCD2.M157 PCD2.M177 PCD2.M257
Anzahl	1024	1024	1024
Zulässige Nummer	0 bis 1023	0 bis 1023	0 bis 1032
Maximale Grösse einer Funktion (ablauf- relevanter Code)	64 KByte - 2 Byte *)	64 KByte - 2 Byte *)	64 KByte - 2 Byte *)

*) Bausteingrösse begrenzt durch Arbeitsspeicher

Die Tabellen listen Anzahl, Nummer und maximale Grösse der Funktionsbausteine, Funktionen und Datenbausteine auf, welche in den einzelnen CPUs der SAIA PCD Serie xx7 angelegt werden können.

Systemfunktionen SFC

SFC-Nr.	SFC-Name	Bedeutung
0	SET_CLK	Uhrzeit stellen
1	READ_CLK	Uhrzeit lesen
2	SET_RTM	Betriebsstundenzähler setzen
3	CTRL_RTM	Betriebsstundenz. starten u. stoppen
4	READ_RTM	Betriebsstundenzähler auslesen
6	RD_SINFO	Startinfo. des aktuellen OB auslesen
13	DPNRM_DG	Slavediagnosedaten lesen
14	DPRD_DAT	Konsistente Nutzdaten lesen
15	DPWR_DAT	Konsistente Nutzdaten schreiben
20	BLKMOV	Variable kopieren
21	FILL	Feld vorbesetzen
22	CREAT_DB	Datenbaustein erzeugen
23	DEL_DB	Datenbaustein löschen
24	TEST_DB	Datenbaustein testen
25	COMPRESS	Anwenderspeicher komprimieren
26	READ_PI	Prozessabbild der Eingänge aktualisieren
27	WRITE_P0	Ausgänge aktualisieren
28	SET_TINT	Uhrzeitalarm stellen
29	CAN_TINT	Uhrzeitalarm stornieren
30	ACT_TINT	Uhrzeitalarm aktivieren
31	QRY_TINT	Uhrzeitalarm abfragen
32	SRT_DINT	Verzögerungsalarm starten

SFC-Nr.	SFC-Name	Bedeutung
33	CAN_DINT	Verzögerungsalarm stornieren
34	QRY_DINT	Verzögerungsalarm abfragen
36	MSK_FLT	Synchronfehlerereignisse maskieren
37	DMSK_FLT	Synchronfehlerereignisse demaskieren
38	READ_ERR	Ereignisstatusregister lesen
39	DIS_IRT	Verwerfen neuer Ereignisse
40	EN_IRT	Verwerfen von Ereignissen aufheben
41	DIS_AIRT	Verzögern von Alarmereignissen
42	EN_AIRT	Verzögerung von Alarmereignissen
43	RE_TRIGR	Zykluszeitüberwachung nachtriggern
44	REPL_VAL	Ersatzwert in AKKU 1 übertragen
46	STP	CPU in STOP überführen
52	WR_USMSG	Anwendereintrag in Diagnosepuffer schreiben
60	GD_SND	GD-Paket senden
61	GD_RCV	GD-Paket übernehmen
64	TIME-TCK	Millisekundentimer auslesen

Systemfunktionen SFC, SAIA® PCD - Serie xx7

SFC-Nr.	SFC-Name	Bedeutung
220	LON_INIT	Initialisierung LON-Interface
221	NV_SEND	Senden einer Netzwerkvariablen SNVT
223	MSG_SEND	Senden einer Message
227	PCD104_RD	Lesen von Dual-Port-RAM
228	PCD104_WR	Schreiben in Dual-Port-RAM
229	PCD104_ST	Status Dual-Port-RAM
230	RD_COMP	Compiler Status lesen
239	WDOG	Watch Dog Relay geschlossen halten
240	COM_RCV	Serielle Schnittstelle empfangen
241	COM_SEND	Serielle Schnittstelle Daten senden
242	COM_STAT	Serielle Schnittstelle Status abfragen
243	COM_INIT	Serielle Schnittstelle initialisieren (ohne Protokoll)
244	COM_SIG	Steuern Modem-Signale

SFC-Nr.	SFC-Name	Bedeutung
245	B_INIT	Serielle Schnittstelle initialisieren (mit Protokoll für SFB 12-14)
248	_INTDIR_	Schneller Zähler mit Drehrichtungsauswertung
250	INP_INT	Interrupt-Eingänge sperren / freigeben
251	INITCNTR	Zähler konfigurieren / starten
252	READCNTR	Zählerstand lesen
253	READ_SSI	SSI-Schnittstelle lesen
254	GRAY2BIN	Umwandlung Gray-Code in Binär-Code

Systemfunktionsbausteine SFB

SFB-Nr.	SFB-Name	Bedeutung
12	BSEND	Daten blockorientiert senden
13	BRCV	Daten blockorientiert empfangen
14	GET	Daten aus remoter CPU lesen

Systemfunktionsbausteine SFB, SAIA® PCD - Serie xx7

SFB-Nr.	SFB-Name	Bedeutung
240	FLASH	DBs in Flash-Speicher schreiben/lesen

Alphabetisches Verzeichnis der Operationen

Operation	Seite
)	14
)MCR	72
+	50
+AR1	51
+AR2	51
+D	44
+I	43
+R	45
-D	44
-I	43
-R	45
*D	44
*I	43
*R	45
/D	44
/I	43
/R	45
=	24
==D	53
==I	52

Operation	Seite
==R	54
<=D	53
<=I	52
<=R	54
<D	53
<I	52
<R	54
<>D	53
<>I	52
<>R	54
>=D	53
>=I	52
>=R	54
>D	53
>I	52
>R	54
ABS	46
ACOS	49
ASIN	49
ATAN	49

Operation	Seite
AUF	65
BE	66
BEA	66
BEB	66
BLD	59
BTD	60
BTI	60
CALL	64
CC	64
CLR	25
COS	49
DEC	58
DTB	61
DTR	60
ENT	58
EXP	48
FN	22
FP	22
FR	28-29
INC	58

Operation	Seite
INVD	63
INVI	63
ITB	61
ITD	60
L	30-35, 41-42
LAR1	39
LAR2	39
LC	35
LEAVE	58
LN	48
LOOP	71
MCR(72
MCRA	72
MCRD	72
MOD	44
NEGD	63
NEGI	63
NEGR	46
NOP	59
NOT	26

Alphabetisches Verzeichnis der Operationen, Fortsetzung

Operation	Seite
O	11, 15, 17, 20-21
O(13
OD	19
ON	11, 17, 20-21
ON(13
OW	19
POP	58
PUSH	58
R	23, 28, 29
RLD	57
RLDA	57
RND	62
RND+	62
RND-	62
RRD	57
RRDA	57
S	23, 29
SA	28
SAVE	26
SE	27

Operation	Seite
SET	25
SI	27
SIN	49
SLD	55
SLW	55
SPA	68
SPB	68
SPBB	68
SPBI	69
SPBIN	69
SPBN	68
SPBNB	68
SPL	71
SPM	70
SPMZ	70
SPN	70
SPO	69
SPP	70
SPPZ	70
SPS	69

Operation	Seite
SPU	70
SPZ	70
SQR	47
SQRT	47
SRD	55
SRW	55
SS	27
SSD	56
SSI	55
SV	27
T	36-38, 41
TAD	58
TAK	58
TAN	49
TAR	40
TAR1	40
TAR2	40
TAW	58
TDB	67
TRUNC	62

Operation	Seite
U	10, 16, 20-21
U(13
UC	64
UD	19
UN	10, 16, 20-21
UN(13
UW	19
X	12, 18, 20-21
X(13
XN	12, 18, 20-21
XN(13
XOD	19
XOW	19
ZR	29
ZV	29

Notizen

Absender:

Firma

Abteilung

Name

Adresse

Tel.

Datum

An

SAIA-Burgess Electronics AG

Bahnhofstrasse 18

CH-3280 Murten (Schweiz)

<http://www.saia-burgess.com>

GB: Electronic Controllers

Befehlsliste PCD Serie xx7

Falls Sie Vorschläge zu SAIA[®] PCD zu machen oder Fehler in diesem Handbuch gefunden haben, sind wir Ihnen für einen kurzen Bericht dankbar.

Notizen