

# GFK-0467J-G

[Buy GE Fanuc Series 90-30 NOW!](#)

## GE Fanuc Manual Series 90-30

Series 90-30/20/Micro Speicherprogrammierbare  
Steuerungen Referenzhandbuch

1-800-360-6802  
sales@pdfsupply.com



# *GE Fanuc Automation*

---

*Speicherprogrammierbare Steuerungen*

*Series 90™ -30/20/Micro  
Speicherprogrammierbare Steuerungen*

*Referenzhandbuch*

*GFK-0467J-GE*

*Mai 1997*

## *Die Begriffe Vorsicht, Achtung und Hinweis, wie sie in dieser Publikation verwendet werden*

### **Vorsicht**

**In dieser Veröffentlichung werden VORSICHT-Hinweise verwendet, um darauf hinzuweisen, daß innerhalb der beschriebenen Geräte gefährliche Spannungen, Ströme, Temperaturen oder andere Bedingungen, die körperliche Schäden hervorrufen können, vorkommen.**

**Wo Unaufmerksamkeit entweder körperliche Schäden oder eine Beschädigung des Gerätes verursachen könnte, werden VORSICHT-Hinweise verwendet.**

### **Achtung**

**ACHTUNG-Hinweise werden dort verwendet, wo das Gerät bei unsachgemäßer Vorgehensweise beschädigt werden könnte.**

### **Hinweis**

HINWEISE sollen nur die Aufmerksamkeit des Lesers auf Informationen lenken, die besonders wichtig für Verständnis und Bedienung des Gerätes sind.

Dieses Dokument stützt sich auf Informationen, die zum Zeitpunkt seiner Veröffentlichung verfügbar waren. Obwohl alle Anstrengungen unternommen wurden, den Inhalt so genau wie möglich zu gestalten, können die hier enthaltenen Informationen nicht den Anspruch erheben, alle Details oder Veränderungen von Software und Hardware abzudecken, oder jede Möglichkeit im Zusammenhang mit Installation, Betrieb oder Wartung zu berücksichtigen. In diesem Dokument können Merkmale beschrieben sein, die nicht in allen Hard- und Softwaresystemen vorhanden sind. Weder General Electric Company noch GE Fanuc Automation übernehmen eine Verpflichtung, Besitzer dieses Dokumentes über nachträglich durchgeführte Änderungen zu informieren.

Weder General Electric Company noch GE Fanuc Automation übernehmen Verantwortung für die Genauigkeit, Vollständigkeit oder Nützlichkeit der in diesem Dokument enthaltenen Informationen.

Bei den folgenden Bezeichnungen handelt es sich um Warenzeichen für Produkte von GE Fanuc Automation North America, Inc.

Alarm Master	GEnet	Modelmaster	Series One
CIMPLICITY	Genius	ProLoop	Series Six
CIMPLICITY PowerTRAC	Genius PowerTRAC	PROMACRO	Series Three
CIMPLICITY 90-ADS	Helpmate	Series Five	VuMaster
CIMSTAR	Logicmaster	Series 90	Workmaster
Field Control			

In diesem Handbuch werden Systembetrieb, Fehlerbehandlung und Logicmaster 90™ Programmieranweisungen für die speicherprogrammierbaren Steuerungen Series 90™-30, Series 90™-20 und Series 90™ Micro beschrieben. Die speicherprogrammierbaren Steuerungen Series 90™-30, Series 90™-20 und Series 90™ Micro gehören alle zur Series 90™ Produktfamilie der speicherprogrammierbaren Steuerungen von GE Fanuc Automation.

## Änderungen in diesem Handbuch

Wir haben den Text an vielen Stellen deutlicher gemacht und um die neuen Funktionen der CPUs vom Ausgabestand 8 erweitert. Die CPUs 351 und 352 des Ausgabestands 8 verfügen zum Beispiel über eine neue Funktion, die einen besseren Speicherschutz ermöglicht (siehe Seite 2-14). Neue Produkte für die Mikro-SPS finden Sie auf Seite 2-41. Weitere Einzelheiten zur Einschaltoutine finden Sie ab Seite 2-30.

Weitere ergänzende Beschreibungen in diesem Handbuch finden Sie insbesondere in den Abschnitten "Programmiergeräte-Kommunikationsfenster" und "Kommunikationsfester-Betriebsarten" in Kapitel 2, "Anwendungsfehler" in Kapitel 3 und "PID" in Kapitel 4.

Die Abschnitte im Anhang E wurden neu geordnet, um die Übersichtlichkeit zu erhöhen. Weitere kleinere Änderungen wurden eingefügt, wo sie notwendig waren.

Darüberhinaus haben wir einige Textbereiche weniger auf Logicmaster bezogen. Dieses Handbuch sollte, unabhängig von der Programmiersoftware, mehr auf die SPS-CPU's der Series 90-30, Series 90-20 und Series 90 Micro bezogen sein.

## Inhalt dieses Handbuchs

**Kapitel 1. Einleitung:** Gibt einen Überblick über die SPS-Systeme Series 90-30, Series 90-20 und Series 90 Micro sowie den Series 90-30/20/Micro-Befehlssatz.

**Kapitel 2. Systembetrieb:** Beschreibt bestimmte Systemabläufe der SPS-Systeme Series 90-30, Series 90-20 und Series 90 Micro. Hierzu gehören Beschreibungen der SPS-Zyklusfolgen, die Abläufe beim Ein- und Ausschalten des Systems, Uhren und Timer, Sicherheitsaspekte, E/A und Fehlerbehandlung. Darüberhinaus finden Sie hier die zum grundlegenden Verständnis der Kontaktplanprogrammierung erforderlichen allgemeinen Informationen.

**Kapitel 3. Fehlerbeschreibung und Fehlerbehebung:** Hier finden Sie Angaben zur Fehlersuche bei den SPS-Systeme Series 90-30, Series 90-20 und Series 90 Micro, sowie eine Erläuterung der Fehlerbeschreibungen in den SPS-Fehlertabellen und der Fehlerkategorien in der E/A-Fehlertabelle.

**Kapitel 4. Series 90-30/20/Micro Befehlsvorrat:** Beschreibt die für die SPS Series 90-30, Series 90-20 und Series 90 Micro verfügbaren Programmieranweisungen. Die Informationen in diesem Kapitel sind in Abschnitte gegliedert, die den Programmfunktionsgruppen entsprechen.

**Anhang A. Befehlsausführungszeiten:** Hier sind für jede einzelne Programmieranweisung der Speicherbedarf in Bytes sowie die Befehlsausführungszeit aufgelistet. Der Speicherbedarf ist die Anzahl Bytes, die von der Funktion in einem Kontaktplanprogramm benötigt wird.

**Anhang B. Interpretation der Fehlertabellen:** Beschreibt, wie Sie die Meldungsstrukturen interpretieren müssen, wenn Sie die Fehlertabellen mit der Logicmaster 90-30/20/Micro Software auslesen.

**Anhang C. Anweisungsmnemonik:** Listet die Kurzbezeichnungen (mnemonische Befehle) auf, die beim Durchsuchen oder Editieren eines Programms zur Anzeige der Programmieranweisungen verwendet werden können.

**Anhang D. Tastenfunktionen:** Auflistung der speziellen Tastaturbelegungen für die Logicmaster 90-30/20/Micro Software.

**Anhang E. Verwendung von Gleitpunktzahlen:** Beschreibung spezieller Aspekte beim Einsatz von Gleitpunktarithmetik.

## Zugehörige Veröffentlichungen

*Logicmaster™ 90 Series 90™ -30/20/Micro Programmiersoftware Anwenderhandbuch (GFK-0466).*

*Logicmaster™ 90 Series 90-30 und 90-20, Wichtige Produktinformationen (GFK-0468).*

*SPS Series 90™ -30, Installationshandbuch (GFK-0356).*

*SPS Series 90™ -20, Installationshandbuch (GFK-0551).*

*Series 90™ -30 E/A-Module, Technische Daten (GFK-0898).*

*Series 90™ Programmierbares Coprozessormodul und Unterstützungssoftware, Anwenderhandbuch (GFK-0255).*

*Series 90™ PCM-Entwicklungssoftware (PCOP), Anwenderhandbuch (GFK-0487).*

*CIMPLICITY™ 90-ADS Alphanumerisches Anzeigesystem, Anwenderhandbuch (GFK-0499).*

*CIMPLICITY™ 90-ADS Alphanumerisches Anzeigesystem, Referenzhandbuch (GFK-0641).*

*Alphanumerisches Anzeige-Coprozessormodul, Datenblatt (GFK-0521).*

*Hand-Programmiergerät für SPS Series 90™ -30 und 90-20, Anwenderhandbuch (GFK-0402).*

*Series 90™ -30 Achsen-Positioniermodul, Standardmodus, Anwenderhandbuch (GFK-0840).*

*Series 90™ -30 Achsen-Positioniermodul, Nachlaufmodus, Anwenderhandbuch (GFK-0781).*

*Series 90™ -30 Schnelles Zählmodul, Anwenderhandbuch (GFK-0293).*

*Series 90™ -30 Genius-Kommunikationsmodul, Anwenderhandbuch (GFK-0412).*

*Genius-Kommunikationsmodul, Datenblatt (GFK-0272).*

*Series 90™ -30 Genius™ Buscontroller, Anwenderhandbuch (GFK-1034).*

*Series 90™ -30 FIP Buscontroller, Anwenderhandbuch (GFK-1038).*

*Series 90™ -30 FIP dezentraler E/A-Scanner, Anwenderhandbuch (GFK-1037).*

*Field Control™, dezentrale E/A und Schnittstellenmodul zum Genius™ -Bus, Anwenderhandbuch (GFK-0825)*

*SPS Series 90™ Micro, Anwenderhandbuch (GFK-1065).*

*SPS Series 90™ PLC Serielle Kommunikation, Anwenderhandbuch (GFK-0582).*



<b>Kapitel 1</b>	<b>Einleitung .....</b>	<b>1-1</b>
<b>Kapitel 2</b>	<b>Systembetrieb .....</b>	<b>2-1</b>
	<b>Abschnitt 1: SPS-Zyklus – Zusammenfassung .....</b>	<b>2-2</b>
	Standard-Programmzyklus .....	2-2
	Berechnung der Zykluszeit .....	2-7
	Beispiel einer Zykluszeitberechnung .....	2-7
	Organisation .....	2-7
	Eingabezyklus .....	2-8
	Anwenderprogramm-Bearbeitungszyklus .....	2-8
	Ausgabezyklus .....	2-8
	Programm-Prüfsummenberechnung .....	2-8
	Programmiergeräte-Kommunikationsfenster .....	2-9
	System-Kommunikationsfenster (ab Modell 331 aufwärts) .....	2-10
	PCM-Kommunikation mit der SPS (ab Modell 331 aufwärts) .....	2-11
	Varianten des Standard-Programmzyklus .....	2-12
	Betriebsart KONSTANTE ZYKLUSZEIT .....	2-12
	SPS-Zyklus in STOP-Modus .....	2-12
	Kommunikationsfenster-Betriebsarten .....	2-12
	Schlüsselschalter an CPUs 351/352: Betriebsartenwechsel und Schutz von Flash Memory .....	2-13
	Schlüsselschalter ab Ausgabestand 7 .....	2-13
	Fehlertabelle mit Schlüsselschalter löschen .....	2-13
	Erweiterter Speicherschutz bei CPUs ab Ausgabestand 8 .....	2-14
	<b>Abschnitt 2: Programmorganisation und Anwenderreferenzen/-daten ...</b>	<b>2-15</b>
	Unterprogrammblöcke (nur SPS Series 90-30) .....	2-15
	Beispiele von Unterprogrammblöcken .....	2-16
	Wie Unterprogrammblöcke aufgerufen werden .....	2-17
	Periodische Unterprogramme .....	2-17
	Anwenderreferenzen .....	2-19
	Transitionen und Overrides .....	2-20
	Datenremanenz .....	2-21
	Datentypen .....	2-22
	System-Statusreferenzen .....	2-23
	Funktionsblockstruktur .....	2-25
	Format der Kontaktplanrelais .....	2-25
	Format der Programm-Funktionsblöcke .....	2-26
	Funktionsblockparameter .....	2-27
	Stromfluß zu und von einer Funktion .....	2-28

	<b>Abschnitt 3: Ein- und Ausschaltsequenzen .....</b>	<b>2-29</b>
	Einschalten .....	2-29
	Ausschalten .....	2-31
	<b>Abschnitt 4: Uhren und Timer .....</b>	<b>2-32</b>
	Betriebszeituhr .....	2-32
	Tageszeituhr .....	2-32
	Zeitüberwachung (Watchdog) .....	2-33
	Timer für konstante Zykluszeit .....	2-33
	Zeitkontakte .....	2-33
	<b>Abschnitt 5: Systemsicherheit .....</b>	<b>2-34</b>
	Paßworte .....	2-34
	Wechsel der Privilegebene .....	2-35
	Sperrn/freigeben von Unterprogrammen .....	2-35
	Permanentes Sperrn eines Unterprogramms .....	2-35
	<b>Abschnitt 6: E/A-System der SPS Series 90-30, 90-20 und Micro .....</b>	<b>2-36</b>
	Modell 30 E/A-Module .....	2-37
	E/A-Datenformate .....	2-39
	Standardbedingungen für die Modell 30 Ausgangsmodule .....	2-39
	Diagnosedaten .....	2-39
	Globaldaten .....	2-39
	Modell 20 E/A-Module .....	2-40
	Micro-SPS .....	2-40
<b>Kapitel 3</b>	<b>Erläuterung und Behebung von Fehlern .....</b>	<b>3-1</b>
	<b>Abschnitt 1: Fehlerbearbeitung .....</b>	<b>3-2</b>
	Alarmprozessor .....	3-2
	Fehlerklassen .....	3-2
	Systemreaktion auf Fehler .....	3-3
	Fehlertabellen .....	3-3
	Fehlerwirkung .....	3-4
	Fehlerreferenzen .....	3-4
	Definitionen der Fehlerreferenz .....	3-5
	Weitere Fehlerauswirkungen .....	3-5
	Anzeige der SPS-Fehlertabelle .....	3-5
	Anzeige der E/A-Fehlertabelle .....	3-5
	Zugriff auf weitere Fehlerdaten .....	3-6



<b>Abschnitt 2: SPS-Fehlertabelle – Erläuterungen</b>	<b>3-7</b>
Fehlerwirkungen	3-8
Verlorenes oder fehlendes Zusatzmodul	3-8
Rückgesetztes, hinzugefügtes oder überzähliges Zusatzmodul	3-8
Diskrepanz bei Systemkonfiguration	3-9
Softwarefehler bei Zusatzmodul	3-9
Programmblock-Prüfsummenfehler	3-10
Signal für niedrige Batteriespannung	3-11
Konstante Zykluszeit überschritten	3-11
Anwendungsfehler	3-11
Anwenderprogramm fehlt	3-12
Verstümmeltes Anwenderprogramm beim Einschalten	3-12
Paßwortfehler	3-12
CPU-Systemsoftwarefehler in der SPS	3-13
Datenübertragungsfehler beim Speichern	3-15
<b>Abschnitt 3: E/A-Fehlertabelle – Erläuterungen</b>	<b>3-16</b>
Verlorenes E/A-Modul	3-17
Hinzugefügtes E/A-Modul	3-17
<b>Kapitel 4</b>	<b>Befehlsvorrat der Series 90-30/20/Micro</b>
<b>Abschnitt 1: Relaisfunktionen</b>	<b>4-1</b>
Kontakte	4-2
Spulen	4-3
Schließerkontakt — —	4-4
Öffnerkontakt — /—	4-4
Beispiel:	4-4
Spule —( )—	4-4
Beispiel:	4-4
Negierte Spule —(/)—	4-4
Beispiel:	4-4
Remanente Spule —(M)—	4-5
Negierte remanente Spule —(/M)—	4-5
Spule für positive Übergänge	4-5
Spulen für negative Übergänge	4-5
Beispiel:	4-5
SET-Spule —(S)—	4-6
RESET-Spule —(R)—	4-6
Beispiel:	4-6
Remanente SET-Spule —(SM)—	4-7
Remanente RESET-Spule —(RM)—	4-7
Verbindungen	4-7
Beispiel:	4-7
Fortsetzspulen (—<+>) und -kontakte (<+>—)	4-8

<b>Abschnitt 2: Timer und Zähler</b> .....	<b>4-9</b>
Für Timer und Zähler benötigte Funktionsblockdaten .....	4-9
ONDTR .....	4-11
Parameter: .....	4-12
Zulässige Speichertypen: .....	4-12
Beispiel: .....	4-13
TMR .....	4-14
Parameter: .....	4-15
Zulässige Speichertypen: .....	4-15
Beispiel: .....	4-16
OFDT .....	4-17
UPCTR .....	4-20
Parameter: .....	4-20
Zulässige Speichertypen: .....	4-21
Beispiel: .....	4-21
DNCTR .....	4-22
Parameter: .....	4-22
Zulässige Speichertypen: .....	4-23
Beispiel: .....	4-23
Beispiel: .....	4-24
 <b>Abschnitt 3: Arithmetische Funktionen</b> .....	 <b>4-26</b>
Grundrechenarten (ADD, SUB, MUL, DIV) .....	4-27
Parameter: .....	4-28
Zulässige Speichertypen: .....	4-28
Beispiel: .....	4-28
Arithmetische Funktionen und Datentypen .....	4-29
MOD (INT, DINT) .....	4-31
Parameter: .....	4-31
Zulässige Speichertypen: .....	4-32
Beispiel: .....	4-32
SQRT (INT, DINT, REAL) .....	4-33
Parameter: .....	4-33
Zulässige Speichertypen: .....	4-34
Beispiel: .....	4-34
Trigonometrische Funktionen (SIN, COS, TAN, ASIN, ACOS, ATAN) ...	4-35
Parameter: .....	4-35
Zulässige Speichertypen: .....	4-36
Beispiel: .....	4-36
Logarithmus- / Exponentialfunktionen (LOG, LN, EXP, EXPT) .....	4-37

Parameter: .....	4-37
Zulässige Speichertypen: .....	4-37
Beispiel: .....	4-38
Bogenmaß-Umrechnung (RAD, DEG) .....	4-39
Parameter: .....	4-39
Zulässige Speichertypen: .....	4-39
Beispiel: .....	4-40
<b>Abschnitt 4: Relationale Funktionen .....</b>	<b>4-41</b>
Parameter: .....	4-42
Zulässige Speichertypen: .....	4-42
Beispiel: .....	4-42
RANGE (INT, DINT, WORD) .....	4-43
Parameter: .....	4-44
Zulässige Speichertypen: .....	4-44
Beispiel 1: .....	4-45
Beispiel 2: .....	4-45
<b>Abschnitt 5: Bitoperationsfunktionen .....</b>	<b>4-46</b>
AND und OR (WORD) .....	4-48
Parameter: .....	4-48
Zulässige Speichertypen: .....	4-49
Beispiel: .....	4-49
XOR (WORD) .....	4-50
Parameter: .....	4-50
Zulässige Speichertypen: .....	4-51
Beispiel: .....	4-51
NOT (WORD) .....	4-52
Parameter: .....	4-52
Zulässige Speichertypen: .....	4-53
Beispiel: .....	4-53
SHL und SHR (WORD) .....	4-54
Parameter: .....	4-55
Zulässige Speichertypen: .....	4-55
Beispiel: .....	4-56
ROL und ROR (WORD) .....	4-57
Parameter: .....	4-57
Zulässige Speichertypen: .....	4-58
Beispiel: .....	4-58
BTST (WORD) .....	4-59

Parameter: .....	4-59
Zulässige Speichertypen: .....	4-60
Beispiel: .....	4-60
BSET und BCLR (WORD) .....	4-61
Parameter: .....	4-61
Zulässige Speichertypen: .....	4-62
Beispiel: .....	4-62
BPOS (WORD) .....	4-63
Parameter: .....	4-63
Zulässige Speichertypen: .....	4-64
Beispiel: .....	4-64
MSKCMP (WORD, DWORD) .....	4-65
<b>Abschnitt 6: Datenverschiebefunktionen .....</b>	<b>4-68</b>
MOVE (BIT, INT, WORD, REAL) .....	4-69
Parameter: .....	4-70
Zulässige Speichertypen: .....	4-70
Beispiel 1: .....	4-71
Beispiel 2: .....	4-71
BLKMOV (INT, WORD, REAL) .....	4-72
Parameter: .....	4-72
Zulässige Speichertypen: .....	4-73
Beispiel: .....	4-73
BLKCLR (WORD) .....	4-74
Parameter: .....	4-74
Zulässige Speichertypen: .....	4-75
Beispiel: .....	4-75
SHFR (BIT, WORD) .....	4-76
Parameter: .....	4-77
Zulässige Speichertypen: .....	4-77
Beispiel 1: .....	4-78
Beispiel 2: .....	4-78
BITSEQ (BIT) .....	4-79
Speicherbedarf der Bitfolgesteuerung .....	4-79
Parameter: .....	4-80
Zulässige Speichertypen: .....	4-81
Beispiel: .....	4-81
COMMREQ .....	4-82
Befehlsblock .....	4-82
Parameter: .....	4-83
Zulässige Speichertypen: .....	4-83
Beispiel: .....	4-84

<b>Abschnitt 7: Tabellenfunktionen</b> .....	<b>4-85</b>
ARRAY_MOVE (INT, DINT, BIT, BYTE, WORD) .....	4-86
Parameter: .....	4-87
Zulässige Speichertypen: .....	4-87
Beispiel 1: .....	4-88
Beispiel 2: .....	4-88
Beispiel 3: .....	4-89
SRCH_EQ und SRCH_NE (INT, DINT, BYTE, WORD) SRCH_GT und SRCH_LT SRCH_GE und SRCH_LE .....	4-90
Parameter: .....	4-91
Zulässige Speichertypen: .....	4-91
Beispiel 1: .....	4-92
Beispiel 2: .....	4-92
<b>Abschnitt 8: Konvertierungsfunktionen</b> .....	<b>4-93</b>
—>BCD-4 (INT) .....	4-94
Parameter: .....	4-94
Zulässige Speichertypen: .....	4-95
Beispiel: .....	4-95
—>INT (BCD-4) .....	4-96
Parameter: .....	4-96
Zulässige Speichertypen: .....	4-97
Beispiel: .....	4-97
—>DINT (REAL) .....	4-98
Parameter: .....	4-98
Zulässige Speichertypen: .....	4-98
Beispiel: .....	4-99
—>REAL (INT, DINT, BCD-4, WORD) .....	4-100
Parameter: .....	4-100
Zulässige Speichertypen: .....	4-100
Beispiel: .....	4-101
—>WORD (REAL) .....	4-102
Parameter: .....	4-102
Zulässige Speichertypen: .....	4-102
Beispiel: .....	4-103
TRUN (INT, DINT) .....	4-104
Parameter: .....	4-104
Zulässige Speichertypen: .....	4-104
Beispiel: .....	4-105

<b>Abschnitt 9: Steuerfunktionen</b> .....	<b>4-106</b>
CALL .....	4-107
Beispiel: .....	4-107
DOIO .....	4-108
Parameter: .....	4-109
Zulässige Speichertypen: .....	4-109
Eingabebeispiel 1: .....	4-110
Eingabebeispiel 2: .....	4-110
Ausgabebeispiel 1: .....	4-111
Ausgabebeispiel 2: .....	4-111
Erweiterte DOIO-Funktion für CPUs ab Modell 331 aufwärts .....	4-112
END .....	4-113
Beispiel: .....	4-113
MCR .....	4-114
Unterschiede zwischen MCR und JUMP .....	4-115
Beispiel: .....	4-116
ENDMCR .....	4-117
Beispiel: .....	4-117
JUMP .....	4-118
Beispiel: .....	4-119
LABEL .....	4-120
Beispiel: .....	4-120
COMMENT .....	4-121
SVCREQ .....	4-122
Parameter: .....	4-123
Zulässige Speichertypen: .....	4-123
Beispiel: .....	4-123
SVCREQ #6: Anzahl Worte in Prüfsumme ändern/lesen .....	4-124
Aktuelle Anzahl Worte lesen: .....	4-124
Neue Wortanzahl einstellen: .....	4-124
Beispiel: .....	4-125
SVCREQ #7: Echtzeituhr lesen/stellen .....	4-126
Beispiel: .....	4-127
Inhalt des Parameterblocks .....	4-128
Datum und Zeit im BCD-Format lesen/einstellen: .....	4-128
Datum und Zeit im verdichteten ASCII-Format mit eingebetteten Doppelpunkten lesen/ändern .....	4-129
SVCREQ #13: SPS abschalten (anhalten) .....	4-130
Beispiel: .....	4-130
SVCREQ #14: Fehlertabellen löschen .....	4-131

	Beispiel: .....	4-131
	SVCREQ #15: Letzten Fehlertableneintrag lesen .....	4-132
	Beispiel 1: .....	4-133
	Beispiel 2: .....	4-134
	SVCREQ #16: Betriebszeituhr lesen .....	4-136
	Beispiel: .....	4-136
	SVCREQ #18: E/A-Override-Zustand lesen .....	4-137
	Beispiel: .....	4-137
	SVCREQ #23: Master-Prüfsumme lesen .....	4-138
	SVCREQ #26/30: E/A abfragen .....	4-139
	SVCREQ #29: Aufgelaufene Netzausfallzeit lesen .....	4-140
	PID .....	4-141
	Parameter: .....	4-142
	Zulässige Speichertypen: .....	4-142
	PID-Parameterblock: .....	4-143
	Arbeitsweise der PID-Anweisung .....	4-144
	.....	4-145
	Interne Parameter im Adreßfeld (RefArray) .....	4-149
	Einstellung von PID-Algorithmus (PIDISA oder PIDIND) und Verstärkung .....	4-149
	CV-Amplitude und Änderungsgrenzen .....	4-150
	Ausführungsintervall und zeitlicher Ablauf der PID-Blöcke .....	4-151
	Festlegen der Prozeßdaten .....	4-151
	Einstellen der Anwenderparameter und Abgleich der Schleifenverstärkung .....	4-152
	Abgleich nach Ziegler und Nichols .....	4-153
	Beispiel eines PID-Aufrufs .....	4-154
<b>Anhang A</b>	<b>Befehlsausführungszeiten .....</b>	<b>A-1</b>
	Befehlsängen bei CPUs 351 und 352 .....	A-8
<b>Anhang B</b>	<b>Fehlerinterpretation mit der Logicmaster 90-30/20/Micro Software .</b>	<b>B-1</b>
	SPS-Fehlertabelle .....	B-1
	E/A-Fehlertabelle .....	B-7
<b>Anhang C</b>	<b>Mnemonische Programmieranweisungen .....</b>	<b>C-1</b>
<b>Anhang D</b>	<b>Tastenfunktionen .....</b>	<b>D-1</b>
<b>Anhang E</b>	<b>Verwendung von Gleitpunktzahlen .....</b>	<b>E-1</b>
	Gleitpunktzahlen .....	E-1
	Internes Format der Gleitpunktzahlen .....	E-2
	Werte der Gleitpunktzahlen .....	E-3
	Eingabe und Anzeige von Gleitpunktzahlen .....	E-4
	Fehler bei Gleitpunktzahlen und Gleitpunkt-Operationen .....	E-5

Tabelle 2-1 Beiträge zur Zyklusdauer .....	2-4
Tabelle 2-2 Beitrag zur E/A-Zykluszeit durch 90-30-Module (in ms) .....	2-5
Tabelle 2-3 Beitrag zur E/A-Zykluszeit durch 90-30-Module bei Modell 351/352 (in ms) .....	2-6
Tabelle 2-4 Zykluszeitberechnung (Beispiel für eine SPS 90-30 Modell 331) .....	2-7
Tabelle 2-5 Registerreferenzen .....	2-19
Tabelle 2-6 Diskrete Referenzen .....	2-19
Tabelle 2-6 Diskrete Referenzen - Fortsetzung .....	2-20
Tabelle 2-7 Datentypen .....	2-22
Tabelle 2-8 System-Statusreferenzen .....	2-23
Tabelle 2-8 System-Statusreferenzen (Fortsetzung) .....	2-24
Tabelle 2-8 System-Statusreferenzen (Fortsetzung) .....	2-25
Tabelle 2-9 Modell 30 E/A-Module .....	2-37
Tabelle 2-9 Modell 30 E/A-Module (Fortsetzung) .....	2-38
Tabelle 2-10 Modell 20 E/A-Module .....	2-40
Tabelle 2-11 Modelle der SPS Series 90 Micro .....	2-40
Tabelle 3-1 Zusammenfassung der Fehler .....	3-3
Tabelle 3-2 Fehlerwirkungen .....	3-4
Tabelle 4-1 Kontakttypen .....	4-2
Tabelle 4-2 Spulentypen .....	4-3
Tabelle 4-3 SVCREQ-Funktion .....	4-122
Tabelle 4-4 Übersicht PID-Parameter .....	4-143
Tabelle 4-5 PID-Funktionsblockdaten .....	4-146
Tabelle A-1 Befehlsausführungszeiten .....	A-2
Tabelle A-1 Befehlsausführungszeiten – Fortsetzung .....	A-3
Tabelle A-1 Befehlsausführungszeiten – Fortsetzung .....	A-4
Tabelle A-1 Befehlsausführungszeiten – Fortsetzung .....	A-5
Tabelle A-1 Befehlsausführungszeiten – Fortsetzung .....	A-6
Tabelle A-1 Befehlsausführungszeiten – Fortsetzung .....	A-7
Tabelle A-1 Befehlsausführungszeiten – Fortsetzung .....	A-8
Tabelle A-2 Befehlslängen bei CPU 351 und 352 .....	A-8
Tabelle B-1 SPS-Fehlergruppen .....	B-3
Tabelle B-2 SPS-Fehlerwirkungen .....	B-4
Tabelle B-3 Alarm-Fehlercodes für SPS-CPU-Softwarefehler .....	B-4
Tabelle B-4 Alarm-Fehlercodes für SPS-CPU-Fehler .....	B-5
Tabelle B-5 SPS-Fehlerdaten – Unzulässiger Boolescher Operationscode erkannt .....	B-6
Tabelle B-6 SPS-Zeitstempel .....	B-6
Tabelle B-7 E/A-Fehlertabelle, Formatanzeigebyte .....	B-8
Tabelle B-8 E/A-Referenzadresse .....	B-8
Tabelle B-9 Speichertyp der E/A-Referenzadresse .....	B-8
Tabelle B-10 E/A-Fehlergruppen .....	B-9
Tabelle B-11 E/A-Fehlerwirkungen .....	B-10
Tabelle B-12 Fehlerspezifische Daten .....	B-10
Tabelle B-13 E/A-Zeitstempel .....	B-10



Die speicherprogrammierbaren Steuerungen Series 90-30, 90-20 und Micro gehören zur Produktfamilie der speicherprogrammierbaren Steuerungen (SPS) von GE Fanuc. Sie lassen sich einfach installieren und konfigurieren, bieten erweiterte Programmierfunktionen und sind kompatibel zur SPS Series 90-70.

Die SPS Series 90-20 bietet eine kostengünstige Plattform für Anwendungen mit einer geringen Anzahl von E/A-Punkten. Sie ist hauptsächlich auf folgende Punkte ausgerichtet:

- Eine kleine SPS, die man einfach benutzen, installieren, erweitern und warten kann.
- Eine kostengünstige familienkompatible SPS.
- Einfachere Systemintegration durch Standard-Kommunikationshardware und -Protokolle.

Die SPS Series 90 Micro bietet ebenfalls eine kostengünstige Plattform für Anwendungen mit einer geringen Anzahl von E/A-Punkten. Die grundsätzliche Zielsetzung der Micro SPS ist die gleiche wie die der Series 90-20, sie bietet darüberhinaus noch folgende Punkte:

- Bei der Micro SPS sind CPU, Stromversorgung und Ein- und Ausgänge in einem einzigen kleinen Gerät integriert.
- Die meisten Modelle besitzen einen schnellen Zähler.
- Durch die Zusammenfassung von CPU, Stromversorgung und Ein- und Ausgängen in einem Gerät ist sie sehr einfach zu konfigurieren.

Die Softwarestrukturen für die SPS Series 90-30 (außer den Modellen 351/352) und Series 90-20 verwenden eine Architektur, die Speicher- und Ausführungsprioritäten im 80188 Mikroprozessor verwaltet. Die Modelle 351 und 352 benutzen hierfür einen 80386 EX, die SPS Series 90 Micro den H8 Mikroprozessor. Diese Operation unterstützt sowohl die Programmausführung als auch die grundlegenden Organisationsarbeiten, wie zum Beispiel Diagnoseroutinen, E/A-Aktualisierung und Alarmbearbeitung. Die Systemsoftware enthält auch Routinen zum Datenaustausch mit dem Programmiergerät, die für Speichern und Laden der Anwenderprogramme, Rückmeldung von Statusinformationen und Steuerung der SPS verwendet werden können.

In der SPS Series 90-30 wird das Anwenderprogramm, das den an die SPS angeschlossenen Prozeß steuert, durch einen dedizierten Befehlsfolge-Coprozessor (ISCP) gesteuert. Bei den Modellen 313 und höher ist der ISCP hardwaremäßig ausgeführt, bei den Modellen 311 und der Micro SPS wird eine softwaremäßige Ausführung benutzt. Der 80188 Mikroprozessor und der ISCP können simultan arbeiten, wodurch der Mikroprozessor den Datenverkehr abwickeln kann, während der ISCP den Hauptteil des Anwenderprogramms bearbeitet. Der Mikroprozessor muß dabei allerdings die nichtbooleschen Funktionsblöcke bearbeiten.

In den speicherprogrammierbaren Steuerungen Series 90-30/20/Micro liegt ein Fehler vor, wenn bestimmte Ausfälle oder Zustände auftreten, die Betrieb und Leistungsfähigkeit des Systems beeinträchtigen. Diese Zustände können die Fähigkeit des Systems mindern, eine Maschine oder einen Prozeß zu steuern. Andere Zustände können nur als Warnung dienen (z.B. die Mel-

dung einer schwachen Batterie, die anzeigt, daß die zur Speicherpufferung erforderliche Spannung zu niedrig ist und die Batterie ausgewechselt werden muß). Der Ausfall oder Zustand wird *Fehler* genannt.

Fehler werden durch eine Alarmbearbeitungsfunktion in der Software bearbeitet, die den Fehler in die SPS-Fehlertabelle oder in die E/A-Fehlertabelle einträgt. Bei Modell 331 und den CPU-Modellen 340/341 werden die Fehler auch mit einem Zeitstempel versehen. Diese Tabellen können mit den Steuerungs- und Statusfunktionen im SPS-Fehlertabellenmenü bzw. im E/A-Fehlertabellenmenü der Logicmaster 90-30/20/Micro Software angezeigt werden.

### **Weitere Referenzangaben: Siehe Anhänge am Ende des Handbuchs**

In Anhang A finden Sie eine Liste der von den einzelnen Programmieranweisungen belegten Speicherbereichen (in Bytes) und der benötigten Ausführungszeiten (in Mikrosekunden).

Anhang B beschreibt, wie Sie beim Auslesen der SPS- und E/A-Fehlertabellen das Meldungsformat interpretieren müssen.

In Anhang C finden Sie eine Liste der in der Logicmaster 90-30/20/Micro Software für die Anweisungen verwendeten mnemonischen Ausdrücke.

In Anhang D finden Sie eine Liste der in der Logicmaster 90-30/20/Micro Software verwendeten speziellen Tastaturbelegungen.

Anhang E vermittelt Ihnen spezielle Überlegungen und Anweisungen, die beim Einsatz von Gleitpunktarithmetik beachtet werden müssen (nur bei den CPUs 351 und 352 der 90-30).

# *Kapitel* *Systembetrieb*

## *2*

---

---

In diesem Kapitel werden bestimmte Systemoperationen der SPS Series 90-30, 90-20 und Micro beschrieben. Hierzu gehören:

- Eine Zusammenfassung der SPS-Zyklusfolgen (siehe Abschnitt 1).
- Programmorganisation und Anwenderreferenzen/-daten (siehe Abschnitt 2).
- Ein- und Ausschalttroutinen (siehe Abschnitt 3).
- Uhren und Timer (siehe Abschnitt 4).
- Systemsicherheit durch Paßworte (siehe Abschnitt 5).
- Modell 30 E/A-Module (siehe Abschnitt 6).

## Abschnitt 1: SPS-Zyklus – Zusammenfassung

Das Programm in einer SPS Series 90-30, 90-20 oder Micro läuft solange zyklisch ab, bis es durch einen Befehl vom Programmiergerät oder von einem anderen Gerät gestoppt wird. Die für die einmalige Ausführung eines Programms erforderliche Abfolge von Operationen wird Zyklus genannt. Zusätzlich zur Bearbeitung des Programms werden in einem Zyklus noch die Daten von den Eingabegeräten erfaßt, Daten an Ausgabegeräte geschickt, interne Organisationsaufgaben erledigt, das Programmiergerät bedient und andere Kommunikationsaufgaben erledigt.

Normalerweise arbeiten die speicherprogrammierbaren Steuerungen Series 90-30, 90-20 und Micro im Modus *STANDARD-PROGRAMMZYKLUS*. Weitere Betriebsarten sind *STOP MIT GESPERRTER E/A*, *STOP MIT FREIGEGBENER E/A* sowie *KONSTANTER ZYKLUS*. Jede dieser Betriebsarten (die in diesem Kapitel beschrieben werden) wird durch externe Ereignisse und Einstellungen der Anwendungskonfiguration gesteuert. Die SPS wählt die Betriebsart zu Beginn jedes Zyklus aus.

### Standard-Programmzyklus

*STANDARD-PROGRAMMZYKLUS* läuft normalerweise unter allen Bedingungen. Die CPU bearbeitet das Anwenderprogramm, aktualisiert die E/A und führt Kommunikation und andere Aufgaben durch. Dies läuft repetitiv im CPU-Zyklus ab. Die Bearbeitungsfolge des Standard-Programmzyklus besteht aus sieben Teilen:

- (1) Organisation Zyklusstart
- (2) Eingabezyklus (Eingänge lesen)
- (3) Anwenderprogramm bearbeiten
- (4) Ausgabezyklus (Ausgänge aktualisieren)
- (5) Programmiergerät bedienen
- (6) Andere Geräte bedienen
- (7) Diagnoseroutinen

Mit Ausnahme des Programmiergeräte-Dienstes werden all diese Schritte in jedem Zyklus durchlaufen. Das Programmiergerät wird nur bedient, wenn ein Modulfehler erkannt oder eine Bedienanforderung vom Programmiergerät gestellt wurde. Abbildung 2-1 zeigt den Ablauf des Standard-Programmzyklus.

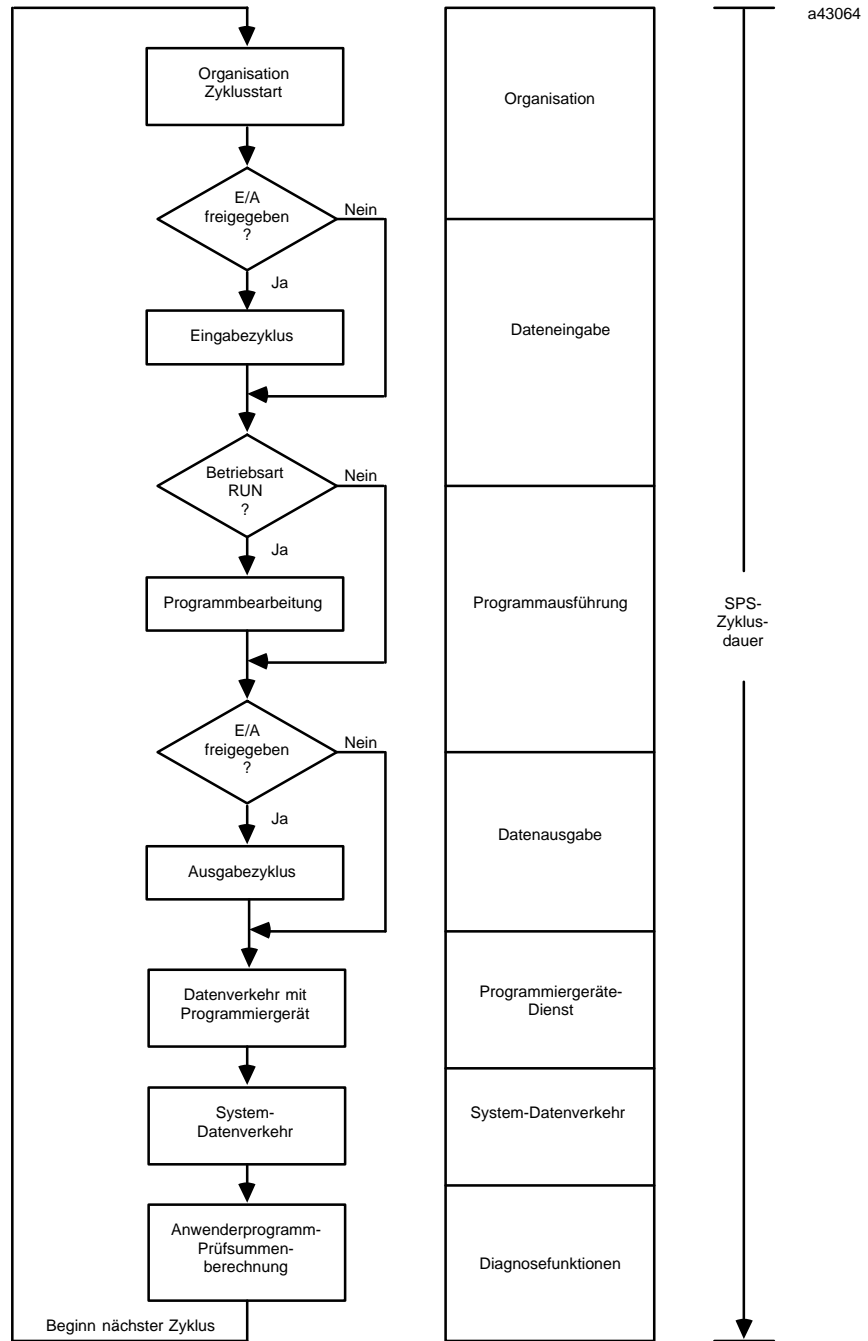


Abbildung 2-1 SPS-Zyklus

Wie Sie in der SPS-Zyklusfolge sehen können, enthält der Zyklus mehrere Elemente, die entsprechend der Darstellung in Tabelle 2-1 zur Gesamt-Zyklusdauer beitragen.

**Tabelle 2-1 Beiträge zur Zyklusdauer**

Zyklus-Element	Beschreibung	Bearbeitungszeit (ms) <sup>4</sup>						
		Micro	211	311/313	331	340/341	351/352	
Organisation	<ul style="list-style-type: none"> <li>● Zyklusdauer berechnen</li> <li>● Start des nächsten Zyklus planen</li> <li>● Modus für nächsten Zyklus festlegen</li> <li>● Fehlerreferenztableaktualisieren</li> <li>● Zeitüberwachungrücksetzen</li> </ul>	0,368	0,898	0,714	0,705	0,424	0,279	
Dateneingabe	Eingangsdaten von Eingangs- und Zusatzmodulen werden empfangen	( <sup>5</sup> )	Siehe Tabelle 2-2 für Beitrag zur Zykluszeit					
Programm-bearbeitung	Anwenderprogramm wird bearbeitet.	Die Bearbeitungszeit ist abhängig von der Programmlänge und den im Programm verwendeten Befehlstypen. Die Befehlsausführungszeiten finden Sie in Anhang A.						
Datenausgabe	Ausgangsdaten werden zu Ausgangs- und Zusatzmodulengeschickt.	0,1656	Siehe Tabelle 2-2 für Beitrag zur Zykluszeit					
Bedienen externer Geräte	Bedianforderung von Programmiergeräten und intelligenten Modulen werden bearbeitet. <sup>1</sup>	HHP	1,93	6,26	4,426	4,524	2,476	0,334
		LM-90	0,380	3,536	2,383	2,454	1,248	0,517
		PCM <sup>2</sup>	—	—	—	3,337	1,943	0,482
Neukonfiguration	Steckplätze mit fehlerhaften Modulen und leere Steckplätze werden überwacht.	— <sup>6</sup>	—	0,458	0,639	0,463	0,319	
Diagnosefunktionen	Überprüfung der Integrität von Anwenderprogramm (die Bearbeitungszeit ist die Zeit, die bei jedem Zyklus für ein in der Prüfsumme eingeschlossenes Wort benötigt wird). <sup>3</sup>	— <sup>7</sup>	0,083	0,050	0,048	0,031	0,010	

1. Der Beitrag externer Geräte zur Zyklusdauer hängt von der Betriebsart des Kommunikationsfensters ab, in dem der Dienst bearbeitet wird. Bei Fenstermodus **LIMITED** [beschränkt] werden für dieses Fenster maximal 6 ms aufgewandt. Bei Fenstermodus **RUN-TO-COMPLETION** [Bearbeitung bis Abschluß] können je nach Anzahl gleichzeitig gestellter Anforderungen bis zu 50 ms für dieses Fenster aufgewandt werden.
2. Diese Messungen wurden durchgeführt mit einem physikalisch vorhandenen PCM, das aber nicht konfiguriert war und in dem keine Anwendertasks abliefen.
3. Die Anzahl der Worte, die in einem Zyklus in die Prüfsumme eingeschlossen werden, kann mit dem Funktionsblock SVCREQ verändert werden.
4. Diese Messungen wurden mit einem leeren Programm und der Standardkonfiguration durchgeführt. Die SPS der Series 90-30 waren in einem leeren Chassis ohne angeschlossene Erweiterungschassis.
5. Die Dateneingabezeit für die Micro SPS kann wie folgt bestimmt werden:  

$$0,365 \text{ ms (fester Zyklus)} + 0,036 \text{ ms (Filterzeit)} \times (\text{Gesamtzykluszeit}) / 0,5 \text{ ms}$$
6. Da die Micro SPS einen festen Satz E/A besitzt, ist eine Neukonfiguration nicht erforderlich.
7. Das Anwenderprogramm der Micro SPS liegt im Flash Memory. Es wird daher nicht auf seine Integrität untersucht.

Tabelle 2-2 Beitrag zur E/A-Zykluszeit durch 90-30-Module (in ms)

Modultyp		CPU Model						
		311/313	331			340/341		
			Hauptchassis	Erweiterungschassis	Dezentrales Chassis	Hauptchassis	Erweiterungschassis	Dezentrales Chassis
Diskrete Eingänge, 8 Punkte		0,076	0,054	0,095	0,255	0,048	0,089	0,249
Diskrete Eingänge, 16 Punkte		0,075	0,055	0,097	0,257	0,048	0,091	0,250
Diskrete Eingänge, 32 Punkte		0,094	0,094	0,126	0,335	0,073	0,115	0,321
Diskrete Ausgänge, 8 Punkte		0,084	0,059	0,097	0,252	0,053	0,090	0,246
Diskrete Ausgänge, 16 Punkte		0,083	0,061	0,097	0,253	0,054	0,090	0,248
Diskrete Ausgänge, 32 Punkte		0,109	0,075	0,129	0,333	0,079	0,114	0,320
Kombinierte Ein-/Ausgänge, 8 Pkt		0,165	0,141	0,218	0,529	0,098	0,176	0,489
Analoge Eingänge, 4 Kanäle		0,151	0,132	0,183	0,490	0,117	0,160	0,462
Analoge Ausgänge, 2 Kanäle		0,161	0,138	0,182	0,428	0,099	0,148	0,392
Schneller Zähler		2,070	2,190	2,868	5,587	1,580	2,175	4,897
APM (1 Achse)		2,330	2,460	3,175	6,647	1,750	2,506	5,899
GCM	Keine Teilnehmer	0,041	0,054	0,063	0,128	0,038	0,048	0,085
	8 64-Punkt-Teiln.	11,420	11,570	13,247	21,288	9,536	10,648	19,485
GCM+	Keine Teilnehmer	0,887	0,967	1,164	1,920	0,666	0,901	1,626
	32 64-Punkt-Teiln.	4,120	6,250	8,529	21,352	5,043	7,146	20,052
PCM 311	Nicht konfiguriert oder keine Anwendung	—	3,350	—	—	1,684	—	—
	128 %R so schnell wie möglich lesen	—	4,900	—	—	2,052	—	—
ADC 311		—	3,340	—	—	1,678	—	—
Analogeingänge (Strom oder Spannung), 16 Kanäle		1,370	1,450	1,937	4,186	1,092	1,570	3,796
S/S-Modul Master	Keine Teilnehmer	1,910	2,030	1,169	1,925	0,678	0,904	1,628
	16 64-Punkt-Teilnehmer	6,020	6,170	8,399	21,291	4,992	6,985	20,010
S/S-Modul Slave	32 Punkte	,206	0,222	0,289	0,689	0,146	0,226	0,636
	64 Punkte	0,331	0,350	0,409	1,009	0,244	0,321	0,926

**Tabelle 2-3 Beitrag zur E/A-Zykluszeit durch 90-30-Module bei Modell 351/352  
(in ms)**

Modultyp	CPU			
	351/352			
	Hauptchassis	Erweiterungschassis	Dezentrales Chassis	
Diskrete Eingänge, 8 Punkte	0,030	0,055	0,206	
Diskrete Eingänge, 16 Punkte	0,030	0,055	0,206	
Diskrete Eingänge, 32 Punkte	0,043	0,073	0,269	
Diskrete Ausgänge, 8 Punkte	0,030	0,053	0,197	
Diskrete Ausgänge, 16 Punkte	0,030	0,053	0,197	
Diskrete Ausgänge, 32 Punkte	0,042	0,070	0,259	
Kombinierte diskrete Ein-/Ausgänge	0,060	0,112	0,405	
Analoge Eingänge, 4 Kanäle	0,075	0,105	0,396	
Analoge Ausgänge, 2 Kanäle	0,058	0,114	0,402	
Analoge Eingänge (Strom oder Spannung), 16 Kanäle	0,978	1,446	3,999	
Analoge Ausgänge, 8 Kanäle	1,274	1,988	4,472	
Kombinierte analoge Ein-/Ausgänge	1,220	1,999	4,338	
Schneller Zähler	1,381	2,106	5,221	
Power Mate APM (1 Achse)	1,527	2,581	6,388	
E/A-Prozessor	1,574	2,402	6,388	
Ethernet-Schnittstelle (kein Anschluß)	0,038	0,041	0,053	
GCM	Keine Teilnehmer	0,911	1,637	5,020
	8 64-Punkt-Teiln.	8,826	16,932	21,179
GCM+	Keine Teilnehmer	0,567	0,866	1,830
	32 64-Punkt-Teiln.	1,714	2,514	5,783
GBC	Keine Teilnehmer	0,798	1,202	2,540
	32 64-Punkt-Teiln.	18,382	25,377	70,777
PCM 311	Nicht konfiguriert oder keine Anwendung	0,476	—	—
	128 %R so schnell wie möglich lesen	0,485	—	—
ADC (keine Task)	0,476	—	—	
Schnittstellenmodul Master	Keine Teilnehmer	0,569	0,865	1,932
	16 Teilnehmer mit 64 Punkten	4,948	7,003	19,908
Schnittstellenmodul Slave	32 Punkte	0,087	0,146	0,553
	64 Punkte	0,154	0,213	0,789



## Berechnung der Zykluszeit

In Tabelle 2-1 sind sieben Elemente dargestellt, die zur SPS-Zykluszeit beitragen. Die Zykluszeit setzt sich zusammen aus festen (Organisation und Diagnosefunktionen) und aus variablen Zeiten. Die variablen Zeiten ändern sich je nach E/A-Konfiguration, Programmgröße und dem Typ des an der SPS angeschlossenen Programmiergerätes.

### Beispiel einer Zykluszeitberechnung

Tabelle 2-4 zeigt an einem Beispiel, wie Sie die Zykluszeit einer SPS Series 90-30 Modell 331 berechnen können.

Für diese Berechnung wurden folgende Module und Anweisungen zugrunde gelegt:

- Eingangsmodule: Fünf Modell 30 Eingangsmodule mit 16 Punkten.
- Ausgangsmodule: Fünf Modell 30 Ausgangsmodule mit 16 Punkten.
- Programmieranweisungen: Ein Programm aus 1200 Schritten mit 700 Booleschen Anweisungen (LD, AND, OR, etc.), 300 Ausgangsmerkern (OUT, OUTM, etc.), und 200 arithmetischen Funktionen (ADD, SUB, usw.).

**Tabelle 2-4 Zykluszeitberechnung (Beispiel für eine SPS 90-30 Modell 331)**

Zykluselement	Berechnung	Zeitanteil		
		Ohne Programmiergerät	mit HHP	mit LM90
Organisation	0,705 ms	0,705 ms	0,705 ms	0,705 ms
Dateneingabe	$0,055 \times 5 = 0,275$ ms	0,275 ms	0,275 ms	0,275 ms
Programmbearbeitung	$700 \times 0,4 \mu\text{s} + 300 \times 0,5 \mu\text{s} + 200 \times 51,2 \mu\text{s} = 10,7$ ms	10,7 ms	10,7 ms	10,7 ms
Datenausgabe	$0,061 \times 4 = 0,244$ ms	0,244 ms	0,244 ms	0,244 ms
Programmiergerät bedienen	0,4 ms + Programmiergerätezeit + 0,6 ms	0 ms	4,524 ms	2,454 ms
Andere Geräte bedienen	Nicht in unserem Beispiel	0 ms	0 ms	0 ms
Neukonfiguration	0,639 ms	0,639 ms	0,639 ms	0,638 ms
Diagnosefunktionen	0,048 ms	0,048 ms	0,048 ms	0,048 ms
SPS-Zykluszeit	Organisation + Dateneingabe + Programmbearbeitung + Datenausgabe + Programmiergerät bedienen + andere Geräte bedienen + Diagnosefunktionen	12,611 ms	17,135 ms	15,065 ms

### Organisation

Im Organisationsteil des Zyklus werden alle Tasks ausgeführt, die für die Vorbereitung des Zyklusstarts benötigt werden. Befindet sich die SPS in der Betriebsart KONSTANTER ZYKLUS, wird der Zyklus solange verzögert, bis die eingestellte Zykluszeit verstrichen ist. War diese Zeit bereits verstrichen, wird der Kontakt OV\_SWP %SA0002 gesetzt und der Zyklus wird unverzüglich fortgesetzt. Als nächstes werden die Timerwerte (1/100, 1/10 und 1 s) aktualisiert, indem der Zeitunterschied zum vorherigen Zyklus und die neue Zykluszeit berechnet werden. Damit keine Genauigkeit verloren geht wird er tatsächliche Zyklusbeginn in Schritten von 100 ms aufgezeichnet. Jeder Timer besitzt ein Restfeld mit der Anzahl von 100-ms-Schritten, die seit der letzten Erhöhung des Timerwerts aufgelaufen sind.

## Eingabezyklus

Die Eingänge werden im Eingabeteil des Zyklus abgefragt, unmittelbar vor der Programmbearbeitung. Während dieses Zyklusteils werden alle Modell 30 Eingangsmodule abgefragt und ihre Daten im %I- (diskrete Eingänge) bzw. %AI- (analoge Eingänge) Speicher abgelegt. Von einem Genius-Kommunikationsmodul empfangene Globaldaten werden im %G-Speicher abgelegt.

Die Module werden in steigender Reihenfolge der Referenzadressen abgefragt, beginnend mit dem Genius-Kommunikationsmodul, dann den diskreten Eingangsmodulen und zuletzt den analogen Eingangsmodulen.

Ist die CPU in STOP-Modus und wurde sie so konfiguriert, daß sie im STOP-Modus die Eingänge nicht abfragt, wird der Eingabezyklus übersprungen.

## Anwenderprogramm-Bearbeitungszyklus

Im Programm-Bearbeitungszyklus findet die eigentliche Programmausführung statt. Die Programmbearbeitung beginnt immer mit der ersten Anweisung im Anwenderprogramm, unmittelbar nach Beendigung des Eingabezyklus, und ergibt einen neuen Satz Ausgabedaten. Die Programmbearbeitung endet, wenn die Anweisung END ausgeführt wird.

Das Anwenderprogramm wird von ISCP und 80C188 Mikroprozessor bearbeitet. Bei den CPUs von Modell 313 aufwärts führt der ISCP die Booleschen Anweisungen aus, während der 80C188 oder 80386EX Timer, Zähler und Funktionsblöcke bearbeitet. Bei den CPUs der Modelle 311 und bei der Series 90-20 bearbeitet der 80C188 alle Booleschen Anweisungen, Timer, Zähler und Funktionsblöcke. In der Micro bearbeitet der H8-Prozessor alle Booleschen Anweisungen und Funktionsblöcke.

Anhang A enthält eine Liste der Ausführungszeiten der einzelnen Programmierfunktionen.

## Ausgabezyklus

Die Aktualisierung der Ausgänge erfolgt unmittelbar nach der Programmbearbeitung im Ausgabezyklus. Hierzu werden Daten aus dem %Q- (für digitale Ausgänge) bzw. dem %AQ- (für analoge Ausgänge) Speicher verwendet. Wurde das Genius-Kommunikationsmodul für das Aussenden von Globaldaten konfiguriert, werden Daten aus dem %G-Speicher zum GCM geschickt. Bei Series 90-20 und Micro werden nur digitale Ausgänge aktualisiert.

Im Ausgabezyklus werden alle Modell 30 Ausgangsmodule in steigender Adreßreihenfolge aktualisiert.

Ist die CPU in STOP-Modus und wurde sie so konfiguriert, daß sie im STOP-Modus die Ausgänge nicht aktualisiert, wird der Eingabezyklus übersprungen. Der Ausgabezyklus ist beendet, wenn alle Ausgabedaten zu allen Modell 30 Ausgangsmodulen geschickt wurden.

## Programm-Prüfsummenberechnung

Am Ende jedes Zyklus wird das Programm einer Prüfsummenberechnung unterzogen. Da eine Berechnung der Prüfsumme für das ganze Programm zu viel Zeit in Anspruch nehmen würde, können Sie im CPU-Detailmenü die Anzahl der Worte (0 bis 32) festlegen, die in die Prüfsumme einbezogen werden.

Stimmt die berechnete Prüfsumme nicht mit der Referenz-Prüfsumme überein, wird ein Programm-Prüfsummenfehler-Merker gesetzt. Hierdurch erfolgt ein Fehlereintrag in die SPS-Fehlertabelle und der SPS-Modus wechselt auf STOP. Das Programmiergeräte-Kommunikationsfenster wird durch einen Fehler in der Prüfsummenberechnung nicht beeinträchtigt.

## Programmiergeräte-Kommunikationsfenster

Dieser Teil des Zyklus ist für den Datenaustausch mit dem Programmiergerät bestimmt. Wurde ein Programmiergerät angeschlossen oder benötigt ein Modul im System eine Neukonfiguration (wurde während der Abfrage nach fehlerhaften Modulen im Zyklus erkannt), führt die CPU das Programmiergeräte-Kommunikationsfenster aus. Dieses Fenster bleibt geschlossen, wenn kein Programmiergerät angeschlossen ist und kein Modul neu konfiguriert werden muß. Pro Zyklus wird jeweils nur ein Module konfiguriert.

Unterstützung wird gegeben für das Hand-Programmiergerät und für andere Programmiergeräte, die über das Series Ninety Protocol (SNP) an den seriellen Port angeschlossen werden können, sowie für den Datenaustausch zwischen Programmiergerät und intelligenten Zusatzmodulen.

In dem standardmäßig eingestellten begrenzten Fenstermodus führt die CPU bei jedem Zyklus eine Operation für das Programmiergerät durch. Das heißt, sie antwortet auf eine Bedienanforderung oder auf einen Tastendruck. Stellt das Programmiergerät eine Anforderung, deren Bearbeitung mehr als 6 ms (oder, je nach CPU, 8 ms – siehe Hinweis) in Anspruch nimmt, wird die Bearbeitung dieser Anforderung so über mehrere Zyklen verteilt, daß jeder Zyklus nur mit maximal 6 ms (oder, je nach CPU, 8 ms – siehe Hinweis) betroffen ist.

### Hinweis

Die Maximaldauer für das Kommunikationsfenster beträgt 6 ms für CPUs ab 340 aufwärts, bzw. 8 ms für die Modelle 311, 313, 323 und 331. Beim Speichern im RUN-Modus kann dieser Wert bis zu 12 ms betragen (s. Seite 2-12).

Abbildung 2-2 zeigt, wie der Programmiergeräte-Kommunikationsteils des Zyklus abläuft.

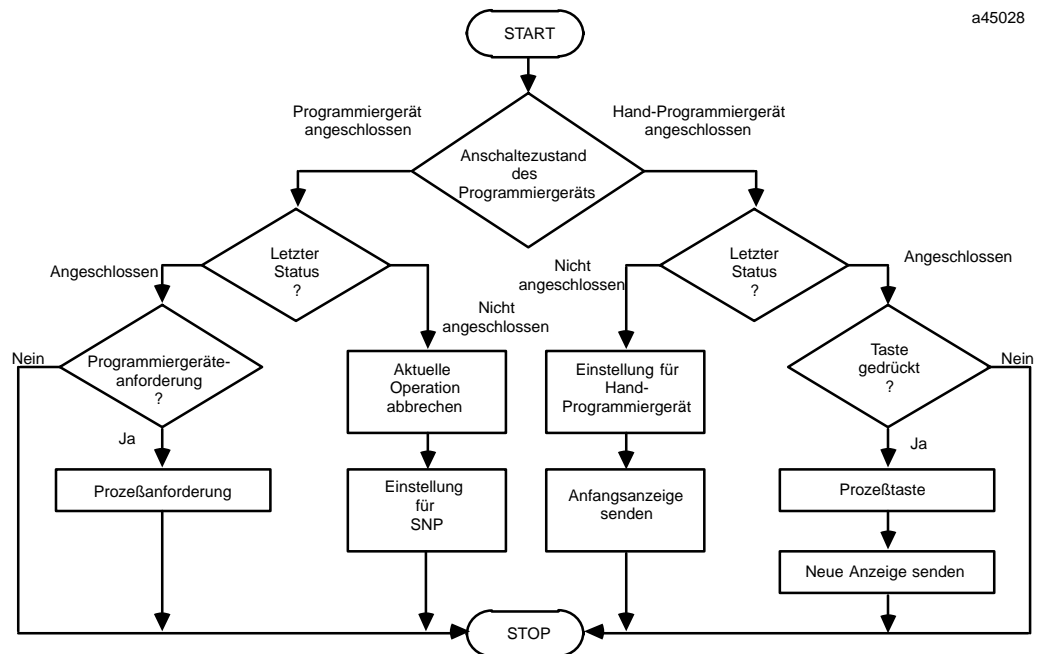


Abbildung 2-2 Programmiergeräte-Kommunikationsfenster – Ablaufdiagramm

### System-Kommunikationsfenster (ab Modell 331 aufwärts)

In diesem Zyklusabschnitt werden die Kommunikationsanforderungen von intelligenten Zusatzmodulen (z.B. PCM) bearbeitet (siehe Blockdiagramm). Die Anforderungen werden in der Reihenfolge ihres Auftretens bearbeitet. Da intelligente Zusatzmodule aber mit umlaufender Berechtigung abgefragt werden, besitzt kein intelligentes Zusatzmodul eine Priorität über ein anderes.

Im standardmäßig eingestellten Modus BEARBEITUNG BIS ABSCHLUSS ist die Länge des System-Kommunikationsfenster auf 50 Millisekunden beschränkt. Stellt ein intelligentes Zusatzmodul eine Anforderung, deren Bearbeitung mehr als 50 ms in Anspruch nimmt, wird die Anforderung so über mehrere Zyklen verteilt, daß kein Zyklus mit mehr als 50 ms daran beteiligt ist.

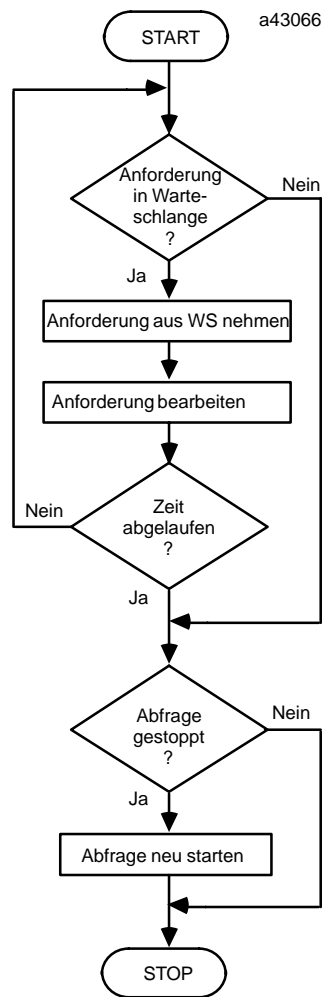


Abbildung 2-3 System-Kommunikation – Ablaufdiagramm

### PCM-Kommunikation mit der SPS (ab Modell 331 aufwärts)

Intelligente Zusatzmodule (z.B. das PCM) können die CPU nicht unterbrechen, wenn Sie bedient werden wollen. Die CPU muß die einzelnen intelligenten Zusatzmodule zyklisch nach Bedienanforderungen abfragen. Diese Abfrage findet während des Zyklus asynchron im Hintergrund statt (siehe Ablaufdiagramm in Abbildung 2-4).

Wird ein intelligentes Zusatzmodul abgefragt und sendet der CPU eine Bedienanforderung, wird die Anforderung zur Bearbeitung im System-Kommunikationsfenster in eine Warteschlange eingereiht.

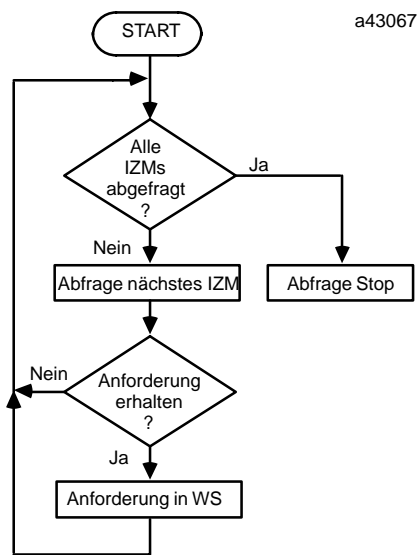


Abbildung 2-4 PCM-Kommunikation mit der SPS

## Varianten des Standard-Programmzyklus

Über die normale Ausführung des Standard-Programmzyklus hinaus können bestimmte Varianten auftreten oder erzwungen werden. Diese Varianten, die in den nächsten Absätzen beschrieben werden, können aus der Programmiersoftware heraus angezeigt und verändert werden.

### Betriebsart KONSTANTE ZYKLUSZEIT

Im Standard-Programmzyklus wird jeder Zyklus so schnell wie möglich abgearbeitet, wobei die einzelnen Zyklen unterschiedlich lang sind. Eine Alternative hierzu ist KONSTANTE ZYKLUSZEIT, bei der jeder Zyklus gleich lang ist. Verwenden Sie diese Betriebsart, wenn Sie E/A-Punkte oder Register zyklisch in gleichen Zeitabständen abfragen wollen (z.B. bei Steuerungsalgorithmen). Sie müssen hierzu die konfigurierte konstante Zykluszeit einstellen, die dann Standard-Zyklusbetriebsart wird und bei jedem Wechsel der SPS von STOP auf RUN zum Tragen kommt. Sie können den Timer für konstante Zykluszeit auf einen Wert zwischen 5 und 200 ms (500 ms bei den CPUs 351 und 352) einstellen. Die Standardeinstellung ist 100 ms.

Wegen Schwankungen in der für die verschiedenen Teile des SPS-Zyklus benötigten Zeit sollten Sie die konstante Zykluszeit auf einen Wert einstellen, der mindestens 10 Millisekunden höher ist als der auf der Statuszeile für die Betriebsart NORMALZYKLUS angezeigte Wert. Hierdurch verhindern Sie, daß Zyklusüberschreitungsfehler im Übermaß auftreten.

Die Betriebsart KONSTANTE ZYKLUSZEIT kann zum Beispiel verwendet werden, wenn die E/A in konstanten Zeitabständen aktualisiert werden muß, oder wenn zwischen Ausgabezyklus und nächstem Eingabezyklus eine bestimmte Zeitspanne verstreichen muß, damit die Eingänge nach dem Empfang von Ausgabedaten aus dem Programm einschwingen können.

Läuft der Timer für konstante Zykluszeit ab, ehe der Zyklus abgeschlossen ist, wird der gesamte Zyklus einschließlich der Fenster beendet. Zu Beginn des nächsten Zyklus wird jedoch ein Zyklusüberschreitungsfehler eingetragen.

### Hinweis

Im Gegensatz zur aktiven konstanten Zykluszeit, die nur im RUN-Modus editiert werden kann, kann die konfigurierte konstante Zykluszeit nur im STOP-Modus editiert werden. Die gemachten Änderungen werden erst wirksam, nachdem Sie die Funktion "Konfiguration vom Programmiergerät in die SPS speichern" ausgeführt haben. Nach der Speicherung wird dies zur Standard-Zyklusbetriebsart.

### SPS-Zyklus in STOP-Modus

Das Anwenderprogramm wird nicht bearbeitet, wenn die SPS im STOP-Modus ist. Sie können wählen, ob die E/A aktualisiert wird. Wurde der Parameter **E/A-Aktualisierung – STOP** im CPU-Detailmenü auf **Ja** eingestellt, kann E/A-Aktualisierung im STOP-Modus durchgeführt werden. Der Datenaustausch mit Programmiergerät und intelligenten Zusatzmodulen wird fortgesetzt. Im STOP-Modus wird die Abfrage fehlerhafter Module und die Modul-Neukonfiguration fortgeführt. Die vom Betriebssystem benutzten Zeitscheiben sind etwas größer als im RUN-Modus (normalerweise etwa 50 ms pro Fenster).

### Kommunikationsfenster-Betriebsarten

Standardmäßig ist das Programmiergeräte-Kommunikationsfenster im "begrenzten" Modus. Das bedeutet, daß jede Anforderung, deren Bearbeitung mehr als 6 ms in Anspruch nimmt, so

über mehrere Zyklen hinweg bearbeitet wird, daß kein Zyklus mehr als 6 ms lang in Anspruch genommen wird. Bei den CPUs 313, 323 und 331 kann beim Speichern im RUN-Modus der Zyklus 12 ms lang beansprucht werden. Der aktive Fenstermodus kann über das Logicmaster-Menü "Zyklussteuerung" verändert werden. Hinweise zur Änderung des aktiven Fenstermodus finden Sie in Kapitel 5 "SPS-Steuerung und Status" von *Logicmaster 90™ Series 90™-30/20/ Micro Programmiersoftware, Anwenderhandbuch* (GFK-0466).

### Hinweis

Wird der Systemfenstermodus auf "begrenzt" umgeschaltet, wird zwar die Auswirkung von Zusatzmodulen (z.B. PCM oder GBC), die mit der SPS über das Systemfenster Daten austauschen, auf die Zykluszeit reduziert, die Reaktion auf deren Anforderungen wird aber auch langsamer.

## Schlüsselschalter an CPUs 351/352: Betriebsartenwechsel und Schutz von Flash Memory

Über einen Schlüsselschalter auf der Vorderseite der CPUs 351 und 352 können Sie für den Flash Memory Schreibschutz einstellen. Stellen Sie hierzu den Schalter auf ON/RUN. Der Flash Memory kann nur überschrieben werden, wenn der Schalter wieder auf OFF zurückgestellt wird.

Ab Ausgabestand 7 der CPUs 351 und 352 besitzt dieser Schlüsselschalter noch eine weitere Funktion: Mit ihm können Sie die SPS in STOP- oder RUN-Modus schalten und nicht fatale Fehler löschen (siehe nächsten Abschnitt).

Ab Ausgabestand 8 der CPUs 351 und 352 besitzt dieser Schlüsselschalter eine erweiterte Speicherschutzfunktion: Sie können nun zwei zusätzliche Arten von Speicherschutz aktivieren (siehe Abschnitt "Speicherschutz ab Ausgabestand 8").

### Schlüsselschalter ab Ausgabestand 7

Im Gegensatz zu den bei früheren Ausgabeständen verfügbaren Schutzmechanismen für Flash Memory, müssen Sie hier erst den Schlüsselschalter über den Parameter "RUN/STOP-Schlüsselschalter" im Konfigurationsmenü der CPU freigeben, damit die CPU über die hier beschriebenen erweiterten Steuerfunktionen verfügen kann.

Beim Übergang der SPS in RUN-Modus verfügt der Schlüsselschalters über die gleichen Schutz- und Prüfmechanismen wie bisher. Das heißt, die SPS kann über den Schlüsselschalter nicht in RUN-Modus geschaltet werden, wenn sie in der Betriebsart STOP/FEHLER ist. Sie können aber nicht fatale Fehler löschen und die SPS dann über den Schlüsselschalter in RUN-Modus schalten.

Enthalten die Fehlertabellen **nicht fatale Fehler** (dies sind Fehler, die die CPU nicht in den Modus STOP/FEHLER versetzen), geht die CPU in den RUN-Modus, wenn Sie den Schlüsselschalter das erste Mal von STOP auf RUN schalten. Die Fehlertabellen werden **nicht** gelöscht.

Enthalten die Fehlertabellen **fatale Fehler** (CPU in STOP/FEHLER-Modus), beginnt die Anzeige CPU RUN beim ersten Umschalten des Schlüsselschalters von STOP auf RUN mit einer Frequenz von 2 Hz zu blinken. Gleichzeitig wird ein 5-Sekunden-Timer angestoßen. Die blinkende RUN-Anzeige meldet, daß die Fehlertabellen fatale Fehler enthalten. In diesem Fall geht die CPU nicht in den RUN-Zustand, selbst wenn der Schlüsselschalter auf RUN steht.

### Fehlertabelle mit Schlüsselschalter löschen

Wenn Sie den Schlüsselschalter innerhalb der 5 Sekunden, die die RUN-Anzeige blinkt, von RUN auf STOP und zurück auf RUN schalten, werden die Fehler gelöscht und die CPU geht in

den RUN-Modus. Die Anzeige hört auf zu blinken und leuchtet dauernd. Der Schalter muß hierbei aber mindestens 1/2 Sekunde lang in der Stellung RUN oder STOP gehalten werden.

### Hinweis

Ist der 5-Sekunden-Timer abgelaufen (die RUN-Anzeige blinkt nicht mehr), bleibt die CPU im Zustand STOP/FEHLER mit Fehlern in der Fehlertabelle. Schalten Sie dann den Schlüsselschalter wieder von STOP auf RUN, wird der Prozeß wiederholt und dies ist der erste Übergang.

Die nachstehende Tabelle gibt eine Zusammenfassung, wie sich die Einstellungen der beiden auf den Schlüsselschalter wirkenden CPU-Parameter (R/S-Schalter und IOScan-Stop) sowie die Einstellung des Schlüsselschalters auf die SPS auswirken.

Parameter R/S-Schalter in CPU-Konfiguration	Stellung des Schlüsselschalters	Parameter IOScan-Stop in CPU-Konfiguration	SPS-Betrieb
AUS	X	X	Alle SPS-Programmiergeräte-Betriebsarten sind erlaubt
EIN	ON/RUN	X	Alle SPS-Programmiergeräte-Betriebsarten sind erlaubt
EIN	OFF/STOP	X	SPS kann nicht auf RUN gehen.
EIN	Schlüsselschalterumschalten von OFF/STOP auf ON/RUN	X	SPS geht auf RUN, wenn keine fatalen Fehler vorhanden sind; andernfalls blinkt RUN LED 5 Sekunden lang.
EIN	Schlüsselschalterumschalten von ON/RUN auf OFF/STOP	Nein	SPS geht auf STOP-NO IO
EIN	Schlüsselschalterumschalten von ON/RUN auf OFF/STOP	Ja	SPS geht auf STOP-IO

X = irrelevant

## Erweiterter Speicherschutz bei CPUs ab Ausgabestand 8

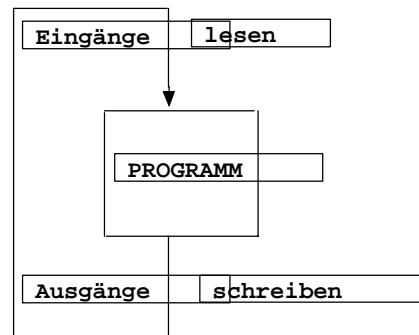
Bei CPUs ab Ausgabestand 8 kann der Schlüsselschalter, zusätzlich zu den zuvor besprochenen Eigenschaften, zum Schutz des RAM eingesetzt werden, so daß der RAM über die Programmiersoftware nicht verändert werden kann. Hierzu muß im Programmierpaket ein Parameter entsprechend eingestellt werden. Bei freigeschaltetem Speicherschutz sind zwei Operationstypen gesperrt: Anwenderprogramm und Konfiguration können nicht verändert werden und Punktdaten dürfen nicht fixiert oder überschrieben werden. Diese Sperre wird aktiviert über das Speicherschutzfeld im Logicmaster-Konfigurationsmenü für die CPU-Module 351 bzw. 352 oder über das Speicherschutzfeld bei den Einstellungen für die CPU-Module 351 bzw. 352 in der Hardwarekonfiguration von CIMPLICITY Control. In beiden Softwarepaketen ist die Standardeinstellung "gesperrt".



## Abschnitt 2: Programmorganisation und Anwenderreferenzen/-daten

Bei der SPS Series 90-30 ist die maximale Programmgröße 6 kB für die Modelle 311 und 313, 16 kB für Modell 331, 32 kB für Modell 340 und 80 kB für Modelle 341, 351 oder 352. Bei der SPS Series 90-20 ist die maximale Programmgröße 2 kB für Modell 211. Bei der SPS Series 90 Micro ist die maximale Programmgröße 6 kB. Bei einer 28-Punkt-Micro sind es 12 kB.

Das Anwenderprogramm enthält Logik, die benutzt wird, wenn das Programm gestartet wird. Maximal sind in einem Programmblock (Haupt- oder Unterprogramm) 3000 Strompfade möglich. Das Programm wird von der SPS repetitiv abgearbeitet.



Eine Liste der bei den einzelnen CPU-Modellen möglichen Programmgrößen und Referenzbereiche finden Sie in GFK-0356, *SPS Series 90-30, Anwenderhandbuch*, und in GFK-0551, *SPS Series 90-20, Anwenderhandbuch*.

Alle Programme beginnen mit einer Variablenvereinbarungstabelle, in der die im Anwenderprogramm zugewiesenen symbolischen Adressen und Referenzbeschreibungen aufgelistet sind.

Mit dem Blockvereinbarungseditor können im Hauptprogramm vereinbarte Unterprogrammblöcke aufgelistet werden.

### Unterprogrammblöcke (nur SPS Series 90-30)

Ein Programm kann während der Bearbeitung Unterprogramme aufrufen. Ehe ein Unterprogramm mit einem Befehl CALL aufgerufen werden kann, muß es über den Blockvereinbarungseditor vereinbart werden. In jedem Programmblock sind bis zu 64 Unterprogrammblockvereinbarungen und 64 CALL-Befehle möglich. Die Maximalgröße eines Unterprogramms beträgt 16 kB oder 3000 Strompfade, wobei Hauptprogramm und alle Unterprogramme die durch das CPU-Modell vorgegebenen Beschränkungen (maximale Größe des Programmspeichers) einhalten müssen.

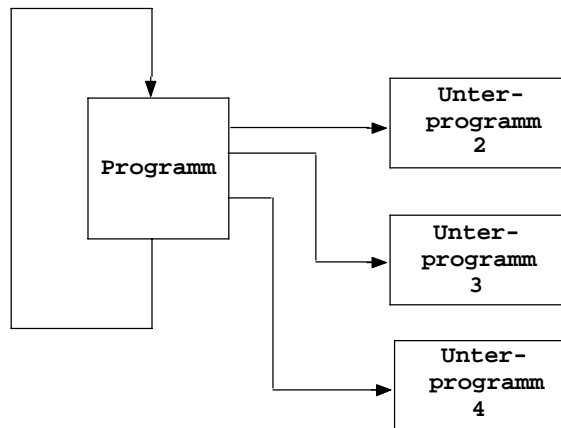
### Hinweis

Weder bei der Series 90-20 noch bei der Micro gibt es Unterprogrammblöcke.

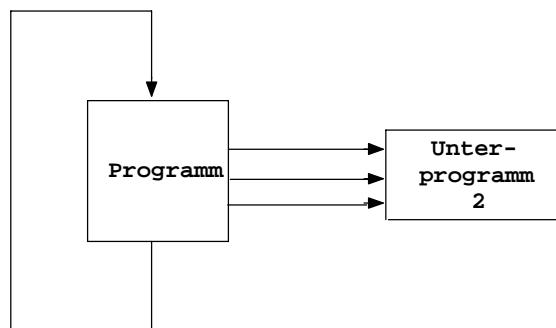
Der Einsatz von Unterprogrammen erfolgt wahlweise. Die Aufteilung eines Programms in kleinere Unterprogramme kann die Programmierung vereinfachen und das Programm verkleinern.

### Beispiele von Unterprogrammblöcken

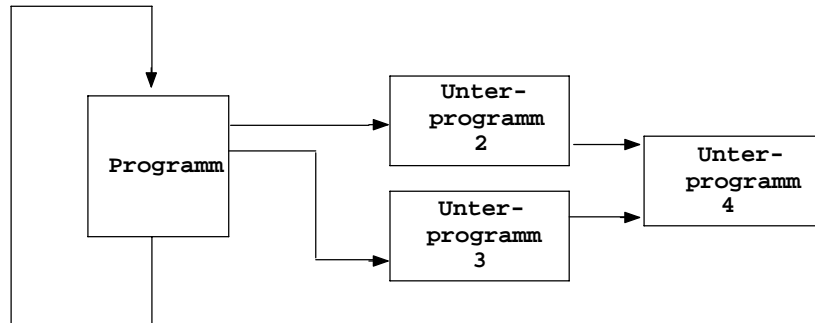
Die Logik eines Programms kann zum Beispiel in drei Unterprogramme aufgeteilt werden. Jedes dieser Unterprogramme kann dann vom Programm entsprechend den Anforderungen aufgerufen werden. In diesem Beispiel kann das Programm nur soviel Logik enthalten, wie zur Steuerung der Unterprogrammabfolge erforderlich ist.



Während des Programmablaufs kann ein Unterprogramm immer wieder aufgerufen werden. Programmteile, die in einem Programm mehrmals ausgeführt werden müssen, sollten daher in einem Unterprogrammblock stehen, der dann jeweils aufgerufen werden kann. Hierdurch wird die Programmlänge reduziert.



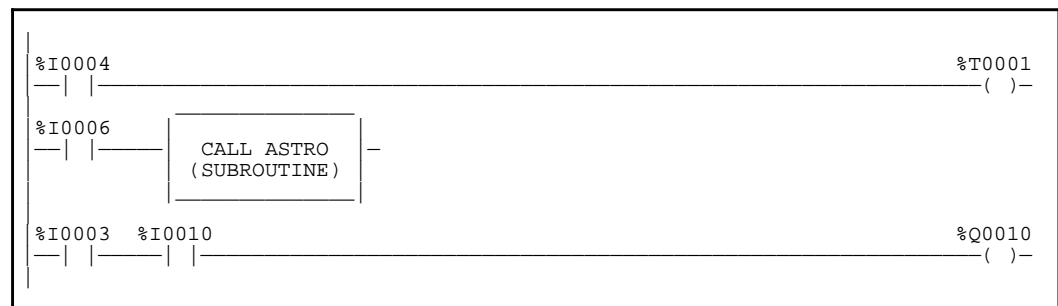
Ein Unterprogramm kann vom Hauptprogramm oder von einem anderen Unterprogramm aus aufgerufen werden. Ein Unterprogramm kann sogar sich selbst aufrufen.



Die SPS trägt einen Fehler "Anwenderstack-Überlauf" ein und geht in die Betriebsart STOP/FEHLER, wenn mehr als acht verschachtelte Aufrufe gemacht werden. Bei der Zählung der Verschachtelungsebenen gilt das Programm als Ebene 1.

### Wie Unterprogrammblöcke aufgerufen werden

Ein Unterprogrammblock wird ausgeführt, wenn er von der Programmlogik im Hauptprogramm oder von einem anderen Block aufgerufen wird.



In diesem Beispiel sehen Sie den Befehl CALL, wie er im aufrufenden Block erscheint.

### Periodische Unterprogramme

Ab Version 4.20 werden bei CPUs ab Modell 340 aufwärts periodische Unterprogramme unterstützt. Beachten Sie bitte folgende Einschränkungen:

- (1) Zeitfunktionsblöcke (TMR, ONDTR und OFDTR) werden in einem periodischen Unterprogramm nicht korrekt ausgeführt. Ein DOIO-Funktionsblock in einem periodischen Unterprogramm, zu dessen Referenzbereich Referenzen gehören, die einem intelligenten E/A-Modul (HSC, Power Mate APM, Genius, usw.) zugeordnet sind, bewirkt, daß die CPU die Datenverbindung zu dem Modul verliert. Die Kontakte FST\_SCN und LST\_SCN (%S1 und %S2) nehmen bei der Ausführung des periodischen Unterprogramms einen Zwischenwert an. Ein periodisches Unterprogramm kann nicht von anderen Unterprogrammen aufgerufen werden.

- 
- (2) Befindet sich im Hauptchassis kein PCM-, CMM- oder ADC-Modul, beträgt die Latenzzeit (d.h. die maximale Zeitspanne zwischen dem Zeitpunkt, zu dem das periodische Unterprogramm ablaufen sollte und dem Zeitpunkt, zu dem es tatsächlich abläuft) des periodischen Unterprogramms etwa 0,35 Millisekunden. Befindet sich ein PCM-, CMM- oder ADC-Modul im Hauptchassis (selbst wenn es nicht konfiguriert oder benutzt wird), kann die Latenzzeit bis zu 2,25 Millisekunden betragen. Aus diesem Grund wird der Einsatz von periodischen Unterprogrammen bei Produkten auf PCM-Basis nicht empfohlen.

## Anwenderreferenzen

Die in einem Anwenderprogramm verwendeten Daten werden entweder als diskrete Referenzen oder als Registerreferenzen gespeichert.

**Tabelle 2-5 Registerreferenzen**

Typ	Beschreibung
%R	Mit dem Präfix %R werden System-Registerreferenzen zugewiesen, in die Programmdateien abgespeichert werden können (z.B. Berechnungsergebnisse).
%AI	Das Präfix %AI steht für ein analoges Eingangsregister. Ihm folgt die Registeradresse der Referenz (z.B. %AI0015). In einem analogen Eingangsregister steht der Wert eines Analogeingangs oder ein anderer Wert.
%AQ	Das Präfix %AQ steht für ein analoges Ausgangsregister. Ihm folgt die Registeradresse der Referenz (z.B. %AQ0056). In einem analogen Ausgangsregister steht der Wert eines Analogausgangs oder ein anderer Wert.

## Hinweis

Sämtliche Registerreferenzen bleiben erhalten, wenn die Versorgungsspannung der CPU ab- und wieder eingeschaltet wird.

**Tabelle 2-6 Diskrete Referenzen**

Typ	Beschreibung
%I	Das Präfix %I steht für diskrete Eingänge. Ihm folgt die Referenzadresse in der Eingangstabelle (z.B. %I00121). Die Referenzen %I liegen in der Eingangs-Statustabelle, in der die Zustände sämtlicher Eingangssignale gespeichert sind, die von den Eingangsmodulen im letzten Eingabezyklus empfangen wurden. Den diskreten Eingangsmodulen wird mit der Konfigurationssoftware oder dem Hand-Programmiergerät eine Referenzadresse zugeordnet. Solange keine Referenzadresse zugeordnet wurde, werden von diesem Modul keine Daten empfangen.
%Q	Das Präfix %Q steht für physikalische diskrete Ausgänge. Die Spulenüberprüfungsfunktion der Logimaster 90-30/20/Micro Software überprüft die %Q-Referenzen auf Mehrfachverwendung. Ab dem Software-Ausgabestand 3 können Sie die Ebene der Spulenüberprüfung einstellen (EINFACH, WARNE MEHRFACH, MEHRFACH). Weitere Informationen zu dieser Funktion finden Sie in GFK-0466, <i>Logimaster 90-30/20/Micro Programmiersoftware, Anwenderhandbuch</i> .  Dem Präfix folgt die Referenzadresse in der Ausgangstabelle (z.B. %IQ00016). Die Referenzen %Q liegen in der Ausgangs-Zustandstabelle, in der die Zustände der Ausgangsreferenzen so gespeichert sind, wie sie zuletzt vom Anwenderprogramm eingestellt wurden. Die Werte der Ausgangs-Zustandstabelle werden am Ende des Programmzyklus an die Ausgangsmodule übertragen.  Den diskreten Ausgangsmodulen wird mit der Konfigurationssoftware oder dem Hand-Programmiergerät eine Referenzadresse zugeordnet. Solange keine Referenzadresse zugeordnet wurde, werden zu diesem Modul keine Daten übertragen. Eine %Q-Referenz kann remanent oder nicht remanent sein. *
%M	Das Präfix %M steht für eine diskrete interne Referenz. Die Spulenüberprüfungsfunktion der Logimaster 90-30/20/Micro Software überprüft die %Q-Referenzen auf Mehrfachverwendung. Ab dem Software-Ausgabestand 3 können Sie die Ebene der Spulenüberprüfung einstellen (EINFACH, WARNE MEHRFACH, MEHRFACH). Weitere Informationen zu dieser Funktion finden Sie in GFK-0466, <i>Logimaster 90-30/20/Micro Programmiersoftware, Anwenderhandbuch</i> . Eine %M-Referenz kann remanent oder nicht remanent sein. *

\* Die Remanenz hängt vom Spulentyp ab. Weitere Informationen hierzu finden Sie auf Seite 2-21.

Tabelle 2-6 Diskrete Referenzen - Fortsetzung

Typ	Beschreibung
%T	<p>Das Präfix %T steht für eine diskrete temporäre Referenz. Diese Referenzen werden nicht auf Mehrfachverwendung überprüft und können daher im gleichen Programm selbst dann mehrfach eingesetzt werden, wenn die Spulenverwendungs-Überprüfungsfunktion aktiviert ist. %T kann zur Vermeidung von Adressierungskonflikten mit den Funktionen Cut/Paste [Einfügen/Ausfügen] und Schreiben und Einfügen von Dateien verwendet werden.</p> <p>Da dieser Speicherbereich nur für den temporären Gebrauch vorgesehen ist, kann er nicht mit remanenten Merkern verwendet werden. Der Inhalt geht bei Spannungsausfall und beim Übergang RUN-STOP-RUN verloren.</p>
%S	<p>Das Präfix %S steht für Systemreferenzen. Mit diesen Referenzen kann auf spezielle SPS-Daten zugegriffen werden (z.B. Zeitglieder, Zyklusdaten, Fehlerdaten). Zu den Systemreferenzen gehören die Referenzen %S, %SA, %SB und %SC.</p> <p>%S, %SA, %SB und %SC können für alle Kontakte verwendet werden.</p> <p>%SA, %SB und %SC können für remanente Spulen –(M)– verwendet werden.</p> <p>%S kann als Wort- oder Bitfolge-Eingangsargument in Funktionen oder Funktionsblöcken verwendet werden.</p> <p>%SA, %SB und %SC können als Wort- oder Bitfolge-Ein- oder Ausgangsargument in Funktionen und Funktionsblöcken verwendet werden.</p>
%G	<p>Das Präfix %G steht für Globalreferenzen. Über diese Referenzen kann auf Daten zugegriffen werden, die von mehreren SPS-System gemeinsam benutzt werden. %G-Referenzen können bei Kontakten und remanenten Spulen verwendet werden, da der %G-Speicher immer remanent ist. %G kann nicht mit nicht-remanenten Spulen verwendet werden.</p>

## Transitionen und Overrides

Den Anwenderreferenzen; %I, %Q, %M und %G sind Transitions- und Overridebits zugeordnet. Die Referenzen %T, %S, %SA, %SB, %SC besitzen Transitionsbits, jedoch keine Overridebits. Die CPU verwendet Transitionsbits für Zähler und Übergangsmarker. Beachten Sie, daß Zähler eine andere Art Transitionsbits verwenden, als dies bei Spulen der Fall ist. Transitionsbits für Zähler werden in der einrichtenden Referenz gespeichert.

In CPUs ab Modell 331 aufwärts können Overridebits gesetzt werden. Bei gesetzten Overridebits können die zugehörigen Referenzen nur über das Programmiergerät verändert werden, nicht vom Programm oder Eingabegerät. Weder die CPU-Modell 323, 321, 313 311 noch die Micro-CPU's unterstützen diskrete Override-Referenzen.

## Datenremanenz

Daten sind "remanent", wenn sie beim Anhalten der SPS gerettet werden. Bei der SPS Series 90-30 werden erhalten Programm, Fehlertabellen und Diagnosedaten, Überschreibungen (Override) und Setzen von Ausgängen, Wortdaten (%R, %AI, %AQ), Bitdaten (%I, %SC, %G, Fehlerbits und reservierte Bits), %Q- und %M-Daten (sofern sie nicht mit nicht-remanenten Merkern verwendet werden), sowie Wortdaten, die in %M und %Q gespeichert sind. Obwohl %SC remanent ist, sind %S, %SA und %SB nicht remanent.

%Q- und %M-Referenzen sind nicht remanent (d.h. sie werden beim Einschalten gelöscht, wenn die SPS von STOP auf RUN umschaltet), wenn Sie mit nicht remanenten Merkern zusammen verwendet werden. Zu den nicht remanenten Merkern gehören Spulen —( )—, negierte Spulen —(/)—, Setzspulen —(S)— und Rücksetzspulen —(R)—.

Werden %Q- und %M-Referenzen zusammen mit remanenten Merkern oder als Funktionsblockausgang verwendet, wird ihr Inhalt über einen Ausfall der Versorgungsspannung hinweg und beim Übergang RUN-STOP-RUN erhalten. Zu den remanenten Merkern gehören remanente Merker —(M)—, negierte remanente Merker —(/M)—, remanente Setzspulen —(SM)— und remanente Rücksetzspulen —(RM)—.

Der Remanenzzustand (remanent oder nicht remanent) einer %Q- oder %M-Referenz wird auf der Grundlage des Merkertyps durch die letzte Programmierung dieser %Q- oder %M-Referenz mit einer Merkeranweisung festgelegt. Wurde zum Beispiel %Q0001 zuletzt als Referenz einer remanenten Spule programmiert, dann sind die Daten von %Q0001 nullspannungssicher. Wurde dagegen %Q0001 zuletzt mit einer nicht remanenten Spule programmiert, sind die %Q0001-Daten nicht nullspannungssicher.

## Datentypen

Es gibt folgende Datentypen:

**Tabelle 2-7 Datentypen**

Type	Name	Beschreibung	Datenformat
INT	Ganze Zahl mit Vorzeichen	Dieser Datentyp belegt 16 Datenbits im Speicher und wird im Zweierkomplement dargestellt. Der zulässige Bereich liegt zwischen -32.768 und +32.767.	<p style="text-align: center;"><b>Register 1</b></p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">S</div> <div style="border: 1px solid black; flex-grow: 1; margin-right: 5px;"></div> <div style="margin-left: 5px;">(16 Bitpositionen)</div> </div> <div style="display: flex; justify-content: space-between; width: 100%; margin-top: 5px;"> <span>16</span> <span>1</span> </div>
DINT	Doppeltgenaue ganze Zahl mit Vorzeichen	Dieser Datentyp belegt 32 Datenbits im Speicher (zwei aufeinanderfolgende Speicherplätze) und wird im Zweierkomplement dargestellt (Bit 32 ist das Vorzeichenbit). Der zulässige Bereich liegt zwischen -2.147.483.648 und +2.147.483.647.	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p><b>Register 2</b></p> <div style="border: 1px solid black; padding: 2px; margin: 0 auto;">S</div> <div style="display: flex; justify-content: space-between; width: 100%; margin-top: 5px;"> <span>32</span> <span>17</span> </div> </div> <div style="text-align: center;"> <p><b>Register 1</b></p> <div style="border: 1px solid black; padding: 2px; margin: 0 auto;"></div> <div style="display: flex; justify-content: space-between; width: 100%; margin-top: 5px;"> <span>16</span> <span>1</span> </div> </div> </div> <p style="text-align: center;">(Wert im Zweierkomplement)</p>
BIT	Bit	Dieser Datentyp ist die kleinste Einheit im Speicher. Er besitzt zwei Zustände, 1 oder 0. Eine Bitfolge kann die Länge N haben.	
BYTE	Byte	Dieser Datentyp hat einen 8-Bit-Wert. Der zulässige Bereich liegt zwischen 0 und 255 (0 bis FF in Hexadezimaldarstellung).	
WORD	Wort	Dieser Datentyp belegt 16 aufeinanderfolgende Bits im Datenspeicher. Diese Bits stellen aber keine Zahl dar, sondern sind unabhängig voneinander. Jedes Bit wird für sich betrachtet und stellt einen eigenen Binärzustand (0 oder 1) dar. Der zulässige Bereich eines Wortwertes liegt zwischen 0 und FFFF	<p style="text-align: center;"><b>Register 1</b></p> <div style="border: 1px solid black; padding: 2px; margin: 0 auto; width: 100%;"></div> <div style="display: flex; justify-content: space-between; width: 100%; margin-top: 5px;"> <span>16</span> <span>1</span> </div> <p style="text-align: right;">(16 Bitpositionen)</p>
BCD-4	Vierstellige BCD-Zahl	Vierstellige BCD-Zahlen belegen 16 Bits im Datenspeicher. Jede BCD-Stelle belegt vier Bits und kann eine Zahl zwischen 0 und 9 darstellen. Die BCD-Codierung der 16 Bits ergibt einen Zahlenbereich zwischen 0 und 9999.	<p style="text-align: center;"><b>Register 1</b></p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">4</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">3</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">2</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">1</div> <div style="margin-left: 5px;">(4 BCD-Stellen)</div> </div> <div style="display: flex; justify-content: space-between; width: 100%; margin-top: 5px;"> <span>16</span> <span>13</span> <span>9</span> <span>5</span> <span>1</span> </div>
REAL	Gleitpunktzahl	Dieser Datentyp belegt 32 aufeinanderfolgende Bits im Speicher (zwei aufeinanderfolgende 16-Bit-Speicherplätze). Der zulässige Bereich liegt zwischen ±1,401298E-45 und ±3,402823E+38.	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p><b>Register 2</b></p> <div style="border: 1px solid black; padding: 2px; margin: 0 auto;">S</div> <div style="display: flex; justify-content: space-between; width: 100%; margin-top: 5px;"> <span>32</span> <span>17</span> </div> </div> <div style="text-align: center;"> <p><b>Register 1</b></p> <div style="border: 1px solid black; padding: 2px; margin: 0 auto;"></div> <div style="display: flex; justify-content: space-between; width: 100%; margin-top: 5px;"> <span>16</span> <span>1</span> </div> </div> </div> <p style="text-align: center;">(Wert im Zweierkomplement)</p>

S = Vorzeichenbit (0 = positiv, 1 = negativ).



## System-Statusreferenzen

Die System-Statusreferenzen der SPS Series 90 sind den %S-, %SA-, %SB- und %SC-Speichern zugeordnet und besitzen jeweils eine symbolische Adresse. Beispiele für Zeitreferenzen sind T\_10MS, T\_100MS, T\_SEC und T\_MIN. Beispiele für Funktionsreferenzen sind FST\_SCN, ALW\_ON und ALW\_OFF.

### Hinweis

%S-Bits können nur gelesen werden. %SA-, %SB- oder %SC-Bits können gelesen und verändert werden.

In der nachstehenden Tabelle 2-8 sind alle in einem Anwenderprogramm möglichen Systemreferenzen zusammengefasst. Bei der Eingabe eines Programms können entweder die Referenz oder die symbolische Adresse verwendet werden. In Kapitel 3 werden Fehler und Fehlerbehebung ausführlicher beschrieben.

Sie können diese speziellen Namen nicht in einem anderen Kontext benutzen.

**Tabelle 2-8 System-Statusreferenzen**

Referenz	Symbolische Adresse	Definition
%S0001	FST_SCN	Wird auf 1 gesetzt, wenn der aktuelle Zyklus der erste Zyklus ist.
%S0002	LST_SCN	Wird von 1 auf 0 gesetzt, wenn der aktuelle Zyklus der letzte Zyklus ist.
%S0003	T_10MS	Zeitkontakt 0,01 Sekunde
%S0004	T_100MS	Zeitkontakt 0,1 Sekunde
%S0005	T_SEC	Zeitkontakt 1,0 Sekunde
%S0006	T_MIN	Zeitkontakt 1,0 Minute
%S0007	ALW_ON	Immer EIN
%S0008	ALW_OFF	Immer AUS
%S0009	SY_FULL	Wird gesetzt, wenn die SPS-Fehlertabelle voll ist. Wird gelöscht, wenn ein Eintrag aus der SPS-Fehlertabelle entfernt wurde oder wenn die SPS-Fehlertabelle gelöscht wurde.
%S0010	IO_FULL	Wird gesetzt, wenn die E/A-Fehlertabelle voll ist. Wird gelöscht, wenn ein Eintrag aus der E/A-Fehlertabelle entfernt wurde oder wenn die E/A-Fehlertabelle gelöscht wurde.
%S0011	OVR_PRE	Wird gesetzt, wenn im %I-, %Q-, %M- oder %G-Speicher ein Override-Zustand besteht.
%S0013	PRG_CHK	Wird gesetzt, wenn die Überprüfung des Hintergrundprogramms läuft.
%S0014	PLC_BAT	Wird gesetzt, um bei einer CPU ab Ausgabestand 4 eine niedrige Batteriespannung anzuzeigen. Die Kontaktreferenz wird einmal pro Zyklus aktualisiert.

Tabelle 2-8 System-Statusreferenzen (Fortsetzung)

Referenz	Symbolische Adresse	Definition
%S0017	SNPXACT	SNP-X-Host ist aktiv an die CPU angeschlossen.
%S0018	SNPX_RD	SNP-X-Host hat Daten von der CPU gelesen.
%S0019	SNPX_WT	SNP-X-Host hat Daten zur CPU geschrieben.
%S0020		Wird gesetzt, wenn eine relationale Funktion mit REAL-Daten erfolgreich ausgeführt wurde. Wird gelöscht, wenn einer der Eingänge keine Zahl ist.
%S0032		Reserviert für Logicmaster 90-30/20/Micro Software.
%SA0001	PB_SUM	Wird gesetzt, wenn die für das Anwenderprogramm berechnete Prüfsumme nicht mit der Referenz-Prüfsumme übereinstimmt. War die Ursache ein vorübergehender Fehler, kann das Bit gelöscht werden, indem das Programm erneut in die CPU gespeichert wird. War die Ursache ein RAM-Fehler, muß die CPU ersetzt werden.
%SA0002	OV_SWP	Wird gesetzt, wenn die SPS erkennt, daß der letzte Zyklus länger als die vom Anwender vorgegebene Zeit war. Wird gelöscht, wenn die SPS erkennt, daß der letzte Zyklus nicht länger als die vom Anwender angegebene Zeit war. Wird auch beim Übergang von STOP nach RUN gelöscht. Nur gültig, wenn die SPS im Modus KONSTANTE ZYKLUSZEIT ist.
%SA0003	APL_FLT	Wird gesetzt, wenn im Anwenderprogramm ein Fehler auftritt. Wird beim Übergang von STOP nach RUN gelöscht.
%SA0009	CFG_MM	Wird gesetzt, wenn beim Einschalten oder Speichern der Konfiguration eine Diskrepanz in der Konfiguration festgestellt wird. Wird beim Einschalten der SPS gelöscht, wenn keine Diskrepanz mehr festgestellt wird oder beim Speichern einer Konfiguration, die mit der Hardware übereinstimmt.
%SA0010	HRD_CPU	Wird gesetzt, wenn die Diagnosefunktionen ein Hardwareproblem erkennen. Wird gelöscht, wenn das CPU-Modul ausgewechselt wird.
%SA0011	LOW_BAT	Wird gesetzt, wenn die Batteriespannung abfällt. Wird gelöscht, wenn die Batterie ausgewechselt und die CPU mit normaler Batteriespannung wieder eingeschaltet wird.
%SA0014	LOS_IOM	Wird gesetzt, wenn die Kommunikation zwischen einem E/A-Modul und der SPS CPU ausfällt. Wird gelöscht, indem das Modul ersetzt und die Versorgungsspannung zum Hauptchassis aus- und wieder eingeschaltet wird.
%SA0015	LOS_SIO	Wird gesetzt, wenn die Kommunikation zwischen einem Zusatzmodul und der SPS CPU ausfällt. Wird gelöscht, indem das Modul ersetzt und die Versorgungsspannung zum Hauptchassis aus- und wieder eingeschaltet wird.
%SA0019	ADD_IOM	Wird gesetzt, wenn ein E/A-Modul in einem Chassis hinzugefügt wird. Wird gelöscht, indem die Versorgungsspannung zum Hauptchassis aus- und wieder eingeschaltet wird, nachdem die Konfigurationsdatei aktualisiert wurde.
%SA0020	ADD_SIO	Wird gesetzt, wenn ein Zusatzmodul in einem Chassis hinzugefügt wird. Wird gelöscht, indem die Versorgungsspannung zum Hauptchassis aus- und wieder eingeschaltet wird, nachdem die Konfigurationsdatei aktualisiert wurde.
%SA0027	HRD_SIO	Wird gesetzt, wenn in einem Zusatzmodul ein Hardwarefehler erkannt wurde. Wird gelöscht, indem die Konfigurationsdatei aktualisiert und die Versorgungsspannung des Hauptchassis aus- und eingeschaltet wird.
%SA0031	SFT_SIO	Wird gesetzt, wenn in einem Zusatzmodul ein nicht behebbarer Fehler aufgetreten ist. Wird gelöscht, indem die Konfigurationsdatei aktualisiert und die Versorgungsspannung des Hauptchassis aus- und eingeschaltet wird.

**Tabelle 2-8 System-Statusreferenzen (Fortsetzung)**

Referenz	Symbolische Adresse	Definition
%SB0010	BAD_RAM	Wird gesetzt, wenn die CPU beim Einschalten einen RAM-Fehler erkennt. Wird gelöscht, wenn die CPU beim Einschalten erkennt, daß der RAM fehlerfrei ist.
%SB0011	BAD_PWD	Wird gesetzt, wenn ein durch Paßwort geschützter Zugriff verletzt wird. Wird gelöscht, wenn die SPS-Fehlertabelle gelöscht wird.
%SB0013	SFT_CPU	Wird gesetzt, wenn die CPU einen nicht behebbaren Fehler in der Software erkennt. Wird gelöscht, indem die SPS-Fehlertabelle gelöscht wird.
%SB0014	STOR_ER	Wird gesetzt, wenn während des Abspeicherns ins Programmiergerät ein Fehler auftritt. Wird gelöscht, wenn ein Speichervorgang fehlerfrei beendet wurde.
%SC0009	ANY_FLT	Wird gesetzt, wenn ein beliebiger Fehler auftritt. Wird gelöscht, wenn beide Fehlertabellen leer sind.
%SC0010	SY_FLT	Wird gesetzt, wenn ein Fehler auftritt, der einen Eintrag in die E/A-Fehlertabelle verursacht. Wird gelöscht, wenn die SPS-Fehlertabelle keinen Eintrag enthält.
%SC0011	IO_FLT	Wird gesetzt, wenn ein Fehler auftritt, der einen Eintrag in die E/A-Fehlertabelle verursacht. Wird gelöscht, wenn die SPS-Fehlertabelle keinen Eintrag enthält.
%SC0012	SY_PRES	Wird gesetzt, wenn die SPS-Fehlertabelle mindestens einen Eintrag enthält. Wird gelöscht, wenn die SPS-Fehlertabelle keinen Eintrag enthält.
%SC0013	IO_PRES	Wird gesetzt, wenn die E/A-Fehlertabelle mindestens einen Eintrag enthält. Wird gelöscht, wenn die E/A-Fehlertabelle keinen Eintrag enthält.
%SC0014	HRD_FLT	Wird gesetzt, wenn ein Hardwarefehler aufgetreten ist. Wird gelöscht, wenn beide Fehlertabellen keinen Eintrag enthalten.
%SC0015	SFT_FLT	Wird gesetzt, wenn ein Softwarefehler aufgetreten ist. Wird gelöscht, wenn beide Fehlertabellen keinen Eintrag enthalten.

## Funktionsblockstruktur

Jeder Strompfad besteht aus einer oder mehreren Programmieranweisungen. Hierbei kann es sich um einfache Relais oder um komplexere Funktionen handeln.

### Format der Kontaktplanrelais

Die Programmiersoftware verfügt über mehrere Arten von Relaisfunktionen, die grundlegend für Stromfluß und logische Verknüpfungen im Programm sorgen. Zu ihnen gehören ein Schließerkontakt (—| |—) und eine negierte Spule (—(/)—). Jeder dieser Relaiskontakte und Spulen besitzt einen Eingang und einen Ausgang. Zusammen ermöglichen sie den Stromfluß durch den Kontakt oder die Spule.

Jedem Relaiskontakt und jeder Spule muß eine Referenz zugeteilt werden, die bei der Auswahl des Relais eingegeben wird. Bei einem Kontakt stellt die Referenz einen Speicherplatz dar, der den Stromfluß in den Kontakt hinein festlegt. Ist im nachstehenden Beispiel die Referenz %I0122 EIN, fließt Strom durch diesen Relaiskontakt.

%I0122  
—| |—

Bei einer Spule stellt die Referenz den Speicherplatz dar, der durch den Stromfluß in die Spule kontrolliert wird. Fließt in diesem Beispiel der Strom in die linke Seite der Spule, wird die Referenz %Q0004 durchgeschaltet.

%Q0004  
-( )-

Sowohl die Programmiersoftware als auch das Hand-Programmiergerät verfügen über eine Spulenverwendungs-Überprüfungsfunktion, die feststellt, ob %Q- oder %M-Referenzen mit Relais-  
spulen oder Funktionsausgängen mehrfach benutzt wurden.

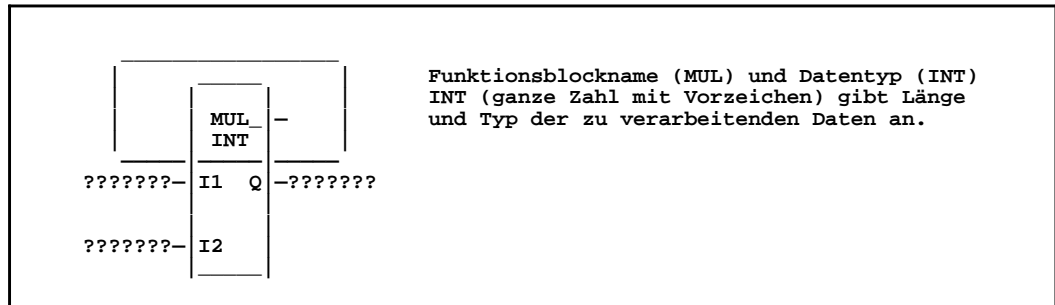
### Format der Programm-Funktionsblöcke

Einige Funktionsblöcke sind sehr einfach aufgebaut. Hierzu gehört die MCR-Funktion, die mit dem abgekürzten Funktionsnamen in eckigen Klammern dargestellt wird:

-[ MCR ]-

Andere Funktionen sind komplexer und können die Eingabe der für die Funktion benötigten Informationen an mehreren Stellen erforderlich machen.

Nachstehend wird der Funktionsblock Multiplikation (MUL) dargestellt. Seine Bestandteile sind typisch für zahlreiche Programmfunktionen in Logicmaster 90-30/20/Micro. Der obere Teil des Funktionsblocks zeigt den Name der Funktion. Er kann auch einen Datentyp anzeigen (hier: Ganze Zahl mit Vorzeichen).

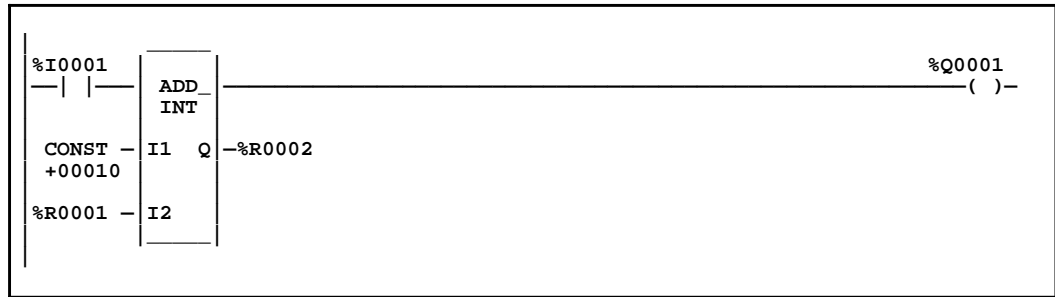


Bei vielen Programmfunktionen können Sie den Datentyp für die Funktion einstellen, nachdem Sie die Funktion ausgewählt haben. Den Datentyp der MUL-Funktion könnten Sie zum Beispiel auf doppelgenaue ganze Zahl mit Vorzeichen abändern. Weitere Informationen zu den Datentypen finden Sie weiter vorne in diesem Kapitel.

## Funktionsblockparameter

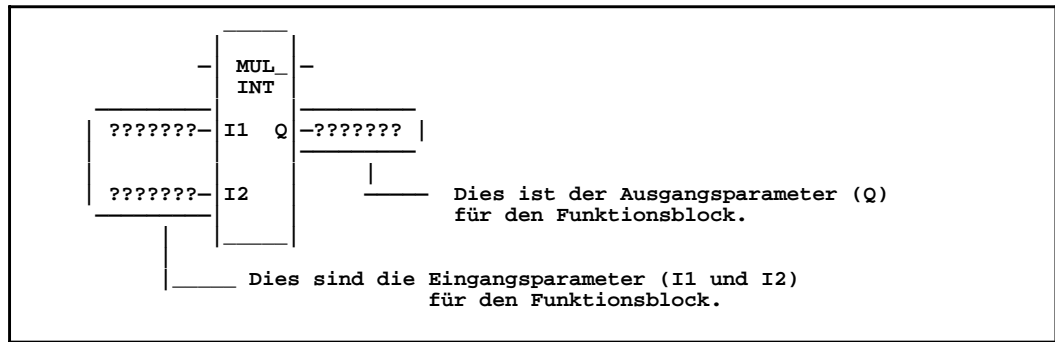
Jede Linie, die in die linke Seite eines Funktionsblocks führt, stellt einen Eingang zu dieser Funktion dar. Es gibt zwei Arten Eingänge in einen Funktionsblock: Konstanten und Referenzen. Eine Konstante ist ein expliziter Wert. Eine Referenz ist die Adresse eines Wertes.

Im nachstehenden Beispiel ist der Eingangsparameter I1 des ADD-Funktionsblocks eine Konstante, während Eingangsparameter I2 eine Referenz darstellt.



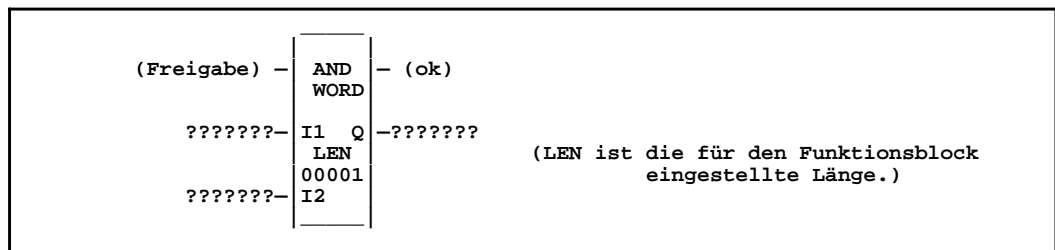
Jede Linie, die den Funktionsblock auf der rechten Seite verläßt, stellt einen Ausgang dar. Aus einem Funktionsblock gibt es nur eine Art Ausgang, die Referenz. Ausgangswerte können niemals in eine Konstante geschrieben werden.

Geben Sie bei dem Fragezeichen auf der linken Seite des Funktionsblocks entweder den Datenwert selbst ein oder aber die Referenzadresse, unter der der Datenwert gefunden werden kann. Am Fragezeichen auf der rechten Seite geben Sie normalerweise eine Referenzadresse für die Daten an, die vom Funktionsblock ausgegeben werden.



Die meisten Funktionsblöcke verändern die Eingangsdaten nicht. Sie schreiben die Ergebnisse der Operation in eine Ausgangsreferenz.

Bei Funktionen, die Tabellen bearbeiten, kann für die Funktion ein Längenwert eingestellt werden. Im nachstehenden Funktionsblock kann für die logische UND-Funktion eine Länge von 256 Worten oder Doppelworten eingestellt werden.

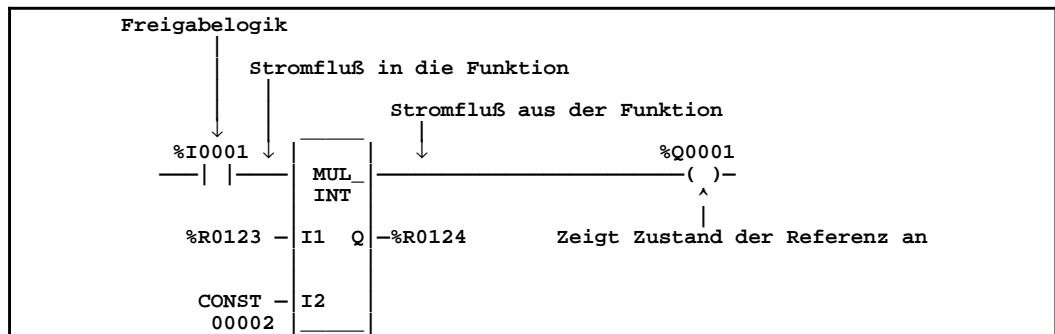


Timer-, Zähler-, BITSEQ- und ID-Funktionen benötigen eine Adresse für die drei Worte im Speicher (Register), in denen der aktuelle Wert, der voreingestellte Wert und ein Steuerwort (Instanz) für die Funktion eingetragen werden.



### Stromfluß zu und von einer Funktion

Der Strom fließt oben links in die Funktion. Häufig wird der Stromfluß zum Funktionsblock durch eine Freigabelogik gesteuert. Ist dies nicht der Fall, wird der Funktionsblock unbedingt in jedem CPU-Zyklus ausgeführt.



### Hinweis

Funktionsblöcke können nicht direkt an die linke Stromschiene angeschlossen werden. Um eine Funktion in jedem Zyklus aufzurufen, können Sie %S7 verwenden, das Bit ALL\_ON (immer EIN), mit einem an der Stromschiene angeschlossenen Schließerkontakt.

Der Strom fließt oben rechts aus der Funktion heraus und kann wahlweise zu anderen Programmelementen oder zu einer Spule geführt werden. Funktionsblöcke leiten den Stromfluß weiter, wenn sie erfolgreich ausgeführt wurden. In diesem Handbuch werden die Durchschaltbedingungen bei den einzelnen Funktionsbeschreibungen erläutert.

## Abschnitt 3: Ein- und Ausschaltsequenzen

Bei der SPS Series 90-30 sind zwei Einschaltsequenzen möglich: Kaltstart und Warmstart. Normalerweise benutzt die CPU die Kaltstartsequenz. Sind jedoch in einem SPS-System ab Modell 331 aufwärts die zwischen Ausschalten und Wiedereinschalten weniger als fünf Sekunden verstrichen, wird die Warmstartsequenz verwendet.

### Einschalten

Bei einem Kaltstart laufen nacheinander folgende Ereignisse ab:

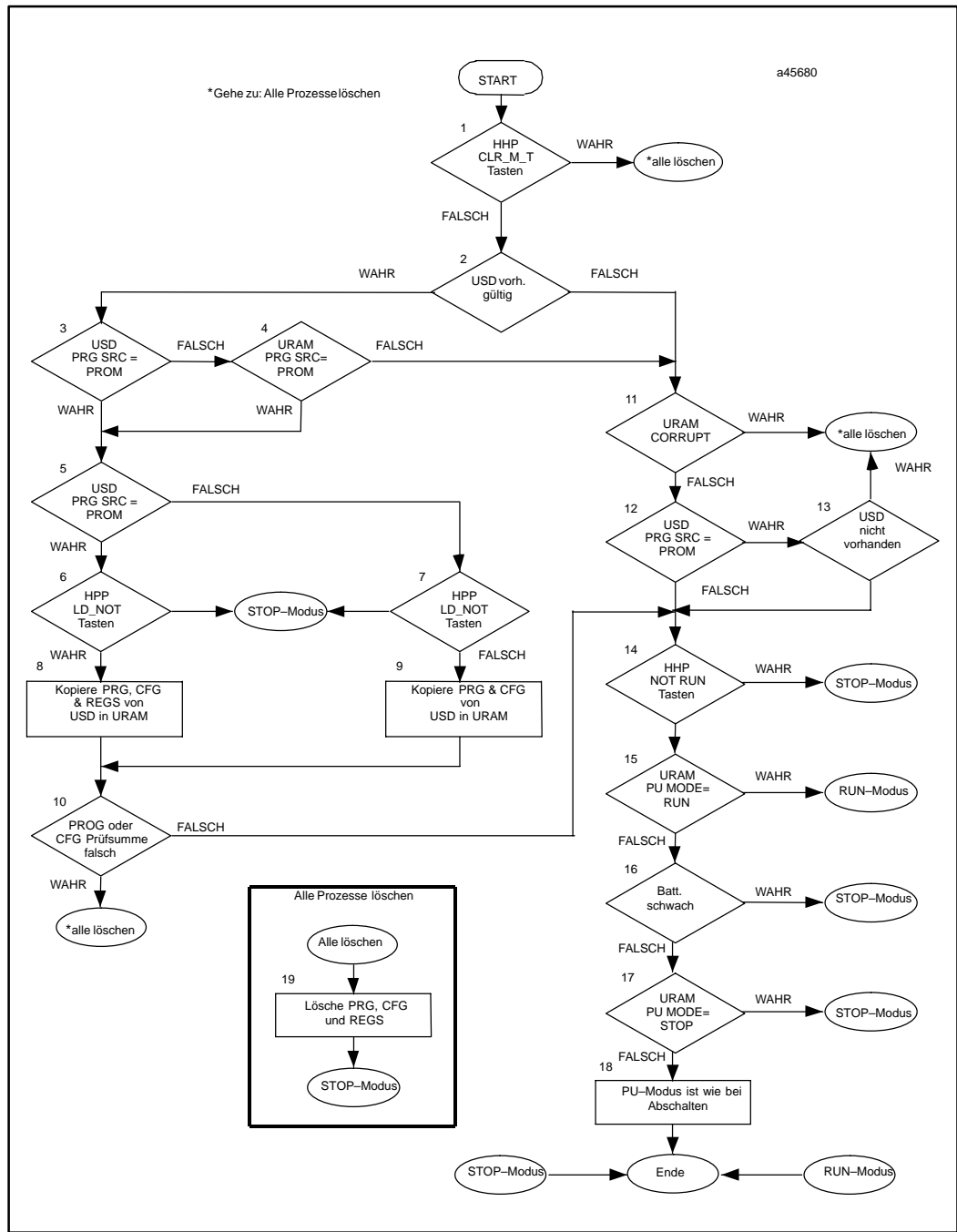
#### Hinweis

Die Abfolge bei einem Warmstart ist die gleiche, lediglich Schritt 1 wird übersprungen.

- (1) Die CPU führt einen Selbsttest durch. Hierzu gehört die Überprüfung eines Teils des batteriegepufferten RAMs um festzustellen, ob der RAM gültige Daten enthält.
- (2) Gibt es ein EPROM, EEPROM oder Flash Memory und gibt die PROM-Einschaltoption an, daß der Inhalt des PROMs benutzt werden soll, wird der Inhalt des PROMs in den RAM kopiert. Gibt es kein EPROM, EEPROM oder Flash Memory, bleibt der RAM unverändert und wird nicht durch den Inhalt des PROMs überschrieben.
- (3) Die CPU fragt die einzelnen Steckplätze im System ab um festzustellen, welche Module vorhanden sind.
- (4) Die Hardwarekonfiguration wird nun mit der Softwarekonfiguration verglichen. Erkannte Diskrepanzen werden als Fehler bewertet und gemeldet. Entspricht ein eingebautes Modul nicht den Angaben in der Softwarekonfiguration, wird dies auch als Fehler bewertet und gemeldet.
- (5) Liegt keine Softwarekonfiguration vor, verwendet die CPU die Standardkonfiguration.
- (6) Die CPU richtet die Kommunikationskanäle zwischen ihr und den intelligenten Modulen ein.
- (7) Im letzten Schritt wird der Modus des ersten Zyklus auf der Grundlage der CPU-Konfiguration bestimmt. Bei RUN läuft der Zyklus so ab, wie unter "Übergang von STOP auf RUN" beschrieben. Abbildung 2-5 auf der nächsten Seite zeigt die Entscheidungsfolge für die CPU bei der Festlegung, ob aus dem PROM kopiert wird und ob der Hochlauf im Modus RUN oder STOP erfolgt.

#### Hinweis

Die Schritte 2 bis 6 gelten nicht für die SPS Series 90 Micro. Informationen zu den Ein- und Ausschaltsequenzen der Micro finden Sie in Kapitel 5 von GFK-1065, *SPS Series 90 Micro, Anwenderhandbuch*.



**Abbildung 2-5 Einschaltsequenz**

Vor der START-Anweisung auf dem Einschalt-Blockschaltbild durchläuft die CPU Einschalt-Diagnosefunktionen, die die verschiedenen von der CPU benutzten Peripheriegeräte testen, und prüft den RAM. Nach Abschluß dieser Diagnosefunktionen werden die internen Datenstrukturen sowie die von der CPU benutzten Peripheriegeräte initialisiert. Anschließend stellt die CPU fest, ob der Anwender-RAM fehlerhaft ist. Ist dies der Fall, werden Anwenderprogramm und Konfiguration gelöscht und auf Standardwerte gesetzt, und es werden alle Anwenderregister gelöscht.



**BEGRIFFE AUS DEM BLOCKSCHALTBILD:**

PRG = Anwenderprogramm

CFG = Anwenderkonfiguration

REGS = Anwenderregister (%I, %Q, %M, %G, %R, %AI, %AQ)

USD = Anwenderspeicher (EEPROM oder Flash Memory)

URAM = nichtflüchtiger Anwender-RAM, der PRG, CFG und REGS enthält

**ERLÄUTERUNGEN ZUM BLOCKSCHALTBILD:**

- (1) Wurden am HHP beim Einschalten die Tasten <CLR> und >M\_T> gedrückt, um den gesamten URAM zu löschen?
- (2) Ist der USD vorhanden (darf nur bei Modellen fehlen, die EEPROM benutzen) und enthält der USD gültige Daten?
- (3) Wurde der Parameter PRG SRC im USD auf PROM eingestellt (d.h. PRG und CFG werden aus dem USD geladen)?
- (4) Wurde der Parameter PRG SRC im URAM auf PROM eingestellt (d.h. PRG und CFG werden aus dem USD geladen)?
- (5) Wurde der Parameter REG SRC im USD auf PROM eingestellt (d.h. REGS werden aus dem USD geladen)?
- (6) Wurden am HHP beim Einschalten die Tasten <LD> und <NOT> gedrückt, um zu verhindern, daß PRG, CFG und REGS vom USD geladen werden?
- (7) Wie (6).
- (8) PRG, CFG und REGS vom USD in URAM kopieren.
- (9) PRG und CFG vom USD in URAM kopieren.
- (10) Sind die gerade vom USD geladenen Prüfsummen von PRG oder CFG ungültig?
- (11) Ist der URAM verstümmelt? Mögliche Ursachen: Abschalten ohne oder mit schwacher Batterie, oder Firmware-Aktualisierung.
- (12) Wurde der Parameter PRG SRC im URAM auf PROM eingestellt (d.h. PRG und CFG werden aus dem USD geladen)?
- (13) Ist der USD vorhanden (nur bei Modellen, die EEPROM benutzen)?
- (14) Wurden am HHP beim Einschalten die Tasten <NOT> und <RUN> gedrückt, um unbedingt im STOP-Modus hochzufahren?
- (15) Wurde der Parameter PWR UP im URAM auf RUN gesetzt?
- (16) Ist die Batteriespannung zu niedrig?
- (17) Wurde der Parameter PWR UP im URAM auf STOP gesetzt?
- (18) Einschaltmodus auf Ausschaltmodus setzen.
- (19) PRG, CFG und REGS löschen.

**Hinweis**

Der erste Teil des Diagramms auf der vorhergehenden Seite gilt nicht für die SPS Series 90 Micro. Informationen zu den Ein- und Ausschaltsequenzen der Micro finden Sie in Kapitel 5 von GFK-1065, *SPS Series 90 Micro, Anwenderhandbuch*.

**Ausschalten**

Eine Systemabschaltung findet statt, wenn die Stromversorgung erkennt, daß die Eingangsspannung länger als eine Periode ausgefallen ist oder wenn die Ausgangsspannung der 5-V-Stromversorgung auf einen Wert von weniger als 4,9 VDC abgefallen ist.

## Abschnitt 4: Uhren und Timer

Zu den in der SPS Series 90-30 enthaltenen Uhren und Timern gehören eine Betriebszeituhr, eine Tageszeituhr (Modelle 331, 340/341, 351/352 und 25-Punkt Micro), eine Zeitüberwachung (Watchdog) und ein Timer für konstante Zykluszeit. Außerdem gibt es noch zwei Arten von Timer-Funktionsblöcken, eine Einschaltverzögerungs-Timer und einen Start-Rücksetz-Timer. Vier zeitgesteuerte Kontakte schalten in Abständen von 0,01 Sekunden, 0,1 Sekunden, 1,0 Sekunde und 1 Minute.

### Betriebszeituhr

Die Betriebszeituhr erfaßt die seit dem Einschalten der CPU verstrichene Zeit in Schritten von 100 Mikrosekunden. Diese Uhr ist nicht nullspannungssicher und läuft bei jedem Einschalten von neuem an. Einmal pro Sekunde unterbricht die Hardware die CPU, damit ein Sekundenwert aufgezeichnet werden kann. Etwa 100 Jahre nach Beginn der Zeiterfassung fängt dieser Sekundenwert wieder bei Null an.

Da die Betriebszeituhr die Basis für die Operationen der Systemsoftware und Timer-Funktionsblöcke bildet, darf sie über das Programmiergerät oder vom Programm aus nicht rückgesetzt werden. Das Anwenderprogramm kann jedoch mit der SVCREQ-Funktion #16 den aktuellen Wert der Betriebszeituhr auslesen.

### Tageszeituhr

Bei der 28-Punkt Micro und ab dem Modell 331 der SPS Series 90-30 wird die Tageszeit durch eine Tageszeituhr verwaltet, die sieben Zeitfunktionen umfaßt:

- Jahr (zwei Stellen)
- Monat
- Tag
- Stunde
- Minute
- Sekunde
- Wochentag

Die Tageszeituhr ist batteriegepuffert und erhält ihren Zustand über einen Spannungsausfall hinweg. Solange die Uhr nicht initialisiert wurde, sind die in ihr enthaltenen Werte jedoch ohne Bedeutung. Das Anwenderprogramm kann mit der SVCREQ-Funktion #7 die Tageszeituhr lesen und einstellen. Die Tageszeituhr kann auch von der CPU-Konfigurationssoftware aus gelesen und eingestellt werden.

Die Tageszeituhr ist so ausgelegt, daß sie die monatlichen und jährlichen Übergänge sowie die Schaltjahre bis zum Jahr 2079 bewältigen kann.

## Zeitüberwachung (Watchdog)

Eine Zeitüberwachung (Watchdog) in der SPS Series 90-30 soll Fehlerzustände erkennen, die zu einer außergewöhnlich langen Zyklusdauer führen. Die Überwachungsfunktion ist fest auf 200 Millisekunden (500 Millisekunden bei CPU 351/352) eingestellt, dieser Wert kann nicht verändert werden. Jeweils zu Beginn eines Zyklus läuft die Zeitüberwachung wieder bei Null an.

Wird der Wert der Zeitüberwachung überschritten, erlischt die OK LED, die CPU wird angehalten und vollständig abgeschaltet und die Ausgänge nehmen ihre voreingestellten Zustände ein. Datenaustausch ist nicht mehr möglich und alle Mikroprozessoren auf allen Modulen sind angehalten. Dieser Zustand kann nur behoben werden, indem Sie die Versorgungsspannung des CPU-Chassis aus- und wieder einschalten.

Bei der 90-20 und bei CPUs ab Modell 340 aufwärts wird beim Ansprechen der Zeitüberwachung die CPU rückgesetzt. Die CPU führt dann die EinschaltRoutine durch, erzeugt einen Zeitüberwachungsfehler und geht in STOP-Modus.

## Timer für konstante Zykluszeit

Der Timer für konstante Zykluszeit steuert die Länge eines Programmzyklus, wenn die SPS Series 90-30 im Modus KONSTANTE ZYKLUSZEIT arbeitet. In dieser Betriebsart benötigt jeder Zyklus die gleiche Zeit. Im allgemeinen ist bei den meisten Anwenderprogrammen die in den einzelnen Zyklen benötigte Zeit für Eingabezyklus, Programmbearbeitungszyklus und Ausgabezyklus nie genau gleich. Der Wert des Timers für konstante Zykluszeit wird über das Programmiergerät im Bereich zwischen 5 Millisekunden und dem Wert der Zeitüberwachung (Watchdog) eingestellt. Der Standardwert ist 100 Millisekunden.

Läuft der Timer für konstante Zykluszeit ab, ehe der Zyklus beendet ist, und wurde beim vorhergehenden Zyklus die Zeit nicht überschritten, trägt die SPS einen Zykluszeitüberschreitungsalarm in die SPS-Fehlertabelle ein. Zu Beginn des nächsten Zyklus setzt die SPS den Fehlerkontakt OV\_SWP, der wieder rückgesetzt wird, wenn die SPS nicht im Modus KONSTANTE ZYKLUSZEIT ist oder wenn beim vorherigen Zyklus die Zeit nicht überschritten wurde.

## Zeitkontakte

Die SPS Series 90 besitzt vier Zeitkontakte mit Einschaltwerten von 0,01 s, 0,1 s, 1,0 s und 1 min. Während der Zyklusbearbeitung ändert sich der Zustand dieser Kontakte nicht. Sie liefern einen Impuls mit gleicher EIN- und AUS-Zeit. Die Kontakte werden adressiert als T\_10MS (0,01 s), T\_100MS (0,1 s), T\_SEC (1,0 s) und T\_MIN (1 min).

Das Taktdiagramm in Abbildung 2-6 zeigt das Schaltspiel dieser Kontakte.

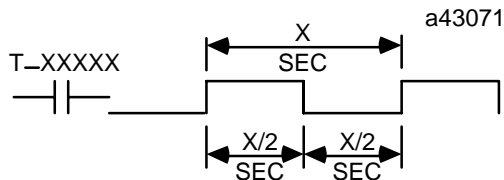


Abbildung 2-6 Taktdiagramm der Zeitkontakte

## Abschnitt 5: Systemsicherheit

Das Sicherheitskonzept der SPS Series 90-20 und 90-30 ist so ausgelegt, daß unberechtigte Änderungen des SPS-Inhalts verhindert werden können. Die SPS kennt vier Sicherheitsebenen. Die erste Ebene, die immer zugänglich ist, gestattet nur das Lesen der SPS-Daten, das Anwenderprogramm kann hier nicht verändert werden. Der Zugriff zu den anderen drei Ebenen wird durch Paßwörter geschützt.

Je höher die Ebene, desto größer die Anzahl der Privilegien. Zusätzlich zu den auf einer Ebene gewährten Privilegien sind die Privilegien aller darunterliegenden Ebenen verfügbar. Die Ebenen und ihre Privilegien sind:

Privileg-ebene	Beschreibung
Ebene 1	Mit Ausnahme der Paßwörter können alle Daten gelesen werden. Hierzu gehören alle Datenspeicher (%I, %Q, %AQ, %R, usw.), Fehlertabellen und alle Programmblocktypen (Daten, Werte und Konstanten). Es dürfen keine Werte in der SPS verändert werden.
Ebene 2	Auf dieser Ebene ist schreibender Zugriff auf die Datenspeicher (%I, %R, usw.) möglich.
Ebene 3	Auf dieser Ebene ist Schreibzugriff auf das Anwenderprogramm in STOP-Modus möglich.
Ebene 4	Dies ist die Standardebene für Systeme, bei denen keine Paßwörter eingestellt sind. Die Standardebene eines Systems mit Paßwörtern ist die jeweils höchste ungeschützte Ebene. In dieser höchsten Ebene besteht Lese- und Schreibzugriff zu allen Speichern und Paßwörtern in RUN- und STOP-Modus. (Konfigurationsdaten können im RUN-Modus nicht verändert werden.)

### Paßwörter

Für jede Privilegebene in der SPS gibt es ein Paßwort. Für den Zugriff auf Ebene 1 kann kein Paßwort eingestellt werden. Sie können für jede Ebene ein eigenes Paßwort verwenden oder das gleiche Paßwort für mehrere Ebenen vereinbaren. Ein Paßwort besteht aus 1 bis 4 ASCII-Zeichen, es kann nur über die Programmiersoftware oder das Hand-Programmiergerät eingegeben oder geändert werden.

Ein Wechsel einer Privilegebene ist nur solange wirksam, wie der Datenverkehr zwischen SPS und Programmiergerät intakt ist. Es sind keine Aktivitäten erforderlich, die Verbindung darf jedoch nicht unterbrochen werden. Werden über einen Zeitraum von 15 Minuten keine Daten ausgetauscht, kehrt das System zur höchsten ungeschützten Ebene zurück.

Wird die Verbindung zur SPS hergestellt, fragt die Programmiersoftware die Schutzzustände der einzelnen Privilegebenen von der SPS ab. Anschließend fordert die Programmiersoftware die SPS auf, auf die höchste ungeschützte Ebene zu gehen, wobei die Programmiersoftware Zugang zur höchsten ungeschützten Ebene erhält, ohne eine bestimmte Ebene anfordern zu müssen. Wird das Hand-Programmiergerät an die SPS angeschlossen, geht die SPS wieder auf die höchste ungeschützte Ebene zurück.

## Wechsel der Privilegebene

Ein Programmiergerät fordert einen Wechsel der Privilegebene, indem es die neue Privilegebene und das Paßwort für diese Ebene liefert. Der Wechsel der Privilegebene wird abgelehnt, wenn das vom Programmiergerät gesendete Paßwort nicht mit dem in der SPS-Paßwortzugriffstabelle gespeicherten Paßwort für die angeforderte Ebene übereinstimmt. In diesem Fall wird die aktuelle Privilegebene beibehalten und es findet kein Wechsel statt. Bei einem Versuch, mit dem Hand-Programmiergerät auf Daten in der SPS zuzugreifen oder diese zu verändern, ohne daß die richtige Privilegebene eingestellt wurde, reagiert das Hand-Programmiergerät mit einer Fehlermeldung und der Zugriff wird verweigert.

## Sperren/freigeben von Unterprogrammen

Unterprogrammblöcke können mit der Block-Verriegelungsfunktion der Programmiersoftware gesperrt oder freigegeben werden. Es gibt zwei Sperrarten:

Sperre	Beschreibung
Anzeige	Bei Sperre können Sie nicht in dieses Unterprogramm zoomen.
Editieren	Bei Sperre können Sie die Daten im Unterprogramm nicht editieren

Sofern ein Unterprogramm nicht permanent für Anzeigen oder Editieren gesperrt ist, können Sie die Sperren im Blockvereinbarungseditor aufheben.

Auf ein für das Anschauen gesperrtes Unterprogramm kann eine Suchen/Ersetzen-Funktion angewandt werden. Wird das gesuchte Objekt in einem solchen Unterprogramm gefunden, erscheint anstelle der Logik eine der folgenden Meldungen auf dem Bildschirm:

`Gefunden in gesperrtem Block <block_name> (Weiter/Ende)`

oder

`Schreiben nicht möglich auf gesperrten Block <block_name> (Weiter/Ende)`

Sie können den Suchvorgang nun fortsetzen oder abbrechen.

Ordner, die gesperrte Unterprogramme enthalten, können gelöscht werden. Werden die Ordnerfunktionen Kopieren, Backup und Restore der Programmiersoftware auf einen Ordner mit gesperrten Unterprogrammen angewandt, dann bleiben diese Sperren erhalten.

## Permanentes Sperren eines Unterprogramms

Zusätzlich zur Anzeige- (VIEW LOCK) und Editiersperre (EDIT LOCK) gibt es noch zwei Arten von permanenten Sperren. Wurde für ein Unterprogramm eine permanente Anzeigesperre (PERMANENT VIEW LOCK) gesetzt, dann können Sie in dieses Unterprogramm nicht mehr hineinzoomen. Wurde eine permanente Editiersperre (PERMANENT EDIT LOCK) gesetzt, dann können Sie das betreffende Unterprogramm nicht mehr editieren.

**Vorsicht**

**Im Gegensatz zu den normalen Sperren VIEW LOCK und EDIT LOCK können gesetzte permanente Sperren nicht mehr aufgehoben werden.**

Eine permanente Editiersperre (PERMANENT EDIT LOCK) kann nur noch in eine permanente Anzeigesperre (PERMANENT VIEW LOCK) umgewandelt werden. Eine permanente Anzeigesperre (PERMANENT VIEW LOCK) kann in keine andere Sperre umgewandelt werden.

## Abschnitt 6: E/A-System der SPS Series 90-30, 90-20 und Micro

Die E/A-Systeme der SPS Series 90-30, 90-20 und Micro bilden die Schnittstelle zwischen der SPS Series 90-30 und den anwenderseitigen Geräten. Die E/A der Series 90-30 wird Modell 30 E/A genannt. Die Module der Modell 30 E/A werden direkt in die Steckplätze des CPU-Chassis und bei der der SPS 90-30 ab Modell 331 aufwärts in die Steckplätze der Erweiterungschassis gesteckt. Bei den E/A-Systemen der Modelle 331, 340 und 341 können bis zu 49 Modell 30 E/A-Module eingebaut werden. Bei den E/A-Systemen Modell 351 und 352 sind bis zu 79 Modell 30 E/A-Module möglich. Ein Chassis der Modelle 311 oder 313 der SPS Series 90-30 mit 5 Steckplätzen kann bis zu 5 Modell 30 E/A-Module aufnehmen, ein Chassis mit 10 Steckplätzen bis zu 10 Module.

Abbildung 2-7 zeigt die E/A-Struktur der SPS Series 90-30.

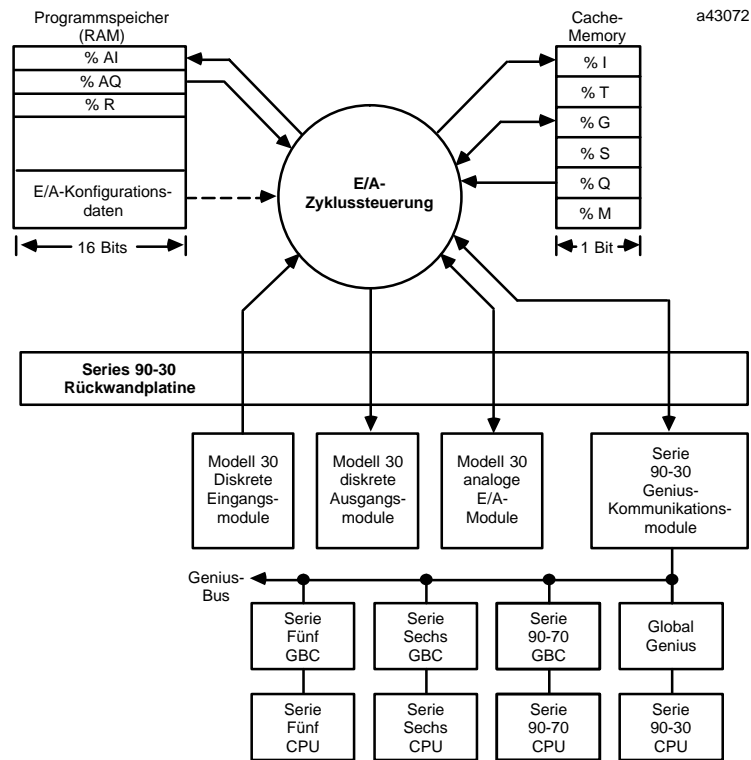


Abbildung 2-7 E/A-Struktur der Series 90-30

### Hinweis

Die Struktur in Abbildung 2-7 ist spezifisch für die 90-30 E/A. Informationen über die Struktur der 90-20 E/A finden Sie in GFK-0551, *SPS Series 90-20, Anwenderhandbuch*. Informationen über die Struktur der Micro E/A finden Sie in GFK-1065, *SPS Series 90 Micro, Anwenderhandbuch*.

## Modell 30 E/A-Module

Es gibt fünf verschiedene Typen Modell 30 E/A-Module: Diskrete Eingangsmodule, diskrete Ausgangsmodule, analoge Eingangsmodule, analoge Ausgangsmodule und Zusatzmodule. In Tabelle 2-9 finden Sie eine Auflistung der Modell 30 E/A-Module mit Bestellnummer, Anzahl E/A-Punkte und einer kurzen Beschreibung.

### Hinweis

Es ist möglich, daß einige der in der Tabelle enthaltenen E/A-Module zum Zeitpunkt der Drucklegung dieses Handbuches noch nicht verfügbar waren. Ihr GE Fanuc Distributor kann Ihnen genaue Auskünfte über die jeweilige Liefersituation geben. Die technischen Daten und Anschlußpläne der einzelnen Modell 30 E/A-Module finden Sie in GFK-0898, *SPS Series 90-30 E/A-Module, Technische Daten*.

**Tabelle 2-9 Modell 30 E/A-Module**

Bestellnummer	Punkte	Beschreibung	Handbuch
<b>Digitale Eingangsmodule</b>			
IC693MDL230	8	120 VAC potentialgetrennt	GFK-0898
IC693MDL231	8	240 VAC potentialgetrennt	GFK-0898
IC693MDL240	16	120 VAC	GFK-0898
IC693MDL241	16	24 VAC/DC positive/negative Logik	GFK-0898
IC693MDL630	8	24 VDC positive Logik	GFK-0898
IC693MDL632	8	125 VDC positive/negative Logik	GFK-0898
IC693MDL633	8	24 VDC negative Logik	GFK-0898
IC693MDL634	8	24 VDC positive/negative Logik	GFK-0898
IC693MDL640	16	24 VDC positive Logik	GFK-0898
IC693MDL641	16	24 VDC negative Logik	GFK-0898
IC693MDL643	16	24 VDC positive Logik, schnell	GFK-0898
IC693MDL644	16	24 VDC negative Logik, schnell	GFK-0898
IC693MDL645	16	24 VDC positive/negative Logik	GFK-0898
IC693MDL646	16	24 VDC positive/negative Logik, schnell	GFK-0898
IC693MDL652	32	24 VDC Position/negative Logik	GFK-0898
IC693MDL653	32	24 VDC positive/negative Logik, schnell	GFK-0898
IC693MDL654	32	5/12 VDC (TTL) positive/negative Logik	GFK-0898
IC693MDL655	32	24 VDC positive/negative Logik	GFK-0898
IC693ACC300	8/16	Eingangssimulator	GFK-0898

Tabelle 2-9 Modell 30 E/A-Module (Fortsetzung)

Bestellnummer	Punkte	Beschreibung	Handbuch
<b>Digitale Ausgangsmodule</b>			
IC693MDL310	12	120 VAC, 0,5 A	GFK-0898
IC693MDL330	8	120/240 VAC, 2 A	GFK-0898
IC693MDL340	16	120 VAC, 0,5 A	GFK-0898
IC693MDL390	5	120/240 VAC potentialgetrennt, 2 A	GFK-0898
IC693MDL730	8	12/24 VDC positive Logik, 2 A	GFK-0898
IC693MDL731	8	12/24 VDC negative Logik, 2 A	GFK-0898
IC693MDL732	8	12/24 VDC positive Logik, 0,5 A	GFK-0898
IC693MDL733	8	12/24 VDC negative Logik, 0,5 A	GFK-0898
IC693MDL734	6	125 VDC positive/negative Logik, 2 A	GFK-0898
IC693MDL740	16	12/24 VDC positive Logik, 0,5 A	GFK-0898
IC693MDL741	16	12/24 VDC negative Logik, 0,5 A	GFK-0898
IC693MDL742	16	12/24 VDC positive Logik, 1 A	GFK-0898
IC693MDL750	32	12/24 VDC negative Logik	GFK-0898
IC693MDL751	32	12/24 VDC positive Logik, 0,3A	GFK-0898
IC693MDL752	32	5/24 VDC (TTL) negative Logik, 0,5 A	GFK-0898
IC693MDL753	32	12/24 VDC positive/negative Logik, 0,5 A	GFK-0898
IC693MDL930	8	Relais, Schließer., 4 A potentialgetrennt	GFK-0898
IC693MDL931	8	Relais, BC, potentialgetrennt	GFK-0898
IC693MDL940	16	Relais, Schließer, 2 A	GFK-0898
<b>Ein-Ausgangsmodule</b>			
IC693MDR390	8/8	24 VDC Input, Relaisausgang	GFK-0898
IC693MAR590	8/8	120 VAC Input, Relaisausgang	GFK-0898
<b>Analogmodule</b>			
IC693ALG220	4 Kanäle	Analogeingang, Spannung	GFK-0898
IC693ALG221	4 Kanäle	Analogeingang, Strom	GFK-0898
IC693ALG222	16	Analogeingang, Spannung	GFK-0898
IC693ALG223	16	Analogeingang, Strom	GFK-0898
IC693ALG390	2 Kanäle	Analogausgang, Spannung	GFK-0898
IC693ALG391	2 Kanäle	Analogausgang, Strom	GFK-0898
IC693ALG392	8 Kanäle	Analogausgang, Strom/Spannung	GFK-0898
IC693ALG442	4/2	Analog, Strom/Spannung Kombination Ein-/Ausgang	GFK-0898
<b>Zusatzmodule</b>			
IC693APU300	—	Schneller Zähler	GFK-0293
IC693CMM311	—	Kommunikations-Coprozessormodul	GFK-0582
IC693PCM300	—	PCM, 160 kB (35 kB MegaBasic Anwenderprogramm)	GFK-0255
IC693PCM301	—	PCM, 192 kB (47 kB MegaBasic Anwenderprogramm)	GFK-0255
IC693PCM311	—	PCM, 640 kB (190 kB MegaBasic Anwenderprogramm)	GFK-0255
IC693ADC311	—	Alphaanzeige-Coprozessor	GFK-0521
IC693BEM331	—	Genius-Buscontroller	GFK-1034
IC693CMM301	—	Genius-Kommunikationsmodul	GFK-0412
IC693CMM302	—	Erweitertes Genius-Kommunikationsmodul	GFK-0695
IC693BEM320	—	Schnittstellenmodul (Slave)	GFK-0631
IC693BEM321	—	Schnittstellenmodul (Master)	GFK-0823
IC693APU301	—	Motion Mate APM, 1-Achsen-Nachlaufmodus	GFK-0781
.. .. .	—	Motion Mate APM, 1-Achsen-Standardmodus	GFK-0840
IC693APU302	—	Motion Mate APM, 2-Achsen-Nachlaufmodus	GFK-0781
.. .. .	—	Motion Mate APM, 2-Achsen-Standardmodus	GFK-0840
IC693MCS001/2	—	Power Mate J Bewegungsteuersystem (1 u. 2 Achsen)	GFK-1256
IC693APU305	—	E/A-Prozessormodul	GFK-1028
IC693CMM321	—	Ethernet-Kommunikationsmodul	GFK-1084



## E/A-Datenformate

Diskrete Ein- und Ausgabedaten werden als Bits im Cache-Memory (Statustabelle) gespeichert. Analoge Ein- und Ausgangsdaten werden als Worte gespeichert und liegen speicherresident in einem für diesen Zweck reservierten Teil des Anwender-RAMs.

## Standardbedingungen für die Modell 30 Ausgangsmodule

In der Standardeinstellung gehen die Ausgänge der Modell 30 Ausgangsmodule beim Einschalten der Versorgungsspannung auf AUS. Sie halten diesen Wert bis zum ersten Ausgabezyklus von der SPS. Analog-Ausgangsmodule können über einen Brückenstecker (neben dem abnehmbaren Klemmenblock des Moduls) so eingestellt werden, daß sie standardmäßig auf Null gehen oder ihren letzten Zustand halten. Da Analog-Ausgangsmodule auch über eine externe Stromquelle versorgt werden können, können sie selbst bei stromloser SPS ihren Betrieb im eingestellten Standardmodus fortsetzen.

## Diagnosedaten

Diagnosebits im %S-Speicher zeigen Ausfälle von E/A-Modulen und Diskrepanzen bei der E/A-Konfiguration an. Es gibt keine Diagnosedaten für einzelnen E/A-Punkte. Weitere Informationen zur Fehlerbearbeitung finden Sie in Kapitel 3 in diesem Handbuch.

## Globaldaten

Mit Genius-Globaldaten können bei der SPS Series 90-30 mit hoher Geschwindigkeit Daten von mehreren CPUs gemeinsam benutzt werden. Mit dem Genius-Buscontroller (IC693BEM331) in der CPU (ab Version 5) und dem erweiterten Genius-Kommunikationsmodul (IC693CMM302) können bis zu 128 Datenbytes per Rundsendung zu anderen SPS oder Computern gesendet werden. Von jedem der bis zu 30 anderen Teilnehmern am Netz können bis zu 128 Bytes empfangen werden. Zum Senden und Empfangen der Daten kann jeder Speichertyp verwendet werden, nicht nur %G-Globalbits. Das ursprüngliche Genius-Kommunikationsmodul (IC693CMM301) war auf feste %G-Adressen beschränkt und konnte pro serieller Busadresse von SBA 16 bis 23 nur 32 Bits austauschen. Dieses erste Modul sollte nicht verwendet werden, da das erweiterte GCM die 50-fache Kapazität besitzt.

Globaldaten können gemeinsam benutzt werden von speicherprogrammierbaren Steuerungen der Serien Fünf, Sechs und 90, die am gleichen Genius-E/A-Bus angeschlossen sind. Es gibt eine vorkonfigurierte Methode der gemeinsamen Datenbenutzung, so daß der Anwender keine Konfiguration durchführen muß.

Globaldaten werden von den Genius-Kommunikationsmodulen implementiert, die Daten auf der Grundlage ihrer seriellen Busadressen zu Genius-E/A-Referenzadressen senden. Jedes Modul kann Globaldaten von bis zu sieben anderen Genius-Kommunikationsmodulen lesen.

## Modell 20 E/A-Module

In Tabelle 2-10 finden Sie eine Auflistung der Modell 20 E/A-Module mit Bestellnummer, Anzahl E/A-Punkte und einer kurzen Beschreibung. Die E/A ist zusammen mit der Stromversorgung in eine Grundplatte integriert. Die technischen Daten und Anschlußpläne der einzelnen Modell 20 E/A-Module finden Sie in GFK-0551, *SPS Series 90-20, Anwenderhandbuch*.

**Tabelle 2-10 Modell 20 E/A-Module**

Bestellnummer	Beschreibung	E/A-Punkte
IC692MAA541	E/A- und Stromversorgungs-Grundmodul, 120 VAC Ein/120 VAC Aus/120 VAC Stromversorgung	16 Ein/12 Aus
IC692MDR541	E/A- und Stromversorgungs-Grundmodul 24 VDC Ein/Relais Aus/120 VAC Stromversorgung	16 Ein/12 Aus
IC692MDR741	E/A- und Stromversorgungs-Grundmodul 24V DC Ein/Relais Aus/240 VAC Stromversorgung	16 Ein/12 Aus
IC692CPU211	CPU-Modul, CPU Modell 211	—

## Micro-SPS

In Tabelle 2-11 finden Sie eine Auflistung der lieferbaren SPS Series 90 Micro mit Bestellnummer, Anzahl E/A-Punkte und einer kurzen Beschreibung. CPU, Stromversorgung und E/A bilden eine komplette Einheit. Die technischen Daten und Anschlußpläne der einzelnen Module finden Sie in GFK-1065, *SPS Series 90 Micro, Anwenderhandbuch*.

**Tabelle 2-11 Modelle der SPS Series 90 Micro**

Bestellnummer	Beschreibung	E/A-Punkte
IC693UDR001	CPU, Stromversorgung und E/A (alles in einer Einheit) Micro-14 Punkte DC Ein/Relais Aus, AC Stromversorgung	8 Ein/6 Aus
IC693UDR002	CPU, Stromversorgung und E/A (alles in einer Einheit) Micro-14 Punkte DC Ein/Relais Aus, DC Stromversorgung	8 Ein/6 Aus
IC693UAA003	CPU, Stromversorgung und E/A (alles in einer Einheit) Micro-14 Punkte AC Ein/AC Aus, AC Stromversorgung	8 Ein/6 Aus
IC693UDR005	CPU, Stromversorgung und E/A (alles in einer Einheit) Micro-28 Punkte DC Ein/Relais Aus, AC Stromversorgung	16 Ein/11 Relais Aus/1 DC Aus)
IC693UDR006	CPU, Stromversorgung und E/A (alles in einer Einheit) Micro-23 Punkte DC Ein/DC Aus, AC Stromversorgung	1 DC Aus/9 Relais Aus/2 Analog Ein/1 Analog Aus)
IC693UAA007	CPU, Stromversorgung und E/A (alles in einer Einheit) Micro-28 Punkte AC Ein/AC Aus, AC Stromversorgung	16 Ein/12 Aus
IC693UDR010	CPU, Stromversorgung und E/A (alles in einer Einheit) Micro-28 Punkte DC Ein/DC Aus, DC Stromversorgung	16 DC Ein/1 DC Aus/11 Relais Aus
IC693UEX011	14-Punkt Erweiterungseinheit DC Ein/RELais Aus, AC Stromversorgung	8 Ein/6 Aus



# Kapitel 3

## *Erläuterung und Behebung von Fehlern*

---

---

Dieses Kapitel bietet Hilfestellung bei der Fehlersuche im SPS-System Series 90-30. Es erläutert die Fehlerbeschreibungen in der SPS-Fehlertabelle und die Fehlerkategorien in den E/A-Fehlertabellen.

In diesem Kapitel werden bei der Erläuterung der einzelnen Fehler die Fehlerbeschreibung für die SPS-Fehlertabelle bzw. die Fehlerkategorie für die E/A-Fehlertabelle aufgelistet. Suchen Sie die Fehlerbeschreibung oder Fehlerkategorie, die dem Eintrag der entsprechenden Fehlertabelle in Ihrem Programmiergerätemenü entspricht. Darunter finden Sie dann eine Beschreibung der Fehlerursache und eine Anleitung, wie Sie den Fehler beheben können.

Kapitel 3 enthält folgende Abschnitte:

<b>Abschnitt</b>	<b>Titel</b>	<b>Beschreibung</b>	<b>Seite</b>
1	Fehlerbearbeitung	Beschreibung der in der SPS Series 90-30 möglichen Fehlertypen und deren Anzeige in den Fehlertabellen. Ebenfalls enthalten sind Beschreibungen der SPS- und E/A-Fehlertabellenanzeige.	3-2
2	SPS-Fehlertabelle – Erläuterungen	Gibt eine Fehlerbeschreibung der einzelnen SPS-Fehler und Anleitungen zur Fehlerbehebung.	3-7
3	E/A-Fehlertabelle – Erläuterungen	Beschreibung der Fehlerkategorien "Verlust von E/A-Modul" und "Hinzufügen von E/A-Modul".	3-16

## Abschnitt 1: Fehlerbearbeitung

In einem SPS-System Series 90-30 spricht man von einem Fehler, wenn bestimmten Ausfälle oder Zustände auftreten, die Betrieb und Leistung des Systems beeinträchtigen. Diese Zustände, wie zum Beispiel der Verlust eines E/A-Moduls oder Chassis, können die Fähigkeit der SPS zur Steuerung einer Maschine oder eines Prozesses beeinträchtigen. Diese Zustände können aber auch Vorteile bringen, wenn zum Beispiel ein neues Modul on-line geschaltet wird und dann benutzt werden kann. In anderen Fällen dienen diese Zustände nur als eine Warnung, wie bei einem Batterieausfallsignal, das anzeigt, daß die für den Speicherschutz eingesetzte Batterie ausgewechselt werden muß.

### Alarmprozessor

Der Zustand oder Ausfall selbst wird Fehler genannt. Empfängt und bearbeitet die CPU einen Fehler, sprechen wir von einem Alarm. Die Software in der CPU, die diese Zustände bearbeitet, wird Alarmprozessor genannt. Die Programmiersoftware bildet für den Alarmprozessor die Schnittstelle zum Anwender. Jeder erkannte Fehler wird in eine Fehlertabelle eingetragen und je nachdem entweder im SPS-Fehlertabellenmenü oder im E/A-Fehlertabellenmenü angezeigt.

### Fehlerklassen

Die SPS Series 90-30 kann mehrere Fehlerklassen erkennen. Hierzu gehören interne Fehler, externe Fehler und Betriebsfehler.

Fehlerklasse	Beispiele
Interne Fehler	Nicht reagierende Module Niedrige Batteriespannung Prüfsummenfehler im Speicher
Externe E/A-Fehler	Verlust von Chassis oder Modul Hinzufügen von Chassis oder Modul
Betriebsfehler	Kommunikationsfehler Konfigurationsfehler Paßwortfehler

## Systemreaktion auf Fehler

Ein Hardwarefehler wird entweder toleriert oder macht eine Abschaltung des Systems erforderlich. Ein E/A-Fehler kann zwar vom SPS-System toleriert werden, aber intolerabel für die Anwendung oder den gesteuerten Prozeß sein. Betriebsfehler werden normalerweise toleriert. Bei der Series 90-30 SPS besitzen die Fehler zwei Attribute:

Attribut	Beschreibung
Betroffene Fehlertabelle	E/A-Fehlertabelle SPS-Fehlertabelle
Fehlerwirkung	Fatal Diagnose Informativ

## Fehlertabellen

In der SPS werden zwei Fehlertabellen unterhalten: Während in die E/A-Fehlertabelle Fehler eingetragen werden, die sich auf das E/A-System beziehen, werden in der SPS-Fehlertabelle alle anderen Fehler protokolliert. In Tabelle 3-1 sind die Fehlergruppen, ihre Fehlerwirkung, die betroffenen Fehlertabellen und die symbolischen Adressen der betroffenen diskreten %S-Systempunkte aufgelistet.

**Tabelle 3-1 Zusammenfassung der Fehler**

Fehlergruppe	Fehlerwirkung	Fehlertabelle	Diskrete Spezial-Fehlerreferenzen			
			io_ftl	any_ftl	io_pres	los_ion
Verlorenes oder fehlendes E/A-Modul	Diagnose	I/O	io_ftl	any_ftl	io_pres	los_ion
Verlorenes oder fehlendes Zusatzmodul	Diagnose	PLC	sy_ftl	any_ftl	sy_pres	los_sio
Diskrepanz bei Systemkonfiguration	Fatal	PLC	sy_ftl	any_ftl	sy_pres	cfg_mm
SPS CPU Hardwarefehler	Fatal	PLC	sy_ftl	any_ftl	sy_pres	hrd_cpu
Programm-Prüfsummenfehler	Fatal	PLC	sy_ftl	any_ftl	sy_pres	pb_sum
Niedrige Batteriespannung	Diagnose	PLC	sy_ftl	any_ftl	sy_pres	low_bat
SPS-Fehlertabelle ist voll	Diagnose	—	sy_full			
E/A-Fehlertabelle ist voll	Diagnose	—	io_full			
Anwendungsfehler	Diagnose	PLC	sy_ftl	any_ftl	sy_pres	apl_ftl
Anwenderprogramm fehlt	Fatal	PLC	sy_ftl	any_ftl	sy_pres	no_prog
Fehler im Anwender-RAM	Fatal	PLC	sy_ftl	any_ftl	sy_pres	bad_ram
Paßwortfehler	Diagnose	PLC	sy_ftl	any_ftl	sy_pres	bad_pwd
Nullsystemkonfiguration	Diagnose	PLC	sy_ftl	any_ftl	sy_pres	nul_cfg
SPS-Softwarefehler	Fatal	PLC	sy_ftl	any_ftl	sy_pres	sft_cpu
SPS-Speicherfehler	Fatal	PLC	sy_ftl	any_ftl	sy_pres	stor_er
Konstante Zykluszeit überschritten	Diagnose	PLC	sy_ftl	any_ftl	sy_pres	ov_swp
Unbekannter SPS-Fehler	Fatal	PLC	sy_ftl	any_ftl	sy_pres	
Unbekannter E/A-Fehler	Fatal	I/O	io_ftl	any_ftl	io_pres	

### Fehlerwirkung

Es gibt zwei Fehlertabellen, die SPS-Fehlertabelle und die E/A-Fehlertabelle. Hierdurch wird es einfacher, einen Fehler zu finden, und die Tabellen werden nicht so lang.

Ein fataler Fehler wird in die entsprechende Tabelle eingetragen, setzt Diagnosevariablen und hält das System an. Ein Diagnosefehler wird in die entsprechende Tabelle eingetragen und setzt Diagnosevariablen. Ein informatorischer Fehler wird nur in die entsprechende Tabelle eingetragen.

Tabelle 3-2 zeigt mögliche Fehlerwirkungen:

**Tabelle 3-2 Fehlerwirkungen**

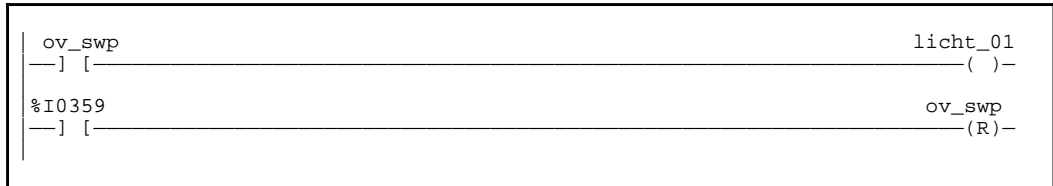
Fehlerwirkung	Reaktion der CPU
Fatal	Fehlereintrag in Fehlertabelle Fehlerreferenzen setzen Auf <b>STOP</b> gehen.
Diagnose	Fehlereintrag in Fehlertabelle Fehlerreferenzen setzen
Informatorisch	Fehlereintrag in Fehlertabelle

Wird ein Fehler erkannt, veranlaßt die CPU die entsprechende Fehlerwirkung. Die Fehlerwirkungen in der SPS Series 90-30 sind nicht konfigurierbar.

### Fehlerreferenzen

Bei den SPS Series 90-30, 90-20 und Micro gibt es einen Typ Fehlerreferenzen, die summarischen Fehlerreferenzen. summarische Fehlerreferenzen werden gesetzt um anzugeben, welcher Fehler aufgetreten ist. Die Fehlerreferenz bleibt solange gesetzt, bis sie vom Anwenderprogramm gelöscht wird oder bis die SPS gelöscht wird.

Das nachfolgende Beispiel zeigt, wie ein Fehlerbit gesetzt und dann wieder gelöscht wird. In dem Beispiel wird die Spule licht\_01 durchgeschaltet, wenn die Zykluszeit überschritten wird. Das Licht und der Kontakt OV\_SWP bleiben solange eingeschaltet, bis der Kontakt %I0359 geschlossen wird.



## Definitionen der Fehlerreferenz

Der Alarmprozessor verwaltet die Zustände von 128 Systembits im %S-Speicher. Mit diesen Fehlerreferenzen kann angezeigt werden, wo ein Fehler aufgetreten ist und von welchem Typ dieser Fehler ist. Fehlerreferenzen sind den %S-, %SA-, %SB- und %SC-Speichern zugeordnet, jede Fehlerreferenz besitzt eine symbolische Adresse. Die Referenzen können bei Bedarf im Anwenderprogramm verwendet werden. Eine Liste der Systemzustandsreferenzen finden Sie in Kapitel 2 dieses Handbuchs.

## Weitere Fehlerauswirkungen

Zwei der zuvor beschriebenen Fehler haben noch Nebenwirkungen, die in der nachstehenden Tabelle beschrieben werden:

Nebenwirkung	Beschreibung
SPS CPU Softwarefehler	Wird ein SPS CPU Softwarefehler protokolliert, geht die CPU der Serie 90-30 bzw. 90-20 unmittelbar in einen speziellen FEHLER-ZYKLUS-Modus. In diesem Modus ist keine Aktivität erlaubt. Zum Lösen dieses Zustands müssen Sie die SPS rücksetzen (d.h. Versorgungsspannung aus- und wieder einschalten).
Fehler beim Speichern von SPS-Folgen	Wird während einer Sequenzspeicherung (Speichern von Programmblöcken und anderen Daten, denen der spezielle Sequenzbeginn-Befehl vorangestellt ist und die mit dem Sequenzende-Befehl enden) die Datenverbindung zu dem die Speicherung durchführenden Programmiergerät unterbrochen oder tritt ein anderer Fehler auf, der das Laden beendet, wird der "Fehler beim Speichern von SPS-Sequenzen" eingetragen. Solange dieser Fehler im System ansteht geht die SPS nicht in RUN-Modus.

## Anzeige der SPS-Fehlertabelle

Im SPS-Fehlertabellenmenü werden SPS-Fehler wie Paßwortfehler, Diskrepanzen bei SPS-Konfiguration, Paritätsfehler und Kommunikationsfehler angezeigt.

Die Betriebsart der Programmiersoftware spielt dabei keine Rolle. Im Off-Line-Modus werden keine Fehler angezeigt. Im On-Line- oder Monitor-Modus werden die SPS-Fehlerdaten angezeigt. Im On-Line-Modus können Fehler gelöscht werden (diese Funktion kann durch ein Paßwort geschützt werden).

Steht ein gelöschter Fehler weiterhin an, wird er nicht nochmals in die Tabelle eingetragen (Ausnahme: Fehler "Niedrige Batteriespannung").

## Anzeige der E/A-Fehlertabelle

Im E/A-Fehlertabellenmenü werden E/A-Fehler wie zum Beispiel Schaltkreisfehler, Adressenkonflikte, fixierte Schaltkreise und Fehler am E/A-Bus angezeigt.

Die Betriebsart der Programmiersoftware spielt keine Rolle. Im Off-Line-Modus, werden keine Fehler angezeigt. Im On-Line- oder Monitor-Modus, werden die E/A-Fehlerdaten angezeigt. Im On-Line-Modus können Fehler gelöscht werden (diese Funktion kann durch ein Paßwort geschützt werden).

Steht ein gelöschter Fehler weiterhin an, wird er nicht nochmals in die Tabelle eingetragen.



## Zugriff auf weitere Fehlerdaten

Die Fehlertabellen enthalten Grundinformationen zu dem Fehler. Weitere Angaben zu den einzelnen Fehlern erhalten Sie ebenso über die Programmiersoftware wie einen hexadezimalen Speicherabzug des Fehlers.

Im letzten Eintrag, Korrekturen, werden für die einzelnen Fehlerbeschreibungen in diesem Kapitel die Aktionen beschrieben, mit denen Sie den Fehler beheben können. Beachten Sie, daß bei einigen dieser Aktionen der folgende Hinweis steht:

**Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.**

Dieser Hinweis bedeutet, daß Sie der Serviceabteilung sagen müssen, was Sie direkt aus der Fehlertabelle ablesen können *und* welche hexadezimale Daten Sie sehen, wenn Sie **Ctrl-F** drücken. Ihre Ansprechpartner in der Serviceabteilung können Ihnen dann weitere Anweisungen zur Fehlerbehebung geben.

## Abschnitt 2: SPS-Fehlertabelle – Erläuterungen

Jede Fehlererläuterung enthält eine Fehlerbeschreibung und Anweisungen, wie der Fehler behoben werden kann. Zahlreiche Fehlerbeschreibungen haben mehrere Ursachen. In diesen Fällen wird der Fehlercode, der bei den zusätzlichen Fehlerinformationen angezeigt wird, zur Unterscheidung unterschiedlicher Fehlerzustände mit gleicher Fehlerbeschreibung benutzt. Der Fehlercode besteht aus den ersten beiden Hexadezimalstellen der fünften Zahlengruppe (siehe nachstehendes Beispiel).

```
01 000000 01030100 0902 0200 000000000000
                        |
                        | Fehlercode (die ersten beiden
                        | Stellen in der fünften Gruppe)
```

Einige Fehler können auftreten, wenn der RAM auf der CPU-Platine der SPS fehlerbehaftet ist. Die gleichen Fehler können auch auftreten, wenn das System abgeschaltet wurde und die Batteriespannung zu niedrig war, um den Speicherinhalt zu erhalten. Um hier zu vermeiden, daß die gleichen Anweisungen häufig wiederholt werden müssen, wird bei möglichen Speicherfehlern als Fehlerursache lediglich angegeben:

**Führen Sie die Korrekturmaßnahmen für Speicherfehler durch.**

Das heißt:

1. Wechseln Sie die Batterie aus, wenn das System abgeschaltet wurde. Die Batteriespannung kann für den Erhalt des Speicherinhalts zu niedrig sein.
2. Wechseln Sie die CPU-Platine der SPS aus. Die Schaltkreise dieser Platine können fehlerhaft sein.

Die nachstehende Tabelle soll Ihnen helfen, schnell eine bestimmte SPS-Fehlerbeschreibung in diesem Abschnitt zu finden. Die einzelnen Einträge sind in der Reihenfolge aufgeführt, wie sie auf dem Programmiergeräte-Bildschirm erscheinen.

Fehlerbeschreibung	Seite
Verlorenes oder fehlendes Zusatzmodul	3-8
Rückgesetztes, hinzugefügtes oder überzähliges Zusatzmodul	3-8
Diskrepanz bei Systemkonfiguration	3-9
Softwarefehler bei Zusatzmodul	3-9
Programmblock-Prüfsummenfehler	3-10
Signal "niedrige Batteriespannung"	3-11
Konstante Zykluszeit überschritten	3-11
Anwendungsfehler	3-11
Anwenderprogramm fehlt	3-12
Verstümmeltes Anwenderprogramm beim Einschalten	3-12
Paßwortfehler	3-12
SPS CPU Systemsoftwarefehler	3-13
Kommunikationsfehler beim Speichern	3-15

## Fehlerwirkungen

Bei einem **fatalen** Fehler geht die SPS am Ende des Zyklus, in dem der Fehler aufgetreten ist, auf STOP. **Diagnosefehler** werden protokolliert und es werden die entsprechenden Fehlerkontakte gesetzt. **Informatorische** Fehler werden nur in die SPS-Fehlertabelle eingetragen.

## Verlorenes oder fehlendes Zusatzmodul

Die Fehlergruppe "**Verlorenes oder fehlendes Zusatzmodul**" tritt auf, wenn ein PCM, CMM oder ADC nicht antwortet. Der Fehler kann beim Einschalten auftreten, wenn das Modul fehlt, oder im Betrieb, wenn das Modul nicht antwortet. Die Fehlerwirkung dieser Gruppe ist **Diagnose**.

<b>Fehlercode:</b>	1, 42
<b>Name:</b>	Software-Rücksetzen von Zusatzmodul fehlgeschlagen
<b>Beschreibung:</b>	CPU der SPS kann nach Software-Rücksetzen den Datenverkehr mit dem Zusatzmodul nicht mehr aufnehmen.
<b>Beseitigung:</b>	<ol style="list-style-type: none"> <li>(1) Versuchen Sie nochmals Software-Rücksetzen.</li> <li>(2) Wechseln Sie das Zusatzmodul aus.</li> <li>(3) Schalten Sie das System ab. Überprüfen Sie die Kabelanschlüsse und den Sitz des PCM im Chassis</li> <li>(4) Wechseln Sie die Kabel aus.</li> </ol>
<b>Fehlercode:</b>	Alle übrigen
<b>Name:</b>	Modulfehler während Konfiguration
<b>Beschreibung:</b>	Die SPS-Betriebssoftware erzeugt dieses Fehlermeldung, wenn ein Modul beim Einschalten oder beim Speichern der Konfiguration ausfällt.
<b>Beseitigung:</b>	<ol style="list-style-type: none"> <li>(1) System abschalten. Wechseln Sie das Modul in diesem Chassis und Steckplatz aus.</li> </ol>

## Rückgesetztes, hinzugefügtes oder überzähliges Zusatzmodul

Die Fehlergruppe "**Rückgesetztes, hinzugefügtes oder überzähliges Zusatzmodul**" tritt auf, wenn ein Zusatzmodul (PCM, ADC, usw.) on-line geht, rückgesetzt wird oder im Chassis festgestellt wird, ohne daß es in der Konfiguration angegeben wurde. Die Fehlerwirkung bei dieser Gruppe ist Diagnose. Drei fehlerspezifische Bytes enthalten zusätzliche Angaben zu dem Fehler.

<b>Beseitigung:</b>	<ol style="list-style-type: none"> <li>(1) Aktualisieren Sie die Konfiguration und nehmen Sie das Modul mit auf.</li> <li>(2) Entfernen Sie das Modul aus dem System.</li> </ol>
---------------------	--

## Diskrepanz bei Systemkonfiguration

Die Fehlergruppe **”Diskrepanz bei Systemkonfiguration”** tritt auf, wenn das in einem Steckplatz eingebaute Modul nicht den Angaben in der Konfigurationsdatei entspricht. Die Fehlerwirkung ist **fatal**.

<b>Fehlercode:</b>	1
<b>Name:</b>	Diskrepanz bei Systemkonfiguration
<b>Beschreibung:</b>	Die SPS-Betriebssoftware (Systemkonfigurierer) erzeugt diesen Fehler, wenn das in einen Steckplatz eingebaute Modul nicht von dem Typ ist, der in der Konfigurationsdatei für diesen Steckplatz angegeben wurde, oder wenn das vorhandene Chassis nicht mit dem konfigurierten Chassistyp übereinstimmt.
<b>Beseitigung:</b>	Diskrepanz feststellen und Modul oder Chassis neu konfigurieren.
<b>Fehlercode:</b>	6
<b>Name:</b>	Diskrepanz bei Systemkonfiguration
<b>Beschreibung:</b>	Wie bei Fehlercode 1 tritt dieser Fehler auf, wenn das in einen Steckplatz eingebaute Modul nicht von dem Typ ist, der in der Konfigurationsdatei für diesen Steckplatz angegeben wurde, oder wenn das vorhandene Chassis nicht mit dem konfigurierten Chassistyp übereinstimmt.
<b>Beseitigung:</b>	Diskrepanz feststellen und Modul oder Chassis neu konfigurieren.
<b>Fehlercode:</b>	18
<b>Name:</b>	Nicht unterstützte Hardware
<b>Beschreibung:</b>	In einer 311, 313 oder 323 oder in einem Erweiterungschassis wurde ein PCM oder PCM-ähnliches Modul eingebaut.
<b>Beseitigung:</b>	PCM oder PCM-ähnliches Modul entfernen oder eine CPU einbauen, die das PCM unterstützt.
<b>Fehlercode:</b>	26
<b>Name:</b>	Modul belegt – Konfiguration von Modul noch nicht angenommen.
<b>Beschreibung:</b>	Das Modul kann derzeit keine neue Konfiguration annehmen, da es mit einem anderen Vorgang beschäftigt ist.
<b>Beseitigung:</b>	Lassen Sie das Modul den aktuellen Vorgang beenden und speichern Sie dann die Konfiguration nochmals.
<b>Fehlercode:</b>	51
<b>Name:</b>	END-Funktion aus SFC-Aktion heraus ausgeführt.
<b>Beschreibung:</b>	Die Lage einer END-Funktion in einem SFC-Programm oder in einem über SFC aufgerufenen Programm verursacht diesen Fehler.
<b>Beseitigung:</b>	END-Funktion aus SFC-Programm bzw. dem über SFC aufgerufenen Programm entfernen.

## Softwarefehler bei Zusatzmodul

Die Fehlergruppe **”Softwarefehler bei Zusatzmodul”** tritt auf, wenn ein nicht behebbarer Softwarefehler in einem PCM- oder ADC-Modul auftritt. Die Fehlerwirkung für diese Gruppe ist **fatal**.

<b>Fehlercode:</b>	Alle
<b>Name:</b>	COMMREQ kommt zu oft
<b>Beschreibung:</b>	Die COMMREQ-Befehle werden schneller zu dem Modul geschickt, als sie dort bearbeitet werden können.
<b>Beseitigung:</b>	SPS-Programm ändern und COMMREQ-Häufigkeit reduzieren.

## Programmblock-Prüfsummenfehler

Die Fehlergruppe "**Programmblock-Prüfsummenfehler**" tritt auf, wenn die CPU der SPS einen Fehler in den durch die SPS empfangenen Programmblöcken erkennt. Sie tritt auch auf, wenn die CPU der SPS einen Prüfsummenfehler im Einschalt-Speichertest oder während der Hintergrundprüfung im RUN-Modus erkennt. Die Fehlerwirkung für diese Gruppe ist **fatal**.

<b>Fehlercode:</b>	Alle
<b>Name:</b>	Programmblock-Prüfsummenfehler
<b>Beschreibung:</b>	Die SPS-Betriebssoftware erzeugt diesen Fehler bei einem verstümmelten programmblock.
<b>Beseitigung:</b>	<ol style="list-style-type: none"><li>(1) SPS-Speicher löschen und Speicherversuch wiederholen.</li><li>(2) Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.</li></ol>

## Signal für niedrige Batteriespannung

Die Fehlergruppe "Signal für niedrige Batteriespannung" tritt auf, wenn die CPU der SPS erkennt, daß die Spannung der Batterie in der SPS-Stromversorgung zu niedrig ist oder wenn ein Modul (z.B. das PCM) zu niedrige Batteriespannung meldet. Die Fehlerwirkung dieser Gruppe ist **Diagnose**.

<b>Fehlercode:</b>	0
<b>Name:</b>	Batterieausfallsignal
<b>Beschreibung:</b>	Die Batterie der CPU (oder eines anderen batteriegepufferten Moduls) ist leer.
<b>Beseitigung:</b>	Batterie auswechseln. Die Versorgungsspannung zum Chassis nicht ausschalten!
<b>Fehlercode:</b>	1
<b>Name:</b>	Signal für niedrige Batteriespannung
<b>Beschreibung:</b>	Batterie in CPU oder einem anderen Modul meldet niedrige Spannung.
<b>Beseitigung:</b>	Batterie auswechseln. Die Versorgungsspannung zum Chassis nicht ausschalten!

## Konstante Zykluszeit überschritten

Die Fehlergruppe "Konstante Zykluszeit überschritten" tritt auf, wenn die CPU der SPS im Modus "konstante Zykluszeit" arbeitet und erkennt, daß die Zykluszeit überschritten wurde. Die zusätzlichen Fehlerdaten enthalten die tatsächliche Zyklusdauer in den ersten beiden Bytes und den Programmnamen in den nächsten acht Bytes. Die Fehlerwirkung der Gruppe ist **Diagnose**.

<b>Beseitigung:</b>	(1) Wert für konstante Zykluszeit erhöhen
	(2) Anwenderprogramm verkürzen.

## Anwendungsfehler

Die Fehlergruppe "Anwendungsfehler" tritt auf, wenn die CPU der SPS einen Fehler im Anwenderprogramm erkennt. Die Fehlerwirkung dieser Gruppe ist **Diagnose**.

<b>Fehlercode:</b>	7
<b>Name:</b>	Unterprogrammschachtelungsüberschritten
<b>Beschreibung:</b>	Unterprogramme können bis zu einer Tiefe von 8 Ebenen geschachtelt werden. Ein Unterprogramm kann ein weiteres Unterprogramm aufrufen, das wieder ein Unterprogramm aufrufen kann, bis eine Schachtelungstiefe von 8 erreicht ist.
<b>Beseitigung:</b>	Programm so ändern, daß Schachtelungstiefe von 8 nicht überschritten wird.
<b>Fehlercode:</b>	1B
<b>Name:</b>	COMMREQ wurde wegen Speichereinschränkung bei der SPS nicht bearbeitet
<b>Beschreibung:</b>	Kommunikationsanforderungen ohne Wartezeit können in der Warteschlange schneller eingetragen werden, als sie verarbeitet werden können (z.B. eine pro Zyklus). Sammeln sich in einer solchen Situation so viele Kommunikationsanforderungen an, daß die SPS weniger als das Speicherminimum zur Verfügung hat, wird die Kommunikationsanforderung zum Fehler erklärt und nicht bearbeitet.
<b>Beseitigung:</b>	Weniger Kommunikationsanforderungen senden oder die im System ausgetauschte Mail auf andere Weise reduzieren.
<b>Fehlercode:</b>	5A
<b>Name:</b>	Anforderung anwenderseitige Abschaltung
<b>Beschreibung:</b>	Die SPS-Betriebssoftware (Funktionsblöcke) erzeugt diesen informatrischen Alarm, wenn im Anwenderprogramm Bedienanforderung #13 (SPS abschalten) bearbeitet wird.
<b>Beseitigung:</b>	Keine. Informatrischer Alarm.

## Anwenderprogramm fehlt

Die Fehlergruppe "**Kein Anwenderprogramm**" tritt auf, wenn die CPU der SPS angewiesen wird, von STOP auf RUN zu gehen oder Daten in die SPS zu speichern und in der SPS kein Anwenderprogramm vorhanden ist. Die CPU der SPS erkennt das Fehlen eines Anwenderprogramms beim Einschalten. Die Fehlerwirkung dieser Gruppe ist **informativ**.

<b>Beseitigung:</b>	Anwenderprogramm vor Übergang in RUN-Modus laden.
---------------------	---

## Verstümmeltes Anwenderprogramm beim Einschalten

Die Fehlergruppe "**Verstümmeltes Anwenderprogramm beim Einschalten**" tritt auf, wenn die CPU der SPS einen Fehler im Anwender-RAM erkennt. Die CPU der SPS bleibt solange in STOP-Modus, bis Anwenderprogramm und Konfigurationsdatei fehlerfrei geladen sind. Die Fehlerwirkung dieser Gruppe ist **fatal**.

<b>Fehlercode:</b>	1
<b>Name:</b>	Verstümmeltes Anwenderprogramm beim Einschalten.
<b>Beschreibung:</b>	Die SPS-Betriebssoftware erzeugt diesen Fehler, wenn sie beim Einschalten verstümmeltes Anwender-RAM erkennt.
<b>Beseitigung:</b>	<ol style="list-style-type: none"> <li>(1) Konfigurationsdatei, Anwenderprogramm und Referenzen (falls vorhanden) neu laden.</li> <li>(2) Batterie in CPU der SPS austauschen.</li> <li>(3) Speichererweiterungsplatine der SPS-CPU austauschen.</li> <li>(4) SPS-CPU austauschen.</li> </ol>
<b>Fehlercode:</b>	2
<b>Name:</b>	Unzulässiger Boolescher Operationscode erkannt
<b>Beschreibung:</b>	Die SPS-Betriebssoftware erzeugt diesen Fehler, wenn sie im Anwenderprogramm eine fehlerhafte Anweisung erkennt.
<b>Beseitigung:</b>	<ol style="list-style-type: none"> <li>(1) Anwenderprogramm und Referenzen (falls vorhanden) neu speichern.</li> <li>(2) Speichererweiterungsplatine der SPS-CPU austauschen.</li> <li>(3) SPS-CPU austauschen.</li> </ol>

## Paßwortfehler

Die Fehlergruppe "Paßwortfehler" tritt auf, wenn die CPU der SPS eine Aufforderung zum Wechsel auf eine neue Privilegeebene erhält und das mit der Anforderung eingegangene Paßwort für diese Ebene nicht gilt. Die Fehlerwirkung dieser Gruppe ist **informativ**.

<b>Beseitigung:</b>	Anforderung mit richtigem Paßwort wiederholen.
---------------------	--

## CPU-Systemsoftwarefehler in der SPS

Die Fehler der Fehlergruppe "CPU-Systemsoftwarefehler" werden von der Betriebssoftware der CPU in der SPS Series 90-30, 90-20 oder Micro erzeugt. Sie können im Systembetrieb an unterschiedlichen Punkten auftreten. Tritt ein fataler Fehler auf, geht die CPU *sofort* in einen speziellen FEHLERZYKLUS-Modus. In dieser Betriebsart ist nur der Datenverkehr mit dem Programmiergerät möglich. Sie können diesen Zustand nur aufheben, indem Sie die Betriebsspannung der SPS aus- und wieder einschalten. Die Fehlerwirkung dieser Gruppe ist **fatal**.

<b>Fehlercode:</b>	1 bis B
<b>Name:</b>	Anwenderspeicher konnte nicht belegt werden
<b>Beschreibung:</b>	Die SPS-Betriebssoftware (Speicherverwaltung) erzeugt diesen Fehler, wenn die Software die Speicherverwaltung auffordert, einen oder mehrere Blöcke aus dem Anwender-RAM zu belegen oder freizugeben, die unzulässig waren. Diese Fehler dürfen in einem Gerät aus der Serienproduktion nicht vorkommen.
<b>Beseitigung:</b>	Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.
<b>Fehlercode:</b>	D
<b>Name:</b>	Kein Systemspeicher verfügbar
<b>Beschreibung:</b>	Die SPS-Betriebssoftware (E/A-Zyklussteuerung) erzeugt diesen Fehler, wenn ihre Anforderung nach einen Systemspeicherblock von der Speicherverwaltung abgelehnt wurde, da im Systemspeicher kein Platz frei ist. Tritt der Fehler während der Ausführung eines DOIO-Funktionsblocks auf, ist er <b>informativ</b> . Tritt er bei der Einschaltinitialisierung oder automatischen Konfiguration auf, ist er jedoch <b>fatal</b> .
<b>Beseitigung:</b>	Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.
<b>Fehlercode:</b>	E
<b>Name:</b>	Systemspeicher konnte nicht freigegeben werden.
<b>Beschreibung:</b>	Die SPS-Betriebssoftware (E/A-Zyklussteuerung) erzeugt diesen Fehler, wenn ihre Anforderung nach Freigabe eines Blocks im Systemspeicher nicht zu dem gewünschten Ergebnis geführt hat. Dieser Fehler kann nur während der Ausführung eines DOIO-Funktionsblocks auftreten.
<b>Beseitigung:</b>	(1) Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten. (2) Führen Sie die Korrekturmaßnahmen für Speicherfehler durch.
<b>Fehlercode:</b>	10
<b>Name:</b>	Unzulässige Abfrageanforderung von E/A-Zyklussteuerung
<b>Beschreibung:</b>	Die SPS-Betriebssoftware (E/A-Zyklussteuerung) erzeugt diesen Fehler, wenn weder Betriebssystem noch DOIO-Funktionsblockabfrage eine vollständige oder teilweise Abfrage der E/A fordern. Dieser Fehler darf in einem Gerät aus der Serienproduktion nicht vorkommen.
<b>Beseitigung:</b>	Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.
<b>Fehlercode:</b>	13
<b>Name:</b>	SPS-Betriebssoftwarefehler
<b>Beschreibung:</b>	Die SPS-Betriebssoftware erzeugt diesen Fehler, wenn bestimmte Probleme bei der SPS-Betriebssoftware auftreten. Dieser Fehler darf in einem Gerät aus der Serienproduktion nicht vorkommen.
<b>Beseitigung:</b>	(1) Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten. (2) Führen Sie die Korrekturmaßnahmen für Speicherfehler durch.



<b>Fehlercode:</b>	14, 27
<b>Name:</b>	VerstümmelterSPS-Programmspeicher
<b>Beschreibung:</b>	Die SPS-Betriebssoftware erzeugt diesen Fehler, wenn bestimmte Probleme bei der SPS-Betriebssoftware auftreten. Dieser Fehler darf in einem Gerät aus der Serienproduktion nicht vorkommen.
<b>Beseitigung:</b>	<ol style="list-style-type: none"> <li>(1) Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.</li> <li>(2) Führen Sie die Korrekturmaßnahmen für Speicherfehler durch.</li> </ol>
<b>Fehlercode:</b>	27 bis 4E
<b>Name:</b>	SPS-Betriebssystemfehler
<b>Beschreibung:</b>	Die SPS-Betriebssoftware erzeugt diesen Fehler, wenn bestimmte Probleme bei der SPS-Betriebssoftware auftreten. Dieser Fehler darf in einem Gerät aus der Serienproduktion nicht vorkommen.
<b>Beseitigung:</b>	Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.
<b>Fehlercode:</b>	4F
<b>Name:</b>	Kommunikationsausfall
<b>Beschreibung:</b>	Die SPS-Betriebssoftware (Bedienanforderungsprozessor) erzeugt diesen Fehler, wenn sie versucht, auf eine Anforderung zu antworten, die Rückwandplatten-Kommunikation erfordert, und dabei eine ablehnende Antwort erhält.
<b>Beseitigung:</b>	<ol style="list-style-type: none"> <li>(1) Überprüfen Sie den Bus auf außergewöhnliche Aktivitäten.</li> <li>(2) Wechseln Sie das intelligente Zusatzmodul aus, an das die Anforderung gerichtet war.</li> </ol>
<b>Fehlercode:</b>	50, 51, 53
<b>Name:</b>	Systemspeicherfehler
<b>Beschreibung:</b>	Die SPS-Betriebssoftware erzeugt diese Fehler, wenn ihre Anforderung nach einem Systemspeicherblock von der Speicherverwaltung abgelehnt wurde, da kein Speicher verfügbar ist oder der Speicher Fehler enthält.
<b>Beseitigung:</b>	<ol style="list-style-type: none"> <li>(1) Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.</li> <li>(2) Führen Sie die Korrekturmaßnahmen für Speicherfehler durch.</li> </ol>
<b>Fehlercode:</b>	52
<b>Name:</b>	Fehler bei Kommunikation über Rückwandplatine
<b>Beschreibung:</b>	Die SPS-Betriebssoftware (Bedienanforderungsprozessor) erzeugt diesen Fehler, wenn sie versucht, auf eine Anforderung zu antworten, die Rückwandplatten-Kommunikation erfordert, und dabei eine ablehnende Antwort erhält.
<b>Beseitigung:</b>	<ol style="list-style-type: none"> <li>(1) Überprüfen Sie den Bus auf außergewöhnliche Aktivitäten.</li> <li>(2) Wechseln Sie das intelligente Zusatzmodul aus, an das die Anforderung gerichtet war.</li> <li>(3) Überprüfen Sie, ob das Parallelkabel des Programmiergerätes richtig angeschlossen ist.</li> </ol>
<b>Fehlercode:</b>	Alle anderen
<b>Name:</b>	Interner Systemfehler der SPS-CPU
<b>Beschreibung:</b>	Ein interner Fehler ist aufgetreten, der in einem Gerät aus der Serienproduktion nicht vorkommen darf.
<b>Beseitigung:</b>	Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.

## Datenübertragungsfehler beim Speichern

Die Fehlergruppe **„Datenübertragungsfehler beim Speichern“** tritt während des Speicherns von Programmblöcken oder anderen Daten zur SPS auf. Der Strom der Befehle und Daten zum Speichern von Programmblöcken und Daten beginnt mit einem speziellen Sequenzanfangsbefehl und endet mit einem Sequenzendebefehl. Wird der Datenverkehr mit dem Programmiergerät, das den Speichervorgang durchführt, unterbrochen oder tritt ein anderer Fehler auf, der das Laden beendet, wird dieser Fehler eingetragen. Solange der Fehler im System ansteht, geht die Steuerung nicht in den RUN-Modus.

Dieser Fehler wird beim Einschalten nicht automatisch gelöscht. Der Anwender muß einen speziellen Befehl zum Löschen des Zustands eingeben. Die Fehlerwirkung dieser Gruppe ist **fatal**.

<b>Beseitigung:</b>	Fehler löschen und erneut versuchen, Programm oder Konfigurationsdatei zu laden.
---------------------	--

### Abschnitt 3: E/A-Fehlertabelle – Erläuterungen

Die E/A-Fehlertabelle enthält Fehlerdaten in drei Klassifizierungen:

- Fehlerkategorie
- Fehlertyp
- Fehlerbeschreibung

Die auf der nächsten Seite beschriebenen Fehler besitzen zwar eine Fehlerkategorie, haben aber weder Fehlertyp noch Fehlergruppe.

Jede Fehlererläuterung enthält eine Fehlerbeschreibung und Anweisungen, wie der Fehler behoben werden kann. Zahlreiche Fehlerbeschreibungen haben mehrere Ursachen. In diesen Fällen wird der Fehlercode, der bei den über **Ctrl-F** zugänglichen zusätzlichen Fehlerinformationen angezeigt wird, zur Unterscheidung unterschiedlicher Fehlerzustände mit gleicher Fehlerbeschreibung benutzt. Weitere Angaben zu der Verwendung von Ctrl-F finden Sie in Anhang B dieses Handbuches. Der Fehlercode besteht aus den ersten beiden Hexadezimalstellen der fünften Zahlengruppe (siehe nachstehendes Beispiel).

```
01 000000 01030100 0902 0200 000000000000
                        |
                        |_____ Fehlercode (die ersten beiden
                                Hexadezimalstellen in der fünften Gruppe)
```

Die nachstehende Tabelle soll Ihnen helfen, schnell eine bestimmte SPS-Fehlerbeschreibung in diesem Abschnitt zu finden. Die einzelnen Einträge sind in der Reihenfolge aufgeführt, wie sie auf dem Programmiergeräte-Bildschirm erscheinen.

## Verlorenes E/A-Modul

Die Fehlerkategorie "**Verlorenes E/A-Modul**" bezieht sich auf diskrete und analoge Modell 30 E/A-Module. Mit dieser Kategorie sind weder Fehlertypen noch Fehlerbeschreibungen verknüpft. Die Fehlerwirkung ist **Diagnose**.

<b>Beschreibung:</b>	Die SPS-Betriebssoftware erzeugt diesen Fehler wenn sie merkt, daß ein Modell 30 E/A-Modul nicht mehr auf Befehle von der SPS-CPU reagiert oder wenn ein leerer Steckplatz laut Konfigurationsdatei ein Modul enthalten müßte.
<b>Beseitigung:</b>	<ol style="list-style-type: none"> <li>(1) Wechseln Sie das Modul aus.</li> <li>(2) Korrigieren Sie die Konfigurationsdatei</li> <li>(3) Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.</li> </ol>

## Hinzugefügtes E/A-Modul

Die Fehlerkategorie "**Hinzugefügtes E/A-Modul**" bezieht sich auf diskrete und analoge Modell 30 E/A-Module. Mit dieser Kategorie sind weder Fehlertypen noch Fehlerbeschreibungen verknüpft. Die Fehlerwirkung ist **Diagnose**.

<b>Beschreibung:</b>	Die SPS-Betriebssoftware erzeugt diesen Fehler, wenn ein zuvor fehlerhaftes E/A-Modul wieder zum normalen Betrieb zurückkehrt.
<b>Beseitigung:</b>	<ol style="list-style-type: none"> <li>(1) Keine Maßnahmen erforderlich, wenn das Modul entfernt oder ausgewechselt wurde oder die Versorgungsspannung zum dezentralen Chassis aus- und wieder eingeschaltet wurde.</li> <li>(2) Aktualisieren Sie die Konfiguration oder entfernen Sie das Modul</li> </ol>
<b>Beschreibung:</b>	Die SPS-Betriebssoftware erzeugt diesen Fehler, wenn sie ein Modell 30 E/A-Modul in einem Steckplatz erkennt, der laut Konfigurationsdatei leer sein sollte.
<b>Beseitigung:</b>	<ol style="list-style-type: none"> <li>(1) Entfernen Sie das Modul (es kann z.B. im falschen Steckplatz stecken).</li> <li>(2) Aktualisieren und speichern Sie die Konfigurationsdatei so, daß sie das überzählige Modul enthält.</li> </ol>



# Kapitel 4

## Befehlsvorrat der Series 90-30/20/Micro

Bei der Programmierung wird ein Anwenderprogramm für eine SPS erzeugt. Die speicherprogrammierbaren Steuerungen von Series 90-30, Series 90-20 und Series 90 Micro besitzen einen gemeinsamen Befehlsvorrat und können daher alle mit dieser Software programmiert werden. In diesem Kapitel werden die zur Erstellung von Kontaktplanprogrammen in den SPS Series 90-30 und 90-20 möglichen Programmieranweisungen beschrieben.

Anweisungen zur Installation der Logicmaster 90-30/20/Micro Software finden Sie in GFK-0466, *Logicmaster 90-30/20/Micro Programmiersoftware, Anwenderhandbuch*. Dort wird erläutert, wie Sie ein Programm erstellen, übertragen, editieren und ausdrucken können.

Unter Konfiguration versteht man den Vorgang, den Hardwaremodulen im System logische Adressen und andere Charakteristiken zuzuordnen. Dies kann entweder vor oder nach der Programmierung mit der Konfigurationssoftware oder dem Hand-Programmiergerät erfolgen (es wird empfohlen, *zuerst zu konfigurieren* und dann erst zu programmieren). Schlagen Sie in GFK-0466, *Logicmaster 90-30/20/Micro Programmiersoftware, Anwenderhandbuch*, nach, wenn Sie sich unschlüssig sind, ob Sie jetzt mit der Programmierung beginnen sollen.

Dieses Kapitel enthält folgende Abschnitte:

Ab-schnitt	Titel	Beschreibung	Seite
1	Relaisfunktionen	Kontakte, Spulen und Verbindungen	4-2
2	Timer und Zähler	Einschaltverzögerungs-Timer, Stoppuhren, Aufwärtszähler und Abwärtszähler	4-9
3	Arithmetische Funktionen	Addition, Subtraktion, Multiplikation, Division, Modulo-Division, Quadratwurzel, trigonometrische Funktionen, Logarithmus-/Exponential-Funktionen und Bogenmaßumrechnung <i>Trigonometrische Funktionen, Logarithmus-/Exponential-Funktionen und Bogenmaßumrechnungs-Funktionen sind nur bei CPU-Modell 352 verfügbar.</i>	4-26
4	Relationale Funktionen	Vergleich zweier Zahlen: Gleich, ungleich, größer als, größer als oder gleich, kleiner als, kleiner als oder gleich.	4-41
5	Bitoperationsfunktionen	Vergleichs- und Verschiebeoperationen mit Bitfolgen.	4-46
6	Datenverschiebefunktionen	Grundlegende Datenverschiebefunktionen.	4-68
7	Tabellenfunktionen	Verwendung von Tabellenfunktionen für die Eingabe von Werten in Tabellen und das Kopieren von Werten aus Tabellen.	4-85
8	Konvertierungsfunktionen	Umwandlung von Daten aus einem Zahlenformat in ein anderes.	4-93
9	Steuerfunktionen	Einsatz der Steuerfunktionen zur Beschränkung der Programmausführung und Veränderung der Art, wie die CPU das Anwenderprogramm bearbeitet.	4-106

## Abschnitt 1: Relaisfunktionen

In diesem Kapitel wird die Verwendung von Kontakten, Spulen und Verbindungen in den Strompfaden eines Kontaktplanprogramms erläutert.

Funktion	Seite
Spulen und negierte Spulen	4-3
Schließer- und Öffnerkontakte	4-2
Remanente und negierte remanente Spulen	4-5
Spulen für positive und negative Übergänge	4-5
Setz- und Rücksetz-Spulen	4-6
Remanente Setz- und Rücksetz-Spulen	4-7
Horizontal- und Vertikalverbindungen	4-7
Fortsetzspulen und -kontakte	4-8

### Kontakte

Mit einem Kontakt wird der Zustand einer Maschine oder einer internen Referenz überwacht. Der Stromfluß über einen Kontakt hängt vom Zustand der überwachten Referenz und vom Kontakttyp ab. Eine Referenz ist EIN, wenn ihr Zustand 1 ist und AUS, wenn ihr Zustand 0 ist.

**Tabelle 4-1 Kontakttypen**

Kontakttyp	Symbol	Kontakt schaltet Stromfluß nach rechts durch,
Schließer	— —	wenn die Referenz EIN ist
Öffner	—/ —	wenn die Referenz AUS ist
Fortsetzkontakt	<+>——	wenn der vorhergehende Fortsetzmerker EIN ist.

## Spulen

Über Spulen werden diskrete Referenzen gesteuert. Der Stromfluß zu einer Spule muß über eine Bedingungslogik gesteuert werden. Spulen verursachen direkt eine Aktion, sie geben keinen Stromfluß nach rechts weiter. Soll als Ergebnis eines Kontaktzustandes ein weiterer Programmteil durchlaufen werden, dann muß für diese Spule eine interne Referenz oder eine Kombination Fortsetzmerker/-kontakt verwendet werden.

Spulen stehen immer ganz rechts in einer Programmzeile. In einem Strompfad sind bis zu acht Spulen möglich.

Die verwendeten Spulentypen hängen davon ab, welche Aktion das Programm durchführen soll. Die Zustände remanenter Spulen bleiben bei einem Ausfall der Versorgungsspannung oder beim Umschalten von STOP- in RUN-Modus erhalten. Die Zustände nicht remanenter Spulen gehen bei der Spannungsrückkehr nach einem Ausfall der Versorgungsspannung oder beim Umschalten von STOP- in RUN-Modus auf Null.

**Tabelle 4-2 Spulentypen**

Spulentyp	Symbol	Strom zur Spule	Ergebnis
Schließer	—( )—	EIN AUS	Schaltet Referenz EIN. Schaltet Referenz AUS.
Negiert	—(/)—	EIN AUS	Schaltet Referenz AUS. Schaltet Referenz EIN.
Remanent	—(M)—	EIN AUS	Schaltet Referenz EIN, remanent. Schaltet Referenz AUS, remanent.
Negiert remanent	—(/M)—	EIN AUS	Schaltet Referenz AUS, remanent. Schaltet Referenz EIN, remanent.
Positiver Übergang	—↑—	AUS→EIN	Ist die Referenz AUS, geht sie für einen Zyklus auf EIN
Negativer Übergang	—↓—	EIN←AUS	Ist die Referenz AUS, geht sie für einen Zyklus auf EIN
Setzen	—(S)—	EIN AUS	Schaltet Referenz EIN, bis sie von —(R)— auf AUS gesetzt wird. Spulenzustand unverändert
Rücksetzen	—(R)—	EIN AUS	Schaltet Referenz AUS, bis sie von —(S)— auf EIN gesetzt wird Spulenzustand unverändert
Remanent setzen	—(SM)—	EIN AUS	Schaltet Referenz EIN, bis sie von —(RM)— auf AUS gesetzt wird; remanent. Spulenzustand unverändert
Remanent Rücksetzen	—(RM)—	EIN AUS	Schaltet Referenz AUS, bis sie von —(SM)— auf EIN gesetzt wird; remanent. Spulenzustand unverändert
Fortsetzmerker	———<+>	EIN AUS	Schaltet nächsten Fortsetzkontakt EIN. Schaltet nächsten Fortsetzkontakt AUS.



### Schließerkontakt —| |—

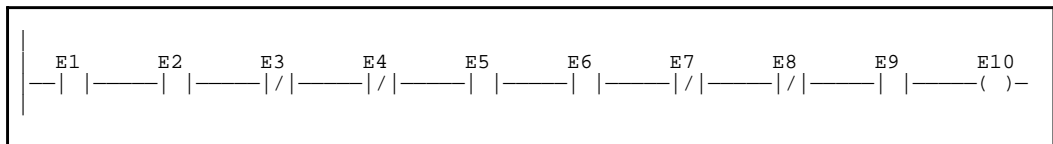
Ein Schließerkontakt arbeitet als Schalter, der den Stromfluß weiterleitet, wenn die zugehörige Referenz EIN (1) ist.

### Öffnerkontakt —|/|—

Ein Öffnerkontakt arbeitet als Schalter, der den Stromfluß weiterleitet, wenn die zugehörige Referenz AUS (0) ist.

**Beispiel:**

Das folgende Beispiel zeigt einen Strompfad mit 10 Elementen, denen die symbolischen Adressen E1 bis E10 zugeordnet sind. Die Spule E10 ist dann durchgeschaltet, wenn die Referenzen E1, E2, E5, E6 und E9 EIN und die Referenzen E3, E4, E7 und E8 AUS sind.

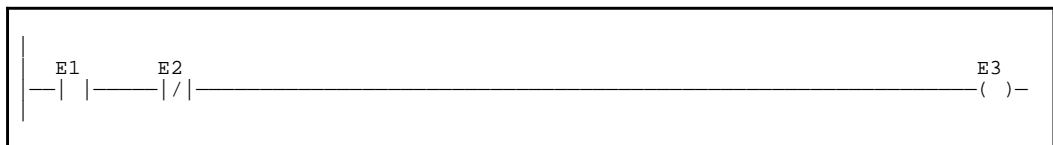


### Spule —( )—

Empfängt eine Spule Stromfluß, schaltet sie den zugehörigen diskreten Ausgang auf EIN. Eine Spule ist nicht remanent und kann daher nicht zusammen mit Systemzustandsreferenzen (%SA, %SB, %SC oder %G) verwendet werden.

**Beispiel:**

Im folgenden Beispiel wird die Spule E3 durchgeschaltet, wenn Referenz E1 EIN und Referenz E2 AUS ist.

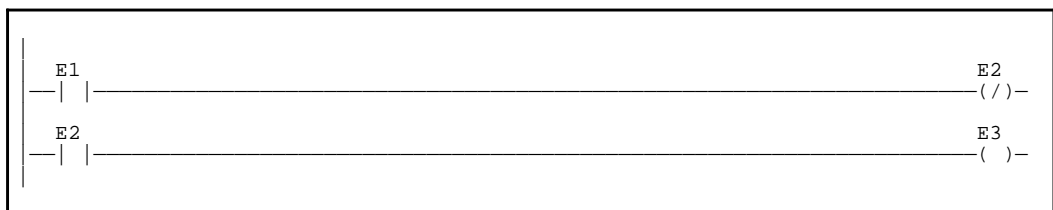


### Negierte Spule —(/)—

Eine negierte Spule schaltet einen diskreten Ausgang durch, wenn sie keinen Stromfluß empfängt. Sie ist nicht remanent und kann daher nicht zusammen mit Systemzustandsreferenzen (%SA, %SB, %SC oder %G) verwendet werden.

**Beispiel:**

Im folgenden Beispiel wird die Spule E3 durchgeschaltet, wenn Referenz E1 AUS ist.



### Remanente Spule —(M)—

Wie bei einer Spule wird bei der remanenten Spule ein diskreter Ausgang durchgeschaltet, wenn die Spule Stromfluß empfängt. Der Zustand der remanenten Spule wird über einen Stromausfall hinweg erhalten. Sie kann daher nicht mit Referenzen aus streng nicht-remanenten Speicherbereichen (%T) verwendet werden.

### Negierte remanente Spule —(/M)—

Die negierte remanente Spule schaltet einen diskreten Ausgang durch, wenn sie keinen Stromfluß empfängt. Der Zustand der remanenten Spule wird über einen Stromausfall hinweg erhalten. Sie kann daher nicht mit Referenzen aus streng nicht-remanenten Speicherbereichen (%T) verwendet werden.

### Spule für positive Übergänge —( )—

Ist die mit einer Spule für positive Übergänge verbundene Referenz AUS, wenn die Spule Stromfluß empfängt, dann wird sie solange durchgeschaltet, bis die Spule erneut bearbeitet wird. (Wird der Strompfad mit dieser Spule bei nachfolgenden Zyklen übersprungen, bleibt sie durchgeschaltet). Diese Spule kann als Wischrelais verwendet werden.

Schreiben Sie nicht von externen Geräten (z.B. PCM, Programmiergerät, ADC, usw.) auf Referenzen, die bei Spulen für positive Übergänge verwendet werden. Dies würde die Wischrelais-Eigenschaft dieser Spule zerstören.

Übergangsspulen können mit Referenzen aus remanenten und nicht-remanenten Speicherbereichen (%Q, %M, %T, %G, %SA, %SB oder %SC) verwendet werden.

### Spulen für negative Übergänge —(-)—

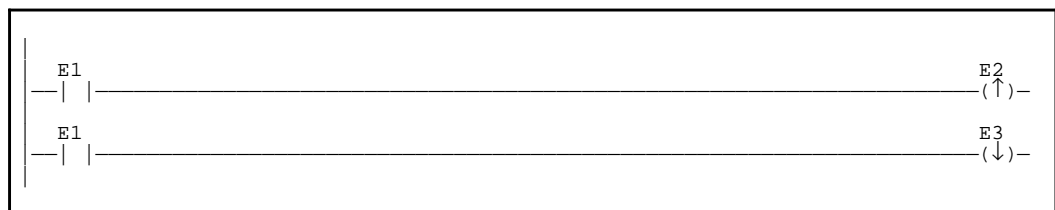
Ist die mit einer Spule für negative Übergänge verbundene Referenz AUS, wenn der Stromfluß zur Spule **unterbrochen** wird, wird sie durchgeschaltet, bis die Spule erneut bearbeitet wird.

Schreiben Sie nicht von externen Geräten (z.B. PCM, Programmiergerät, ADC, usw.) auf Referenzen, die bei Spulen für negative Übergänge verwendet werden. Dies würde die Wischrelais-Eigenschaft dieser Spule zerstören.

Übergangsspulen können mit Referenzen aus remanenten und nicht-remanenten Speicherbereichen (%Q, %M, %T, %G, %SA, %SB oder %SC) verwendet werden.

#### Beispiel:

Schaltet bei dem folgenden Beispiel die Referenz E1 von AUS nach EIN, dann fließt Strom zu den Spulen E2 und E3 und E2 wird für die Dauer eines logischen Zyklus durchgeschaltet. Schaltet E1 von EIN nach AUS, dann wird der Stromfluß zu E2 und E3 unterbrochen und die Spule E3 wird für die Dauer eines Zyklus durchgeschaltet.



## SET-Spule —(S)—

SET und RESET sind nicht-remanente Spulen, die den Zustand einer Referenz auf EIN oder AUS halten ("verriegeln") können. Wird der Stromfluß zu einer SET-Spule geschaltet, bleibt ihre Referenz solange EIN (unabhängig davon, ob die Spule selbst Stromfluß empfängt), bis sie durch Stromfluß zu einer RESET-Spule (-(R)-) rückgesetzt wird.

SET-Spulen schreiben ein undefiniertes Ergebnis in das Übergangsbitt für die angegebene Referenz (siehe "Transitionen und Overrides" in Kapitel 2).

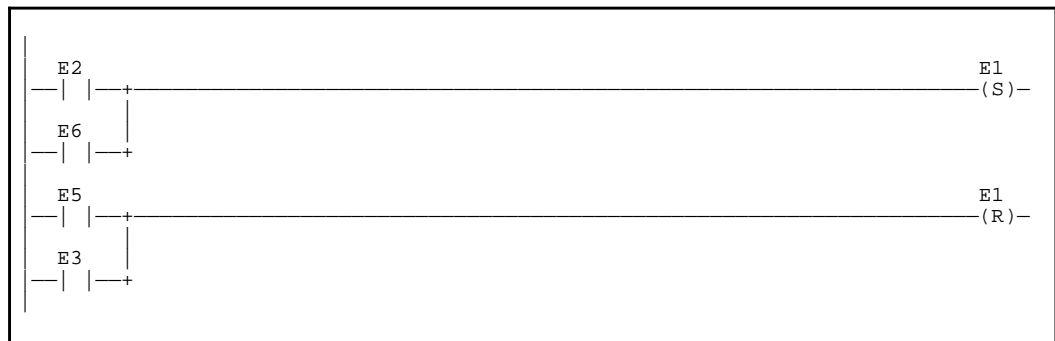
## RESET-Spule —(R)—

Empfängt die RESET-Spule Stromfluß, wird eine diskrete Referenz auf AUS geschaltet. Diese Referenz bleibt solange AUS, bis sie durch eine SET-Spule wieder eingeschaltet wird. Die im Programm zuletzt bearbeitete Spule einer SET-/RESET-Kombination hat jeweils Vorrang.

RESET-Spulen schreiben ein undefiniertes Ergebnis in das Übergangsbitt für die angegebene Referenz (siehe "Transitionen und Overrides" in Kapitel 2).

### Beispiel:

Beim folgenden Beispiel wird die Spule E1 durchgeschaltet, wenn Referenz E2 oder E6 EIN ist. Die Spule E1 wird abgeschaltet, wenn Referenz E5 oder E3 EIN ist.



### Hinweis

Bei Spulenüberprüfung EINFACH können sie eine bestimmte %M- oder %Q-Referenz zwar mit nur einer Spule benutzen, aber Sie können sie gleichzeitig mit einer SET- und einer RESET-Spule einsetzen. Bei Spulenüberprüfung WARNEN MEHRFACH oder MEHRFACH können Sie die einzelnen Referenzen mit mehreren Spulen, SET-Spulen und RESET-Spulen verwenden. Bei Mehrfachbenutzung könnte eine Referenz entweder durch eine SET-Spule oder eine normale Spule EIN geschaltet werden und durch eine RESET-Spule oder eine normale Spule AUS geschaltet werden.

### Remanente SET-Spule —(SM)—

Die Funktion remanenter SET- und RESET-Spulen ist ähnlich wie die der normalen SET- und RESET-Spulen, ihre Zustände bleiben jedoch bei einem Ausfall der Versorgungsspannung oder beim Umschalten der SPS von STOP- in RUN-Modus erhalten. Eine remanente SET-Spule schaltet einen diskreten Ausgang durch, wenn die Spule Stromfluß empfängt. Die Referenz bleibt durchgeschaltet, bis sie durch eine remanente RESET-Spule wieder rückgesetzt wird.

Remanente SET-Spulen schreiben ein undefiniertes Ergebnis in das Übergangsbitt für die angegebene Referenz (siehe "Transitionen und Overrides" in Kapitel 2).

### Remanente RESET-Spule —(RM)—

Empfängt die remanente RESET-Spule Stromfluß, dann wird eine diskrete Referenz auf AUS geschaltet. Diese Referenz bleibt solange AUS, bis sie durch eine remanente SET-Spule wieder eingeschaltet wird. Der Zustand dieser Spule bleibt bei einem Ausfall der Versorgungsspannung oder beim Umschalten von STOP- in RUN-Modus erhalten.

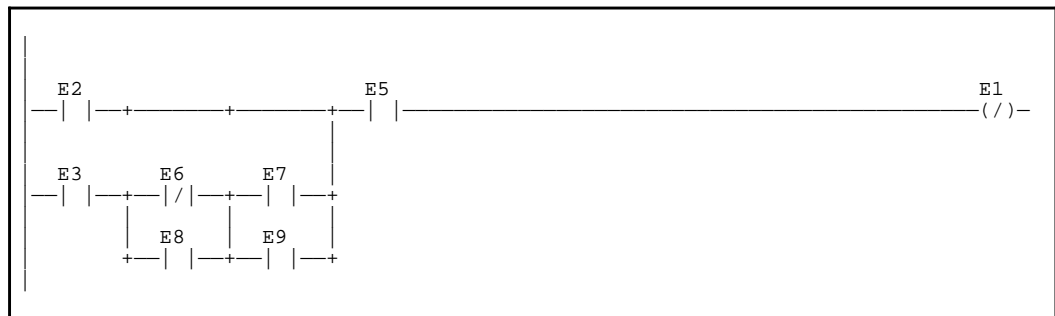
Remanente RESET-Spulen schreiben ein undefiniertes Ergebnis in das Übergangsbitt für die angegebene Referenz (siehe "Transitionen und Overrides" in Kapitel 2).

### Verbindungen

Über Horizontal- und Vertikalverbindungen werden in einer Kontaktplanzeile Funktionen miteinander verbunden. Sie vervollständigen in einer Programmzeile den Weg für den Stromfluß von links nach rechts.

#### Beispiel:

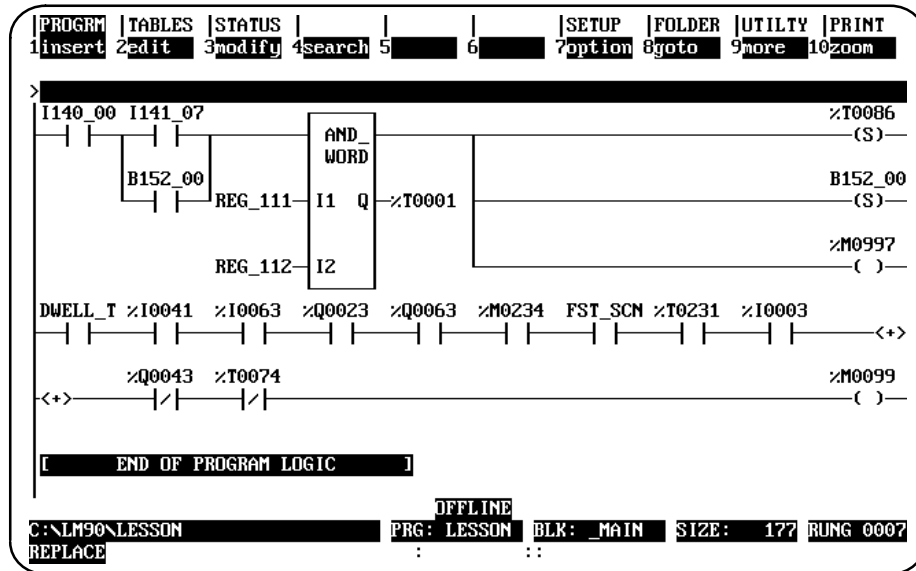
Im folgenden Beispiel werden zwei Horizontalverbindungen dazu verwendet, die Kontakte E2 und E5 miteinander zu verbinden. Die Verknüpfung der Kontakte E3, E6, E7, E8 und E9 mit E2 wird über eine Vertikalverbindung hergestellt.



### Fortsetzspulen (——<+>) und -kontakte (<+>——)

Mit Fortsetzspulen (——<+>) und Fortsetzkontakten (<+>——) wird die Kontaktplanlogik über die Grenze von zehn Spalten hinweg fortgeführt. Der Zustand der zuletzt ausgeführten Fortsetzspule wird als Stromflußzustand für den als nächstes ausgeführten Fortsetzkontakt verwendet. Wird im Stromfluß keine Fortsetzspule vor einem Fortsetzkontakt ausgeführt, dann ist der Zustand dieses Kontakts stromlos.

Pro Strompfad sind nur eine Fortsetzspule und ein Fortsetzkontakt zulässig. Der Fortsetzkontakt muß in Spalte 1 stehen, die Fortsetzspule in Spalte 10. Die nachstehende Abbildung zeigt ein Beispiel für eine Fortsetzspule und einen Fortsetzkontakt.



## Abschnitt 2: Timer und Zähler

In diesem Kapitel wird die Verwendung von Zeitgliedern zur Einschaltverzögerung und Zeiterfassung sowie Auf- und Abwärtszählern beschrieben. Die mit diesen Funktionen verknüpften Daten sind nullspannungssicher.

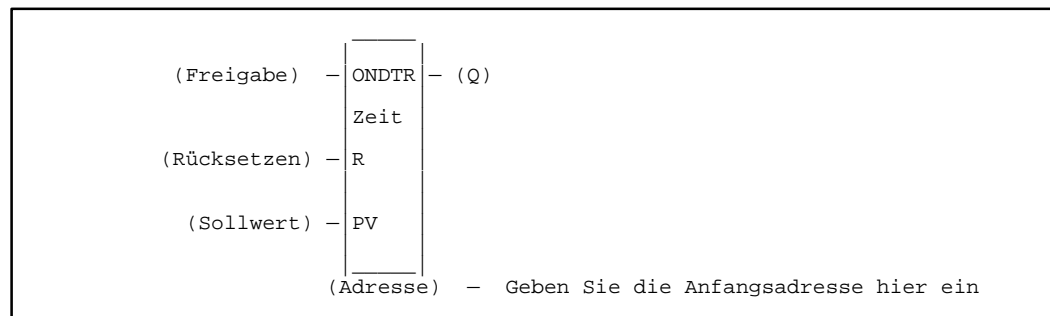
Abkürzung	Funktion	Seite
ONDTR	RemanenterEinschaltverzögerungs-Timer	4-11
TMR	EinfacherEinschaltverzögerungs-Timer	4-14
OFDT	Ausschaltverzögerungs-Timer	4-17
UPCTR	Aufwärtszähler	4-22
DNCTR	Abwärtszähler	4-22

### Für Timer und Zähler benötigte Funktionsblockdaten

Jeder Timer und jeder Zähler belegt im %R-Speicher drei Worte (Register), um die folgenden Daten abzuspeichern:

Istwert (CV)	Wort 1
Sollwert (PV)	Wort 2
Steuerwort	Wort 3

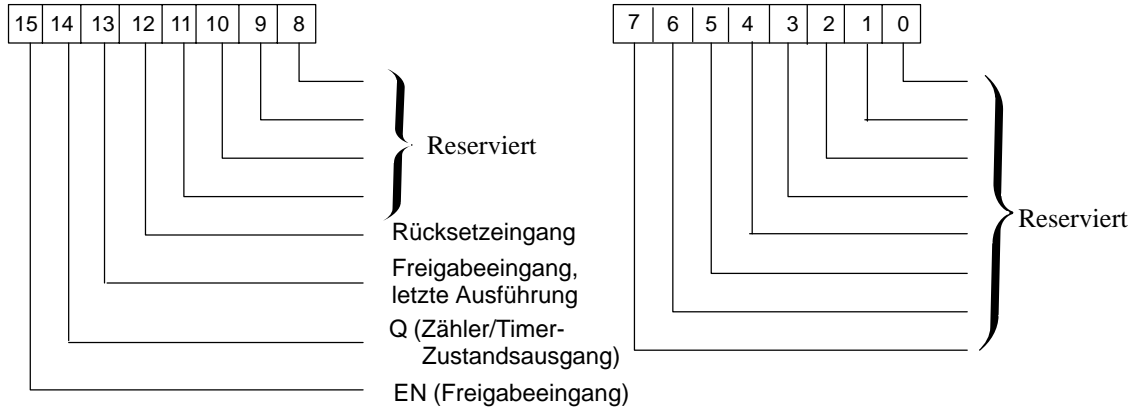
Bei der Eingabe eines Timers oder Zählers müssen Sie immer eine Anfangsadresse für diese drei Worte (Register) unmittelbar unterhalb der Funktionsgraphik eingeben. Zum Beispiel:



### Hinweis

Verwenden Sie für die 3 Worte Timer/Zählerblöcke keine aufeinanderfolgenden Register. Logicmaster prüft nicht, ob sich Registerblöcke überschneiden. Timer und Zähler funktionieren nicht, wenn Sie mit dem aktuellen Wert eines Blocks den Sollwert des vorherigen Blocks überschreiben.

Im Steuerwort werden die Zustände der Booleschen Ein- und Ausgänge des zugehörigen Funktionsblocks in folgendem Format abgelegt:



## ONDTR

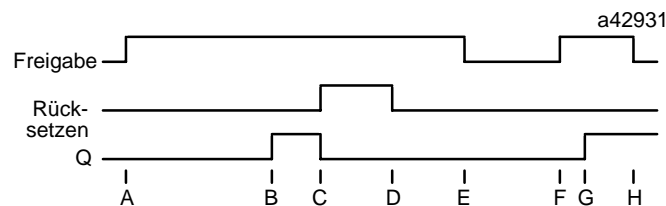
Ein remanenter Timer zur Einschaltverzögerung (ONDTR) inkrementiert seinen Wert, solange er Stromfluß empfängt, und hält den Wert, wenn der Stromfluß stoppt. Die Zeitzählung erfolgt dabei in Schritten von 1/10 (Standardeinstellung), 1/100 oder 1/1000 Sekunden in einem Bereich zwischen 0 und 32.767 Zeiteinheiten. Dieser Timer ist nullspannungssicher, beim Einschalten wird er nicht automatisch initialisiert.

Die ONDTR-Funktion beginnt mit der Zeitzählung (Istwert), wenn sie erstmals Stromfluß empfängt. Der Istwert wird aktualisiert, wenn der Timer im Kontaktplanprogramm angetroffen wird.

### Hinweis

Empfängt der gleiche Timer an mehreren Stellen (mit der gleichen Adresse) während eines CPU-Zyklus Stromfluß, dann sind alle Zeitwerte gleich.

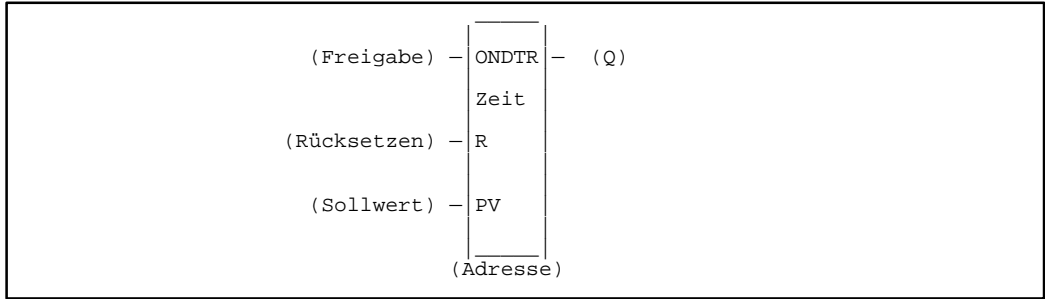
Der Ausgang Q wird durchgeschaltet, wenn der Istwert gleich dem Sollwert ist oder diesen übersteigt. Der Istwert im Timer wird bis zum Maximalwert erhöht, solange der Stromfluß zu dem Zeitglied besteht. Ist der Maximalwert erreicht, wird er gehalten und Ausgang Q bleibt unabhängig vom Zustand des Freigabeeingangs durchgeschaltet.



- A = Das Freigabesignal (ENABLE) wird aktiv, der Timer beginnt mit der Zeitzählung.
- B = Der Istwert erreicht den Sollwert (PV) der Ausgang Q wird durchgeschaltet.
- C = Das Rücksetzsignal (RESET) wird aktiv Q wird abgeschaltet, der Istwert rückgesetzt.
- D = Das Rücksetzsignal (RESET) wird zurückgenommen, der Timer beginnt wieder mit der Zählung.
- E = Das Freigabesignal (ENABLE) wird zurückgenommen, die Zeitzählung gestoppt. Der Istwert bleibt erhalten.
- F = Das Freigabesignal (ENABLE) wird wieder aktiv, der Timer fährt mit der Zeitzählung fort.
- G = Der Istwert erreicht den Sollwert (PV), der Ausgang Q wird durchgeschaltet. Der Timer fährt mit der Zeitzählung fort, bis entweder das Freigabesignal zurückgenommen wird, RESET aktiv wird oder der Istwert den Maximalwert erreicht.
- H = Das Freigabesignal (ENABLE) wird zurückgenommen, die Zeitzählung wird angehalten.

Wird der Stromfluß zum Timer unterbrochen, dann wird die Erhöhung des Istwerts angehalten und der Istwert bleibt erhalten. Die Erhöhung des Istwerts wird bei dem erhaltenen Wert fortgesetzt, wenn der Stromfluß zum Timer wieder einsetzt. Bei Stromfluß zum Rücksetzeingang (R) wird der Istwert auf Null gesetzt und der Ausgang Q wird abgeschaltet. Bei SPS-Modellen 351 und 352 wird das Ausgangssignal zurückgenommen, wenn die Freigabe zu ONDTR auf L-Pegel ist, PV=0 und der Rücksetzeingang R Stromfluß empfängt. Bei den SPS-Modellen 311 bis 341 ist unter den gleichen Voraussetzungen der Ausgang aktiv.





**Parameter:**

Parameter	Beschreibung
Adresse	<p>Die ONDTR-Funktion benutzt drei aufeinanderfolgende Worte (Register) im %R-Speicher, um folgende Werte zu speichern:</p> <ul style="list-style-type: none"> <li>• Istwert (CV) = Wort 1.</li> <li>• Sollwert (PV) = Wort 2.</li> <li>• Steuerwort = Wort 3.</li> </ul> <p>Bei der Eingabe der ONDTR-Funktion müssen Sie die Speicheradresse dieser drei aufeinanderfolgenden Worte (Register) unmittelbar unter der Funktionsgraphik angeben.</p> <p><b>Hinweis:</b> Verwenden Sie diese Adresse nicht bei anderen Anweisungen.</p> <p><b>Achtung:</b> Adreßüberschneidungen führen zu Störungen im Timerbetrieb.</p>
Freigabe	Der aktuelle Timerwert wird erhöht, wenn der Freigabeeingang Stromfluß erhält.
R	Der Istwert wird auf Null gesetzt, wenn Eingang R Stromfluß erhält.
PV	Der Wert von PV wird in den Sollwert des Timers kopiert, wenn der Timer freigegeben oder rückgesetzt wird.
Q	Der Ausgang Q wird durchgeschaltet, wenn der Istwert größer als oder gleich dem Sollwert ist.

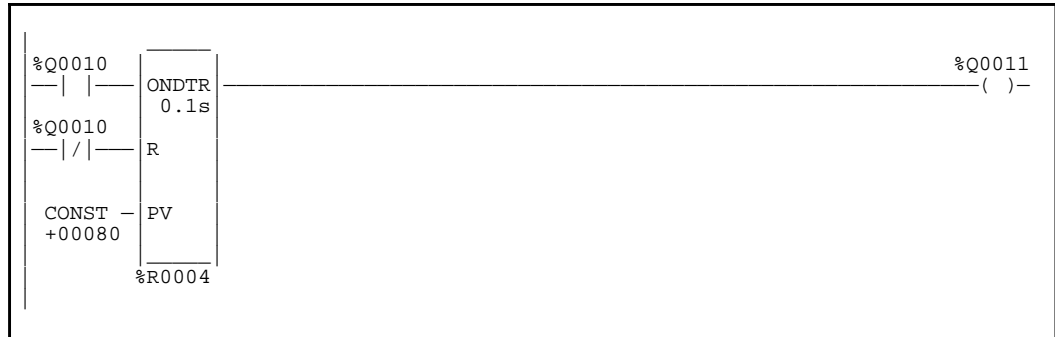
**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Adresse								•				
Freigabe	•											
R	•											
PV		•	•	•	•		•	•	•	•	•	•
Q	•											•

- Zulässiger Datentyp oder Platz, an dem Strom durch die Funktion fließen kann.

**Beispiel:**

In dem folgenden Beispiel wird ein remanenter Timer zur Einschaltverzögerung dazu verwendet, ein Signal (%Q0011) zu erzeugen, das 8,0 Sekunden nach Einschalten von %Q0010 aktiv wird und das abgeschaltet wird, wenn %Q0010 abschaltet.



## TMR

Der einfache Timer zur Einschaltverzögerung (TMR) inkrementiert seinen Wert, solange er Stromfluß empfängt, und setzt den Wert auf Null, wenn der Stromfluß stoppt. Die Zeitzählung erfolgt dabei in Schritten von 1/10 (Standardeinstellung), 1/100 oder 1/1000 Sekunde in einem Bereich zwischen 0 und 32.767 Zeiteinheiten. Der Zustand dieses Timers ist nullspannungssicher, beim Einschalten erfolgt keine automatische Initialisierung.

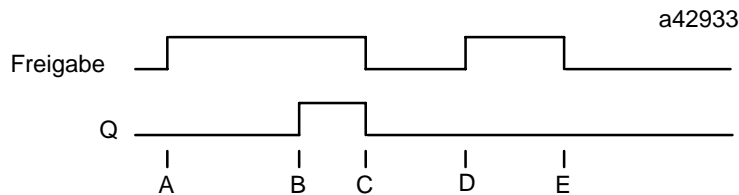
Die TMR-Funktion beginnt mit der Zeitzählung (Istwert), wenn sie Stromfluß empfängt. Der Istwert wird kontinuierlich aktualisiert, wenn der Timer im Kontaktplanprogramm angetroffen wird, und gibt so die gesamte Zeitdauer an, die der Timer seit dem letzten Rücksetzen freigegeben war.

### Hinweis

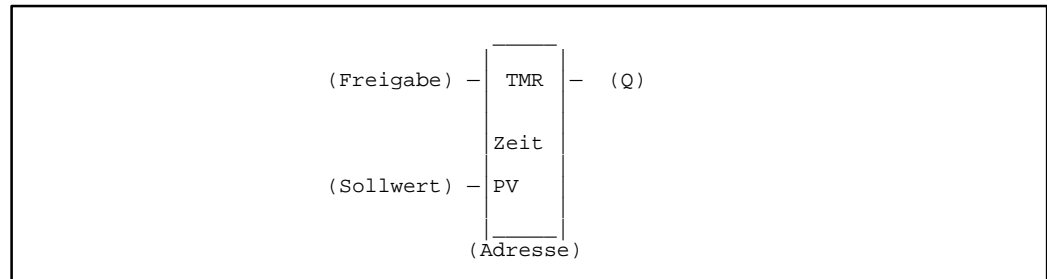
Empfängt der gleiche Timer an mehreren Stellen (mit der gleichen Adresse) während eines CPU-Zyklus Stromfluß, dann sind alle Zeitwerte gleich.

Diese Aktualisierung dauert solange, wie die Freigabelogik EIN bleibt. Die Funktion schaltet den Stromfluß nach rechts durch, wenn der Istwert den Sollwert (PV) erreicht hat. Der Timer erhöht den Istwert solange, bis der Maximalwert erreicht ist. Wechselt der Freigabeparameter von EIN nach AUS, stoppt der Timer und der Istwert wird auf Null rückgesetzt.

Der Zustand der TMR-Funktion ist nullspannungssicher, beim Einschalten wird sie nicht automatisch initialisiert.



- A = Das Freigabesignal (ENABLE) wird aktiv, der Timer beginnt mit der Zeitzählung.
- B = Der Istwert erreicht den Sollwert (PV), der Ausgang Q wird durchgeschaltet. Der Timer fährt mit der Zeitzählung fort.
- C = Das Freigabesignal (ENABLE) wird zurückgenommen, die Zeitzählung wird angehalten. Der Istwert wird gelöscht.
- D = Das Freigabesignal (ENABLE) wird aktiv, der Timer beginnt mit der Zeitzählung.
- E = Das Freigabesignal (ENABLE) wird zurückgenommen ehe der Istwert den Sollwert (PV) erreicht. Q bleibt abgeschaltet, die Zeitzählung wird angehalten und auf Null gesetzt.



**Parameter:**

Parameter	Beschreibung
Adresse	<p>Die TMR-Funktion benutzt drei aufeinanderfolgende Worte (Register) im %R-Speicher, um folgende Werte zu speichern:</p> <ul style="list-style-type: none"> <li>• Istwert (CV) = Wort 1.</li> <li>• Sollwert (PV) = Wort 2.</li> <li>• Steuerwort = Wort 3.</li> </ul> <p>Bei der Eingabe der TMR-Funktion müssen Sie die Speicheradresse dieser drei aufeinanderfolgenden Worte (Register) unmittelbar unter der Funktionsgraphik angeben.</p> <p><b>Hinweis:</b> Verwenden Sie diese Adresse nicht bei anderen Anweisungen.</p> <p><b>Achtung:</b> Adreßüberschneidungen führen zu Störungen im Timerbetrieb.</p>
Freigabe	Der aktuelle Timerwert wird erhöht, wenn der Freigabeeingang Stromfluß erhält. Der Istwert wird auf Null gesetzt und Q abgeschaltet, wenn die TMR-Funktion nicht freigegeben ist.
PV	Der Wert von PV wird in den Sollwert des Timers kopiert, wenn der Timer freigegeben oder rückgesetzt wird.
Q	Der Ausgang Q wird durchgeschaltet, wenn der Istwert größer als oder gleich dem Sollwert ist.

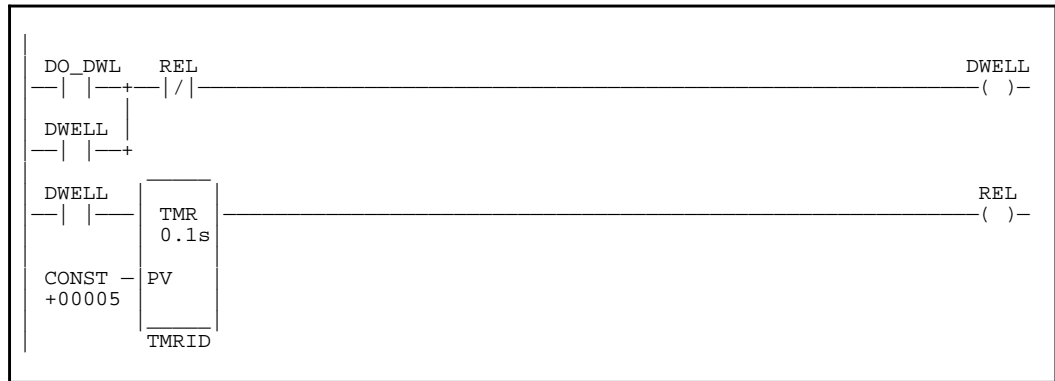
**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Adresse								•				
Freigabe	•											
PV		•	•	•	•		•	•	•	•	•	•
Q	•											•

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

**Beispiel:**

Im folgenden Beispiel wird ein Timer zur Einschaltverzögerung (mit der Adresse) TMRID dazu verwendet, die Einschaltdauer einer Spule DWELL zu steuern. Wird der Schließkontakt (Taster) DO\_DWL geschlossen, wird die Spule DWELL durchgeschaltet. Der Haltekontakt der Spule DWELL hält die Spule DWELL durchgeschaltet (nachdem der Kontakt DO\_DWL losgelassen wurde) und startet auch den Timer TMRID. Wenn TMRID seinen Sollwert von einer halben Sekunde erreicht hat, schaltet die Spule REL durch und unterbricht dadurch den Haltezustand der Spule DWELL. Der Kontakt DWELL unterbricht den Stromfluß zu TMRID, setzt dessen Istwert zurück und schaltet die Spule REL ab. Der Kreis ist dann wieder für die nächste Betätigung des Kontaktes DO\_DWL bereit.



## OFDT

Der Ausschaltverzögerungs-Timer (OFDT) erhöht seinen Wert, wenn der Stromfluß abgeschaltet ist, und wird durch Stromfluß auf Null gesetzt. Die Zeit kann in 1/10 (Standardeinstellung), 1/100 oder 1/1000 Sekunde im Bereich zwischen 0 und +32.767 Zeiteinheiten gemessen werden. Der Zustand dieses Timers ist nullspannungssicher, beim Einschalten erfolgt keine automatische Initialisierung.

Empfängt der OFDT erstmalig Stromfluß, wird der Stromfluß nach rechts durchgeschaltet und der Istwert (PV) auf Null gesetzt. (Der OFDT benutzt Wort 1 (Register) als Speicherplatz für CV – siehe Abschnitt "Parameter:" auf der nächsten Seite.) Der Ausgang bleibt durchgeschaltet, solange die Funktion Stromfluß erhält. Wird der Stromfluß von links unterbrochen, gibt sie weiterhin Stromfluß nach rechts ab und der Timer beginnt mit der Zeitzählung im Istwert.

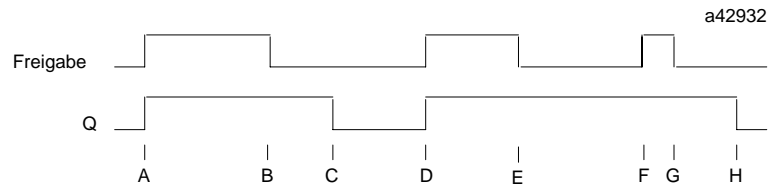
### Hinweis

Empfängt der gleiche Timer an mehreren Stellen (mit der gleichen Adresse) während eines CPU-Zyklus Stromfluß, dann sind alle Zeitwerte gleich.

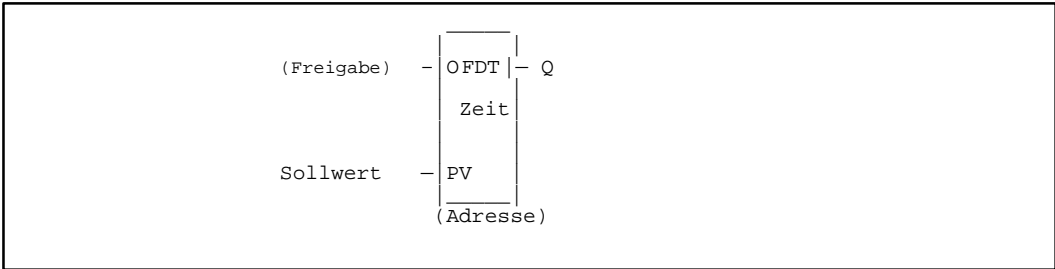
Der OFDT schaltet den Stromfluß nicht durch, wenn der Sollwert Null oder negativ ist.

Jedesmal, wenn die Funktion mit abgeschalteter Freigabelogik aufgerufen wird, wird der Istwert aktualisiert und gibt so die Zeit an, die seit dem Abschalten des Timers verstrichen ist. Erreicht der Istwert (CV) den Sollwert (PV), unterbricht die Funktion den Stromfluß nach rechts. Der Timer hält dann mit der Zeitzählung an – siehe Teil C im nachstehenden Ablaufdiagramm.

Der Sollwert wird auf Null rückgesetzt, wenn die Funktion wieder Stromfluß erhält.



- A = Freigabesignal (ENABLE) und Q werden beide aktiv; der Timer wird rückgesetzt (CV = 0).
- B = Das Freigabesignal (ENABLE) wird zurückgenommen, der Timer beginnt mit der Zählung.
- C = CV erreicht PV; Q wird zurückgenommen; der Timer stoppt die Zeitzählung.
- D = Das Freigabesignal (ENABLE) wird aktiv, der Timer wird rückgesetzt (CV = 0).
- E = Das Freigabesignal (ENABLE) wird zurückgenommen, der Timer beginnt mit der Zählung.
- F = Das Freigabesignal (ENABLE) wird aktiv, der Timer wird rückgesetzt (CV = 0).
- G = Das Freigabesignal (ENABLE) wird zurückgenommen, der Timer beginnt mit der Zählung.
- H = CV erreicht PV; Q wird zurückgenommen; der Timer stoppt die Zeitzählung.



Wird der OFDT in einem Programmblock verwendet, der nicht in jedem Zyklus aufgerufen wird, zählt der Timer die Zeit zwischen den Aufrufen des Programmblocks (sofern er nicht rückgesetzt wird). Das heißt, daß er funktioniert wie ein Timer in einem Programm mit einem viel langsameren Zyklus als der Timer im Hauptprogrammblock. Bei Programmblöcken, die für längere Zeit inaktiv sind, muß der Timer so programmiert werden, daß diese Aufholfunktion berücksichtigt wird. Wird zum Beispiel ein Timer in einem Programmblock rückgesetzt und wird der Programmblock für vier Minuten nicht aufgerufen (ist inaktiv), sind bereits vier Minuten aufgelaufen, wenn der Programmblock aufgerufen wird. Wird der Timer nicht zuerst rückgesetzt, dann wird er bei der Freigabe mit diesen vier Minuten beaufschlagt.

**Parameter:**

Parameter	Beschreibung
Adresse	Die OFDT-Funktion benutzt drei aufeinanderfolgende Worte (Register) im %R-Speicher, um folgende Werte zu speichern: <ul style="list-style-type: none"> <li>• Istwert (CV) = Wort 1.</li> <li>• Sollwert (PV) = Wort 2.</li> <li>• Steuerwort = Wort 3.</li> </ul> Bei der Eingabe der OFDT-Funktion müssen Sie die Speicheradresse dieser drei aufeinanderfolgenden Worte (Register) unmittelbar unter der Funktionsgraphik angeben. <b>Hinweis:</b> Verwenden Sie diese Adresse nicht bei anderen Anweisungen. <b>Achtung:</b> Adreßüberschneidungen führen zu Störungen im Timerbetrieb.
Freigabe	Der Istwert des Timers wird erhöht, wenn der Freigabeeingang aktiviert wird.
Zeit	Zeit (P1) gibt die von den Registern benutzte Zeiteinheit (Millisekunden usw.) an.
PV	Der Wert von PV wird in den Sollwert des Timers kopiert, wenn der Timer freigegeben oder rückgesetzt wird. Bei einer Registerreferenz (%R) für PV wird der Parameter PV als das zweite Wort des Adreßparameters angegeben. Beispiel: Ein Adreßparameter %R00001 würde %R00002 als PV-Parameter benutzen.
Q	Der Ausgang Q wird durchgeschaltet, wenn der Istwert kleiner als der Sollwert ist. Der Zustand von Q ist nullspannungssicher, beim Einschalten erfolgt keine automatische Initialisierung.

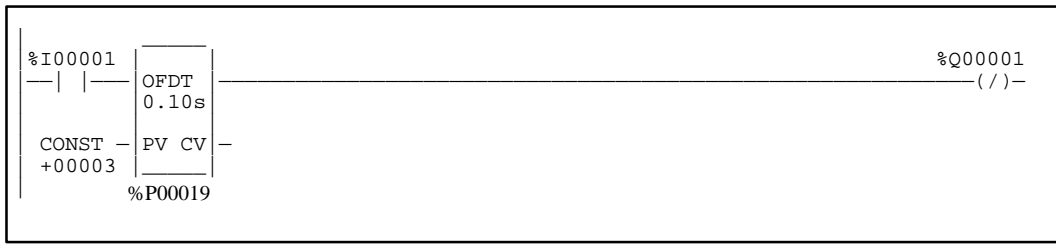
**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Adresse								•				
Freigabe	•											
PV	•	•	•	•	•		•	•	•	•	•	•
Q	•											•

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

**Beispiel:**

Im nachstehenden Beispiel wird mit einem OFDT-Timer ein Ausgang (%Q0001) immer dann abgeschaltet, wenn ein Eingang (%I0001) eingeschaltet wird. Der Ausgang wird 0,3 Sekunden nach Abschalten des Eingangs wieder eingeschaltet.

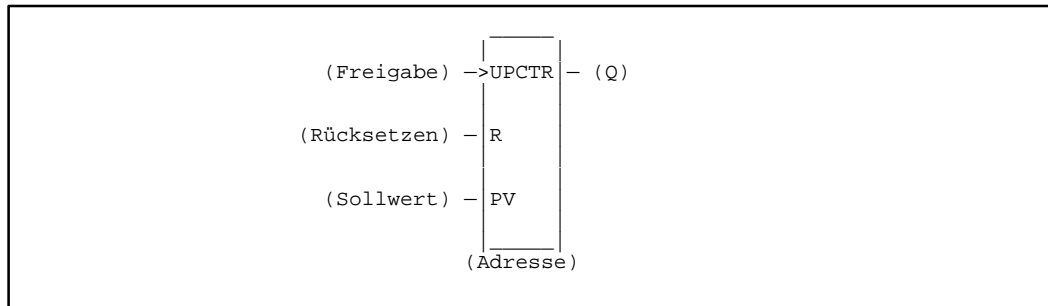




## UPCTR

Mit der Funktion des Aufwärtszählers (UPCTR) wird bis zu einem vorgegebenen Wert vorwärts gezählt. Der Wertebereich beträgt dabei zwischen 0 und +32.767 Zählwerten. Wird der Rücksetzeingang des Aufwärtszählers aktiviert, dann wird der Istwert auf Null gesetzt. Wechselt der Zähler von AUS auf EIN, wird der Istwert um 1 erhöht. Der Istwert kann über den Sollwert (PV) hinaus erhöht werden. Der Ausgang ist durchgeschaltet, solange der Istwert größer als der Sollwert oder gleich ist.

Der Zustand der UPCTR-Funktion ist nullspannungssicher, beim Einschalten findet keine automatische Initialisierung statt.



### Parameter:

Parameter	Beschreibung
Adresse	<p>Die UPCTR-Funktion benutzt drei aufeinanderfolgende Worte (Register) im %R-Speicher, um folgende Werte zu speichern:</p> <ul style="list-style-type: none"> <li>• Istwert (CV) = Wort 1.</li> <li>• Sollwert (PV) = Wort 2.</li> <li>• Steuerwort = Wort 3.</li> </ul> <p>Bei der Eingabe der UPCTR-Funktion müssen Sie die Speicheradresse dieser drei aufeinanderfolgenden Worte (Register) unmittelbar unter der Funktionsgraphik angeben.</p> <p><b>Hinweis:</b> Verwenden Sie diese Adresse nicht bei anderen Aufwärts- oder Abwärtszählern oder bei anderen Anweisungen. Dies führt zu Störungen im Betrieb.</p> <p><b>Achtung:</b> Adreßüberschneidungen führen zu Störungen im Zählerbetrieb.</p>
Freigabe	Bei einem positiven Übergang am Freigabeeingang wird der Istwert um 1 erhöht.
R	Der Istwert wird auf Null rückgesetzt, wenn R Stromfluß empfängt.
PV	Der Wert von PV wird in den Sollwert des Zählers kopiert, wenn der Zähler freigegeben oder rückgesetzt wird.
Q	Der Ausgang Q wird durchgeschaltet, wenn der Istwert kleiner als der Sollwert oder gleich ist.

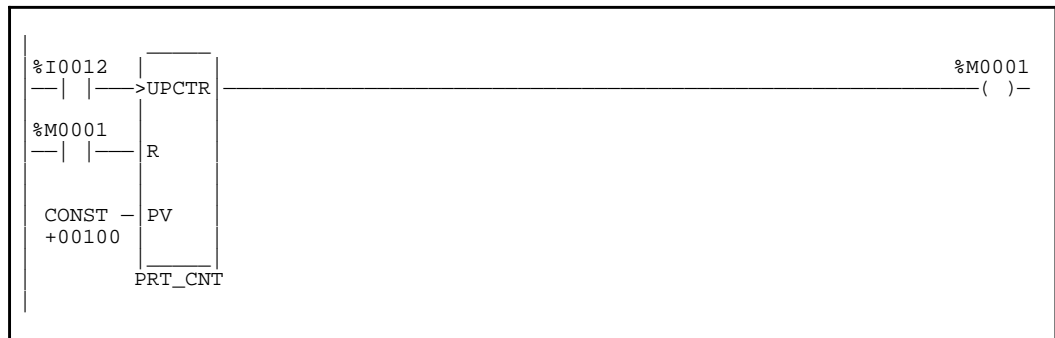
### Zulässige Speichertypen:

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Adresse								•				
Freigabe	•											
R	•											
PV		•	•	•	•		•	•	•	•	•	•
Q	•											•

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

### Beispiel:

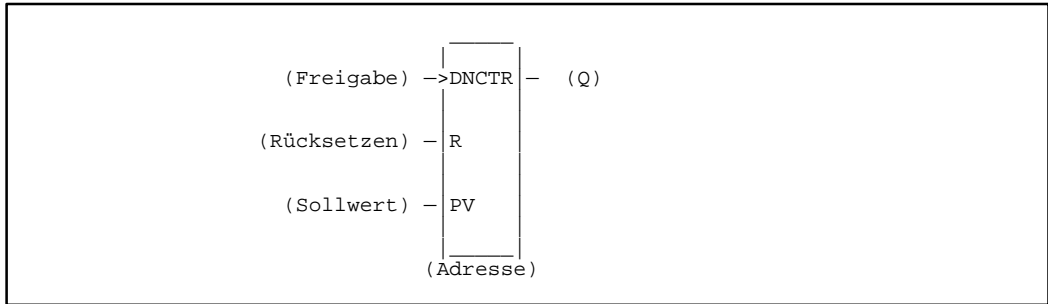
Im nachstehenden Beispiel wird der Aufwärtszähler PRT\_CNT jedesmal um 1 weitergeschaltet, wenn der Eingang %I0012 von AUS auf EIN wechselt. Der interne Merker %M0001 wird durchgeschaltet, wenn 100 Einheiten gezählt wurden. Der Zählwert wird auf Null gesetzt, wenn %M0001 EIN ist.



## DNCTR

Die Funktion des Abwärtszählers (DNCTR) zählt von einem vorgegebenen Wert aus rückwärts. Der kleinste einstellbare Sollwert beträgt dabei 0, der größte +32.767. Der kleinste Istwert beträgt -32.768. Wird der Rücksetzeingang des Abwärtszählers aktiviert, dann wird der Istwert auf den Sollwert (PV) gesetzt. Wechselt der Freigabeeingang von AUS auf EIN, dann wird der Istwert um 1 erniedrigt. Der Ausgang wird durchgeschaltet, wenn der Istwert kleiner oder gleich Null ist.

Der Istwert der DNCTR-Funktion ist nullspannungssicher, beim Einschalten wird die Funktion nicht automatisch initialisiert.



### Parameter:

Parameter	Beschreibung
Adresse	<p>Die DNCTR-Funktion benutzt drei aufeinanderfolgende Worte (Register) im %R-Speicher, um folgende Werte zu speichern:</p> <ul style="list-style-type: none"> <li>• Istwert (CV) = Wort 1.</li> <li>• Sollwert (PV) = Wort 2.</li> <li>• Steuerwort = Wort 3.</li> </ul> <p>Bei der Eingabe der DNCTR-Funktion müssen Sie die Speicheradresse dieser drei aufeinanderfolgenden Worte (Register) unmittelbar unter der Funktionsgraphik angeben.</p> <p><b>Hinweis:</b> Verwenden Sie diese Adresse nicht bei anderen Aufwärts- oder Abwärtszählern oder bei anderen Anweisungen. Dies führt zu Störungen im Betrieb.</p> <p><b>Achtung:</b> Adreßüberschneidungen führen zu Störungen im Zählerbetrieb.</p>
Freigabe	Bei einem positiven Übergang am Freigabeeingang wird der Istwert um 1 erniedrigt.
R	Der Istwert wird auf den Sollwert gesetzt, wenn R Stromfluß empfängt.
PV	Der Wert von PV wird in den Sollwert des Zählers kopiert, wenn der Zähler freigegeben oder rückgesetzt wird.
Q	Der Ausgang Q wird durchgeschaltet, wenn der Istwert kleiner oder gleich Null ist.

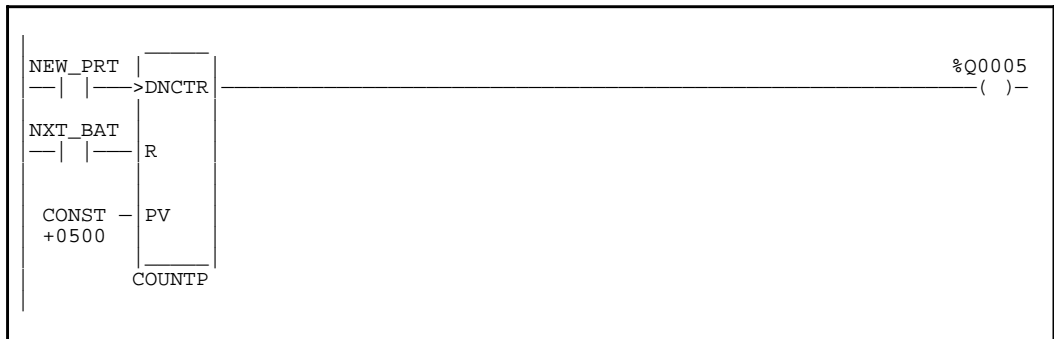
**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Adresse								•				
Freigabe	•											
R	•											
PV		•	•	•	•		•	•	•	•	•	•
Q	•											•

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

**Beispiel:**

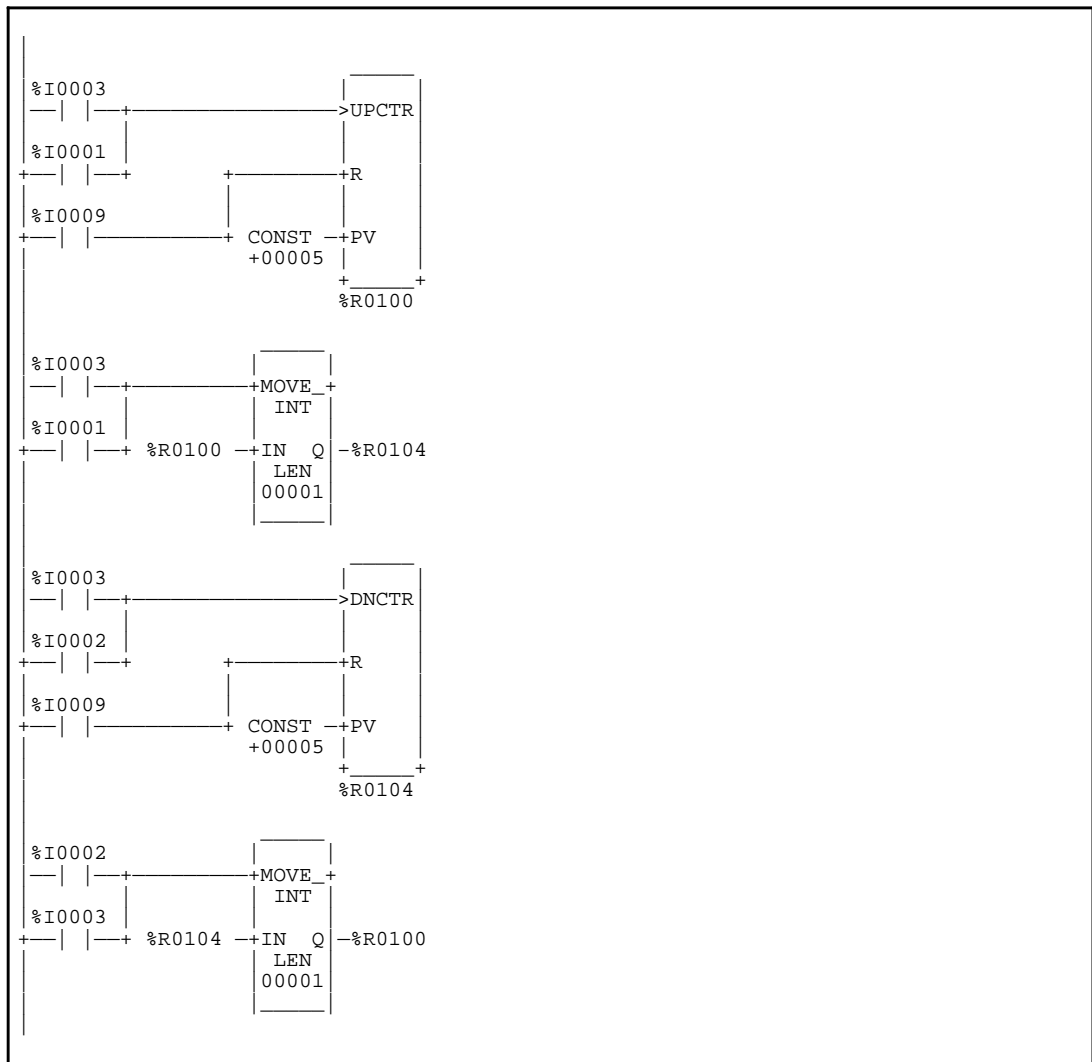
Im folgenden Beispiel zählt der mit COUNTP bezeichnete Abwärtszähler 500 neue Teile, ehe der Ausgang %Q0005 durchgeschaltet wird.



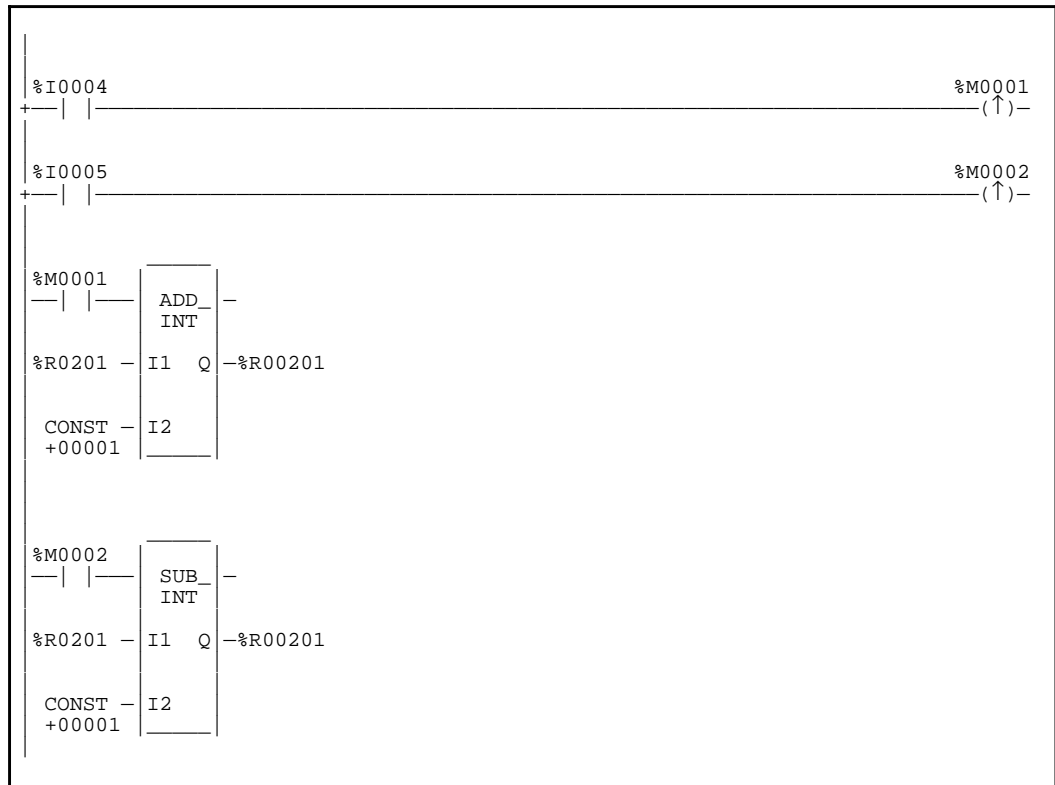
**Beispiel:**

Im folgenden Beispiel wird die SPS dazu eingesetzt, die Anzahl der in einem Zwischenlager enthaltenen Teile zu verfolgen. Mit dem Befehlsvorrat der Series 90-30/20/Micro kann diese Aufgabe auf zwei verschiedene Arten gelöst werden.

Bei der ersten Methode werden ein Auf- und ein Abwärtszähler mit gemeinsam benutztem Register für den Istwert eingesetzt. Wird ein Teil in das Zwischenlager eingebracht, wird der Aufwärtszähler um 1 erhöht. Hierdurch erhöht sich der aktuelle Wert für die Anzahl im Zwischenlager gespeicherter Teile um 1. Verläßt ein Teil das Zwischenlager, wird der Abwärtszähler um 1 erniedrigt und verkleinert dadurch den aktuellen Wert für die Anzahl im Zwischenlager gespeicherter Teile um 1. Um einen Konflikt mit dem gemeinsam genutzten Register zu vermeiden, verwenden die beiden Zähler unterschiedliche Registeradressen. Zählt ein Register, muß sein Istwert an das aktuelle Werteregister des anderen Zählers angepaßt werden.



Bei der nachstehend gezeigten zweiten Methode wird die Lagerverwaltung mit ADD- und SUB-Funktionen durchgeführt.



## Abschnitt 3: Arithmetische Funktionen

In diesem Kapitel werden die arithmetischen Funktionen des Befehlssatzes der Serie 90-30/20/Micro beschrieben.

Abkürzung	Funktion	Beschreibung	Seite
ADD	Addition	Addition zweier Zahlen	4-27
SUB	Subtraktion	Subtraktion einer Zahl von einer anderen	4-27
MUL	Multiplikation	Multiplikation zweier Zahlen	4-27
DIV	Division	Division einer Zahl durch eine andere, im Ergebnis steht ein Quotient	4-27
MOD	Modulo-Division	Division einer Zahl durch eine andere, im Ergebnis steht ein Rest	4-31
SQRT	Quadratwurzel	Quadratwurzel ziehen aus einer ganzen oder reellen Zahl	4-33
SIN, COS, TAN, ASIN, ACOS, ATAN	Trigonometrische Funktionen †	Führt die entsprechende Funktion mit dem an Eingang IN anliegenden reellen Wert durch.	4-35
LOG, LN, EXP, EXPT	Logarithmus-/ Exponential- funktionen †	Führt die entsprechende Funktion mit dem an Eingang IN anliegenden reellen Wert durch.	4-37
RAD, DEG	Bogenmaß- umwandlung †	Führt die entsprechende Funktion mit dem an Eingang IN anliegenden reellen Wert durch.	4-39

† Trigonometrische Funktionen, Logarithmus-/Exponentialfunktionen und Bogenmaß-Umwandlungsfunktionen stehen nur bei den CPU-Modellen 352 zur Verfügung.

### Hinweis

Division und Modulo-Division sind ähnliche Funktionen, die sich in ihrem Ausgangswert unterscheiden. Während die Division einen Quotienten ermittelt, ermittelt die Modulo-Division einen Rest.

## Grundrechenarten (ADD, SUB, MUL, DIV)

Zu den Grundrechenarten gehören Addition, Subtraktion, Multiplikation und Division. Empfängt eine dieser Funktionen Stromfluß, wird die entsprechende arithmetische Funktion mit den Eingangsparametern I1 und I2 durchgeführt. Diese beiden Parameter müssen vom gleichen Datentyp sein. Der Ausgangswert Q ist vom gleichen Datentyp wie I1 und I2.

### Hinweis

Die DIV-Funktion rundet immer ab auf die nächste ganze Zahl.  
(Beispiel: 24 DIV 5 = 4.)

Die Grundrechenfunktionen benutzen die folgenden Datentypen:

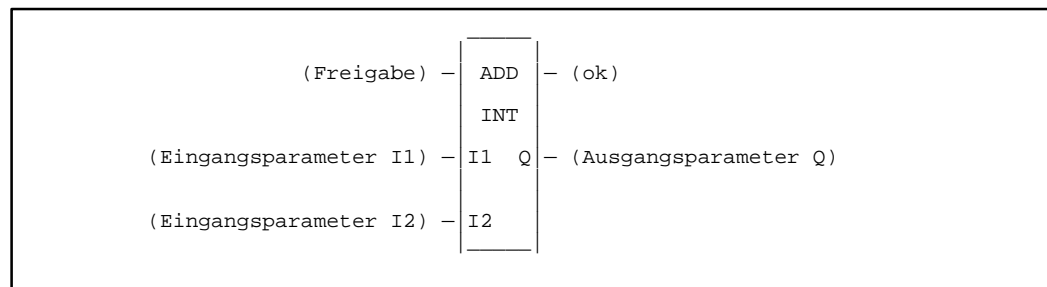
Datentyp	Beschreibung
INT	Ganze Zahl mit Vorzeichen
DINT	Doppeltgenaue ganze Zahl mit Vorzeichen
REAL	Gleitpunktzahl

### Hinweis

Den Datentyp REAL gibt es nur bei der CPU 352.

Die Standardbelegung (ganze Zahl mit Vorzeichen) kann nach Anwahl der Funktion verändert werden. Weitere Informationen über Datentypen finden Sie in Abschnitt 2 von Kapitel 2 dieses Handbuchs.

Ergibt die Operation von INT oder DINT einen Überlauf, wird die Ausgangsreferenz auf den für diesen Datentyp größtmöglichen Wert gesetzt. Bei Zahlen mit Vorzeichen zeigt das Vorzeichen die Richtung des Überlaufs an. Ergibt die Operation keinen Überlauf (und liegen die Eingangswerte im zulässigen Bereich), wird der OK-Ausgang durchgeschaltet, andernfalls ist er AUS. Bei vorzeichenbehafteten oder doppeltgenauen ganzen Zahlen richtet sich das Vorzeichen des Ergebnisses der DIV- und MUL-Funktionen nach den Vorzeichen von I1 und I2.





**Parameter:**

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
I1	I1 enthält eine Konstante oder Referenz für den ersten in der Operation verwendeten Wert (I1 steht auf der linken Seite der mathematischen Gleichung. Zum Beispiel: I1 — I2).
I2	I2 enthält eine Konstante oder Referenz für den zweiten in der Operation verwendeten Wert (I2 steht auf der rechten Seite der mathematischen Gleichung. Zum Beispiel: I1 — I2).
ok	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion ohne Überlauf ausgeführt wurde und keine unzulässige Operation aufgetreten ist.
Q	Der Ausgang Q enthält das Ergebnis der Operation.

**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%A Q	const	keine
enable	•											
I1		o	o	o	o		o	•	•	•	•†	
I2		o	o	o	o		o	•	•	•	•†	
ok	•											•
Q		o	o	o	o		o	•	•	•		

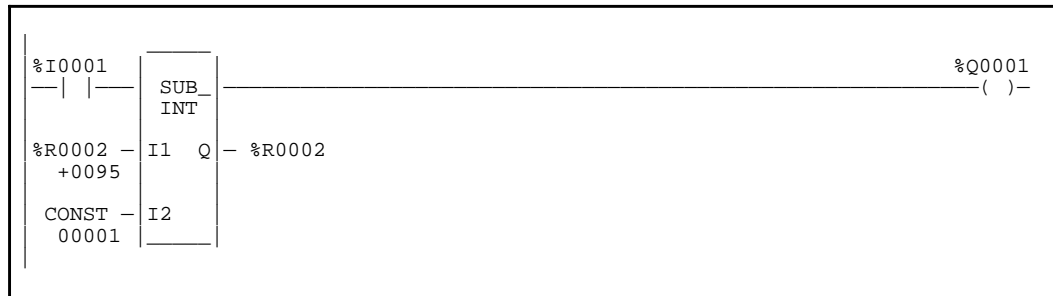
- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- o Referenz nur für INT-Daten zulässig, nicht für DINT oder REAL.
- † Bei doppelgenauen ganzzahligen Operationen mit Vorzeichen sind Konstanten auf Werte zwischen -32.768 und +32.767 beschränkt.

**Hinweis**

INT ist der Standardtyp für 16-Bit- oder Einzelregister-Operanden. Drücken Sie **F10**, um die Typeneinstellung auf DINT, 32 Bit Doppelwort, oder REAL (nur CPU 352) umzuschalten. Während die INT-Werte der SPS ein einzelnes 16-Bit-Register belegen, benötigen die DINT-Werte von %R, %AI und %AQ zwei aufeinanderfolgende Register, wobei die niedrigstwertigen 16 Bits im ersten Wort und die höchstwertigen 16 Bits im zweiten Wort stehen. REAL-Werte (nur CPU 352) belegen ebenfalls ein 32-Bit-Doppelregister, wobei das Vorzeichen im höchstwertigen Bit von Exponent und Mantisse gefolgt wird.

**Beispiel:**

In dem folgenden Beispiel wird jedesmal, wenn der Eingang %I0001 gesetzt wird, der ganzzahlige Inhalt von %R0002 um 1 erniedrigt. Die Spule %Q0001 wird immer dann durchgeschaltet, wenn die Subtraktion ohne Überlauf ausgeführt wurde.



## Arithmetische Funktionen und Datentypen

Funktion	Operation	Anzeige als
ADD INT	$Q (16 \text{ Bits}) = I1 (16 \text{ Bits}) + I2 (16 \text{ Bits})$	5-stellige Dezimalzahl mit Vorzeichen
ADD DINT	$Q (32 \text{ Bits}) = I1 (32 \text{ Bits}) + I2 (32 \text{ Bits})$	8-stellige Dezimalzahl mit Vorzeichen
ADD REAL *	$Q (32 \text{ Bits}) = I1 (32 \text{ Bits}) + I2 (32 \text{ Bits})$	7-stellige Dezimalzahl mit Vorzeichen und Nachkommastellen
SUB INT	$Q (16 \text{ Bits}) = I1 (16 \text{ Bits}) - I2 (16 \text{ Bits})$	5-stellige Dezimalzahl mit Vorzeichen
SUB DINT	$Q (32 \text{ Bits}) = I1 (32 \text{ Bits}) - I2 (32 \text{ Bits})$	8-stellige Dezimalzahl mit Vorzeichen
SUB REAL *	$Q (32 \text{ Bits}) = I1 (32 \text{ Bits}) - I2 (32 \text{ Bits})$	7-stellige Dezimalzahl mit Vorzeichen und Nachkommastellen
MUL INT	$Q (16 \text{ Bits}) = I1 (16 \text{ Bits}) \cdot I2 (16 \text{ Bits})$	5-stellige Dezimalzahl mit Vorzeichen
MUL DINT	$Q (32 \text{ Bits}) = I1 (32 \text{ Bits}) \cdot I2 (32 \text{ Bits})$	8-stellige Dezimalzahl mit Vorzeichen
MUL REAL *	$Q (32 \text{ Bits}) = I1 (32 \text{ Bits}) \cdot I2 (32 \text{ Bits})$	7-stellige Dezimalzahl mit Vorzeichen und Nachkommastellen
DIV INT	$Q (16 \text{ Bits}) = I1 (16 \text{ Bits}) / I2 (16 \text{ Bits})$	5-stellige Dezimalzahl mit Vorzeichen
DIV DINT	$Q (32 \text{ Bits}) = I1 (32 \text{ Bits}) / I2 (32 \text{ Bits})$	8-stellige Dezimalzahl mit Vorzeichen
DIV REAL *	$Q (32 \text{ Bits}) = I1 (32 \text{ Bits}) / I2 (32 \text{ Bits})$	7-stellige Dezimalzahl mit Vorzeichen und Nachkommastellen

## Hinweis

Die Datentypen der Eingangs- und Ausgangsdaten müssen gleich sein. Im Gegensatz zur Series 90-70 sind hier bei den Funktionen MUL und DIV keine gemischten Typen möglich. Beispiel: MUL INT von zwei 16-Bit-Eingangswerten ergibt ein 16-Bit-Produkt, keinen 32-Bit-Wert. Soll bei MUL DINT ein 32-Bit-Ergebnis herauskommen, müssen beide Eingangswerte 32-Bit-Werte sein. Bei DIV INT liefert ein 16-Bit-I1 ein 16-Bit-Ergebnis, während DIV DINT ein 32-Bit-I1 durch ein 32-Bit-I2 teilt und ein 32-Bit-Ergebnis liefert.

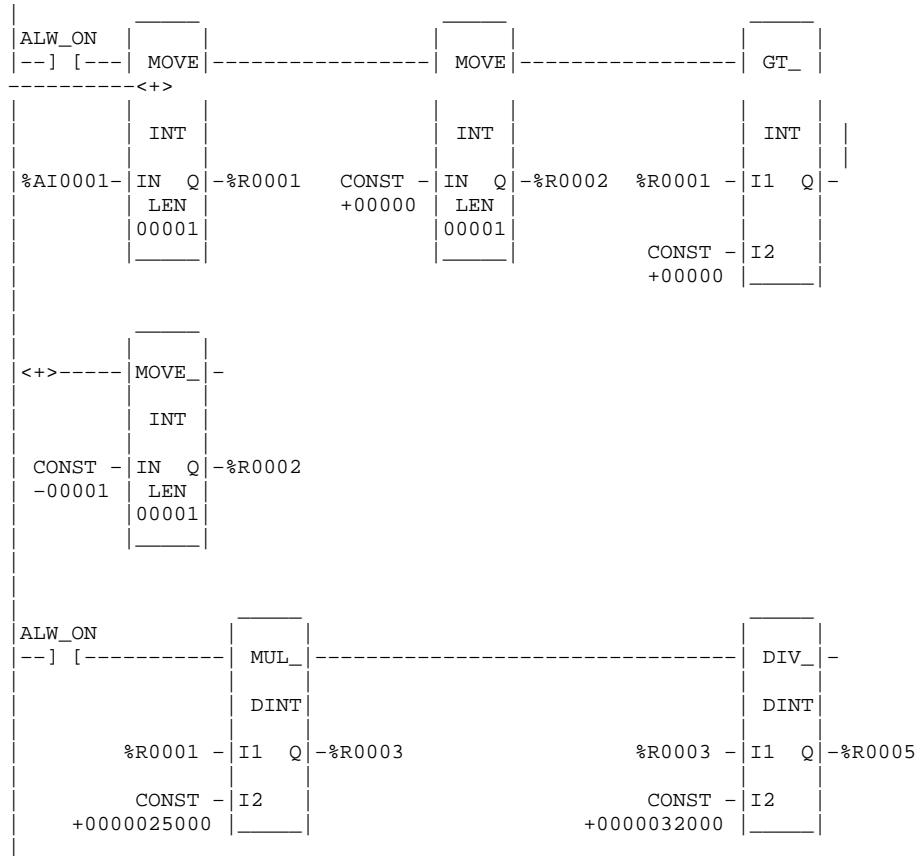
Diese Funktionen schalten den Stromfluß durch, wenn kein arithmetischer Überlauf entsteht. Bei einem Überlauf ist das Ergebnis der größte Wert mit richtigem Vorzeichen, und es findet kein Stromfluß statt.

Vermeiden Sie Überläufe beim Einsatz der MUL- und DIV-Funktionen. Beachten Sie beim Konvertieren von INT- Werten in DINT-Werte, daß die CPU Standard-Zweierkomplemente verwendet, bei denen das Vorzeichen im höchsten Bit des zweiten Wortes steht. Sie müssen das Vorzeichen des unteren 16-Bit-Wortes prüfen und es in das zweite 16-Bit-Wort übertragen. Ist

das höchstwertige Bit in einem 16-Bit-INT-Wort 0 (positiv), müssen Sie eine Null in das zweite Wort schieben. Ist das höchstwertige Bit in einem 16-Bit-Wort -1 (negativ), müssen Sie -1 (Hexa 0FFFFh) in das zweite Wort schieben. Die Konvertierung von DINT nach INT ist einfacher, das das niedrigwertige 16-Bit-Wort (erstes Register) den INT-Teil des 32-Bit-DINT-Wortes darstellt. Die oberen 16 Bits oder das zweite Wort müssen entweder 0 (positiv) oder -1 (negativ) sein, andernfalls ist die DINT-Zahl zu groß, um auf 16 Bits konvertiert zu werden.

Eine häufige Anwendung ist die Skalierung analoger Eingangswerte mit einer MUL-Operation, die von einer DIV- und möglicherweise einer ADD-Operation gefolgt wird. Bei einem Bereich bis 320000 ergibt die Verwendung von MUL INT einen Überlauf. Die Verwendung eines %AI-Wertes für MUL DINT funktioniert ebenfalls nicht, da der 32-Bit-Eingangswert I1 gleichzeitig zwei Analogeingänge kombiniert. Sie müssen den Analogeingang in das untere Wort eines Doppelregisters schreiben, dann das Vorzeichen prüfen und daraufhin das zweite Register auf 0 (wenn positiv) oder -1 (wenn negativ) setzen. Verwenden Sie das Doppelregister mit MUL DINT um ein 32-Bit-Produkt für die folgende DIV-Funktion zu erzielen.

Mit den folgenden Programmbeispiel können Sie einen ±10-V-Eingang %AI auf ±250000 technische Einheiten in %R5 skalieren.



## MOD (INT, DINT)

Mit der Modulfunktion wird ein Wert durch einen anderen Wert vom gleichen Datentyp dividiert. Das Ergebnis ist ein Rest. Das Vorzeichen des Ergebnisses ist immer gleich dem Vorzeichen des Eingangsparameters I1.

Die MOD-Funktion verarbeitet die folgenden Datentypen:

Datentyp	Beschreibung
INT	Ganze Zahl mit Vorzeichen
DINT	Doppeltgenaue ganze Zahl mit Vorzeichen

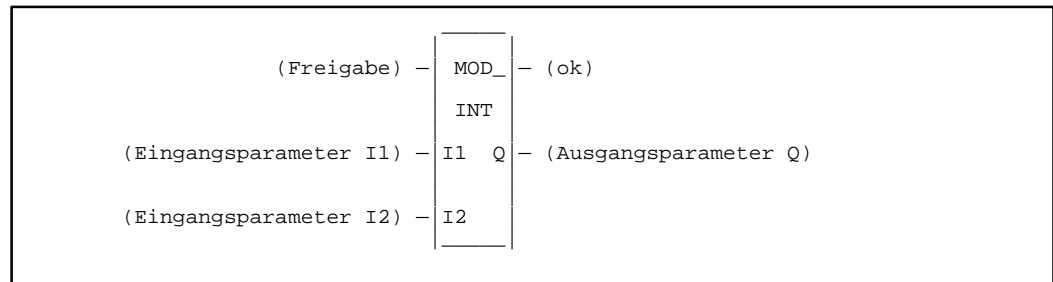
Die Standardbelegung (ganze Zahl mit Vorzeichen) kann nach Anwahl der Funktion verändert werden. Weitere Informationen über Datentypen finden Sie in Abschnitt 2 von Kapitel 2 dieses Handbuchs.

Wenn die Modulfunktion Stromfluß erhält, dividiert sie den Wert I1 durch dem Wert I2. Beide Werte müssen vom gleichen Datentyp sein. Der Ausgangswert Q berechnet sich zu:

$$Q = I1 - ((I1 \text{ DIV } I2) * I2)$$

DIV ergibt dabei eine ganze Zahl. Q ist vom gleichen Datentyp wie die Eingangsparameter I1 und I2.

Der OK-Ausgang ist immer EIN, wenn die Funktion Stromfluß erhält und nicht versucht wurde, durch Null zu dividieren. In einem solchen Fall wird der OK-Ausgang auf AUS gesetzt.



### Parameter:

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
I1	I1 enthält eine Konstante oder Referenz für den Dividenten.
I2	I2 enthält eine Konstante oder Referenz für den Divisor.
ok	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion ohne Überlauf ausgeführt wurde.
Q	Der Ausgang Q enthält den Rest der Division von I1 durch I2.

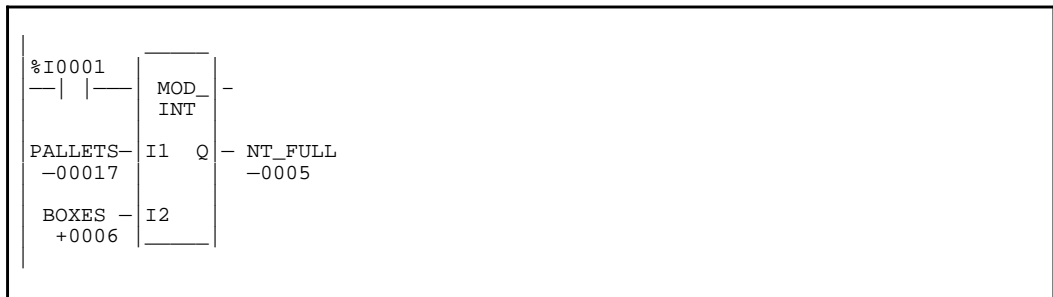
**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%A Q	const	keine
Freigabe	•											
I1		o	o	o	o		o	•	•	•	•†	
I2		o	o	o	o		o	•	•	•	•†	
ok	•											•
Q		o	o	o	o		o	•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- o Referenz nur für INT-Daten zulässig, nicht für DINT.
- † Bei doppelgenauen ganzzahligen Operationen mit Vorzeichen sind Konstanten auf Werte zwischen -32.768 und +32.767 beschränkt.

**Beispiel:**

Bei dem folgenden Beispiel wird jedesmal, wenn der Eingang %I0001 gesetzt wird, der Rest der ganzzahligen Division von PALLETS durch BOXES in NT-FUL eingetragen.



## SQRT (INT, DINT, REAL)

Mit der Quadratwurzelfunktion (SQRT) können Sie die Quadratwurzel aus einem Wert ziehen. Empfängt die Funktion Stromfluß, wird der Wert von Ausgang Q auf den ganzzahligen Teil der Quadratwurzel vom Eingang IN gesetzt. Die Datentypen von Ausgang Q und Eingang IN müssen gleich sein.

Die SQRT-Funktion verwendet folgende Datentypen:

Datentyp	Beschreibung
INT	Ganze Zahl mit Vorzeichen
DINT	Doppeltgenaue ganze Zahl mit Vorzeichen
REAL	Gleitpunkt

### Hinweis

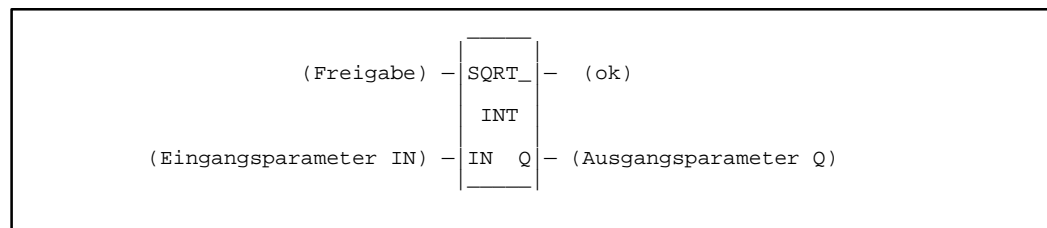
Den Datentyp REAL gibt es nur bei der CPU 352.

Die Standardbelegung (ganze Zahl mit Vorzeichen) kann nach Anwahl der Funktion verändert werden. Weitere Informationen über Datentypen finden Sie in Abschnitt 2 von Kapitel 2 dieses Handbuchs.

Der Ausgang OK wird durchgeschaltet, wenn die Funktion ohne Überlauf durchgeführt wurde und keine der folgenden unzulässigen REAL-Operationen auftrat:

- IN < 0
- IN ist keine Zahl

In den anderen Fällen wird OK abgeschaltet.



### Parameter:

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält eine Konstante oder Referenz für den Wert, dessen Quadratwurzel berechnet werden soll. Ist IN kleiner als Null, wird die Funktion nicht durchgeschaltet.
ok	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion ohne Überlauf ausgeführt wurde und keine unzulässige Operation aufgetreten ist.
Q	Ausgang Q enthält die Quadratwurzel von IN.

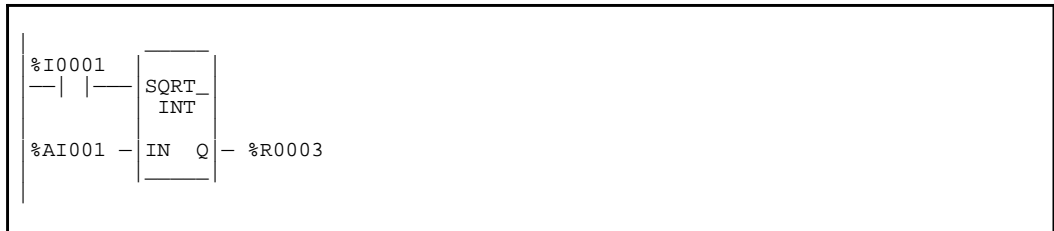
**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN		o	o	o	o		o	•	•	•	•†	
ok	•											•
Q		o	o	o	o		o	•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- o Referenz nur für INT-Daten zulässig, nicht für DINT oder REAL.
- † Bei doppelgenauen ganzzahligen Operationen mit Vorzeichen sind Konstanten auf Werte zwischen -32.768 und +32.767 beschränkt.

**Beispiel:**

Im nachstehenden Beispiel wird die Quadratwurzel aus der in in %AI001 stehenden ganzen Zahl in das in %R0003 stehende Ergebnis eingetragen, wenn %I0001 EIN ist.



## Trigonometrische Funktionen (SIN, COS, TAN, ASIN, ACOS, ATAN)

Die Funktionen SIN, COS und TAN ermitteln den Sinus, Kosinus und Tangens des an ihrem Eingang anliegenden Wertes. Empfängt eine der Funktionen Stromfluß, berechnet sie den entsprechenden trigonometrischen Wert (Sinus, Kosinus oder Tangens) von IN (in Radiant) und speichert das Ergebnis im Ausgang Q. IN und Q sind beide Gleitpunktwerte.

Die Funktionen ASIN, ACOS und ATAN ermitteln die Umkehrfunktionen des Sinus, Kosinus und Tangens des an ihrem Eingang anliegenden Wertes. Empfängt eine der Funktionen Stromfluß, berechnet sie den entsprechenden trigonometrischen Wert (Arkussinus, Arkuskosinus oder Arkustangens) von IN (in Radiant) und speichert das Ergebnis im Ausgang Q. IN und Q sind beide Gleitpunktwerte.

Bei den Funktionen SIN, COS und TAN ist ein weiterer Eingangsbereich möglich:  
 $-2^{63} < IN < +2^{63}$ , ( $2^{63} = 9,22 \times 10^{18}$ )

Der Eingangsbereich der Funktionen ASIN und ACOS ist schmal:  $-1 \leq IN \leq 1$ . Hat der Parameter IN einen gültigen Wert, liegt das Ergebnis der Funktion ASIN\_REAL im Bereich:

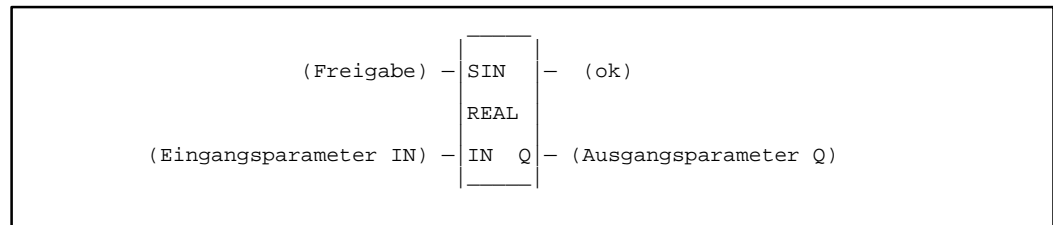
$$\text{ASIN (IN)} = -p/2 \leq Q \leq +p/2$$

Das Ergebnis Q der Funktion ACOS\_REAL liegt im Bereich:

$$\text{ACOS (IN)} = 0 \leq Q \leq p$$

Bei der Funktion ATAN ist der größte Eingangsbereich möglich:  $-1 \leq IN \leq +1$ . Hat der Parameter IN einen gültigen Wert, liegt das Ergebnis der Funktion ATAN\_REAL im Bereich:

$$\text{ATAN (IN)} = -p/2 \leq Q \leq +p/2$$



### Hinweis

Die trigonometrischen Funktionen sind nur bei der CPU 352 verfügbar.

### Parameter:

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält eine Konstante oder eine Referenz auf den reellen Wert für die Operation.
ok	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion ohne Überlauf ausgeführt wurde, keine unzulässige Operation aufgetreten ist und IN eine Zahl ist.
Q	Ausgang Q enthält den trigonometrischen Wert von IN.



**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN								•	•	•	•	
ok	•											•
Q								•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

**Beispiel:**

Im nachstehenden Beispiel wird der Kosinus des Werte in %R0001 in %R0033 eingetragen.

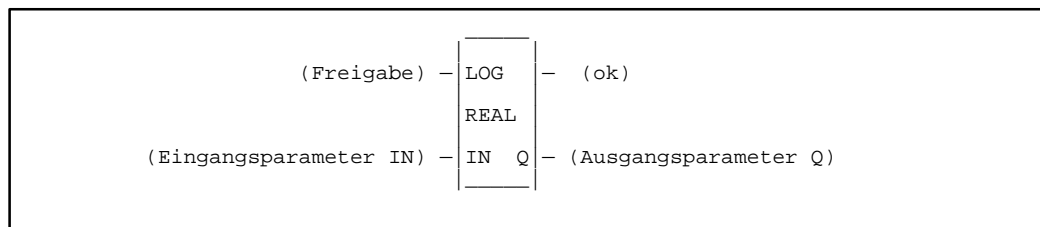


## Logarithmus- / Exponentialfunktionen (LOG, LN, EXP, EXPT)

Die Funktionen LOG, LN und EXP besitzen zwei Eingangsparameter und zwei Ausgangsparameter. Wenn diese Funktionen Stromfluß empfangen, führen sie mit dem am Eingang IN anliegenden reellen Wert die entsprechende Logarithmus- oder Exponentialfunktion aus und legen das Ergebnis im Ausgang Q ab.

- Bei der Funktion LOG wird der dekadische Logarithmus von IN in Q abgelegt.
- Bei der Funktion LN wird der natürliche Logarithmus von IN in Q abgelegt.
- Bei der Funktion EXP wird  $e$  zu der an IN angegebenen Potenz erhoben und das Ergebnis in Q abgelegt.
- Bei der Funktion EXPT wird der Wert am Eingang I1 zu der an I2 angegebenen Potenz erhoben und das Ergebnis in Q abgelegt (die Funktion EXPT besitzt drei Eingangs- und zwei Ausgangsparameter).

Der Ausgang OK wird durchgeschaltet, sofern IN eine Zahl und nicht negativ ist.



### Parameter:

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält den reellen Wert für die Operation.
ok	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion ohne Überlauf ausgeführt wurde, keine unzulässige Operation aufgetreten ist und IN eine Zahl und nicht negativ ist.
Q	Ausgang Q enthält den Logarithmus- bzw. Exponentialwert von IN.

### Hinweis

Die Funktionen LOG, LN, EXP und EXPT sind nur bei der CPU 352 verfügbar.

### Zulässige Speichertypen:

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN *								•	•	•	•	
ok	•											•
Q								•	•	•		

- \* Bei der Funktion EXPT wird Eingang IN durch die Eingangsparameter I1 und I2 ersetzt.
- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

**Beispiel:**

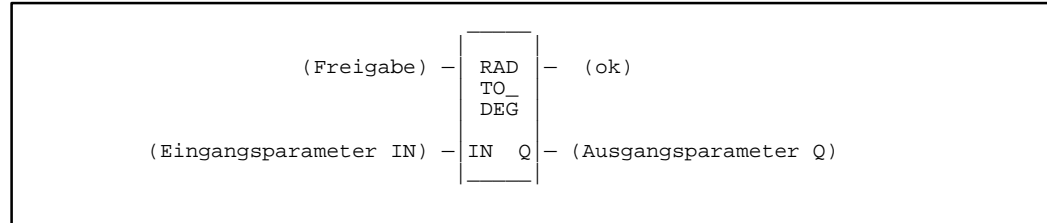
Im nachstehenden Beispiel wird der Wert von %AI001 zur 2,5–ten Potenz erhoben. Das Ergebnis wird in %R001 eingetragen.

```
-----|-----|
|                  |-----|
|                  | REAL  |
|                  |-----|
| %AI001-          | I1  Q  |-%R001
|                  |-----|
| CONST -         | IN  Q  |
| 2.50000E+00     |-----|
```

## Bogenmaß-Umrechnung (RAD, DEG)

Empfängt die Funktion Stromfluß, wird die entsprechende Konvertierung (RAD\_TO\_DEG oder DEG\_TO\_RAD) des reellen Wertes am Eingang IN durchgeführt. Das Ergebnis wird in den Ausgang Q eingetragen.

Der Ausgang OK wird durchgeschaltet, sofern IN eine Zahl und nicht negativ ist.



### Parameter:

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält den reellen Wert für die Operation.
ok	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion ohne Überlauf ausgeführt wurde und IN eine Zahl ist.
Q	Ausgang Q enthält den konvertierten Wert von IN.

### Hinweis

Die Funktionen RAD und DEG sind nur bei der CPU 352 verfügbar.

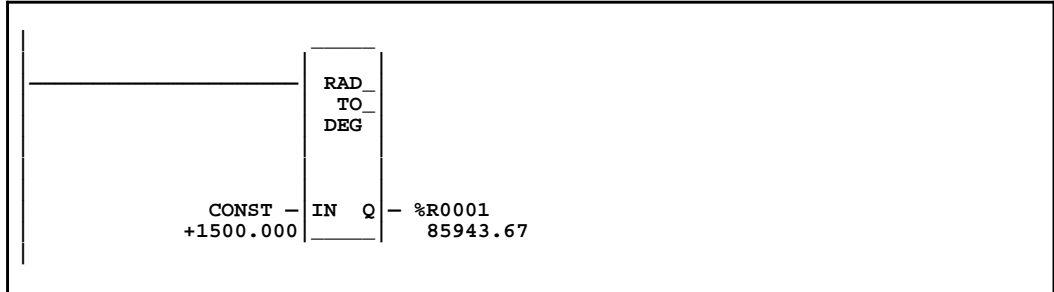
### Zulässige Speichertypen:

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN								•	•	•	•	
ok	•											•
Q								•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

**Beispiel:**

Im nachstehenden Beispiel wird +1500 in Grad umgewandelt und in %R0001 eingetragen.



## Abschnitt 4: Relationale Funktionen

Mit relationalen Funktionen werden zwei Zahlen miteinander verglichen. In diesem Kapitel werden folgende relationale Funktionen beschrieben:

Abkürzung	Funktion	Beschreibung	Seite
EQ	Gleich	Überprüft, ob zwei Zahlen gleich sind.	4-41
NE	Ungleich	Überprüft, ob zwei Zahlen ungleich sind.	4-41
GT	Größer als	Überprüft, ob eine Zahl größer als eine andere ist.	4-41
GE	Größer als oder gleich	Überprüft, ob eine Zahl größer als eine andere oder mit dieser gleich ist	4-41
LT	Kleiner als	Überprüft, ob eine Zahl kleiner als eine andere ist.	4-41
LE	Kleiner als oder gleich	Überprüft, ob eine Zahl kleiner als eine andere oder mit dieser gleich ist	4-41
RANGE	Bereich	Stellt fest, ob eine Zahl in einem bestimmten Bereich liegt (verfügbar für CPUs ab Ausgabestand 4.5)	4-43

Mit relationalen Funktionen wird das Verhältnis zweier Werte zueinander bestimmt. Empfängt die Funktion Stromfluß, vergleicht sie Eingangsparameter I1 mit Eingangsparameter I2. Der Datentyp beider Parameter muß gleich sein.

Relationale Funktionen verwenden folgende Datentypen:

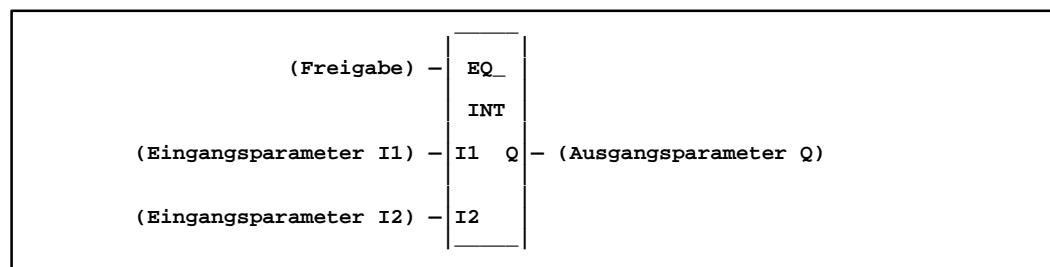
Datentyp	Beschreibung
INT	Ganze Zahl mit Vorzeichen
DINT	Doppeltgenaue ganze Zahl mit Vorzeichen
REAL	Gleitpunktzahl

### Hinweis

Der Datentyp REAL ist nur in der CPU 352 verfügbar. Der Funktionsblock RANGE akzeptiert den Datentyp REAL nicht. Das Bit %S0020 wird gesetzt, wenn eine relationale Funktion mit REAL-Daten erfolgreich ausgeführt wird. Das Bit wird gelöscht, wenn einer der Eingänge keine Zahl ist.

Die Standardbelegung (ganze Zahl mit Vorzeichen) kann nach Anwahl der Funktion in ganze Zahl mit Vorzeichen, doppeltgenaue ganze Zahl mit Vorzeichen, oder reelle Zahl verändert werden. Wollen Sie Daten eines anderen Typs oder zwei unterschiedliche Datentypen miteinander vergleichen, müssen Sie zunächst die entsprechende Konvertierungsfunktion (siehe Abschnitt 8 dieses Kapitels) anwenden, um die Daten in eine der ganzzahligen Datentypen umzuwandeln.

Erfüllen die Eingangsparameter I1 und I2 die angegebene Relation, empfängt Ausgang Q Stromfluß und wird durchgeschaltet; andernfalls wird er auf AUS (0) gesetzt.



**Parameter:**

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
I1	I1 enthält eine Konstante oder Referenz für den ersten zu vergleichenden Wert. (I1 steht auf der linken Seite der relationalen Gleichung; zum Beispiel I1 < I2).
I2	I2 enthält eine Konstante oder Referenz für den zweiten zu vergleichenden Wert. (I2 steht auf der rechten Seite der relationalen Gleichung; zum Beispiel I1 < I2).
Q	Der Ausgang Q wird durchgeschaltet, wenn I1 und I2 der angegebenen Relation entsprechen.

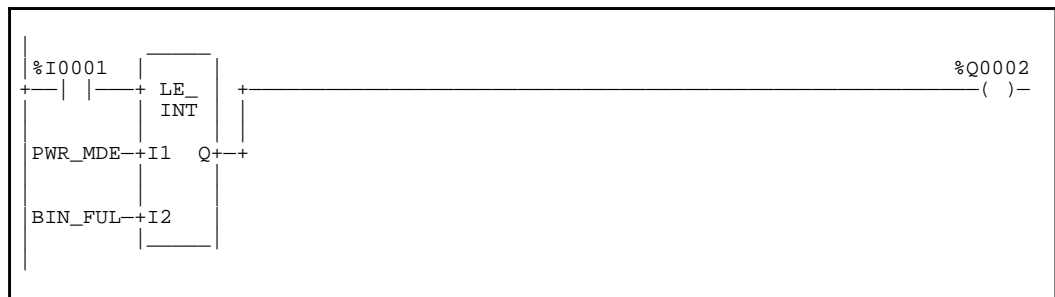
**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
I1		o	o	o	o		o	•	•	•	•‡	
I2		o	o	o	o		o	•	•	•	•‡	
Q	•											•

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- o Referenz nur für INT-Daten zulässig, nicht für DINT oder REAL.
- ‡ Bei doppelgenauen ganzzahligen Operationen mit Vorzeichen sind Konstanten auf ganzzahlige Werte beschränkt.

**Beispiel:**

Bei dem folgenden Beispiel werden jedesmal, wenn der Eingang %I0001 gesetzt wird, die beiden doppelgenauen ganzzahligen Werte mit Vorzeichen PWR\_MDE und BIN\_FUL miteinander verglichen. Sind die beiden Werte gleich, wird die Spule %Q0002 eingeschaltet.



## RANGE (INT, DINT, WORD)

Mit der RANGE-Funktion können Sie feststellen, ob ein Wert in einem von zwei Zahlen festgelegten Bereich liegt.

### Hinweis

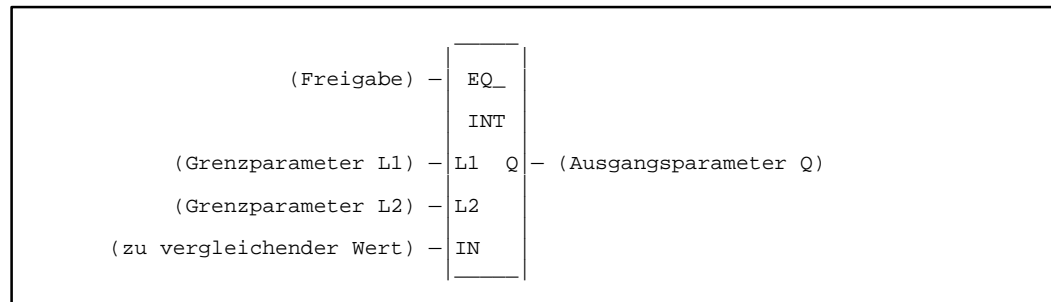
**Diese Funktion ist nur verfügbar in CPUs ab Ausgabestand 4.41.**

Die RANGE-Funktion verwendet folgende Datentypen:

Datentyp	Beschreibung
INT	Ganze Zahl mit Vorzeichen
DINT	Doppeltgenaue ganze Zahl mit Vorzeichen
WORD	Wort

Die Standardbelegung (ganze Zahl mit Vorzeichen) kann nach Anwahl der Funktion verändert werden. Weitere Informationen über Datentypen finden Sie in Abschnitt 2 von Kapitel 2 dieses Handbuchs.

Ist die Funktion freigegeben, vergleicht der RANGE-Funktionsblock den Wert im Eingangsparameter IN mit dem durch die Grenzparameter L1 und L2 festgelegten Bereich. Liegt der Wert in diesem Bereich (einschließlich der Grenzwerte), wird Ausgangsparameter Q durchgeschaltet (1). Im anderen Fall ist Q AUS (0).



### Hinweis

Die Grenzparameter L1 und L2 stellen die Endpunkte des Bereichs dar. Keiner dieser Parameter hat eine Bedeutung als Minimal-/Maximalwert oder oberer/unterer Grenzwert. Ein gewünschter Bereich von 0 bis 100 kann daher definiert werden mit L1 = 0 und L2 = 100 oder aber mit L1 = 100 und L2 = 0.



**Parameter:**

Parameter	Beschreibung
Freigabe	Die Operation wird ausgeführt, wenn die Funktion freigegeben wird.
L1	L1 enthält den Anfangspunkt des Bereichs.
L2	L2 enthält den Endpunkt des Bereichs.
IN	IN enthält den Wert, der mit dem durch L1 und L2 festgelegten Bereich verglichen wird.
Q	Ausgang Q wird aktiviert, wenn der Wert aus IN innerhalb dem durch L1 und L2 festgelegten Bereich liegt (einschließlich der Grenzwerte)

**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%A Q	const	keine
enable	•											
L1		o	o	o	o		o	•	•	•	•‡	
L2		o	o	o	o		o	•	•	•	•‡	
IN		o	o	o	o		o	•	•	•		
Q	•											•

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- o Referenz nur für INT- oder WORD-Daten zulässig, nicht für DINT.
- ‡ Bei doppeltgenauen ganzzahligen Operationen mit Vorzeichen sind Konstanten auf ganzzahlige Werte beschränkt.

**Beispiel 1:**

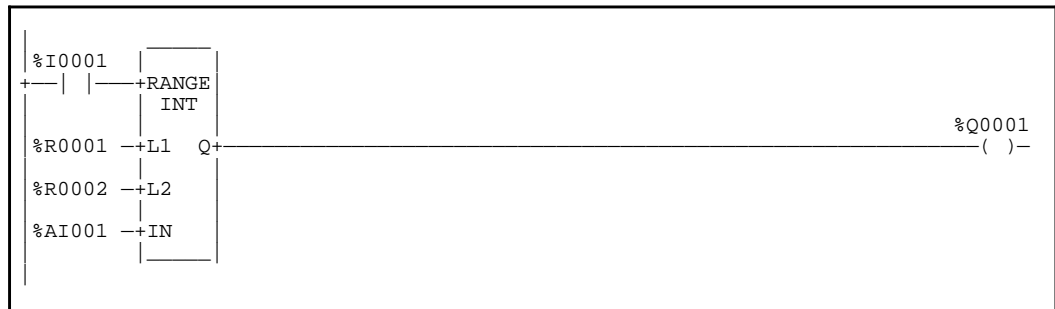
Im nachstehenden Beispiel wird überprüft, ob %AI001 in dem durch die beiden Konstanten 0 und 100 festgelegten Zahlenbereich liegt.



RANGE Wahrheitswerttabelle				
Freigabe- zustand %I0001	Wert von L1 Konstante	Wert von L2 Konstante	Wert von IN %AI001	Zustand von Q %Q0001
EIN	100	0	< 0	AUS
EIN	100	0	0 — 100	EIN
EIN	100	0	> 100	AUS
AUS	100	0	—	AUS

**Beispiel 2:**

In diesem Beispiel wird untersucht, ob %AI001 in dem durch zwei Registerwerte definierten Bereich liegt.



RANGE Wahrheitswerttabelle				
Freigabe- zustand %I0001	Wert von L1 %R0001	Wert von L2 %R0002	Wert von IN %AI001	Zustand von Q %Q0001
EIN	500	0	< 0	AUS
EIN	500	0	0 — 500	EIN
EIN	500	0	> 500	AUS
AUS	500	0	—	AUS

## Abschnitt 5: Bitoperationsfunktionen

Bitoperationsfunktionen führen vergleichende, logische und verschiebende Operationen mit Bitfolgen durch. Die Funktionen AND, OR, XOR und NOT bearbeiten ein einzelnes Wort. Die restlichen Bitoperationsfunktionen können mehrere Worte bearbeiten, wobei die maximale Länge einer Folge 256 Worte beträgt. Sämtliche Bitoperationsfunktionen benötigen Daten vom Typ WORD.

Obwohl die Daten im 16-Bit-Raster angegeben werden müssen, bearbeiten diese Funktionen die Daten als fortlaufendes Bitmuster, wobei Bit 1 des ersten Wortes das niedrigstwertige Bit (LSB) darstellt. Das letzte Bit des letzten Wortes ist das höchstwertige Bit (MSB). Geben Sie zum Beispiel drei Datenworte ab der Referenz %R0100 an, dann werden diese als 48 zusammenhängende Bits behandelt.

%R0100	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	← Bit 1 (LSB)
%R0101	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	
%R0102	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	
	↑																
	(MSB)																

### Hinweis

Überlappende Adreßbereiche bei Ein- und Ausgangsreferenzen in Mehrwortfunktionen können zu unvorhersehbaren Ergebnissen führen

In diesem Abschnitt werden folgende Bitoperationsfunktionen beschrieben:

<b>Abkürzung</b>	<b>Funktion</b>	<b>Beschreibung</b>	<b>Seite</b>
AND	Logisch UND	Sind ein Bit in Bitfolge I1 und das entsprechende Bit in Bitfolge I2 beide 1, wird in der gleichen Bitposition der Ausgangsbitfolge Q eine 1 eingetragen.	4-48
OR	Logisch ODER	Sind ein Bit in Bitfolge I1 und/oder das entsprechende Bit in Bitfolge I2 beide 1, wird in der gleichen Bitposition der Ausgangsbitfolge Q eine 1 eingetragen.	4-48
XOR	Logische Antivalenz	Sind ein Bit in Bitfolge I1 und das entsprechende Bit in Bitfolge I2 voneinander verschieden, wird in der gleichen Bitposition der Ausgangsbitfolge Q eine 1 eingetragen.	4-50
NOT	Logische Invertierung	Die einzelnen Bits in der Ausgangsbitfolge Q werden auf den entgegengesetzten Zustand der entsprechenden Bits in der Bitfolge I1 gesetzt.	4-52
SHL	Nach links verschieben	Alle Bits in einem Wort oder einer Wortfolge werden um die angegebene Anzahl Plätze nach links verschoben.	4-54
SHR	Nach rechts verschieben	Alle Bits in einem Wort oder einer Wortfolge werden um die angegebene Anzahl Plätze nach rechts verschoben.	4-54
ROL	Nach links rotieren	Alle Bits in einer Folge werden um die angegebene Anzahl Plätze nach links rotiert.	4-57
ROR	Nach rechts rotieren	Alle Bits in einer Folge werden um die angegebene Anzahl Plätze nach rechts rotiert.	4-57
BTST	Bit testen	Untersucht ein Bit in einer Bitfolge darauf, ob es momentan 1 oder 0 ist.	4-59
BSET	Bit auf 1 setzen	Setzt ein Bit in einer Bitfolge auf 1.	4-61
BCLR	Bit auf 0 setzen	Setzt ein Bit in einer Bitfolge auf 0.	4-61
BPOS	Bitposition	Ortet ein auf 1 gesetztes Bit in einer Bitfolge.	4-63
MSKCMP	Vergleich mit Maske	Vergleicht den Inhalt zweier getrennter Bitfolgen mit der Möglichkeit, Bits zu maskieren (verfügbar für CPUs ab Ausgabestand 4.5).	4-65

## AND und OR (WORD)

Beginnend mit dem jeweils niedrigstwertigen Bit (LSB) untersuchen die Funktionen AND und OR bei jedem Zyklus, bei dem sie Stromfluß erhalten, die einzelnen Bits in den Bitfolgen I1 und I2.

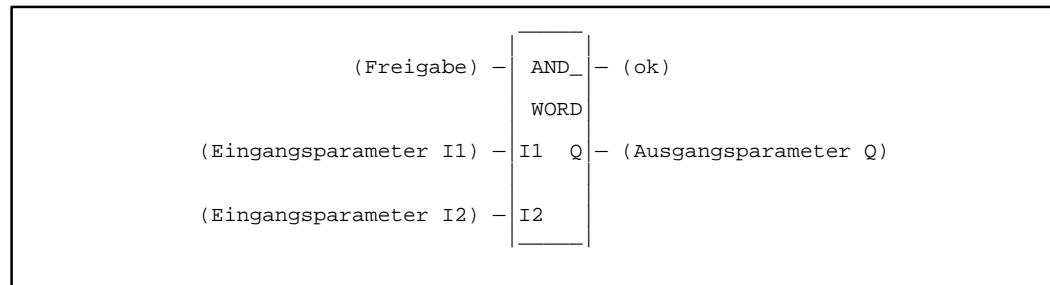
Sind zwei einander entsprechende Bits beide "1", dann trägt die AND-Funktion an der zugehörigen Stelle in der Ausgangsbitfolge Q eine "1" ein. Ist auch nur eines der beiden Bits "0", dann wird an der zugehörigen Stelle in der Ausgangsbitfolge Q eine "0" eingetragen.

Die AND-Funktion ist bei der Erstellung von Masken oder Menüs hilfreich, bei denen nur bestimmte Bits durchgelassen werden (die einer "1" in der Maske entsprechen) und alle anderen Bits auf "0" gesetzt werden. Mit dieser Funktion kann auch ein bestimmter Bereich im Wortspeicher gelöscht werden, indem die Bits mit einer anderen Bitfolge, die nur Nullen enthält, über die AND-Funktion verknüpft werden. Die angegebenen Bitfolgen I1 und I2 können sich überlappen.

Ist eines der beiden Bits "1", dann trägt die OR-Funktion an der zugehörigen Stelle in der Ausgangsbitfolge Q eine "1" ein. Sind beide Bits "0", wird an der zugehörigen Stelle in der Ausgangsbitfolge Q eine "0" eingetragen.

Die OR-Funktion ist hilfreich bei der Kombination von Bitfolgen und bei der Steuerung mehrerer Ausgänge durch eine einfache logische Struktur. Die Funktion entspricht der Parallelschaltung zweier Relaiskontakte für jedes in der Bitfolge enthaltene Bit. Mit ihr können Anzeigelampen aus Eingangszuständen heraus angesteuert oder Statuslampen zum Blinken gebracht werden.

Die Funktion schaltet den Stromfluß jedesmal nach rechts durch, wenn sie Stromfluß erhält.



### Parameter:

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
I1	I1 enthält eine Konstante oder Referenz für das erste Wort der ersten Folge.
I2	I2 enthält eine Konstante oder Referenz für das erste Wort der zweiten Folge.
ok	Der OK-Ausgang wird durchgeschaltet, wenn der Freigabeeingang aktiviert ist.
Q	Der Ausgang Q enthält das Ergebnis der Operation.

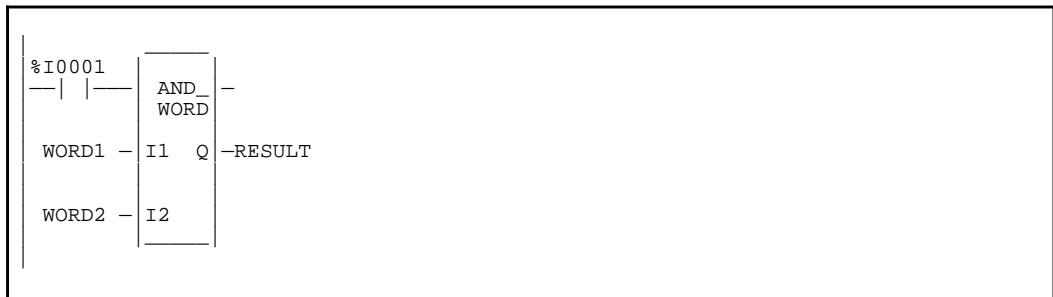
### Zulässige Speichertypen:

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%A Q	const	keine
Freigabe	•											
I1		•	•	•	•	•	•	•	•	•	•	
I2		•	•	•	•	•	•	•	•	•	•	
ok	•											•
Q		•	•	•	•	•†	•	•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- † Nur %SA, %SB, oder %SC; %S kann nicht verwendet werden.

### Beispiel:

Bei dem folgenden Beispiel werden jedesmal, wenn der Eingang %I0001 gesetzt wird, die durch die symbolischen Adressen WORD1 und WORD2 spezifizierten 16 Bit langen Bitfolgen untersucht. Das Ergebnis der logischen AND-Verknüpfung wird in der Ausgangs-Bitfolge RESULT eingetragen.



WORD1	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	0
WORD2	1	1	0	1	1	1	0	0	0	0	0	0	1	1	1	1
RESULT	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0

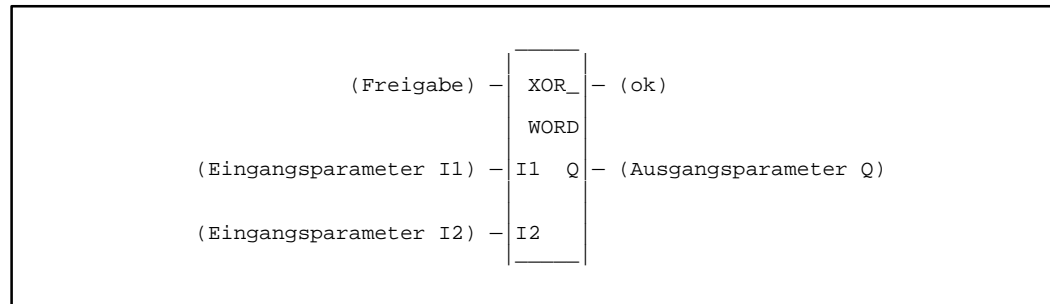
## XOR (WORD)

Die XOR-Funktion (Antivalenz) vergleicht die einzelnen einander entsprechenden Bits in den Bitfolgen I1 und I2 miteinander. Unterscheiden sich diese beiden Bits, dann wird an der zugehörigen Stelle in der Ausgangsbitfolge Q eine "1" eingetragen.

Beginnend mit dem jeweils niedrigstwertigen Bit (LSB) untersucht die XOR-Funktion bei jedem Zyklus, bei dem sie Stromfluß erhält, die einzelnen einander entsprechenden Bits in den Bitfolgen I1 und I2. Ist nur eines der beiden Bits "1", dann wird an der zugehörigen Stelle in der Ausgangsbitfolge Q eine "1" eingetragen. Die XOR-Funktion schaltet den Stromfluß jedesmal dann nach rechts durch, wenn sie Stromfluß erhält.

Beginnen die Bitfolge I2 und die Ausgangsbitfolge Q mit der gleichen Referenz, dann wird bei einer "1" in der Bitfolge I1 bei jedem Zyklus, bei dem die Funktion Stromfluß erhält, das entsprechende Bit in der Bitfolge I2 zwischen "0" und "1" umgeschaltet. Es können längere Zyklen programmiert werden, indem der Stromfluß zur Funktion mit der doppelten gewünschten Blinkfrequenz gepulst wird. Der Stromflußimpuls sollte mindestens einen Zyklus lang sein (Wischrelais oder automatisch rücksetzender Timer).

Mit der XOR-Funktion kann ein schneller Vergleich zweier Bitfolgen durchgeführt werden oder es kann eine Bitgruppe mit einer Frequenz von einem EIN-Zustand pro zwei Zyklen zum Blinken angeregt werden.



### Parameter:

Parameter	Description
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
I1	I1 enthält eine Konstante oder Referenz für das erste Wort, das über die XOR-Funktion verknüpft wird.
I2	I2 enthält eine Konstante oder Referenz für das zweite Wort, das über die XOR-Funktion verknüpft wird.
ok	Der OK-Ausgang wird durchgeschaltet, wenn das Freigabesignal (enable) aktiv ist.
Q	Der Ausgang Q enthält das Ergebnis der logischen XOR-Verknüpfung von I1 und I2.

### Zulässige Speichertypen:

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
I1		•	•	•	•	•	•	•	•	•	•	
I2		•	•	•	•	•	•	•	•	•	•	
ok	•											•
Q		•	•	•	•	•†	•	•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- † Nur %SA, %SB, oder %SC; %S kann nicht verwendet werden.

### Beispiel:

Bei dem folgenden Beispiel wird jedesmal, wenn der Eingang %I0001 gesetzt wird, die durch die symbolische Adresse WORD3 dargestellte Bitfolge gelöscht (auf Null gesetzt).



I1 (WORD3)	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	0
I2 (WORD3)	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	0

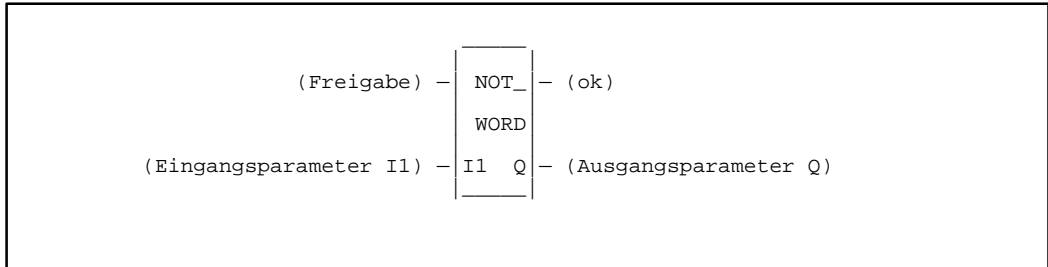
Q (WORD3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-----------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



## NOT (WORD)

Die NOT-Funktion trägt in die einzelnen Stellen der Ausgangs-Bitfolge Q den invertierten Wert der Bits der Eingangsbitfolge I1 ein.

Bei jedem Zyklus, bei dem die Funktion Stromfluß erhält, werden alle Bits verändert, so daß die Ausgangs-Bitfolge Q ein gespiegeltes Abbild der Bitfolge I1 ist. Die NOT-Funktion schaltet den Stromfluß jedesmal dann nach rechts durch, wenn sie Stromfluß erhält.



### Parameter:

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
I1	I1 enthält eine Konstante oder Referenz für das Wort, das invertiert werden soll.
ok	Der OK-Ausgang wird jedesmal durchgeschaltet, wenn das Freigabesignal (enable) aktiv ist.
Q	Der Ausgang Q enthält die Negation von I1.

### Zulässige Speichertypen:

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
I1		•	•	•	•	•	•	•	•	•	•	
ok	•											•
Q		•	•	•	•	•†	•	•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- † Nur %SA, %SB, oder %SC; %S kann nicht verwendet werden.

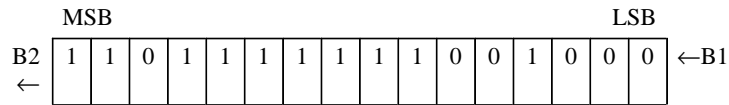
### Beispiel:

Bei dem folgenden Beispiel wird jedesmal, wenn der Eingang %I0001 gesetzt wird, die durch die symbolische Adresse TAC dargestellte Bitfolge in die invertierte Bitfolge CAT umgewandelt.



## SHL und SHR (WORD)

Die SHL-Funktion verschiebt alle Bits eines Wortes oder einer Gruppe um eine vorgegebene Anzahl Stellen nach links. Wird die Funktion ausgeführt, wird die angegebene Anzahl Bits aus der Ausgangsbitfolge nach links herausgeschoben. Die am oberen Ende (MSB) herausgeschobene Anzahl Bits wird am unteren Ende (LSB) hineingeschoben.



Die SHR-Funktion verschiebt alle Bits eines Wortes oder einer Gruppe um eine vorgegebene Anzahl Stellen nach rechts. Wird die Funktion ausgeführt, wird die angegebene Anzahl Bits aus der Ausgangsbitfolge nach rechts herausgeschoben. Die am unteren Ende (LSB) herausgeschobene Anzahl Bits wird am oberen Ende (MSB) hineingeschoben.



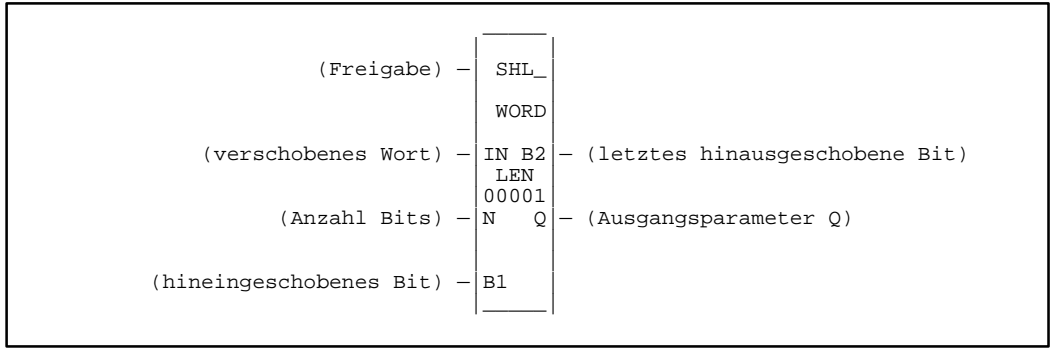
Die Stringlänge kann bei beiden Funktionen zwischen 1 und 256 Worten eingestellt werden. Die Anzahl (N) der Plätze, die für die Verschiebung angegeben wird, muß größer als 0 und kleiner als die Anzahl der in der Bitfolge enthaltenen Bits (LEN \* 16) sein. Ist dies der Fall, wird der Ausgang Q mit Kopien des Eingangsbits (B1) gefüllt und das Eingangsbit wird in den Ausgangs-Stromfluß (B2) kopiert. Ist die Anzahl der zu verschiebenden Bits 0, erfolgt keine Verschiebung. Das Eingangsfeld wird in das Ausgangsfeld kopiert und das Eingangsbit (B1) wird in den Ausgangs-Stromfluß kopiert.

Die in den Anfang der Bitfolge eingeschobenen Bits werden über den Eingangsparameter B1 festgelegt. Ist die für die Verschiebung angegebene Anzahl Stellen größer als 1, haben alle eingeschobenen Bits den gleichen Wert (0 oder 1). Es sind hier möglich:

- Der Boolesche Ausgangswert einer anderen Programmfunktion.
- Nur Einsen. Hierzu müssen Sie die Spezialreferenz mit der symbolischen Adresse ALW\_ON als Eingabe zu Eingang B1 angeben.
- Nur Nullen. Hierzu müssen Sie die Spezialreferenz mit der symbolischen Adresse ALW\_OFF als Eingabe zu Eingang B1 angeben.

Die Funktionen SHL und SHR schalten den Stromfluß nach rechts durch, solange die angegebene Anzahl verschobener Bits nicht Null oder größer als die Gesamtlänge der Folge ist.

Der Funktionsausgang Q ist die verschobene Kopie der Eingangsbitfolge. Wünschen Sie eine Verschiebung der Eingangsbitfolge, muß der Ausgangsparameter Q die gleiche Speicheradresse wie der Eingangsparameter IN verwenden. Die gesamte verschobene Bitfolge wird jedesmal geschrieben, wenn die Funktion Stromfluß empfängt. Ausgangswert B2 ist das letzte hinausgeschobene Bit. Wurden z.B. vier Bits verschoben, dann ist B2 das vierte hinausgeschobene Bit.



**Parameter:**

Parameter	Beschreibung
Freigabe	Die Verschiebung wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält das erste Wort, das verschoben werden soll.
N	N enthält die Anzahl Plätze (Bits), um die das Feld verschoben werden soll.
B1	B1 enthält den Bitwert, der in das Feld eingeschoben werden soll.
B2	B2 enthält den Wert des letzten aus dem Feld herausgeschobenen Bits.
Q	Der Ausgang Q enthält das erste Wort des verschobenen Feldes.
LEN	LEN ist die Anzahl der Worte in dem verschobenen Feld.

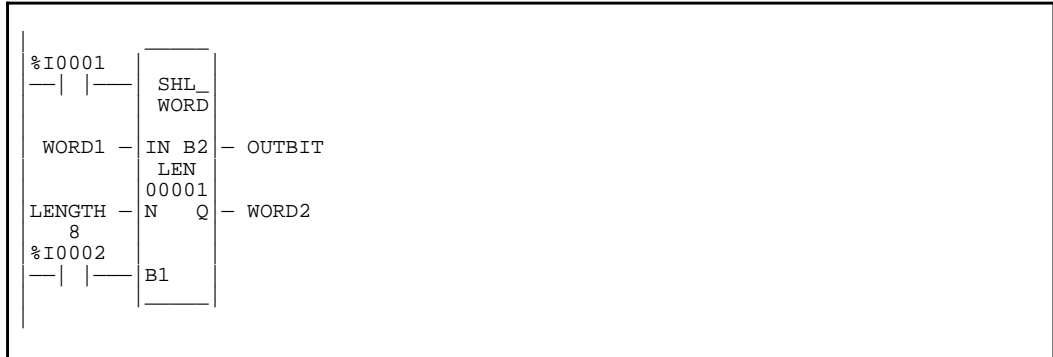
**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN		•	•	•	•	•	•	•	•	•		
N		•	•	•	•		•	•	•	•	•	
B1	•											
B2	•											•
Q		•	•	•	•	•†	•	•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- † Nur %SA, %SB, oder %SC; %S kann nicht verwendet werden.

**Beispiel:**

Im folgenden Beispiel wird jedesmal, wenn Eingang %I0001 gesetzt wird, die durch die symbolische Adresse WORD2 dargestellte Ausgangsbitfolge als Kopie der um die durch die symbolische Adresse LENGTH angegebene Anzahl Stellen nach links verschobene Bitfolge WORD1 erstellt. Die resultierenden leeren Bitstellen am Anfang der Ausgangsbitfolge werden mit dem Wert von %I0002 aufgefüllt.



## ROL und ROR (WORD)

Mit der ROL-Funktion (rotiere nach links) können alle Bits in einer Bitfolge um eine vorgegebene Anzahl Stellen nach links verschoben werden. Wird die Funktion ausgeführt, wird die angegebene Anzahl Bits auf der linken Seite aus der Eingangsbitfolge herausgeschoben und auf der rechten Seite wieder hineingeschoben.

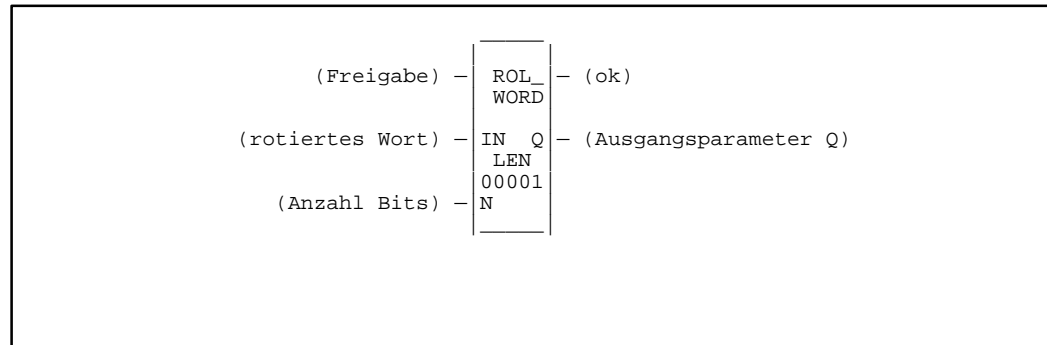
Mit der ROR-Funktion (rotiere nach rechts) können alle Bits in einer Bitfolge um eine vorgegebene Anzahl Stellen nach rechts verschoben werden. Wird die Funktion ausgeführt, wird die angegebene Anzahl Bits auf der rechten Seite aus der Eingangsbitfolge herausgeschoben und auf der linken Seite wieder hineingeschoben.

Die Stringlänge kann bei beiden Funktionen zwischen 1 und 256 Worten eingestellt werden.

Die Anzahl der Plätze, die für die Rotation angegeben wird, muß größer als 0 und kleiner als die Anzahl der in der Bitfolge enthaltenen Bits sein. Ist dies nicht der Fall, erfolgt keine Verschiebung und es findet kein Stromfluß statt.

Die Funktionen ROL und ROR schalten den Stromfluß nach rechts durch, solange die angegebene Anzahl rotierter Bits nicht größer als die Gesamtlänge des Strings oder kleiner als Null ist.

Das Ergebnis wird in die Ausgangsbitfolge Q eingetragen. Wünschen Sie eine Rotation der Eingangsbitfolge, muß der Ausgangsparameter Q die gleiche Speicheradresse wie der Eingangsparameter IN benutzen. Die gesamte rotierte Bitfolge wird bei jedem Zyklus, in dem sie Stromfluß empfängt, geschrieben.



### Parameter:

Parameter	Beschreibung
Freigabe	Die Rotation wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält das erste Wort, das rotiert werden soll.
N	N enthält die Anzahl Plätze, um die das Feld rotiert werden soll.
ok	Der OK-Ausgang wird durchgeschaltet, wenn die Rotation aktiviert ist und die Rotationslänge nicht größer als die Feldlänge ist.
Q	Der Ausgang Q enthält das erste Wort des rotierten Feldes.
LEN	LEN ist die Anzahl der Worte in dem rotierten Feld.

**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN		•	•	•	•	•	•	•	•	•		
N		•	•	•	•		•	•	•	•	•	
ok	•											•
Q		•	•	•	•	•†	•	•	•	•		

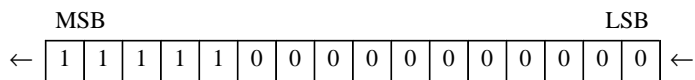
- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- † Nur %SA, %SB, oder %SC; %S kann nicht verwendet werden.

**Beispiel:**

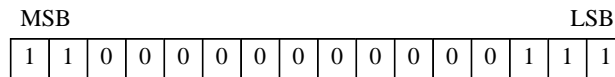
Bei dem folgenden Beispiel wird jedesmal, wenn Eingang %I0001 gesetzt wird, die Eingangsbitfolge %R0001 um 3 Bits rotierend verschoben. Das Ergebnis wird in %R0002 abgelegt. Nachdem die Funktion ausgeführt wurde, bleibt die Eingangsbitfolge %R0001 unverändert. Wird die gleiche Referenz für IN und Q verwendet, findet eine Rotation auf der Stelle statt.



%R0001:



%R0002 (nachdem %I0001 gesetzt wurde):

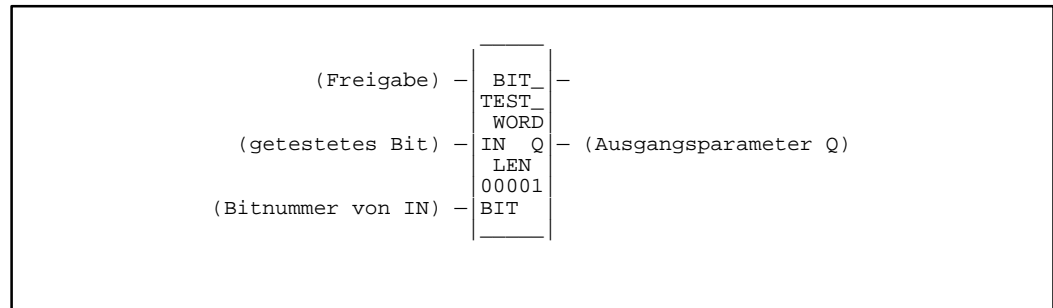


## BTST (WORD)

Mit der BTST-Funktion (Bit testen) kann ein Bit in einer Bitfolge daraufhin untersucht werden, ob es momentan 1 oder 0 ist. Das Ergebnis des Tests wird in Ausgang Q eingetragen.

Bei jedem Zyklus, bei dem sie Stromfluß empfängt, setzt die BTST-Funktion ihren Ausgang Q auf den gleichen Zustand, den das angegebene Bit innehat. Wird die Bitnummer nicht durch eine Konstante, sondern durch ein Register angegeben, dann kann der gleiche Funktionsblock in aufeinanderfolgenden Zyklen unterschiedliche Bits testen. Liegt der Wert von BIT außerhalb des zulässigen Bereichs ( $1 \leq \text{BIT} \leq (16 * \text{LEN})$ ), dann wird Q auf AUS gesetzt.

Es kann eine Bitfolgenlänge zwischen 1 und 256 Worten angegeben werden.



### Parameter:

Parameter	Beschreibung
Freigabe	Der Bittest wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält das erste Datenwort, das bearbeitet werden soll.
BIT	BIT enthält die Nummer des Bits von IN, das getestet werden soll. Zulässiger Bereich ist 1 bis (16*LEN).
OK	Der Ausgang OK wird durchgeschaltet, wenn der Freigabeeingang aktiv ist und BIT größer als die Bitfolgelänge oder Null ist.
Q	Der Ausgang Q wird durchgeschaltet, wenn das getestete Bit "1" war.
LEN	LEN ist die Anzahl Worte in der getesteten Bitfolge



**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN		•	•	•	•	•	•	•	•	•		
BIT		•	•	•	•		•	•	•	•	•	
Q	•											•

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

**Beispiel:**

Bei dem folgenden Beispiel wird jedesmal, wenn der Eingang %I0001 gesetzt wird, das Bit an der in der Referenz PICKBIT angegebenen Adresse getestet. Das Bit ist ein Teil der Bitfolge PRD\_CDE. Ist das Bit 1, werden der Ausgang Q und die Spule %Q0001 durchgeschaltet.

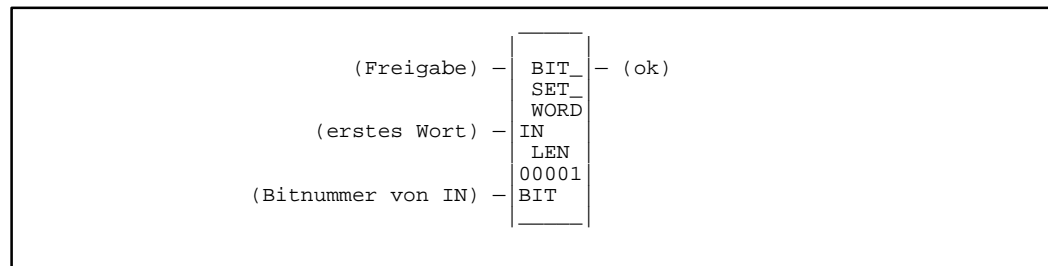


## BSET und BCLR (WORD)

Mit der Funktion BSET (Bit auf 1 setzen) kann ein Bit in einer Bitfolge auf 1 gesetzt werden. Mit der Funktion BCLR (Bit auf 0 setzen) kann ein Bit in einer Bitfolge auf 0 gesetzt werden.

Bei jedem Zyklus, bei dem diese Funktionen Stromfluß empfangen, setzen sie das angegebene Bit auf 1 (BSET) bzw. auf 0 (BCLR). Wird die Bitnummer nicht durch eine Konstante, sondern durch eine Variable (Register) angegeben, kann der gleiche Funktionsblock in aufeinanderfolgenden Zyklen unterschiedliche Bits setzen.

Es kann eine Bitfolgenlänge zwischen 1 und 256 Worten angegeben werden. Die Funktion schaltet den Stromfluß nach rechts durch, wenn der Wert von BIT nicht außerhalb des zulässigen Bereichs ( $1 \leq \text{BIT} \leq (16 * \text{LEN})$ ) liegt. Liegt BIT außerhalb dieses Bereichs, dann wird OK auf AUS gesetzt.



### Parameter:

Parameter	Beschreibung
Freigabe	Die Bitoperation wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält das erste Datenwort, das bearbeitet werden soll.
BIT	BIT enthält die Nummer des Bits von IN, das auf 1 gesetzt werden soll. Zulässiger Bereich ist 1 bis (16*LEN)..
ok	Der Ausgang OK wird durchgeschaltet, wenn der Freigabeeingang durchgeschaltet ist.
LEN	LEN ist die Anzahl Worte in der Bitfolge.

**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN		•	•	•	•	†	•	•	•	•		
BIT		•	•	•	•		•	•	•	•	•	
ok	•											•

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- † Nur %SA, %SB, oder %SC; %S kann nicht verwendet werden.

**Beispiel:**

Bei dem folgenden Beispiel wird jedesmal, wenn der Eingang %I0001 gesetzt wird, Bit 12 der bei Referenz %R0040 beginnenden Folge auf 1 gesetzt.



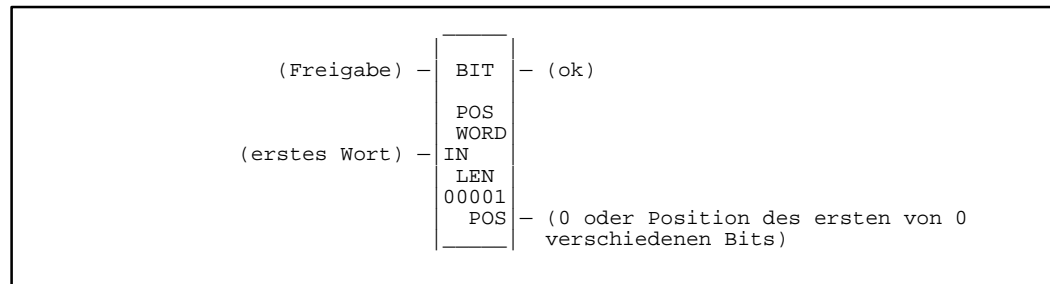
## BPOS (WORD)

Mit der BPOS-Funktion kann ein Bit, das auf 1 gesetzt ist, in einer Bitfolge gefunden werden.

Bei jedem Zyklus, bei dem sie Stromfluß empfängt, untersucht die Funktion die bei IN beginnende Bitfolge. Hält die Funktion mit der Suche inne, hat sie entweder ein Bit gefunden, das auf 1 gesetzt wurde, oder sie hat das Ende der Bitfolge erreicht.

POS wird auf die Position innerhalb der Bitfolge eingestellt, in der das erste von Null verschiedene Bit steht. POS wird auf Null gesetzt, wenn kein von Null verschiedenes Bit gefunden wurde.

Es kann eine Bitfolgenlänge zwischen 1 und 256 Worten angegeben werden. Die Funktion schaltet den Stromfluß immer dann nach rechts durch, wenn der Freigabeeingang EIN ist.



### Parameter:

Parameter	Beschreibung
Freigabe	Die Bitsuche wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält das erste Datenwort, das bearbeitet werden soll.
ok	Der Ausgang OK wird durchgeschaltet, wenn der Freigabeeingang durchgeschaltet ist.
POS	Wird ein auf 1 gesetztes Bit gefunden, steht hier die Position dieses Bits; wird kein auf 1 gesetztes Bit gefunden, steht hier Null.
LEN	LEN ist die Anzahl Worte in der Bitfolge

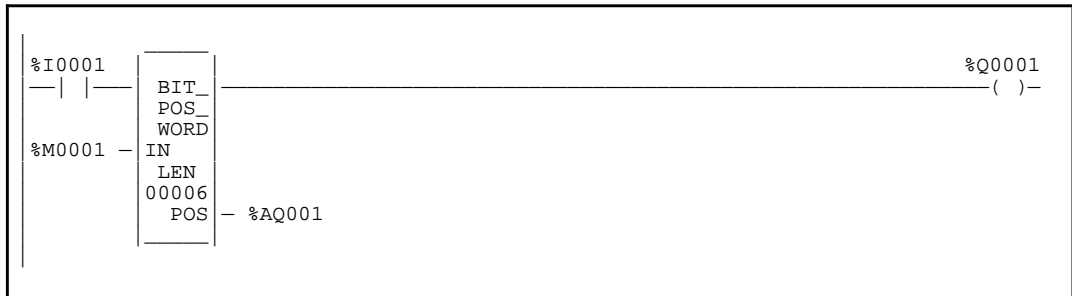
**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN		•	•	•	•	•	•	•	•	•		
POS		•	•	•	•		•	•	•	•		
ok	•											•

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

**Beispiel:**

Wird im nachstehenden Beispiel %I0001 gesetzt, dann wird die bei %M0001 beginnende Bitfolge solange durchsucht, bis ein auf 1 gesetztes Bit gefunden wurde oder 6 Worte durchsucht wurden. Die Spule %Q0001 wird durchgeschaltet. Wird ein auf 1 gesetztes Bit gefunden, dann wird seine Position innerhalb der Bitfolge in %AQ001 eingetragen. Ist Bit %M0001 0 und Bit %M0002 1, wenn %I0001 gesetzt wird, dann wird in %AQ001 der Wert 2 eingetragen.



## MSKCMP (WORD, DWORD)

Mit der MSKCMP-Funktion (Vergleich mit Maske) können Sie den Inhalt zweier getrennter Bitfolgen miteinander vergleichen und dabei bestimmte Bits maskieren (*diese Funktion ist nur verfügbar für CPUs ab Ausgabestand 4.41*). Die Länge der zu vergleichenden Bitfolgen wird durch den Parameter LEN festgelegt (LEN gibt die Anzahl von 16-Bit-Worten für die Funktion MSKCMPW bzw. die Anzahl von 32-Bit-Worten für die Funktion MSKCMPD an).

Wird über die Freigabelogik Stromfluß an den Freigabeeingang EN angelegt, beginnt die Funktion mit dem Vergleich der jeweils zusammengehörenden Bits in den beiden Bitfolgen und führt ihn solange fort, bis entweder ein Unterschied gefunden oder das Ende der Bitfolge erreicht wurde.

Mit dem BIT-Eingang wird die Bitnummer gespeichert, an der der nächste Vergleich beginnen soll (eine 0 gibt das erste Bit in der Bitfolge an). Mit dem Eingang BN wird die Bitnummer gespeichert, bei der der letzte Vergleich stattgefunden hat (eine 1 gibt das erste Bit in der Bitfolge an). Verwenden Sie für BIN und BN die gleiche Referenz, beginnt der Vergleich an der auf einen Unterschied folgenden Bitposition bzw., wenn kein Unterschied gefunden wurde, beim nächsten Aufruf wieder am Anfang.

Soll der nächste Vergleich an einer anderen Stelle der Bitfolge beginnen, können Sie für BIT und BN unterschiedliche Referenzen eingeben. Liegt der Wert von BIT außerhalb der Bitfolge, wird BIT vor dem Beginn des nächsten Vergleichs auf 0 gesetzt.

### Alle Bits in I1 und I2 sind gleich

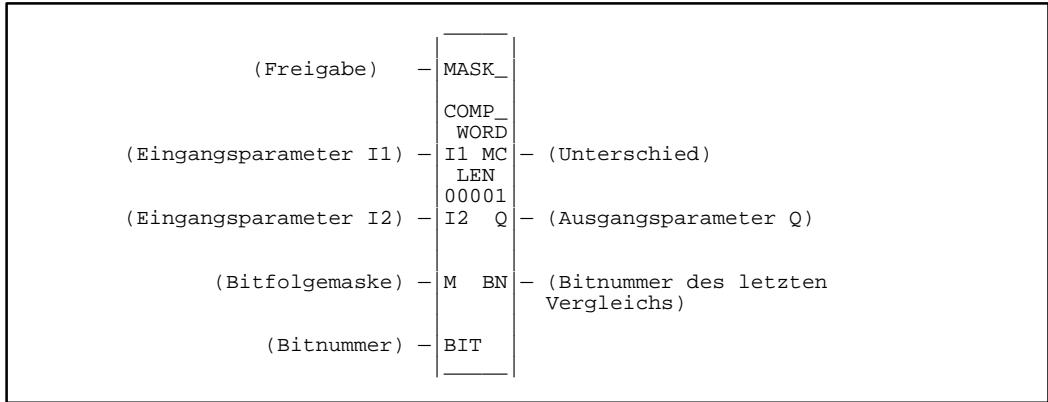
Stimmen alle zusammengehörenden Bits der Bitfolgen I1 und I2 überein, werden der Unterschiedsausgang MC auf 0 und BN auf die höchste Bitnummer der Eingangsfolge gesetzt. Der Vergleich wird dann angehalten. Beim nächsten Aufruf von MSKCMPW wird BN wieder auf 0 rückgesetzt.

### Es wurde ein Unterschied festgestellt

Sind zwei miteinander verglichene Bits unterschiedlich, überprüft die Funktion das Bit mit der entsprechenden Nummer in der Bitfolge M (der Maske). Steht an dieser Stelle in der Maske eine 1, wird der Vergleich fortgesetzt, bis ein weiterer Unterschied gefunden wird oder das Ende der Eingangsbitfolge erreicht ist.

Wird ein Unterschied festgestellt und ist das entsprechende Maskenbit eine 0, führt die Funktion folgende Aktivitäten durch:

1. Das entsprechende Maskenbit in M wird auf 1 gesetzt.
2. Der Unterschiedsausgang MC wird auf 1 gesetzt.
3. Die Ausgangsfolge Q wird so aktualisiert, daß sie dem neuen Inhalt der Maskenfolge M entspricht.
4. Der Bitnummernausgang BN wird auf die Nummer des unterschiedlichen Bits gesetzt.
5. Der Vergleich wird gestoppt.



**Parameter:**

Parameter	Beschreibung
Freigabe	Freigabelogik für Funktionsfreigabe
I1	Referenz für die erste zu vergleichende Bitfolge
I2	Referenz für die zweite zu vergleichende Bitfolge
M	Referenz für die Bitfolgenmaske
BIT	Referenz für die Bitnummer, bei der der nächste Vergleich beginnen soll
MC	Anwenderprogramm zur Bestimmung eines Unterschieds
Q	Ausgangskopie der Maskenbitfolge (M)
BN	Nummer des Bits, bei dem der letzte Vergleich stattfand
LEN	LEN ist die Anzahl Worte in der Bitfolge.

**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
I1		o	o	o	o	o	o	•	•	•		
I2		o	o	o	o	o	o	•	•	•		
M		o	o	o	o	o†	o	•	•	•		
BIT		•	•	•	•	•	•	•	•	•	•	
LEN											•‡	
MC	•											•
Q		o	o	o	o	o†	o	•	•	•		
BN		•	•	•	•	•	•	•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- o Zulässige Referenz nur für WORD-Daten; nicht zulässig für DWORD
- † Nur %SA, %SB, %SC; %S kann nicht verwendet werden
- ‡ Maximale Konstante von 4095 für WORD und 2047 für DWORD

### Beispiel:

Im folgenden Beispiel wird der Funktionsblock MSKCMPW ausgeführt, wenn %I0001 EIN ist. %M0001 bis %M0016 werden mit %M0017 bis %M0032 verglichen. Die Bits %M0033 bis %M0048 enthalten die Maskenwerte. Der Wert in %R0001 legt fest, an welcher Bitposition in den beiden Bitfolgen der Vergleich beginnt. Der Inhalt der beiden Referenzen sieht vor der Ausführung des Funktionsblocks wie folgt aus:

(I1) – %M0001 = 6C6Ch =

0	1	1	0	1	1	0	0	0	1	1	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(I2) – %M0017 = 606Fh =

0	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(M/Q) – %M0033 = 000Fh =

0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(BIT/BN) – %R0001 = 0

(MC) – %Q0001 = AUS

Nach Ausführung des Funktionsblockes sieht der Inhalt dieser Referenzen wie folgt aus:

(I1) – %M0001 =

0	1	1	0	1	1	0	0	0	1	1	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(I2) – %M0017 =

0	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

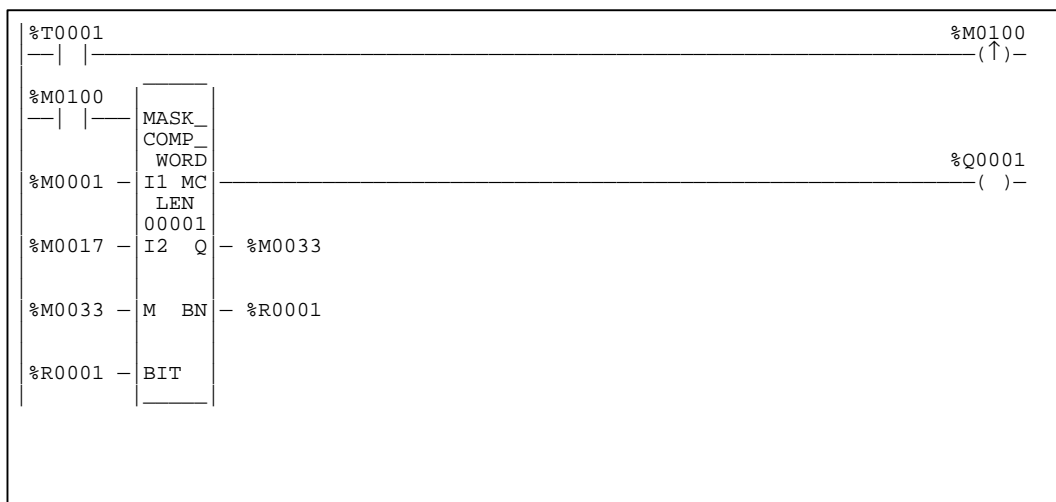
(M/Q) – %M0033

0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(BIT/BN) – %R0001 = 8

(MC) – %Q0001 = EIN

### Kontaktplandarstellung



Beachten Sie, daß wir in dem vorstehenden Beispiel den Kontakt FST\_SCN dazu benutzt haben, eine einzige Ausführung zu erzwingen. Im anderen Fall würde der Vergleich mit Maske wiederholt werden und nicht unbedingt das gleiche Ergebnis erbringen.



## Abschnitt 6: Datenverschiebefunktionen

Die Datenverschiebungsfunktionen vermitteln die grundlegenden Möglichkeiten zur Bewegung von Daten. In diesem Kapitel werden die folgenden Datenverschiebungsfunktionen beschrieben:

Abkürzung	Funktion	Beschreibung	Seite
MOVE	Kopieren	Diese Funktion kopiert Daten bitweise. Die maximale Länge beträgt 256 Worte, bei MOVE_BIT 256 Bits. Daten können ohne vorherige Konvertierung in einen anderen Datentyp kopiert werden.	4-69
BLKMOV	Block kopieren	Diese Funktion kopiert einen Block von sieben Konstanten in eine angegebene Speicheradresse. Die Konstanten werden als Teil der Funktion eingegeben.	4-72
BLKCLR	Block löschen	Diese Funktion überschreibt den Inhalt eines Datenblocks mit Nullen. Sie erlaubt das Löschen eines bit- (%I, %Q, %M, %G oder %T) oder wortstrukturierten (%R, %AI oder %AQ) Speicherbereichs. Maximal sind 256 Worte erlaubt.	4-74
SHFR	Schieberegister	Schiebt ein oder mehrere Datenworte in eine Tabelle. Maximal sind 256 Worte erlaubt.	4-76
BITSEQ	Bitfolgesteuerung	Schiebt eine Bitfolge durch ein Bitfeld. Maximal sind 256 Worte erlaubt.	4-79
COMMREQ	Kommunikationsanforderung	Diese Funktion ermöglicht dem Programm den Datenaustausch mit einem intelligenten Modul, wie zum Beispiel dem Genius-Kommunikationsmodul oder einem programmierbaren Coprozessormodul.	4-82

## MOVE (BIT, INT, WORD, REAL)

Mit der MOVE-Funktion können Sie Daten bitweise von einer Stelle zur andern kopieren. Da die Daten im Bitformat kopiert werden, können Quelle und Ziel unterschiedliche Datentypen besitzen.

Die MOVE-Funktion besitzt zwei Ein- und zwei Ausgangsparameter. Erhält die Funktion Stromfluß, kopiert sie Daten bitweise vom Eingangsparameter IN zum Ausgangsparameter Q. Werden Daten von einer Stelle im diskreten Speicher zu einer anderen Stelle kopiert (z.B. vom %I-Speicher zum %T-Speicher), wird die mit den diskreten Speicherelementen verbundene Transitionsinformation aktualisiert um anzuzeigen, ob diskrete Speicherelemente durch die MOVE-Operation ihren Zustand verändert haben. Die Daten an der Eingangsreferenz werden nur verändert, wenn in Quelle und Ziel eine Überlappung stattfindet.

Beim BIT-Typ gelten andere Aspekte. Umfaßt ein im Q-Parameter spezifiziertes BIT-Feld nicht alle Bits eines Bytes, werden die mit diesem Byte verknüpften Transitionsbits (die nicht in dem Feld sind) gelöscht, wenn MOVE\_BIT Stromfluß erhält.

Der IN-Eingang der Funktion kann entweder eine Referenz für die zu kopierenden Daten oder eine Konstante sein. Wird eine Konstante angegeben, wird der konstante Wert an der von der Ausgangsreferenz angegebenen Stelle eingetragen. Wird zum Beispiel für IN der Wert 4 angegeben, wird diese 4 in die von der Ausgangsreferenz Q angegebene Speicheradresse eingetragen werden. Ist die Länge größer als 1 und wurde eine Konstante angegeben, wird die Konstante ab der durch den Ausgang Q angegebene Speicheradresse über die angegebene Länge eingetragen. Wurden z.B. die Konstante 9 als Eingabewert für IN sowie eine Länge von 4 angegeben, dann wird die 9 in die durch den Ausgang Q festgelegte Speicheradresse sowie die drei darauffolgenden Adressen eingetragen.

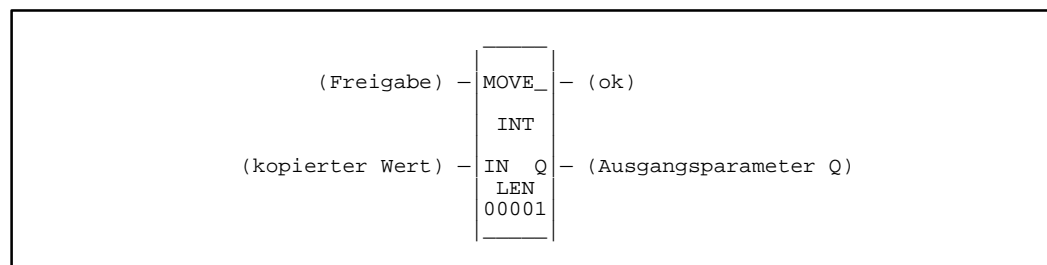
Der Operand LEN gibt an die Anzahl

- Worte die bei MOVE\_INT und MOVE\_WORD kopiert werden
- Bits, die bei MOVE\_BIT kopiert werden
- reellen Werte, die bei MOVE\_REAL kopiert werden.

### Hinweis

Der Datentyp REAL ist nur bei der CPU 352 verfügbar.

Die Funktion schaltet den Stromfluß nach rechts durch, wenn sie Stromfluß empfängt.



**Parameter:**

Parameter	Beschreibung
Freigabe	Das Kopieren wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält den Wert, der kopiert werden soll. Bei MOVE_BIT kann eine beliebige diskrete Referenz verwendet werden, die nicht im Byteraster zu liegen braucht. Es werden jedoch im On-Line-Betrieb 16 Bits ab der angegebenen Referenzadresse angezeigt.
ok	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion freigegeben ist.
Q	Der an IN anstehende Wert wird in Q eingetragen wenn die Funktion ausgeführt wird. Bei MOVE_BIT kann eine beliebige diskrete Referenz verwendet werden, die nicht im Byteraster zu liegen braucht. Es werden jedoch im On-Line-Betrieb 16 Bits ab der angegebenen Referenzadresse angezeigt.
LEN	LEN gibt an, wieviele Worte oder Bits kopiert werden sollen. Bei MOVE_WORD und MOVE_INT muß LEN zwischen 1 und 256 Worten liegen. Ist IN eine Konstante, muß bei MOVE_BIT LEN zwischen 1 und 16 Bits liegen; in den anderen Fällen zwischen 1 und 256.

**Hinweis**

Bei den CPUs 351 und 352 arbeiten die Funktionen MOVE\_INT und MOVE\_WORD nicht einwandfrei, wenn sich die Parameter IN und Q überlappen.

Beachten Sie auch, daß die CPU 352 derzeit die einzige CPU der Series 90-30 ist, die mit Gleitpunktarithmetik ausgestattet ist. Somit ist diese CPU die einzige, die MOVE\_REAL ausführen kann.

**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN		•	•	•	•	o	•	•	•	•	•	
ok	•											•
Q		•	•	•	•	o†	•	•	•	•		

**Hinweis: Bei REAL-Daten sind nur die Typen %R, %AI und %AQ zulässig.**

- Zulässige Referenz für BIT-, INT-, oder WORD-Daten oder Platz, an dem Strom durch die Funktion fließen kann.  
Bei MOVE\_BIT brauchen die diskreten Anwenderreferenzen %I, %Q, %M und %T nicht im Byteraster zu liegen.
- o Zulässige Referenz nur für BIT- oder WORD-Daten; nicht zulässig für INT
- † Nur %SA, %SB, %SC; %S kann nicht verwendet werden

### Beispiel 1:

Wird der Freigabeeingang %Q0014 EIN, werden 48 Bits von der Speicheradresse %M0001 in die Speicheradresse %M0033 kopiert. Die Verschiebung wird korrekt durchgeführt, obwohl bei Quelle und Ziel 16 Bits überlappen (außer bei den CPUs 351 und 352, wie zuvor erwähnt).



#### Vor Ausführung der MOVE-Funktion:

INPUT (%M0001 bis %M0048)

	1															
%M0016	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
%M0032	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
%M0048	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

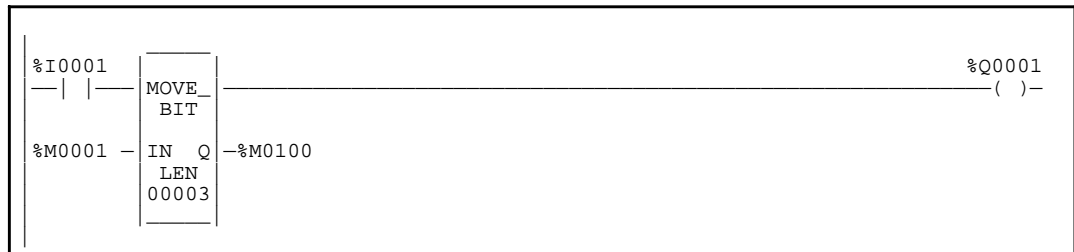
#### Nach Ausführung der MOVE-Funktion:

INPUT (%M0001 bis %M0048)

	33															
%M0048	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
%M0064	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
%M0080	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

### Beispiel 2:

Wird in diesem Beispiel %I0001 gesetzt, werden die drei Bits %M0001, %M0002 und %M0003 nach %M0100, %M0101 und %M0102 verschoben. Die Spule %Q0001 wird durchgeschaltet.



## BLKMOV (INT, WORD, REAL)

Mit der BLKMOV-Funktion können Sie einen Block von sieben Konstanten zu einer angegebenen Speicheradresse kopieren.

### Hinweis

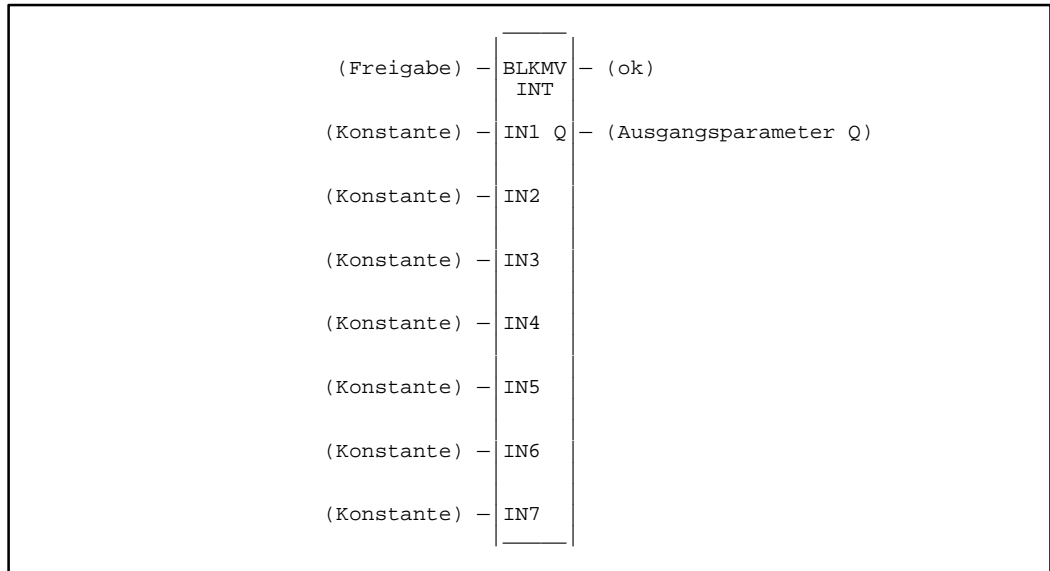
Der Datentyp REAL ist nur bei der CPU 352 verfügbar.

Die BLKMOV-Funktion besitzt acht Ein- und zwei Ausgangsparameter. Erhält die Funktion Stromfluß, kopiert sie die Konstanten in einen zusammenhängenden Speicherbereich, der bei der im Ausgang Q spezifizierten Adresse beginnt. Ausgang Q kann nicht der Eingang einer anderen Programmfunktion sein.

### Hinweis

Bei BLKMOV\_INT werden die Werte von IN1 – IN7 als Dezimalwerte mit Vorzeichen angezeigt. Bei BLKMOV\_WORD werden IN1 – IN7 in Hexadezimaldarstellung angezeigt. Bei BLKMOV\_REAL werden IN1 – IN7 als reelle Zahlen angezeigt.

Die Funktion schaltet den Stromfluß jedesmal nach rechts durch, wenn sie Stromfluß erhält.



### Parameter:

Parameter	Beschreibung
Freigabe	Der Block wird kopiert, wenn die Funktion freigegeben ist.
IN1–IN7	IN1-IN7 enthalten sieben Konstanten.
ok	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion freigegeben ist.
Q	Ausgang Q enthält die erste ganze Zahl des kopierten Feldes. IN1 wird zu Q kopiert.

### Zulässige Speichertypen:

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
enable	•											
IN1 — IN7											•	
ok	•											•
Q		•	•	•	•	o†	•	•	•	•		

**Hinweis:** Bei REAL-Daten sind nur die Typen %R, %AI und %AQ zulässig.

- Zulässige Referenz für Platz, an dem Strom durch die Funktion fließen kann.
- o Zulässige Referenz nur für WORD-Daten; nicht zulässig für INT
- † Nur %SA, %SB, %SC; %S kann nicht verwendet werden

### Hinweis

Die CPU 352 ist derzeit die einzige CPU der Series 90-30, die mit Gleitpunktarithmetik ausgestattet ist und somit die einzige CPU, die BLKMV\_REAL ausführen kann.

### Beispiel:

Beim folgenden Beispiel kopiert die BLKMOV-Funktion die sieben Eingangskonstanten in die Speicheradressen %R0010 bis %R0016, wenn der durch die symbolische Adresse FST\_SCN gekennzeichnete Freigabeeingang EIN ist.

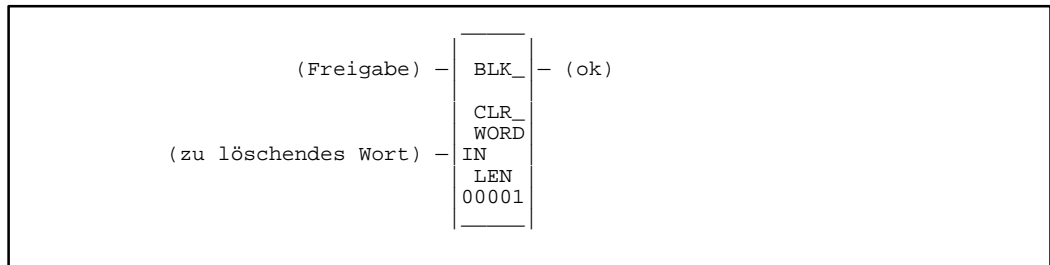


## BLKCLR (WORD)

Mit der BLKCLR-Funktion (Block löschen) kann ein spezifizierter Datenblock mit Nullen gefüllt werden.

Die BLKCLR-Funktion besitzt zwei Ein- und einen Ausgangsparameter. Erhält die Funktion Stromfluß, schreibt sie die Speicheradresse, die mit der durch IN festgelegten Adresse beginnt, mit Nullen voll. Stehen die gelöschten Daten im diskreten Speicher (%I, %Q, %M, %G oder %T), wird die mit den diskreten Speicherelementen verbundene Transitionsinformation ebenfalls gelöscht.

Die Funktion schaltet den Stromfluß jedesmal nach rechts durch, wenn sie Stromfluß erhält.



### Parameter:

Parameter	Beschreibung
Freigabe	Das Feld wird gelöscht, wenn die Funktion freigegeben ist.
IN	IN enthält den ersten Wert des Feldes, das gelöscht werden soll.
ok	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion freigegeben ist.
LEN	LEN muß zwischen 1 und 256 Worten sein.

**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN		•	•	•	•	•†	•	•	•	•		
ok	•											•

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- † Nur %SA, %SB, %SC; %S kann nicht verwendet werden

**Beispiel:**

Im folgenden Beispiel werden beim Einschalten 32 Worte des %Q-Speichers (512 Punkte), beginnend mit %Q0001, mit Nullen gefüllt.





## SHFR (BIT, WORD)

Mit der Schieberegisterfunktion (SHFR) können ein oder mehrere Datenworte oder Datenbits von einer Referenzadresse in einen angegebenen Speicherbereich verschoben werden. So kann zum Beispiel ein Wort in einen Speicherbereich mit einer angegebenen Länge von fünf Worten eingeschoben werden. Als Ergebnis dieses Vorgangs wird am Ende des Speicherbereichs ein anderes Datenwort herausgeschoben.

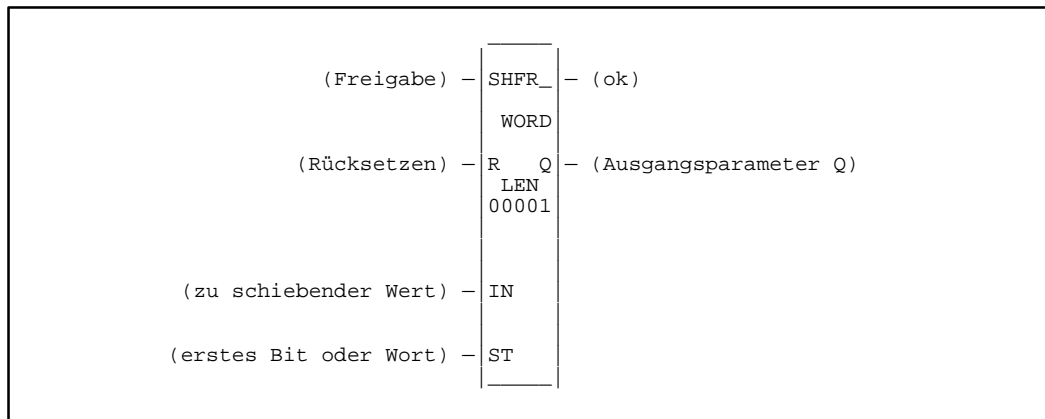
### Hinweis

Bei der Zuweisung von Referenzadressen können überlappende Ein- und Ausgangsreferenzadreibereiche in Mehrwortfunktionen zu unvorhersehbaren Reaktionen führen.

Die SHFR-Funktion besitzt vier Ein- und zwei Ausgangsparameter. Der Rücksetzeingang (R) hat Priorität über den Funktionsfreigabeeingang. Ist der Rücksetzeingang aktiv, werden ab dem Schieberegister (ST) über die für LEN angegebene Länge alle Referenzen mit Nullen gefüllt.

Erhält die Funktion Stromfluß, ohne daß der Rücksetzeingang aktiv ist, werden die Daten im Schieberegister um ein Element (entweder ein Bit oder ein Wort) zur nächst höheren Referenz hin verschoben. Das letzte Element des Schieberegisters wird in Q geschoben. Die höchste Referenz des Schieberegisterelements von IN wird in das nun freie Element ab ST eingeschoben. Auf den Inhalt des Schieberegisters kann im gesamten Programmablauf zugegriffen werden, da er an absoluten Adressen im logisch adressierbaren Speicherbereich liegt.

Die SHFREG-Funktion schaltet den Stromfluß nach rechts durch, wenn sie über die Freigabelogik Stromfluß erhält.



**Parameter:**

Parameter	Beschreibung
Freigabe	Die Verschiebung wird durchgeführt, wenn der Freigabeeingang aktiviert ist und R nicht.
R	Das Schieberegister bei ST wird mit Nullen gefüllt, wenn R aktiviert wird.
IN	IN enthält den Wert, der ins erste Bit oder Wort des Schieberegisters eingeschoben werden soll. Bei SHFR_BIT kann eine beliebige diskrete Referenz verwendet werden, die nicht im Byteraster zu liegen braucht. Es werden jedoch im On-Line-Betrieb 16 Bits ab der angegebenen Referenzadresse angezeigt.
ST	ST enthält das erste Wort oder Bit des Schieberegisters. Bei SHFR_BIT kann eine beliebige diskrete Referenz verwendet werden, die nicht im Byteraster zu liegen braucht. Es werden jedoch im On-Line-Betrieb 16 Bits ab der angegebenen Referenzadresse angezeigt.
ok	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion freigegeben ist.
Q	Der Ausgang Q enthält das aus dem Schieberegister herausgeschobene Bit oder Wort. Bei SHFR_BIT kann eine beliebige diskrete Referenz verwendet werden, die nicht im Byteraster zu liegen braucht. Es werden jedoch im On-Line-Betrieb 16 Bits ab der angegebenen Referenzadresse angezeigt.
LEN	LEN legt die Länge des Schieberegisters fest. Für SHFR_WORD muß LEN zwischen 1 und 256 Worten liegen. Für SHFR_BIT muß LEN zwischen 1 und 256 Bits liegen.

**Zulässige Speichertypen:**

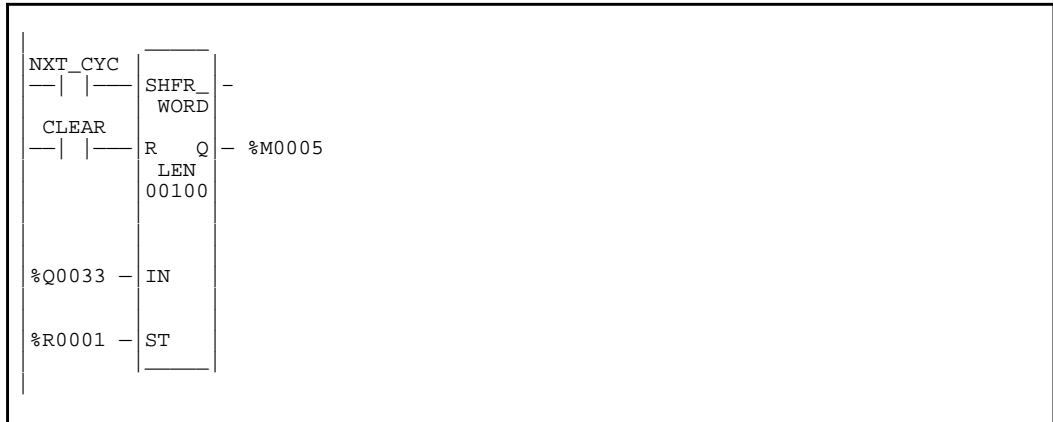
Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
enable	•											
R	•											
IN		•	•	•	•	•	•	•	•	•	•	
ST		•	•	•	•	•†	•	•	•	•		
ok	•											•
Q		•	•	•	•	•†	•	•	•	•		

- Zulässige Referenz für BIT- oder WORD-Daten oder Platz, an dem Strom durch die Funktion fließen kann.  
Bei SHFR\_BIT brauchen die diskreten Anwenderreferenzen %I, %Q, %M und %T nicht im Byteraster zu liegen.
- † Nur %SA, %SB, %SC; %S kann nicht verwendet werden

**Beispiel 1:**

Bei dem folgenden Beispiel wird die Funktion auf die Registerspeicher-Adressen %R0001 bis %R0100 angewandt. Die Schieberegisterworte werden auf Null gesetzt, wenn die Rücksetzreferenz CLEAR aktiv ist.

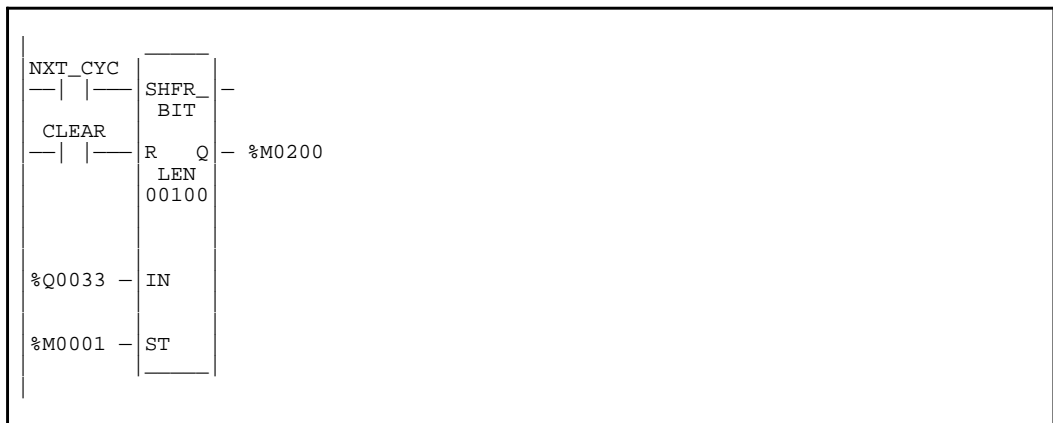
Ist die Referenz NXT\_CYC aktiv und ist CLEAR nicht aktiv, dann wird das Wort von Adresse %Q0033 der Ausgangs-Zustandstabelle in das Schieberegister geschoben. Das aus dem Schieberegister von %R0100 herausgeschobene Wort wird im Ausgang %M0005 abgelegt.



**Beispiel 2:**

Bei diesem Beispiel wird die Funktion auf die Speicheradressen %M0001 bis %M0100 angewandt. Ist die Rücksetz-Referenz CLEAR aktiv, dann füllt die SHFR-Funktion %M0001 bis %M0100 mit Nullen.

Ist NXT\_CYC aktiv und CLEAR inaktiv, dann verschiebt die SHFR-Funktion die Daten aus %M0001 bis %M0100 um ein Bit nach unten. Das Bit in %Q0033 wird nach %M0001 geschoben, während das aus %M0100 herausgeschobene Bit in %M0200 eingetragen wird.



## BITSEQ (BIT)

Mit der Bitfolgesteuerung (BITSEQ) kann eine Bitfolge durch eine Bitfeld verschoben werden. Die BITSEQ-Funktion besitzt fünf Ein- und einen Ausgangsparameter. Das Ergebnis der Funktion hängt von dem vorhergehenden Wert des Parameters EN ab:

R aktueller Durchlauf	EN voriger Durchlauf	EN aktueller Durchlauf	BITSEQ-Funktion
AUS	AUS	AUS	BITSEQ-Funktion wird nicht ausgeführt
AUS	AUS	EIN	BITSEQ-Funktion erhöht/erniedrigt um 1
AUS	EIN	AUS	BITSEQ-Funktion wird nicht ausgeführt
AUS	EIN	EIN	BITSEQ-Funktion wird nicht ausgeführt
EIN	EIN/AUS	EIN/AUS	BITSEQ-Funktion wird rückgesetzt

Der Rücksetzeingang (R) überschreibt den Freigabeeingang (EN) und bewirkt immer ein Rücksetzen der BITSEQ-Funktion. Ist der Rücksetzeingang aktiv, dann wird die aktuelle Schrittnummer auf den über den Schrittnummerparameter eingegebenen Wert gesetzt. Wurde keine Schrittnummer eingegeben, wird der Schritt auf 1 gesetzt. Mit Ausnahme des Bits, auf das der aktuelle Schritt zeigt und das auf 1 gesetzt wird, werden sämtliche Bits in der Bitfolgesteuerung auf 0 gesetzt.

Ist EN aktiv und R nicht, dann wird das Bit gelöscht, auf das der aktuelle Schritt zeigt. Die aktuelle Schrittnummer wird je nach Richtungsparameter entweder erhöht oder erniedrigt. Danach wird das Bit, auf das die neue Schrittnummer zeigt, auf 1 gesetzt.

- Die Schrittnummer wird auf 1 rückgesetzt, wenn sie beim Erhöhen außerhalb des Bereichs ( $1 \leq \text{Schrittnummer} \leq \text{LEN}$ ) gerät.
- Die Schrittnummer wird auf LEN rückgesetzt, wenn sie beim Erniedrigen außerhalb des Bereichs ( $1 \leq \text{Schrittnummer} \leq \text{LEN}$ ) gerät.

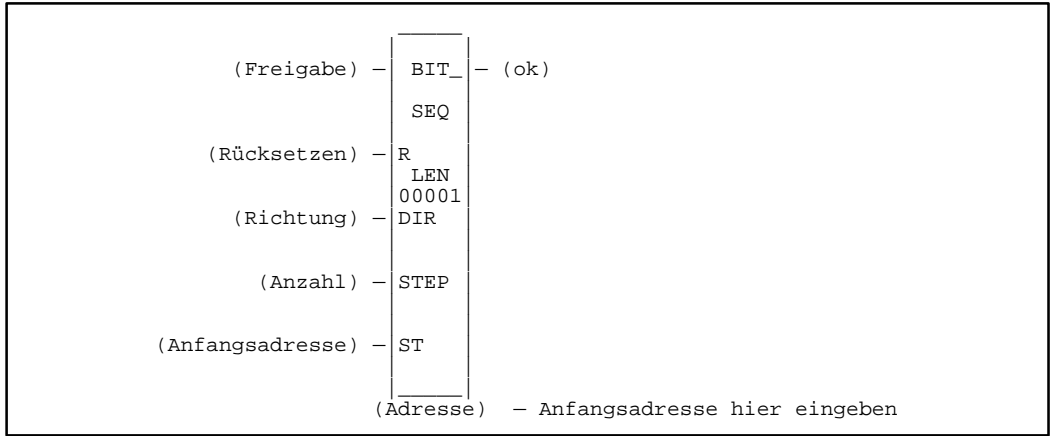
Die Angabe des ST-Parameters ist nicht zwingend. Wird er nicht verwendet, dann arbeitet die BITSEQ-Funktion wie beschrieben, es werden jedoch keine Bits gelöscht oder gesetzt. Die BITSEQ-Funktion schaltet lediglich die aktuelle Schrittnummer durch den zulässigen Bereich hindurch.

### Speicherbedarf der Bitfolgesteuerung

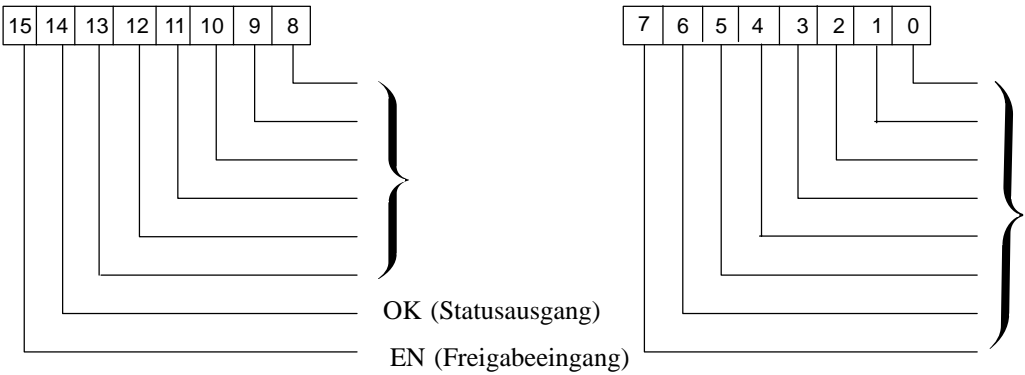
Jede Bitfolgesteuerung belegt im %R-Speicher drei Worte (Register), um die folgenden Daten abzuspeichern:

aktuelle Schrittnummer	Wort 1
Länge der Bitfolge (in Bits)	Wort 2
Steuerwort	Wort 3

Bei der Eingabe der BITSEQ-Funktion müssen Sie eine Anfangsadresse für diese drei Worte unmittelbar unterhalb der Funktionsgraphik eingeben (siehe Beispiel auf der nächsten Seite).



Im Steuerwort werden die Zustände der Booleschen Ein- und Ausgänge des zugehörigen Funktionsblocks in folgendem Format abgelegt:



	Beschreibung
Adresse	Gibt die Speicheradresse von aktuellem Schritt und Länge sowie der letzte Freigabe- und OK-Zustände an.
Freigabe	Die Bitfolge wird verschoben, wenn die Funktion freigegeben wird, nachdem sie beim letzten Zyklus nicht freigegeben war und R nicht durchgeschaltet ist.
R	Wird R aktiviert, dann wird die Schrittnummer der BITSEQ-Funktion auf den Wert in STEP (Standard = 1) gesetzt und mit Ausnahme des aktuellen Schrittnummernbits werden alle Bits der Folge auf Null gesetzt..
DIR	Wird DIR aktiviert, dann wird die Schrittnummer vor dem Verschieben um 1 erhöht. Ist DIR nicht durchgeschaltet, dann wird sie erniedrigt.
STEP	Die Schrittnummer wird auf diesen Wert eingestellt, wenn R aktiviert wird.
ST	ST enthält das erste Wort der Bitfolgesteuerung
ok	Der OK-Ausgang wird bei jeder Freigabe der Funktion durchgeschaltet.
LEN	LEN muß zwischen 1 und 256 Bits sein.

## Hinweis

Bei der BITSEQ-Funktion prüft die Spulenverwendungs-Überprüfungsfunktion 16 Bits des ST-Parameters, selbst wenn LEN kleiner als 16 ist.

### Zulässige Speichertypen:

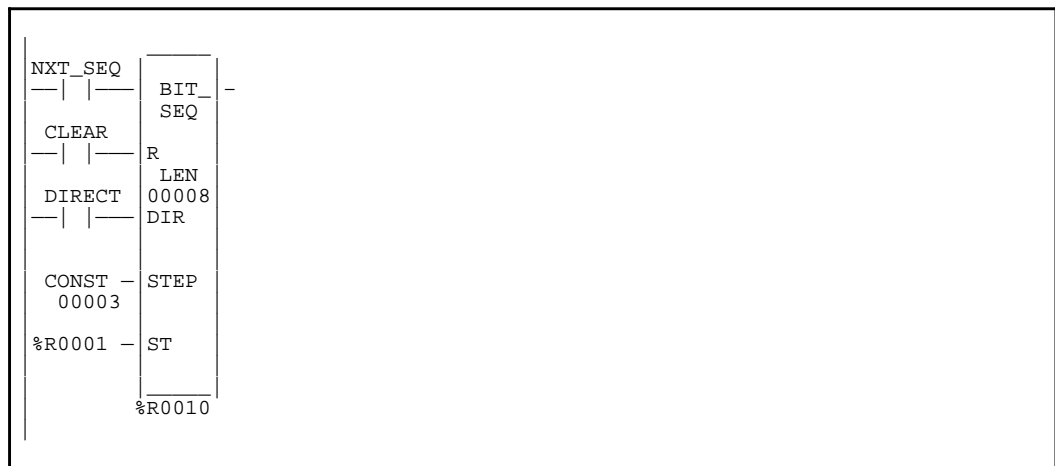
Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Adresse								•				
Freigabe	•											
R	•											
DIR	•											
STEP		•	•	•	•		•	•	•	•	•	•
ST		•	•	•	•	•†	•	•	•	•		•
ok	•											•

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- † Nur %SA, %SB, %SC; %S kann nicht verwendet werden

### Beispiel:

Beim folgenden Beispiel verwendet die BITSEQ-Funktion den Registerspeicher %R0001. Die statischen Daten sind in den Registern %R0010, %R0011 und %R0012 abgelegt. Ist CLEAR aktiv, dann wird die Funktion rückgesetzt und der aktuelle Schritt wird auf Schrittnummer 3 eingestellt. Die ersten acht Bits von %R0001 werden auf Null gesetzt.

Ist NXT\_SEQ aktiv und CLEAR nicht, dann wird das Bit für Schrittnummer 3 gelöscht und das Bit für Schrittnummer 2 oder 4 (hängt davon ab, ob DIR durchgeschaltet ist) wird gesetzt.



## COMMREQ

Die COMMREQ-Funktion wird eingesetzt, wenn das Programm mit einem intelligenten Modul (z.B. Genius-Kommunikationsmodul oder programmierbares Coprozessormodul) Daten austauschen will.

### Hinweis

Auf den folgenden Seiten wird lediglich das Format der COMMREQ-Funktion dargestellt. Für jeden Gerätetyp brauchen Sie zusätzliche Angaben bei der Programmierung der COMMREQ-Funktion, die Sie in der Gerätedokumentation zu den einzelnen Modulen finden.

Die COMMREQ-Funktion besitzt drei Ein- und einen Ausgangsparameter. Erhält eine COMMREQ-Funktion Stromfluß, dann wird eine Befehlsdatenblock an das intelligente Modul geschickt. Dieser Befehlsblock beginnt mit der über den Parameter IN spezifizierten Referenz. Das Gerät, mit dem Daten ausgetauscht werden sollen, wird durch seine in SYSID angegebene Chassis- und Steckplatznummer identifiziert.

Die COMMREQ-Funktion sendet entweder eine Meldung und wartet auf Antwort oder sie sendet eine Meldung und fährt ohne Antwort fort. Gibt der Befehlsblock an, daß das Programm nicht auf eine Antwort wartet, dann wird der Inhalt des Befehlsblocks zu dem Empfangsgerät gesendet und die Programmausführung direkt fortgesetzt (der Zeitüberwachungswert wird ignoriert). Diese Betriebsart ist der **NOWAIT**-Modus.

Gibt der Befehlsblock an, daß das Programm auf eine Antwort wartet, wird der Inhalt des Befehlsblocks zu dem Empfangsgerät gesendet und die CPU wartet auf eine Antwort. Die maximale Wartezeit wird im Befehlsblock angegeben. Antwortet das Gerät nicht innerhalb dieser Zeit, wird die Programmausführung fortgesetzt. Diese Betriebsart ist der **WAIT**-Modus.

Die Funktion schaltet Stromfluß durch, wenn die vorgegebene Zeitdauer nicht überschritten wurde (Timeout) und keine Nullzeit als Überwachungszeit angegeben wurde. Der Funktionsfehlerausgang (FT) wird gesetzt, wenn

1. die angegebene Zieladresse nicht existiert (SYSID);
2. die angegebene Task für das Gerät nicht zulässig ist (TASK);
3. die Datenlänge 0 beträgt;
4. die Status-Pointeradresse (Teil des Befehlsblocks) des Geräts nicht existiert. Ursache kann eine fehlerhafte Wahl des Speichertyps sein oder eine Adresse innerhalb dieses Speichertyps, die außerhalb des zulässigen Bereichs liegt.

### Befehlsblock

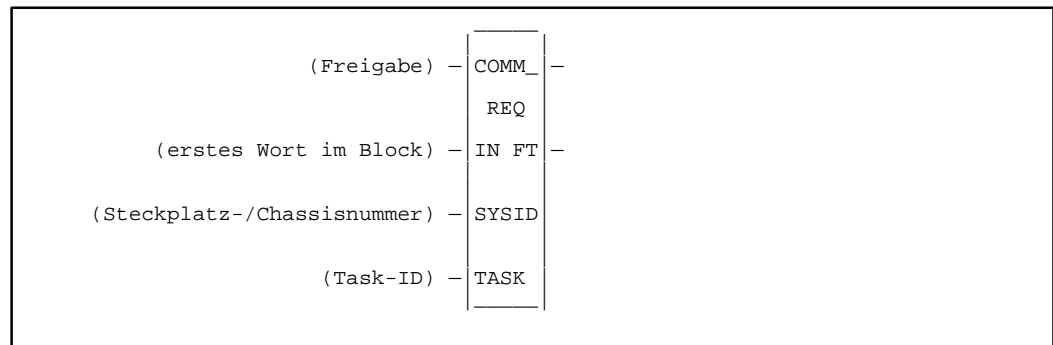
Der Befehlsblock liefert dem intelligenten Modul Informationen zu dem auszuführenden Befehl.

Die Adresse des Befehlsblocks wird für den IN-Eingang zur COMMREQ-Funktion angegeben. Bei dieser Adresse kann es sich um einen beliebigen wortstrukturierten Speicherbereich (%R, %AI oder %AQ) handeln. Die Länge des Befehlsblocks hängt von der zum Gerät gesendeten Datenmenge ab.

Der Befehlsblock besitzt folgende Struktur:

Länge (in Worten)	Adresse
Anzeige "Warten/nicht warten"	Adresse + 1
StatuszeigerSpeicher	Adresse + 2
StatuszeigerOffset	Adresse + 3
Pausenüberwachungswert	Adresse + 4
Max. Kommunikationsdauer	Adresse + 5
Datenblock	Adresse + 6
	bis Adresse + 133

Die für den Befehlsblock erforderlichen Angaben können mit der entsprechenden Programmierfunktion in die angegebenen Speicherbereiche eingetragen werden.



**Parameter:**

Parameter	Beschreibung
Freigabe	Die Kommunikationsanforderung wird ausgeführt, wenn die Funktion aktiviert ist.
IN	IN enthält das erste Wort des Befehlsblocks.
SYSID	SYSID enthält die Chassisnummer (höchstwertiges Byte) und die Steckplatznummer (niedrigwertiges Byte) des Zielgeräts.
TASK	Dieser Parameter enthält die Task-Kennung vom Prozeß des Zielgeräts.
FT	FT wird durchgeschaltet, wenn die Kommunikationsanforderung erfolglos ist.

**Zulässige Speichertypen:**

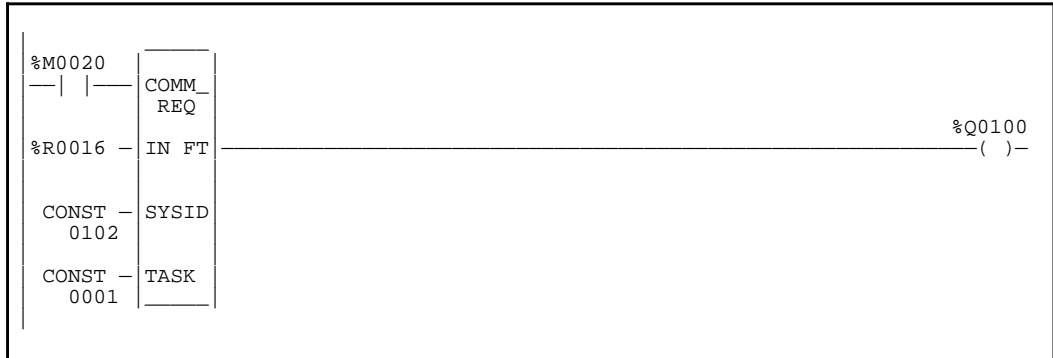
Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN								•	•	•		
SYSID		•	•	•	•		•	•	•	•	•	
TASK								•	•	•	•	
FT	•											•

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.



**Beispiel:**

Ist im folgenden Beispiel der Freigabeeingang %M0020 EIN, wird der ab Adresse %R0016 abgelegte Befehlsblock zur Kommunikations-Task 1 des Geräts übertragen, das in Steckplatz 2 von Chassis 1 in der SPS eingebaut ist. Tritt ein Fehler bei der Bearbeitung der COMMREQ-Funktion auf, wird %Q0100 durchgeschaltet.



**Hinweis**

Bei Systemen ohne Erweiterungschassis muß SYSID des Hauptchassis Null sein.

## Abschnitt 7: Tabellenfunktionen

Mit den Tabellenfunktionen können Sie folgende Funktionen ausführen:

Abkürzung	Funktion	Beschreibung	Seite
ARRAY_MOVE	Feld kopieren	Kopiert eine vorgegebene Anzahl Datenelemente von einem Quellfeld zu einem Zielfeld	4-86
SRCH_EQ	Suche gleiche	Sucht nach allen Feldwerten, die zu einem angegebenen Wert gleich sind.	4-90
SRCH_NE	Suche ungleiche	Sucht nach allen Feldwerten, die von einem angegebenen Wert verschieden sind.	4-90
SRCH_GT	Suche größer als	Sucht nach allen Feldwerten, die größer als ein angegebener Wert sind.	4-90
SRCH_GE	Suche größer als oder gleich	Sucht nach allen Feldwerten, die größer als ein angegebener Wert oder gleich sind.	4-90
SRCH_LT	Suche kleiner als	Sucht nach allen Feldwerten, die kleiner als ein angegebener Wert sind.	4-90
SRCH_LE	Suche kleiner als oder gleich	Sucht nach allen Feldwerten, die kleiner als ein angegebener Wert oder gleich sind.	4-90

Die maximal zulässige Länge bei diesen Funktionen beträgt 32.767 Bytes oder Worte bzw. 262.136 Bits (Bits nur für ARRAY\_MOVE).

Tabellenfunktionen können auf folgende Datentypen angewandt werden:

Datentyp	Beschreibung
INT	Ganze Zahl mit Vorzeichen
DINT	Doppelt genaue ganze Zahl mit Vorzeichen
BIT *	Bit
BYTE	Byte .
WORD	Wort

\* Nur für ARRAY\_MOVE.

Die Standardbelegung (ganze Zahl mit Vorzeichen) kann nach Anwahl der Funktion verändert werden. Um andere Datentypen oder Daten mit zwei unterschiedlichen Typen vergleichen zu können, müssen Sie zunächst die entsprechende Datenkonvertierungsfunktion (beschrieben in Abschnitt 8) einsetzen, um eine Umwandlung in die oben genannten Datentypen vorzunehmen.

## ARRAY\_MOVE (INT, DINT, BIT, BYTE, WORD)

Mit der Funktion ARRAY\_MOVE (Feld kopieren) können Sie eine angegebene Anzahl Datenelemente von einem Quellfeld in ein Zielfeld kopieren.

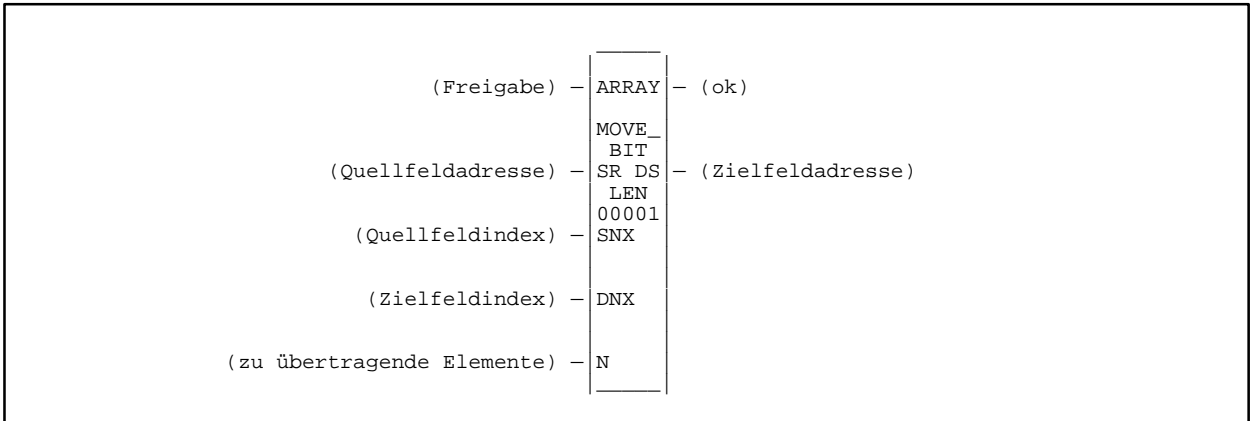
Die Funktion ARRAY\_MOVE besitzt fünf Ein- und zwei Ausgangsparameter. Empfängt die Funktion Stromfluß, wird ab der indizierten Adresse (SR + SNX — 1) die Anzahl der Datenelemente in der Zählanzeige (N) aus dem Eingangsfeld herausgezogen. Die Datenelemente werden in das Ausgangsfeld mit der indizierten Adresse (DS + DNX — 1) geschrieben. Der Operand LEN gibt an, aus wieviel Elementen die Felder bestehen.

Wird bei ARRAY\_MOVE\_BIT für die Parameter der Anfangsadresse von Quell- und/oder Zielfeld wortstrukturierter Speicher gewählt, ist das niedrigstwertige Bit des angegebenen Wortes das erste Bit des Feldes. Der angezeigte Wert enthält unabhängig von der Feldlänge immer 16 Bits.

Die Indizes einer ARRAY\_MOVE-Anweisung sind auf Basis 1. Beim Einsatz von ARRAY\_MOVE darf kein Element außerhalb der Quell- und Zielfelder (die durch ihre Anfangsadresse und Länge spezifiziert wurden) adressiert werden.

Der OK-Ausgang empfängt Stromfluß, wenn nicht eine der folgenden Umstände vorliegt:

- Der Freigabeeingang ist AUS.
- (N + SNX — 1) ist größer als LEN.
- (N + DNX — 1) ist größer als LEN.



**Parameter:**

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
SR	SR enthält die Anfangsadresse des Quellfeldes. Für ARRAY_MOVE_BIT kann eine beliebige Referenz verwendet werden, die nicht im Byteraster zu liegen braucht. Es werden jedoch im On-Line-Betrieb 16 Bits ab der angegebenen Referenzadresse angezeigt.
SNX	SNX enthält den Index des Quellfeldes.
DNX	DNX enthält den Index des Quellfeldes.
N	N bietet eine Zählanzeige.
ok	Der OK-Ausgang wird durchgeschaltet, wenn der Freigabeeingang aktiviert ist.
DS	DS enthält die Anfangsadresse des Zielfeldes. Für ARRAY_MOVE_BIT kann eine beliebige Referenz verwendet werden, die nicht im Byteraster zu liegen braucht. Es werden jedoch im On-Line-Betrieb 16 Bits ab der angegebenen Referenzadresse angezeigt.
LEN	LEN gibt die Anzahl Elemente ab SR und DS an, aus denen die Felder bestehen.

**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
SR		o	o	o	o	n †	o	•	•	•		
SNX		•	•	•	•		•	•	•	•	•	
DNX		•	•	•	•		•	•	•	•	•	
N		•	•	•	•		•	•	•	•	•	
ok	•											•
DS		o	o	o	o	†	o	•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.  
Bei ARRAY\_MOVE\_BIT brauchen die diskreten Anwenderreferenzen %I, %Q, %M und %T nicht im Byteraster zu liegen.
- o Zulässige Referenz nur für INT, BIT, BYTE oder WORD, nicht zulässig für DINT.
- n Zulässiger Datentyp nur für BIT, BYTE oder WORD, nicht zulässig für INT oder DINT.
- † Nur %SA, %SB, %SC; %S kann nicht verwendet werden

**Beispiel 1:**

Ist in diesem Beispiel %R100=3, dann werden %R0003 — %R0007 des Feldes %R0001 — %R0016 gelesen und in %R0104 — %R0108 des Feldes %R0100 — %R0115 geschrieben.

```

%I0001  ARRAY -
        MOVE_
        WORD
%R0001  SR DS - %R0100
        LEN
        00016
%R100   SNX

CONST   DNX
        00005

CONST   N
        00005
    
```

**Beispiel 2:**

Unter Benutzung von Bitspeicher für SR und DS wird %M0011 — %M0017 des Feldes %M0009 — %M0024 gelesen und in %Q0026 — %Q0032 des Feldes %Q0022 — %Q0037 geschrieben.

```

%I0001  ARRAY -
        MOVE_
        BIT
%M0009  SR DS - %Q0022
        LEN
        00016
CONST   SNX
        00003

CONST   DNX
        00005

CONST   N
        00007
    
```

**Beispiel 3:**

Unter Benutzung von Wortspeicher für SR und DS wird aus dem Feld mit allen sechzehn Bits von %R0001 und vier Bits von %R0002 der Teil vom drittniedrigsten Bit von %R0001 bis einschließlich dem zweitniedrigsten Bit von %R0002 gelesen und dann in den Bereich vom fünftniedrigsten Bit von %R0100 bis einschließlich dem viertniedrigsten Bit von %R0101 des Feldes geschrieben, das alle 16 Bits von %R0100 und vier Bits von %R0101 enthält.

```

%I0001  |-----|
--| |---| ARRAY  |-----|
        |         | MOVE_ |
        |         | BIT   |
%R0001  |-----| SR DS  |-----| %R0100
        |         | LEN   |
        |         | 00020 |
CONST   |-----| SNX   |
00003   |         |
        |         |
CONST   |-----| DNX   |
00005   |         |
        |         |
CONST   |-----| N     |
00016   |         |
    
```

## SRCH\_EQ und SRCH\_NE (INT, DINT, BYTE, WORD) SRCH\_GT und SRCH\_LT SRCH\_GE und SRCH\_LE

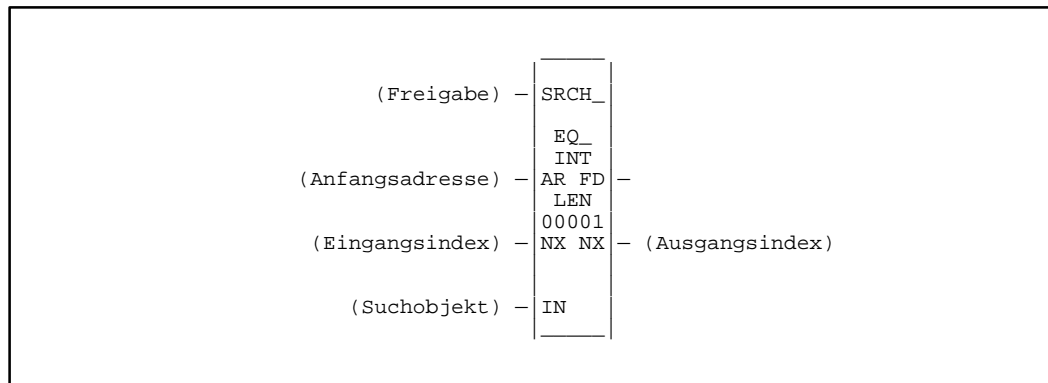
Mit den nachstehend aufgeführten Suchfunktionen können Sie nach Feldwerten suchen.

Abkürzung	Funktion	Beschreibung
SRCH_EQ	Suche gleiche	Sucht nach allen Feldwerten, die zu einem angegebenen Wert gleich sind.
SRCH_NE	Suche ungleiche	Sucht nach allen Feldwerten, die von einem angegebenen Wert verschieden sind.
SRCH_GT	Suche größer als	Sucht nach allen Feldwerten, die größer als ein angegebener Wert sind.
SRCH_GE	Suche größer als oder gleich	Sucht nach allen Feldwerten, die größer als ein angegebener Wert oder gleich sind.
SRCH_LT	Suche kleiner als	Sucht nach allen Feldwerten, die kleiner als ein angegebener Wert sind.
SRCH_LE	Suche kleiner als oder gleich	Sucht nach allen Feldwerten, die kleiner als ein angegebener Wert oder gleich sind.

Jede Funktion besitzt vier Ein- und zwei Ausgangsparameter. Empfängt die Funktion Stromfluß, wird das Feld ab (AR + Eingang NX) durchsucht. Der Startpunkt setzt sich zusammen aus der Anfangsadresse des Feldes (AR) plus dem Index zu diesem Feld (Eingang NX).

Die Suche wird durchgeführt, bis das Feldelement des Suchobjekts (IN) gefunden oder das Feldende erreicht ist. Wird ein Feldelement gefunden, wird der Ausgangsparameter (FD) durchgeschaltet und der Ausgangsparameter (Ausgang NX) auf die Relativposition dieses Elements im Feld gesetzt. Wird vor Erreichen des Feldendes kein passendes Feldelement gefunden, wird Ausgangsparameter FD auf AUS gesetzt und der Ausgangsparameter (Ausgang NX) auf Null.

Zulässige Werte für Eingang NX sind 0 bis LEN-1. Um die Suche beim ersten Element zu beginnen, muß NX auf Null gesetzt werden. Da bei der Ausführung dieser Wert um 1 erhöht wird, ergibt sich für Ausgang NX ein Wertebereich zwischen 1 und LEN. Liegt der Wert von Eingang NX außerhalb des Bereichs (<0 oder ≥LEN), wird sein Wert auf den Standardwert 0 gesetzt.



**Parameter:**

Parameter	Beschreibung
Freigabe	Die Operation wird ausgeführt, wenn die Funktion freigegeben ist.
AR	AR enthält die Anfangsadresse des durchsuchten Feldes.
Eingang NX	Eingang NX enthält den Index zu dem Feld, bei dem die Suche beginnen soll.
IN	IN enthält das gesuchte Objekt.
Ausgang NX	Ausgang NX enthält die Position des Suchziels im Feld.
FD	FD zeigt an, daß ein Feldelement gefunden wurde und die Funktion erfolgreich war.
LEN	LEN gibt die Anzahl Elemente ab AR an, aus denen das durchsuchte Feld besteht. Der mögliche Bereich sind 1 bis 32.767 Bytes oder Worte.

**Zulässige Speichertypen:**

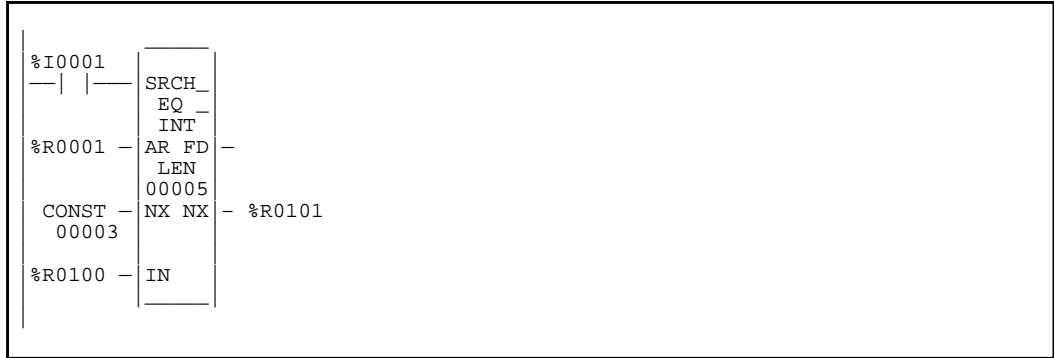
Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	none
Freigabe	•											
AR		o	o	o	o	n	o	•	•	•		
NX Ein		•	•	•	•		•	•	•	•	•	
IN		o	o	o	o	n	o	•	•	•	•	
NX Aus		•	•	•	•		•	•	•	•		
FD	•											•

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- o Zulässige Referenz nur für INT, BYTE oder WORD, nicht zulässig für DINT.
- n Zulässige Referenz nur für BYTE oder WORD, nicht zulässig für INT oder DINT.



**Beispiel 1:**

Das Feld AR ist definiert als Speicheradressen %R0001 – %R0005. Ist EN EIN, wird der Teil des Feldes zwischen %R0004 und %R0005 nach einem Element durchsucht, dessen Wert gleich IN ist. Sind %R0001 = 7, %R0002 = 9, %R0003 = 6, %R0004 = 7, %R0005 = 7 und %R0100 = 7, beginnt die Suche bei %R0004 und endet bei %R0004, wenn FD aktiviert wird. In %R0101 wird eine 4 eingetragen..



**Beispiel 2:**

Das Feld AR ist definiert als Speicheradressen %AI0001 – %AI0016. Die Werte dieser Feldelemente sind 100, 20, 0, 5, 90, 200, 0, 79, 102, 80, 24, 34, 987, 8, 0 und 500. Zunächst steht in %AQ0001 der Wert 5. Ist EN aktiviert, wird das Feld bei jedem Zyklus nach einer Übereinstimmung mit dem Wert 0 von IN durchsucht. Im ersten Zyklus beginnt die Suche bei %AI0006. Eine Übereinstimmung wird gefunden bei %AI0007. Hierauf wird FD durchgeschaltet und %AQ0001 auf 7 gesetzt. Im zweiten Zyklus beginnt die Suche bei %AI0008. Eine Übereinstimmung wird gefunden in %AI0015. FD bleibt somit EIN und %AQ0001 wird auf 15 gesetzt. Der nächste Zyklus beginnt bei %AI0016. Da jetzt das Feldende ohne Übereinstimmung erreicht wird, geht FD auf AUS und %AQ0001 wird auf Null gesetzt. Im nächsten Zyklus beginnt die Suche wieder am Anfang des Feldes.



## Abschnitt 8: Konvertierungsfunktionen

Mit den Konvertierungsfunktionen können Sie das Format eines Datenelements umwandeln. Zahlreiche Programmanweisungen, wie z.B. die arithmetischen Funktionen, verlangen die Verwendung eines bestimmten Datentyps. In diesem Kapitel werden folgende Konvertierungsfunktionen beschrieben:

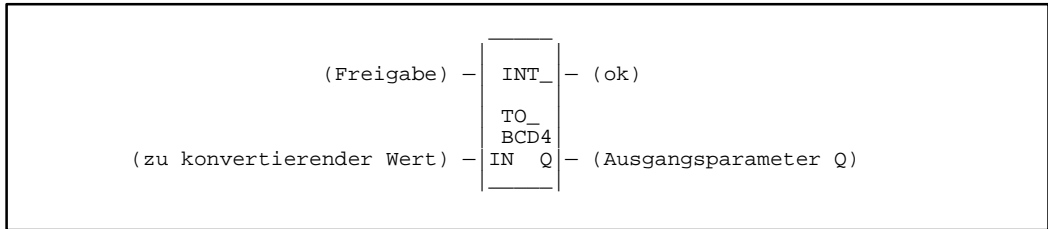
Abkürzung	Funktion	Beschreibung	Seite
BCD-4	Umwandlung in BCD-4	Konvertiert eine vorzeichenbehaftete ganze Zahl in 4-stelliges BCD-Format.	4-94
INT	Umwandlung in ganze Zahl mit Vorzeichen	Konvertiert eine Zahl im BCD-4-Format in eine vorzeichenbehaftete ganze Zahl.	4-96
DINT	Umwandlung in doppelgenaue ganze Zahl mit Vorzeichen	Konvertiert eine Zahl im REAL-Format in eine doppelgenaue ganze Zahl mit Vorzeichen.	4-98
REAL	Umwandlung in REAL	Konvertiert eine Zahl im Format INT, DINT, BCD-4 oder WORD in das REAL-Format.	4-100
WORD	Umwandlung in WORD	Konvertiert eine Zahl im REAL-Format in das WORD-Format.	4-102
TRUN	Abschneiden	Rundet die reelle Zahl gegen Null hin.	4-104

—>BCD-4 (INT)

Die BCD-4-Funktion gibt das vierstellige BCD-Äquivalent einer vorzeichenbehafteten ganzen Zahl aus. Die ursprünglichen Daten werden dabei nicht verändert. Die Ausgangsdaten können direkt als Eingangsdaten einer anderen Funktion verwendet werden.

Die Konvertierung in BCD-Format ist zum Beispiel erforderlich für die Ansteuerung von BCD-codierten LED-Anzeigen oder für die Einstellung externer Geräte (z.B. schneller Zähler).

Erhält die Funktion Stromfluß, führt sie die Konvertierung durch und stellt das Ergebnis am Ausgang Q bereit. Solange das Ergebnis der Konvertierung im Bereich zwischen 0 und 9999 liegt schaltet die Funktion den Stromfluß durch, wenn sie aktiviert wird.



**Parameter:**

Parameter	Beschreibung
Freigabe	Die Konvertierung wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält eine Referenz für den ganzzahligen Wert, der in BCD-4 konvertiert werden soll.
ok	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion fehlerfrei ausgeführt wurde.
Q	Ausgang Q enthält das BCD-4-Äquivalent des in IN spezifizierten Wertes.

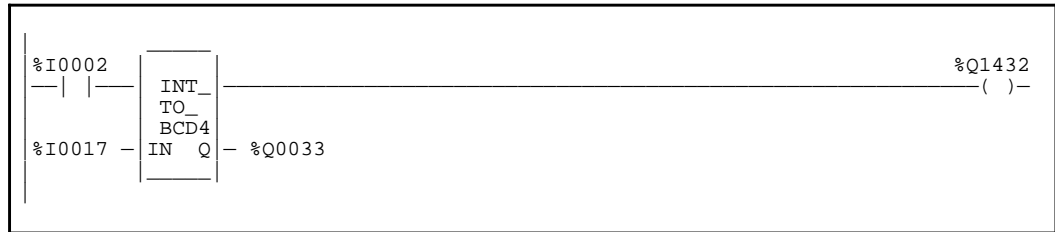
### Zulässige Speichertypen:

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN		•	•	•	•		•	•	•	•	•	
ok	•											•
Q		•	•	•	•		•	•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

### Beispiel:

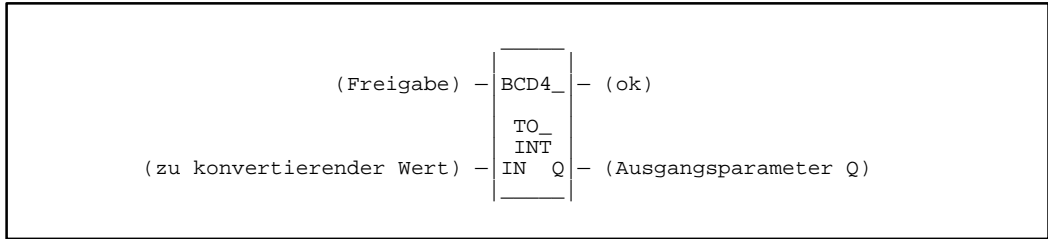
Wird im folgenden Beispiel der Eingang %I0002 gesetzt und liegt kein Fehler vor, wird die an den Eingängen %I0017 bis %I0032 anliegende vorzeichenbehaftete ganze Zahl in vier BCD-Stellen umgewandelt. Das Ergebnis wird in den Speicheradressen %Q0033 bis %Q0048 abgelegt. Mit der Spule %Q01432 kann überprüft werden, ob die Konvertierung erfolgreich war.



—>INT (BCD-4)

Die INT-Funktion gibt das ganzzahlige Äquivalent eines 4-stelligen BCD-Wertes aus. Die ursprünglichen Daten werden dabei nicht verändert. Die Ausgangsdaten können direkt als Eingangsdaten einer anderen Funktion verwendet werden.

Erhält die Funktion Stromfluß, dann führt sie die Konvertierung durch und stellt das Ergebnis am Ausgang Q bereit. Solange die Daten innerhalb des zulässigen Bereichs liegen, schaltet die Funktion den Stromfluß immer durch, wenn sie aktiviert wird.



**Parameter:**

Parameter	Beschreibung
Freigabe	Die Konvertierung wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält eine Referenz für den BCD-4 Wert, der in einen ganzzahligen Wert konvertiert werden soll.
ok	Solange die Daten innerhalb des zulässigen Bereichs liegen, wird der OK-Ausgang durchgeschaltet, wenn der Freigabeeingang aktiviert wird.
Q	Ausgang Q enthält die ganzzahlige Form des ursprünglichen Wertes aus IN.

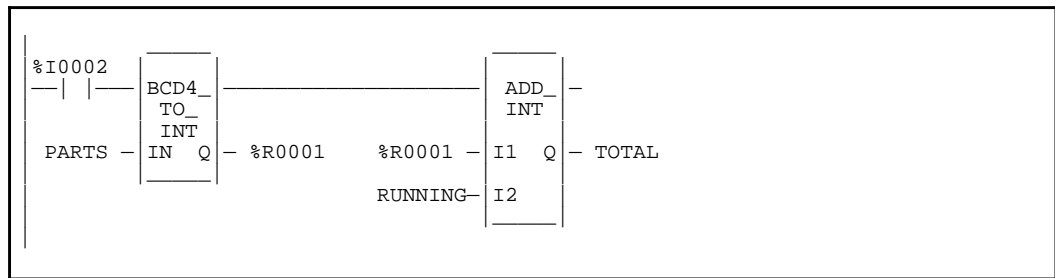
### Zulässige Speichertypen:

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN		•	•	•	•		•	•	•	•	•	
ok	•											•
Q		•	•	•	•		•	•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

### Beispiel:

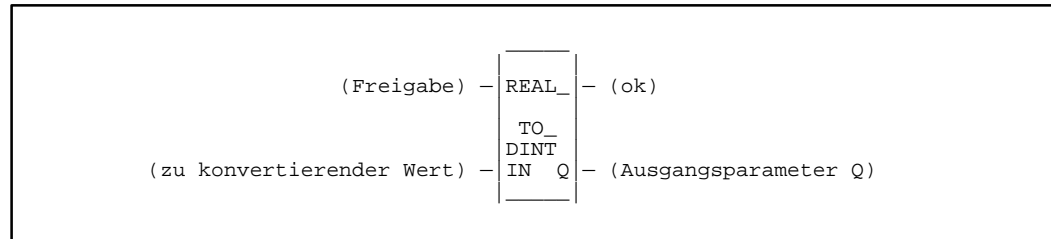
Wird im folgenden Beispiel der Eingang %I0002 gesetzt, dann wird der vierstellige BCD-Wert in PARTS in eine vorzeichenbehaftete ganze Zahl konvertiert, die an die ADD-Funktion weitergegeben wird. Hier wird der Wert zu der durch die Referenz RUNNING spezifizierten vorzeichenbehafteten ganzen Zahl addiert. Das Ergebnis wird von der ADD-Funktion an die Referenz TOTAL ausgegeben.



—>**DINT (REAL)**

Die DINT-Funktion gibt das vorzeichenbehaftete doppelgenaue ganzzahlige Äquivalent eines Wertes im REAL-Format aus. Die ursprünglichen Daten werden dabei nicht verändert. Die Ausgangsdaten können direkt als Eingangsdaten einer anderen Funktion verwendet werden.

Erhält die Funktion Stromfluß, dann führt sie die Konvertierung durch und stellt das Ergebnis am Ausgang Q bereit. Solange der Wert im REAL-Format innerhalb des zulässigen Bereichs liegt, schaltet die Funktion den Stromfluß immer durch, wenn sie aktiviert wird.



**Parameter:**

Parameter	Beschreibung
Freigabe	Die Konvertierung wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält eine Referenz für den Wert, der in einen doppelgenauen ganzzahligen Wert konvertiert werden soll.
ok	Solange die REAL-Daten innerhalb des zulässigen Bereichs liegen, wird der OK-Ausgang durchgeschaltet, wenn der Freigabeeingang aktiviert wird.
Q	Ausgang Q enthält die vorzeichenbehaftete doppelgenaue ganzzahlige Form des ursprünglichen Wertes aus IN.

**Hinweis**

Bei der Wandlung von REAL nach DINT kann Genauigkeit verlorengehen, da REAL 24 signifikante Bits besitzt.

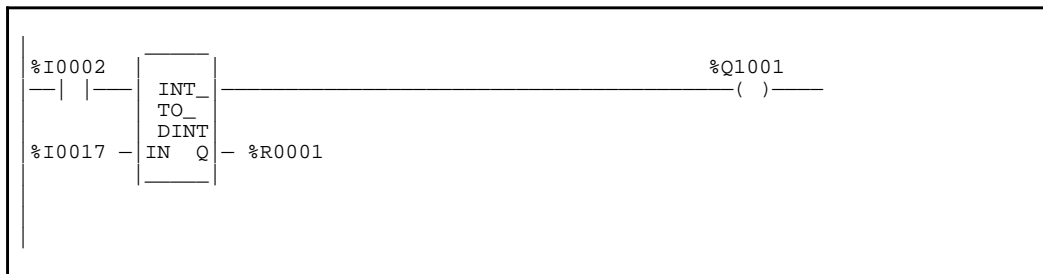
**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN		o	o	o	o		o	•	•	•	•	
ok	•											•
Q								•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

**Beispiel:**

Wird im folgenden Beispiel der Eingang %I0002 gesetzt, dann wird der ganzzahlige Wert in der Eingangsadresse %I0017 in eine vorzeichenbehaftete doppelgenaue ganze Zahl konvertiert. Das Ergebnis wird in die Adresse %R0001 eingetragen. Der Ausgang %Q1001 wird durchgeschaltet, wenn die Funktion erfolgreich ausgeführt wurde.



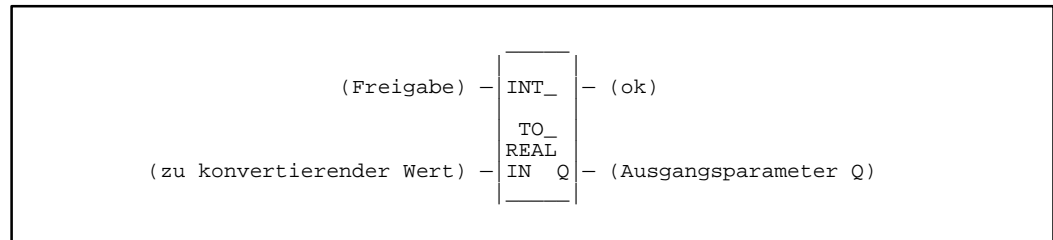


—>REAL (INT, DINT, BCD-4, WORD)

Diese Funktion gibt den reellen Wert der Eingangsdaten aus. Die ursprünglichen Daten werden dabei nicht verändert. Die Ausgangsdaten können direkt als Eingangsdaten einer anderen Funktion verwendet werden.

Erhält die Funktion Stromfluß, dann führt sie die Konvertierung durch und stellt das Ergebnis am Ausgang Q bereit. Solange die angegebene Konvertierung einen Wert innerhalb des zulässigen Bereichs ergibt, schaltet die Funktion den Stromfluß immer durch, wenn sie aktiviert wird.

Bei der Wandlung von DINT nach REAL kann Genauigkeit verlorengehen, da die Anzahl der signifikanten Bits auf 24 reduziert wird.



**Parameter:**

Parameter	Beschreibung
Freigabe	Die Konvertierung wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält eine Referenz für den ganzzahligen Wert, der in einen REAL-Wert konvertiert werden soll.
ok	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion fehlerfrei ausgeführt wird
Q	Ausgang Q enthält die REAL-Form des ursprünglichen Wertes aus IN.

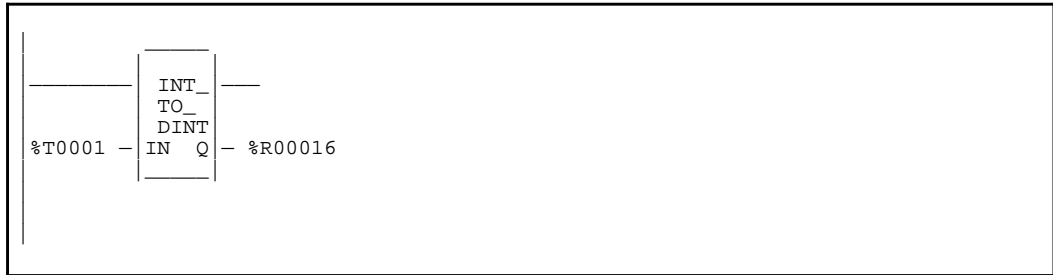
**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN		o	o	o	o		o	•	•	•	•	
ok	•											•
Q								•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- o Nicht zulässig für DINT\_TO\_REAL

**Beispiel:**

Im folgenden Beispiel beträgt der ganzzahlige Wert des Eingangs IN 678. Das Ergebnis (678.000) wird in %T0016 abgelegt.



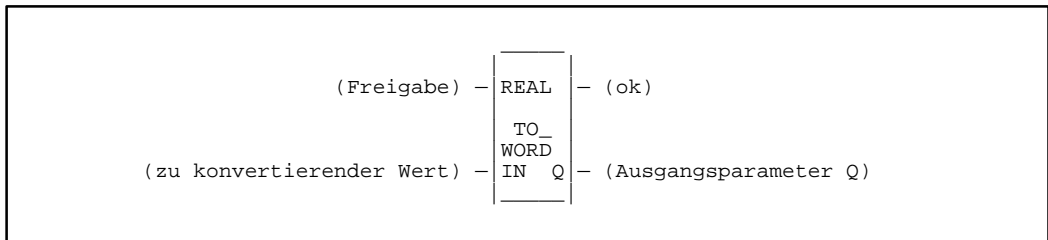
—>WORD (REAL)

Diese Funktion gibt das WORD-Äquivalent von REAL-Daten aus. Die ursprünglichen Daten werden dabei nicht verändert.

**Hinweis**

Diese Funktion ist nur in der CPU 352 verfügbar.

Erhält die Funktion Stromfluß, dann führt sie die Konvertierung durch und stellt das Ergebnis am Ausgang Q bereit. Solange die angegebene Konvertierung einen Wert innerhalb des zulässigen Bereichs (0 bis FFFFh) ergibt, schaltet die Funktion den Stromfluß immer durch, wenn sie aktiviert wird.



**Parameter:**

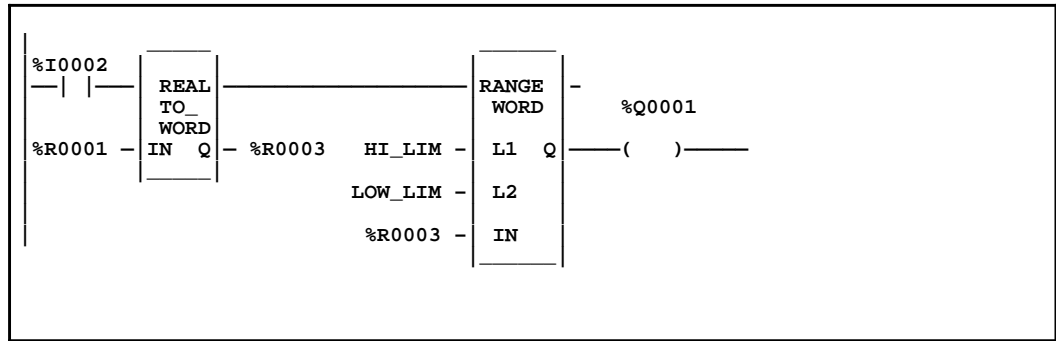
Parameter	Beschreibung
Freigabe	Die Konvertierung wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält eine Referenz für den Wert, der in das WORD-Format konvertiert werden soll.
ok	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion fehlerfrei ausgeführt wird
Q	Ausgang Q enthält die vorzeichenlose ganzzahlige Form des ursprünglichen Wertes aus IN.

**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN								•	•	•	•	
ok	•											•
Q		•	•	•	•		•	•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

**Beispiel:**



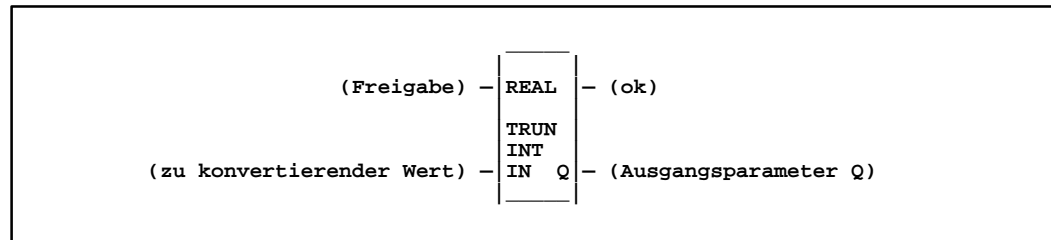
## TRUN (INT, DINT)

Mit dieser Funktion kann ein reeller Wert gegen Null hin gerundet werden. Die ursprünglichen Daten werden dabei nicht verändert. Die Ausgangsdaten können direkt als Eingangsdaten einer anderen Funktion verwendet werden.

### Hinweis

Die CPU 352 ist die einzige CPU der Series 90-30, die Gleitpunktarithmetik unterstützt. Die TRUN-Funktion hat daher für die übrigen CPUs der Series 90-30 keine Bedeutung.

Erhält die Funktion Stromfluß, dann führt sie die Konvertierung durch und stellt das Ergebnis am Ausgang Q bereit. Solange die angegebene Konvertierung einen Wert innerhalb des zulässigen Bereichs ergibt und eine Zahl ist, schaltet die Funktion den Stromfluß immer durch, wenn sie aktiviert wird.



### Parameter:

Parameter	Beschreibung
Freigabe	Die Konvertierung wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält eine Referenz für den REAL-Wert, der abgeschnitten werden soll.
ok	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion fehlerfrei ausgeführt wird, der Wert innerhalb des zulässigen Bereichs liegt und eine Zahl ist.
Q	Ausgang Q enthält den gekürzten INT- oder DINT-Wert des ursprünglichen Wertes aus IN.

### Hinweis

Bei der Wandlung von REAL nach DINT kann Genauigkeit verlorengehen, da REAL 24 signifikante Bits besitzt.

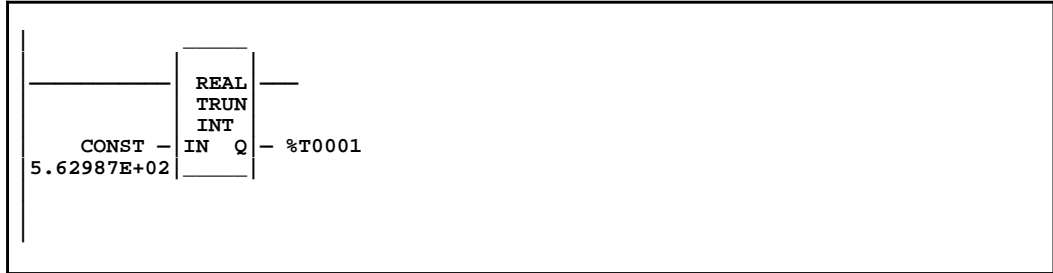
### Zulässige Speichertypen:

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
IN								•	•	•	•	
ok	•											•
Q		o	o	o	o		o	•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- o Nur für REAL\_TRUN\_INT zulässig

**Beispiel:**

Im folgenden Beispiel wird die angezeigte Konstante abgeschnitten und das ganzzahlige Ergebnis 562 wird in %T0001 eingetragen.



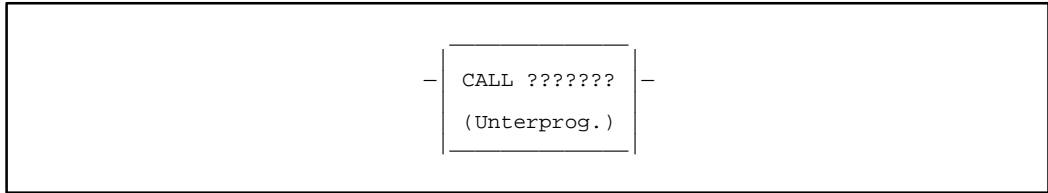
## Abschnitt 9: Steuerfunktionen

In diesem Kapitel werden die Steuerfunktionen beschrieben, mit denen die Programmausführung eingeschränkt und der von der CPU verfolgte Weg der Programmausführung abgeändert werden kann. Die Funktionsweise des CPU-Zyklus wird in Abschnitt 1 von Kapitel 1 erläutert.

Funktion	Beschreibung	Seite
CALL	Leitet die Programmbearbeitung auf einen Unterprogrammblock um.	4-107
DOIO	Aktualisiert während eines Zyklus einen bestimmten Bereich von Ein- oder Ausgängen unmittelbar. (Enthält eine DO I/O-Funktion Referenzadressen eines Moduls, werden sämtliche Ein- oder Ausgänge dieses Moduls aktualisiert, teilweise Aktualisierungen eines Moduls werden nicht durchgeführt). Wahlweise kann eine Kopie der bearbeiteten E/A im internen Speicher und nicht bei den echten Eingangspunkten abgelegt werden.	4-108
END	Bildet ein temporäres Programmende. Die Programmbearbeitung beginnt beim ersten Strompfad und endet entweder beim letzten Strompfad oder bei der END-Anweisung, was immer zuerst kommt. Diese Anweisung ist hilfreich beim Austesten eines Programms.	4-113
MCR und MCRN	Programmierung einer Hauptsteuerrelaisfunktion. Die MCR-Funktion veranlaßt, daß alle Strompfade zwischen MCR und dem nächsten END MCR ohne Stromfluß ausgeführt werden. Die Logicmaster 90-30/20/Micro Software unterstützt zwei MCR-Funktionstypen, die geschachtelte Form (MCRN) und die ungeschachtelte Form (MCR).	4-114
ENDMCR und ENDMCRN	Zeigt an, daß für die Ausführung der nachfolgenden Logik wieder Stromfluß erforderlich ist. Die Logicmaster 90-30/20/Micro Software unterstützt zwei Funktionstypen, die geschachtelte (ENDMCRN) und die ungeschachtelte Form (ENDMCR).	4-117
JUMP und JUMPN	Bewirkt, daß bei der Programmausführung zu einer angegebenen Stelle (durch eine Marke gekennzeichnet; siehe LABEL) im Programm gesprungen wird. Die Logicmaster 90-30/20/Micro Software unterstützt zwei Typen der JUMP-Funktion, die geschachtelte Form (JUMPN) und die ungeschachtelte Form (JUMP).	4-118
LABEL und LABELN	Gibt das Ziel einer JUMP-Anweisung an. Die Logicmaster 90-30/20/Micro Software unterstützt zwei Typen der LABEL-Funktion, die geschachtelte Form (LABELN) und die ungeschachtelte Form (LABEL).	4-120
COMMENT	Legt einen Kommentar (Strompfadkommentar) im Programm an. Nachdem die Anweisung programmiert wurde, kann der Text durch "Zoomen" der Anweisung eingegeben werden.	4-121
SVCREQ	Anforderung einer der folgenden SPS-Spezialdienste: <ul style="list-style-type: none"> <li>● Taskzustand und Anzahl Worte in Prüfsumme verändern/lesen</li> <li>● Echtzeituhrlesen/stellen</li> <li>● SPS abschalten</li> <li>● Fehlertabellenlöschen</li> <li>● Letzten Fehlertabelleneintrag lesen</li> <li>● Betriebszeituhrlesen</li> <li>● E/A-Override-Zustandlesen</li> <li>● Master-Prüfsumme lesen</li> <li>● E/A abfragen</li> <li>● Aufgelaufene Netzausfallzeit lesen</li> </ul>	4-122
PID	Es gibt zwei PID-Algorithmen: <ul style="list-style-type: none"> <li>● Standard-ISA-PID-Algorithmus (PIDISA).</li> <li>● Unabhängiger PID-Algorithmus (PIDIND).</li> </ul>	4-141

## CALL

Mit der CALL-Funktion verzweigen Sie die Programmausführung zu einem in der Funktion angegebenen Unterprogrammblock.



Erhält die CALL-Funktion Stromfluß, verzweigt sie den Zyklus unmittelbar zum gewünschten Unterprogrammblock, der dann bearbeitet wird. Nach Abschluß des Unterprogrammblocks kehrt die Steuerung wieder zu dem Punkt zurück, der unmittelbar nach der CALL-Anweisung steht.

### Beispiel:

Das folgende Beispiel zeigt die CALL-Anweisung, wie sie im aufrufenden Block erscheint. Wenn Sie den Cursor auf die Anweisung setzen und dann **F10** drücken, zoomen Sie die Anzeige in das Unterprogramm.



### Hinweis

Die Micro SPS enthält keine Unterprogramme. Die CALL-Funktion kann daher mit der Micro SPS nicht benutzt werden.



## DOIO

Mit der DO I/O-Funktion können Ein- und Ausgänge bei laufendem Programm für einen Zyklus aktualisiert werden. Die Funktion kann auch dazu benutzt werden, um bestimmte E/A zusätzlich zum normalen E/A-Zyklus während des Programms zu aktualisieren.

Werden Eingangsreferenzen angegeben, können durch die Funktion die neuesten Eingangswerte für das Programm erfaßt werden. Werden Ausgangsreferenzen angegeben, aktualisiert DO I/O die Ausgänge auf der Grundlage der neuesten im E/A-Speicher abgelegten Werte. Bei der E/A werden immer komplette E/A-Module aktualisiert, falls erforderlich berichtigt die SPS während der Funktionsausführung die Referenzen.

Die DO I/O-Funktion besitzt vier Ein- und einen Ausgangsparameter. Erhält die Funktion Stromfluß und sind Eingangsreferenzen angegeben, werden die Eingangspunkte zwischen der Anfangsreferenz ST und END abgefragt. Wurde für ALT eine Referenz angegeben, wird eine Kopie des neuen Eingangswertes ab dieser Referenz im Speicher abgelegt, die realen Eingangspunkte werden nicht aktualisiert. ALT muß genauso groß sein wie der abgefragte Referenztyp. Wird für ST und END eine diskrete Referenz benutzt, muß ALT ebenfalls diskret sein. Wird für ALT keine Referenz angegeben, werden die realen Eingangspunkte aktualisiert.

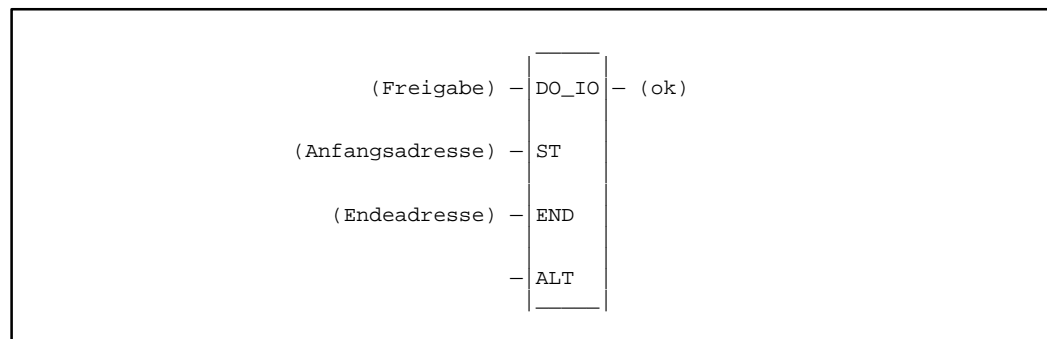
Erhält die Funktion Stromfluß und sind Ausgangsreferenzen angegeben, werden die Ausgangspunkte zwischen der Anfangsreferenz ST und END zu den Ausgangsmodulen übertragen. Sollen Ausgangswerte von einem von %Q oder %AQ verschiedenen internen Speicher an die Ausgangsmodulen gegeben werden, kann die Anfangsreferenz für ALT angegeben werden. Der Ausgabebereich, der zu den Ausgangsmodulen übertragen wird, ist durch die Anfangsreferenz ST und die Endreferenz END bestimmt.

Die Funktion wird solange ausgeführt, bis entweder alle Eingänge des gewählten Bereiches abgefragt wurden oder alle Ausgänge auf den E/A-Modulen aktualisiert wurden. Die Programmausführung kehrt dann zur nächsten Funktion nach der DO I/O-Funktion zurück.

Enthält der Referenzbereich ein Zusatzmodul (HSC, APM, usw.), werden alle Ein- (%I und %AI) bzw. Ausgangsdaten (%Q und %AQ) für dieses Modul abgefragt. Bei der Abfrage von Zusatzmodulen wird der ALT-Parameter ignoriert. Im Referenzbereich darf kein erweitertes GCM-Modul liegen.

Die Funktion gibt Stromfluß nach rechts weiter, wenn sie Stromfluß erhält, mit Ausnahme von:

- Nicht alle Referenzen des angegebenen Typs sind im gewählten Bereich vorhanden.
- Die CPU kann die von der Funktion angelegte temporäre E/A-Liste nicht ordnungsgemäß verarbeiten.
- Im Bereich befinden sich E/A-Module, die mit einem Fehler "E/A-Verlust" verknüpft sind.



**Parameter:**

Parameter	Beschreibung
Freigabe	Bei Funktionsfreigabe wird ein begrenzter Ein- oder Ausgabezyklus durchgeführt.
ST	ST ist die Anfangsadresse eines Satzes von Ein- oder Ausgangspunkten oder Worten, die bedient werden sollen.
END	END ist die Endadresse eines Satzes von Ein- oder Ausgangspunkten oder Worten, die bedient werden sollen.
ALT	Beim Eingabezyklus gibt ALT die Adresse an, in der die abgefragten Eingangspunkte/Worte gespeichert werden sollen. Beim Ausgabezyklus gibt ALT die Adresse an, von der die Ausgangspunkte/Worte abgeholt werden können, die zu den E/A-Modulen geschickt werden. Bei CPUs ab Modell 331 kann der ALT-Parameter eine Auswirkung auf die Ausführungsgeschwindigkeit des DOIO-Funktionsblockes haben (siehe nachstehenden Hinweis und Abschnitt über die erweiterte DOIO-Funktion für CPUs ab Modell 331 auf Seite 4-112).
ok	Der OK-Ausgang wird durchgeschaltet, wenn die Ein- oder Ausgangszyklen normal ausgeführt wurden.

**Hinweis**

Bei CPUs ab Modell 331 aufwärts kann der ALT-Parameter der DOIO-Funktion dazu benutzt werden, den Steckplatz eines einzelnen Moduls im Hauptchassis einzugeben. Wird dies gemacht, wird der DOIO-Funktionsblock in 80 Mikrosekunden bearbeitet (anstatt 236 Mikrosekunden, wenn der Block ohne den ALT-Parameter programmiert wurde). Es findet keine Überprüfung zur Verhinderung von Adreßkonflikten oder Moduldiskrepanzen statt.

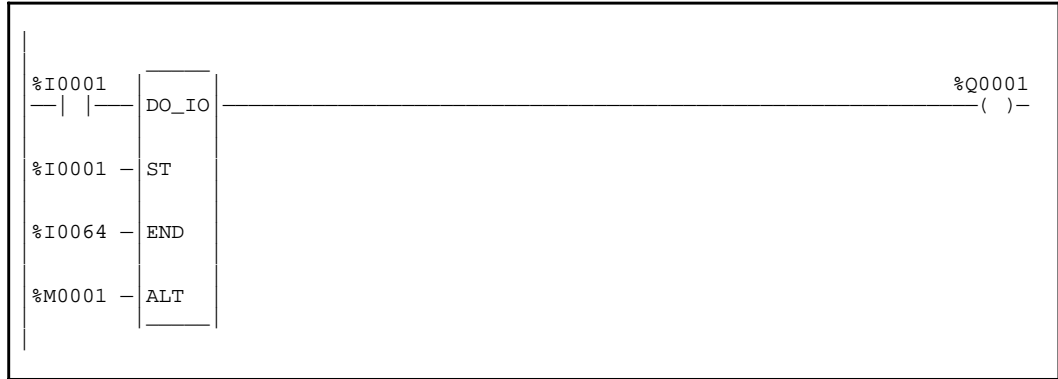
**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
ST		•	•						•	•		
END		•	•						•	•		
ALT		•	•	•	•		•	•	•	•		•
ok	•											•

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

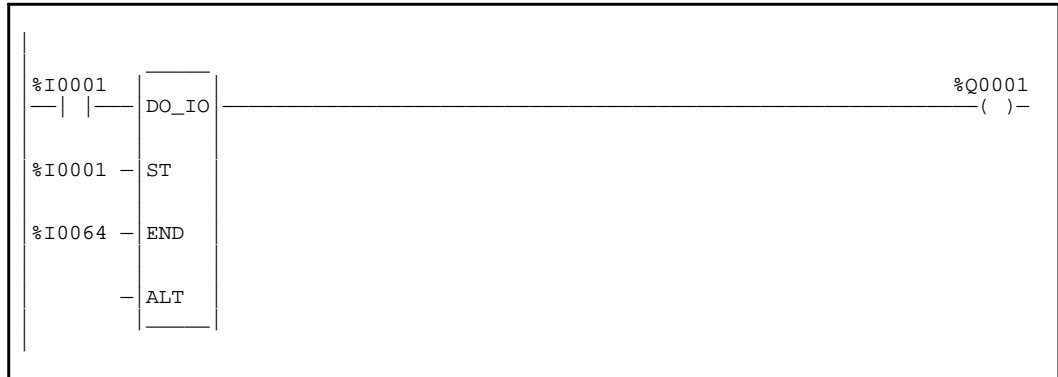
### Eingabebeispiel 1:

Wird im folgenden Beispiel der Freigabeeingang %I0001 aktiviert, dann werden die Referenzen %I0001 bis %I0064 abgefragt und %Q0001 wird durchgeschaltet. Eine Kopie der abgefragten Eingänge wird im internen Speicher ab Referenz %M0001 bis Referenz %M0064 abgelegt. Die echten Eingangspunkte werden nicht aktualisiert. Mit dieser Funktionsart können die aktuellen Werte der Eingangspunkte mit den Werten der Eingangspunkte am Anfang des Zyklus verglichen werden.



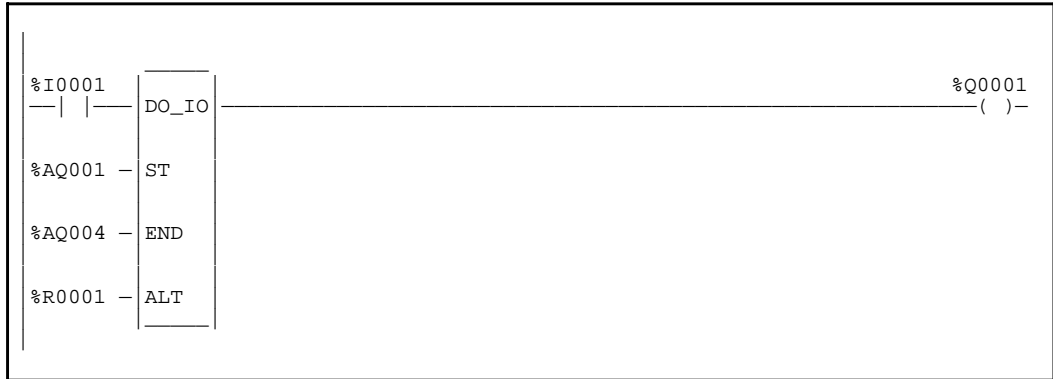
### Eingabebeispiel 2:

Wird im folgenden Beispiel der Freigabeeingang %I0001 aktiviert, dann werden die Referenzen %I0001 bis %I0064 abgefragt und %Q0001 wird durchgeschaltet. Die abgefragten Eingänge werden im Eingangszustandsspeicher ab Referenz %M0001 bis Referenz %M0064 abgelegt. Mit dieser Funktionsart können Eingangspunkte während des Programmausführungsteils eines CPU-Zyklus mehrmals abgefragt werden.



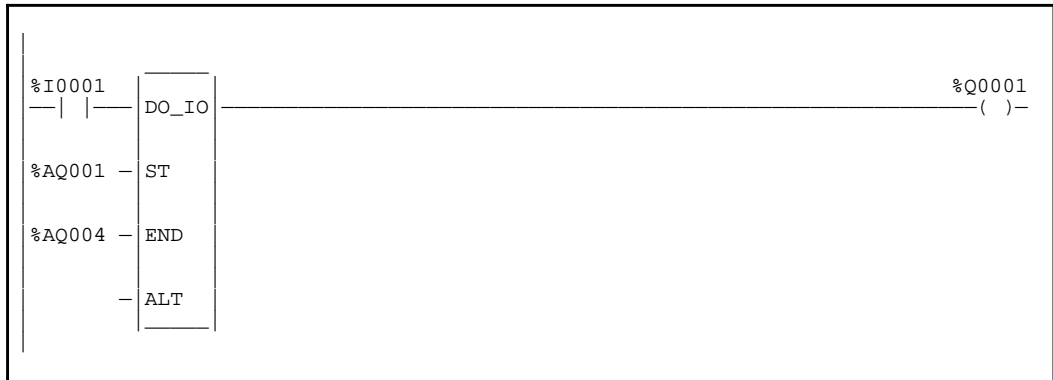
### Ausgabebeispiel 1:

Wird im folgenden Beispiel der Freigabeeingang %I0001 aktiviert, dann werden die Referenzen %R0001 bis %R0004 in die analogen Ausgangskanäle %AQ001 bis %AQ004 eingetragen und %Q0001 wird durchgeschaltet. Die Werte aus %AQ001 bis %AQ004 werden nicht zu den analogen Ausgangsmodulen übertragen.



### Ausgabebeispiel 2:

Wird im folgenden Beispiel der Freigabeeingang %I0001 aktiviert, dann werden die Werte aus den Referenzen %AQ001 bis %AQ004 in die analogen Ausgangskanäle %AQ001 bis %AQ004 eingetragen und %Q0001 wird durchgeschaltet.



## Erweiterte DOIO-Funktion für CPUs ab Modell 331 aufwärts

### Achtung

**Wird die erweiterte DOIO-Funktion in einem Programm verwendet, darf dieses Programm nicht mit einer Version der Logicmaster 90-30/20 Software geladen werden, die älter als 4.01 ist.**

Die erweiterte Version der DOIO-Funktion (E/A-Aktualisierung) ist verfügbar für CPUs ab Modell 331 und Ausgabestände ab 4.20. Die erweiterte Version der DOIO-Funktion kann nur auf ein einziges diskretes Ein- oder Ausgangsmodul mit 8, 16 oder 32 Punkten angewandt werden.

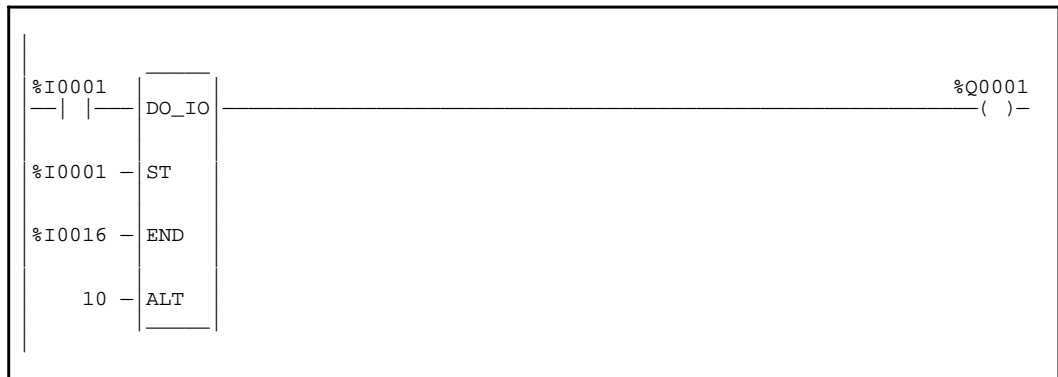
Mit dem ALT-Parameter wird der Steckplatz des Moduls im Hauptchassis identifiziert. Eine Konstante 2 in diesem Parameter gibt der CPU zum Beispiel an, daß die erweiterte Version des DOIO-Funktionsblocks auf das in Steckplatz 2 eingebaute Modul angewandt werden soll.

### Hinweis

Die erweiterte DOIO-Funktion überprüft nur, ob das Modul in dem angegebenen Steckplatz funktionstüchtig ist.

Die erweiterte DOIO-Funktion kann nur für Module im Hauptchassis verwendet werden. Der Wert des ALT-Parameters muß daher zwischen 2 und 5 (Chassis mit 5 Steckplätzen) bzw. 2 und 10 (Chassis mit 10 Steckplätzen) liegen.

Anfangs- und Endreferenzen müssen vom Typ %I oder %Q sein. Diese Referenzen geben die erste und letzte Referenz an, für die das Modul konfiguriert wurde. Wurde zum Beispiel ein Eingangsmodul mit 16 Punkten für %I0001 bis %I0016 in Steckplatz 10 eines Chassis mit 10 Steckplätzen konfiguriert, muß der ST-Parameter %I0001 sein, der END- Parameter %I0016 und de ALT-Parameter 10 (siehe nachstehende Abbildung).



Die Tabelle vergleicht die Ausführungszeiten eines normalen DOIO-Funktionsblocks für ein E/A-Modul mit 8, 16 oder 32 Punkten mit denen eines erweiterten DOIO-Funktionsblocks.

Modul	Ausführungszeit normale DOIO	Ausführungszeit erweiterte DOIO
Diskretes Eingangsmodul mit 8 Punkten	224 Mikrosekunden	67 Mikrosekunden
Diskretes Ausgangsmodul mit 8 Punkten	208 Mikrosekunden	48 Mikrosekunden
Diskretes Eingangsmodul mit 16 Punkten	224 Mikrosekunden	68 Mikrosekunden
Diskretes Ausgangsmodul mit 16 Punkten	211 Mikrosekunden	47 Mikrosekunden
Diskretes Eingangsmodul mit 32 Punkten	247 Mikrosekunden	91 Mikrosekunden
Diskretes Ausgangsmodul mit 32 Punkten	226 Mikrosekunden	50 Mikrosekunden

## END

Die END-Funktion ermöglicht ein temporäres Programmende. Die Programmbearbeitung beginnt beim ersten Strompfad und endet entweder beim letzten Strompfad oder bei der END-Anweisung, was immer zuerst kommt.

Die END-Funktion beendet die Programmausführung unbedingt. Nach der END-Funktion darf im Strompfad nichts mehr stehen. Nach der END-Funktion wird keine Logik mehr bearbeitet und die Programmsteuerung wird für den nächsten Zyklus an den Programmanfang übergeben.

Die END-Funktion ist hilfreich beim Austesten, da sie die Ausführung der nach ihr stehenden Logik verhindert.

Bei der Logicmaster-Programmiersoftware zeigt eine Marke [ END OF PROGRAM LOGIC ] (Programmende) das Ende der Programmausführung an. Diese Marke wird verwendet, wenn im Programm keine END-Funktion programmiert wurde.

```
—[ END ]
```

### Beispiel:

Im folgenden Beispiel wurde ein END programmiert, um den aktuellen Zyklus zu beenden.

```
STOP  
—[ END ]
```

### Hinweis

Bei CPUs ab Ausgabestand 7 entsteht ein Fehler "END Function Executed from SFC Action" [Ende-Funktion aus SFC-Aktion ausgeführt], wenn Sie eine END-Funktion in SFC oder in ein von SFC aufgerufenes Programm platzieren. Bei CPUs bis zum Ausgabestand 7 gabe es hierbei Probleme im Ablauf, aber keine Fehlermeldung. Weitere Angaben zu diesem Fehler finden Sie unter "Diskrepanz bei Systemkonfiguration" in Abschnitt 2 von Kapitel 3.

## MCR

Sämtliche Strompfade zwischen der Hauptsteuerrelaisfunktion MCR und dem zugehörigen ENDMCR werden mit negativer Logik ausgeführt. Mit der MCR-Funktion kann ein Teil der Programmlogik übergangen werden. Nach der zu MCR gehörenden Funktion ENDMCR wird die normale Programmausführung wieder aufgenommen. Im Gegensatz zu der JUMP-Funktion können MCR-Funktionen nur in Vorwärtsrichtung verwendet werden. Die Anweisung ENDMCR muß in der Programmabfolge nach der MCR-Funktion auftreten.

Logicmaster 90-30/20/Micro unterstützt zwei Formen der MCR-Funktion, eine geschachtelte und eine ungeschachtelte Form. Die ungeschachtelte Form (MCR) ist seit Ausgabestand 1 der Software verfügbar.

### Hinweis

CPU-Modelle ab 351 kennen keine ungeschachtelte Form (MCR). Mit diesen CPUs 351 und 352 dürfen Sie nur die geschachtelte Form (MCRN) verwenden.

Zu jeder ENDMCR-Anweisung kann nur eine MCR-Anweisung gehören. Der Bereich einer ungeschachtelten MCR/ENDMCR-Funktion darf sich nicht mit dem Bereich eines anderen MCR/ENDMCR- oder JUMP/LABEL-Paares überschneiden. Eine ungeschachtelte MCR-Funktion darf nicht im Wirkungsbereich eines anderen MCR/ENDMCR- oder JUMP/LABEL-Paares liegen. Darüberhinaus darf ein JUMP/LABEL-Paar oder ein MCR/ENDMCR-Paar nicht im Wirkungsbereich eines MCR/ENDMCR-Paares liegen.

### Hinweis

Die ungeschachtelte MCR-Funktion ist die einzige Hauptsteuerrelaisfunktion, die in Ausgabe 1 der SPS Series 90-30 verwendet werden kann. Die geschachtelte Version sollte für alle neuen Anwendungen benutzt werden.

Die geschachtelte Form der MCR-Funktion trägt die Bezeichnung MCRN und ist ab Ausgabestand 2 der SPS Series 90-30 verfügbar. Eine MCRN-Funktion kann im Rahmen der entsprechenden Regeln in anderen MCRN-Funktionen geschachtelt sein. Eine MCRN-Anweisung und die zugehörige ENDMCRN-Anweisung müssen vollständig innerhalb eines anderen MCRN/ENDMCRN-Paares liegen.

Solange eine MCRN-Funktion im Verhältnis zu anderen MCRN-Funktionen ordnungsgemäß geschachtelt wurde und nicht im Wirkungsbereich ungeschachtelter MCR- und JUMP-Funktionen liegt, kann sie an beliebiger Stelle im Programm angeordnet sein.

### Hinweis

Bei CPUs 351 dürfen Sie nur eine MCRN für jede ENDMCRN verwenden.

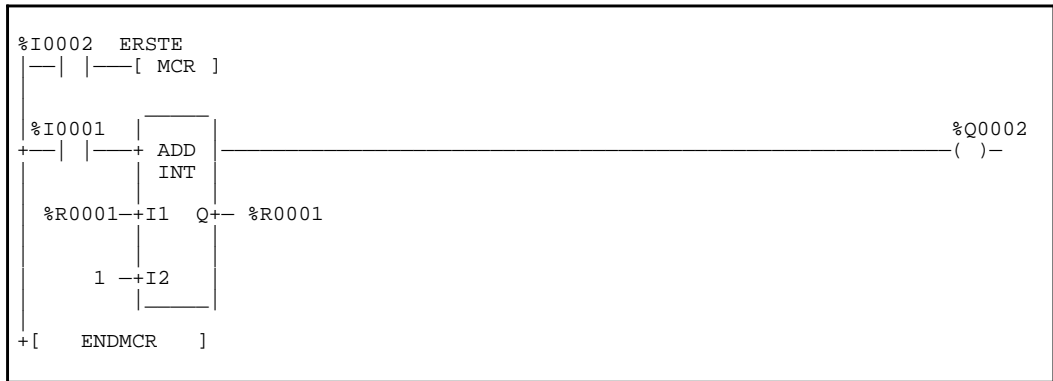
Zu einer ENDMCRN-Anweisung können mehrere MCRN-Funktionen gehören (mit der oben erwähnten Ausnahme der CPUs 351 und 352). Diese Struktur entspricht den geschachtelten JUMP-Anweisungen, bei denen mehrere JUMP dem gleichen LABEL zugeordnet sein können. Die Unterschiede zwischen JUMP-Funktion und MCR-Funktion werden im Abschnitt "Unterschiede zwischen MCR und JUMP" auf Seite 4-115 beschrieben.

Beide Formen der MCR-Funktion besitzen die gleichen Parameter. Sie besitzen beide einen Booleschen Freigabeeingang (EN) sowie eine Bezeichnung, über die sie identifiziert werden. Diese Bezeichnung wird am Ende mit der ENDMCR-Anweisung wiederholt. Weder MCR noch MCRN besitzen Ausgänge, daher ist nach einer MCR-Funktion keine weitere Funktion im Strompfad mehr möglich.

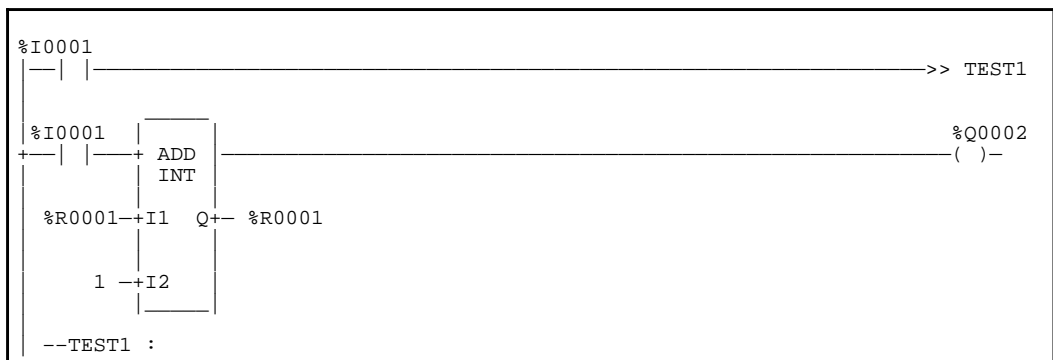


### Unterschiede zwischen MCR und JUMP

Bei der MCR-Funktion werden Funktionsblöcke im Wirkungsbereich der MCR-Funktion ohne Stromfluß ausgeführt und Spulen werden abgeschaltet. Ist im nachstehenden Beispiel %I0002 EIN, wird die MCR-Funktion freigegeben. Bei freigegebener MCR-Funktion wird, selbst wenn %I0001 EIN ist, der ADD-Funktionsblock ausgeführt, ohne daß Strom fließt (d.h. er addiert keine 1 zu %R0001), und %Q0001 wird abgeschaltet.



Bei der JUMP-Funktion werden die Funktionsblöcke zwischen JUMP und LABEL nicht ausgeführt und Spulen sind nicht betroffen. Im nachstehenden Beispiel wird der Sprung ausgeführt, wenn %I0002 EIN ist. Da der Programmteil zwischen JUMP und LABEL übersprungen wird, wird %Q0001 nicht beeinflusst (d.h. er bleibt EIN, wenn er zuvor EIN war, bzw. AUS, wenn er AUS war).





**Beispiel:**

Wird bei dem folgenden Beispiel %I0002 aktiviert und damit der Stromfluß zur MCR-Funktion durchgeschaltet, wird die Programmausführung bis zur zugeordneten Funktion ENDMCR mit negativem Stromfluß fortgesetzt. Sind %I0001 und %I0003 EIN, geht %Q0001 AUS und %Q0003 bleibt EIN.



# ENDMCR

Mit ENDMCR wird nach einer MCR-Funktion wieder der normale Programmablauf aufgenommen. Nach einer aktiven MCR-Funktion veranlaßt die zugehörige ENDMCR-Funktion, daß das Programm wieder mit normalem Stromfluß fortgesetzt wird. War die zugehörige MCR-Funktion nicht aktiv, dann hat die ENDMCR-Funktion keine Wirkung.

Logicmaster 90-30/20/Micro unterstützt zwei Formen der ENDMCR-Funktion, eine geschachtelte und eine ungeschachtelte Form. Die ungeschachtelte Form (ENDMCR) muß mit der ungeschachtelten MCR-Funktion (MCR) verwendet werden. Die geschachtelte Form (ENDMCRN) muß mit der geschachtelten MCR-Funktion (MCRN) verwendet werden.

Die ENDMCR-Funktion besitzt einen negierten Booleschen Eingang (EN). Die Funktionsfreigabe muß von der Stromschiene kommen, die Ausführung kann nicht bedingt erfolgen. Die ENDMCR-Funktion besitzt auch eine Bezeichnung, mit der sie identifiziert und der (den) entsprechenden MCR(s) zugeordnet wird. Sie besitzt keine Ausgänge, daher ist nach einer ENDMCR-Funktion keine weitere Funktion im Strompfad mehr möglich.

```
???????          ???????
-[ ENDMCR      ] oder  -[ ENDMCRN   ]
```

### Beispiel:

Die ENDMCR-Funktion im folgenden Beispiel beendet den Bereich "CLEAR" der MCR-Funktion.

Beispiel einer ungeschachtelten ENDMCR-Funktion:

```
| CLEAR
| -[ ENDMCR      ]
```

Beispiel einer geschachtelten ENDMCR-Funktion:

```
| CLEAR
| -[ ENDMCRN    ]
```

## JUMP

Mit der JUMP-Funktion kann ein Teil des Programms übersprungen werden. Die Programmausführung wird an der durch LABEL markierten Stelle fortgesetzt. Ist JUMP aktiv, werden alle Spulen innerhalb des Wirkungsbereiches der Funktion eingefroren. Hierzu gehören auch Merker, die mit Timern, Zählern, Haftrelais und Relais verknüpft sind.

Logicmaster 90-30/20/Micro unterstützt zwei Formen der JUMP-Funktion, eine geschachtelte und eine ungeschachtelte Form. Die ungeschachtelte Form, die seit Software-Ausgabestand 1 verfügbar ist, hat die Form  $\text{---}\rightarrow\text{ LABEL01}$ , wobei LABEL01 die Bezeichnung der entsprechenden ungeschachtelten LABEL-Anweisung ist.

Bei ungeschachtelten JUMP-Funktionen kann immer nur eine JUMP-Anweisung einer LABEL-Anweisung zugeordnet werden. Der Sprung kann entweder nach vorne oder nach hinten ausgeführt werden.

Der Bereich einer ungeschachtelten JUMP/LABEL-Funktion darf sich nicht mit dem Bereich eines anderen MCR/ENDMCR- oder JUMP/LABEL-Paares überschneiden. Eine ungeschachtelte JUMP-Funktion sowie die zugehörige LABEL-Funktion dürfen nicht im Wirkungsbereich eines anderen MCR/ENDMCR- oder JUMP/LABEL-Paares liegen. Darüberhinaus darf ein MCR/ENDMCR-Paar oder ein anderes JUMP/LABEL-Paar nicht im Wirkungsbereich eines ungeschachtelten JUMP/LABEL-Paares liegen.

### Hinweis

Die ungeschachtelte JUMP-Funktion ist die einzige Sprungfunktion, die in Ausgabe 1 der SPS Series 90-30 verwendet werden kann. Die geschachtelte JUMP-Funktion kann (und sollte) für alle neuen Anwendungen verwendet werden.

Beachten Sie auch, daß die 351 und die höheren CPUs nur geschachtelte Sprünge unterstützt, keine ungeschachtelten Sprünge.

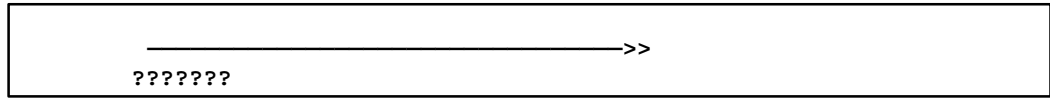
Die geschachtelte Form der JUMP-Anweisung hat die Form  $\text{---N---}\gg\text{ LABEL01}$ , wobei LABEL01 die Bezeichnung der entsprechenden ungeschachtelten LABEL-Anweisung ist. Die geschachtelte Form der JUMP-Funktion ist ab Ausgabestand 2 der Logicmaster 90-30/20/Micro Software und SPS-Firmware verfügbar.

Solange eine geschachtelte JUMP-Anweisung nicht im Wirkungsbereich einer ungeschachtelten MCR- oder JUMP-Funktion liegt, darf sie an beliebiger Stelle im Programm angelegt werden.

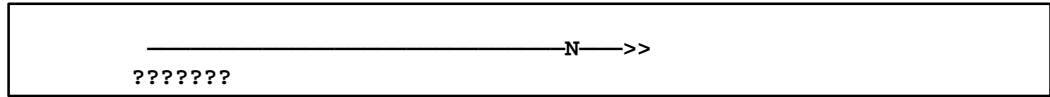
Zu einer geschachtelten LABEL-Anweisung können mehrere geschachtelten JUMP-Funktionen gehören. Geschachtelte JUMP-Funktionen können vorwärts oder rückwärts ausgeführt werden.

Beide JUMP-Funktionen werden immer in die Spalten 9 und 10 der aktuellen Strompfadzeile eingetragen. Nach einer JUMP-Funktion ist keine weitere Funktion im Strompfad mehr möglich. Der Stromfluß erfolgt direkt von der Anweisung zu dem Strompfad mit der angegebenen Bezeichnung.

Ungeschachtelte JUMP-Funktion:



Geschachtelte JUMP-Funktion:



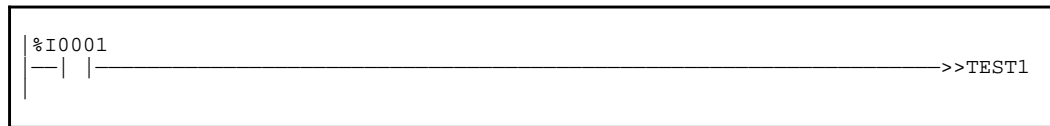
**Achtung**

**Um Endlosschleifen zu vermeiden, muß ein Rückwärtssprung immer eine Bedingung enthalten.**

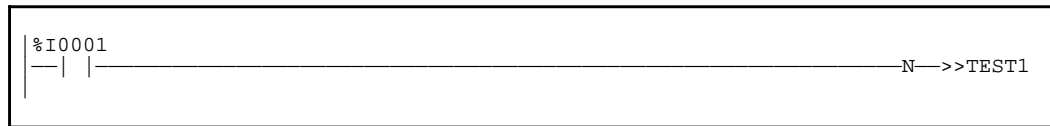
**Beispiel:**

In den folgenden Beispielen wird der Stromfluß immer dann zu der LABEL-Funktion TEST1 weitergeschaltet, wenn die JUMP-Funktion TEST1 aktiv wird.

Beispiel einer ungeschachtelten JUMP-Funktion:



Beispiel einer geschachtelten JUMP-Funktion:



## LABEL

Die LABEL-Funktion bildet das Ziel einer JUMP-Funktion. Mit ihr wird die normale Programmbearbeitung nach einer JUMP-Funktion wieder aufgenommen.

Innerhalb eines Programms darf eine bestimmte LABEL-Bezeichnung nur einmal vorkommen. Programme ohne vollständige JUMP/LABEL-Paare können zwar erstellt und in der SPS gespeichert werden, sind aber nicht ablauffähig.

Logicmaster 90-30/20/Micro unterstützt zwei Formen der LABEL-Funktion, eine geschachtelte und eine ungeschachtelte Form. Die ungeschachtelte Form, LABEL01:, muß mit der ungeschachtelten JUMP-Funktion, ——>> LABEL01, verwendet werden. Die geschachtelte Form, LABEL01:(geschachtelt), muß mit der geschachtelten JUMP-Funktion, —N——>> LABEL01, verwendet werden.

Die LABEL-Funktion, die nur für sich allein in einem Strompfad stehen darf, besitzt weder Ein- noch Ausgänge.

Ungeschachtelte LABEL-Funktion:

```
???????:
```

Geschachtelte LABEL-Funktion:

```
???????:(nested)
```

### Beispiel:

In den folgenden Beispielen wird der Stromfluß von der JUMP-Funktion TEST1 bei der LABEL-Funktion TEST1 fortgesetzt.

Beispiel einer ungeschachtelten LABEL-Funktion:

```
| TEST1 :
```

Beispiel einer geschachtelten LABEL-Funktion:

```
| TEST1 :(nested)
```

## COMMENT

Mit der COMMENT-Funktion können Erläuterungen (Strompfadkommentare) in ein Programm eingefügt werden. Ein Kommentar kann dabei bis zu 2048 Zeichen umfassen. Die entsprechende Darstellung im Kontaktplanprogramm ist:

```
(* COMMENT *)
```

Der Text kann gelesen oder editiert werden, indem der Cursor auf (\* COMMENT \*) gesetzt und **F10** (Zoom) gedrückt wird. Kommentartexte können auch ausgedruckt werden.

Bei längeren Kommentaren kann der Zusatztext auch in einer anderen Datei abgelegt werden. Hierzu müssen Sie folgendermaßen vorgehen:

1. Erstellen Sie den Kommentar:
  - A. Geben Sie Text ein bis zu dem Punkt, an dem der Text von der anderen Datei anfangen soll.
  - B. Setzen Sie den Cursor auf den Anfang einer neuen Zeile und geben dann **\I** oder **\i** ein, den Laufwerk-Kennbuchstaben gefolgt von einem Doppelpunkt, das Unterverzeichnis bzw. den Ordner und den Dateinamen. Zum Beispiel:

```
\I d:\text\commnt1
```

Die Laufwerkennung kann weggelassen werden, wenn Datei und Programmordner auf dem gleichen Laufwerk liegen.

- C. Fahren Sie mit der Programmbearbeitung fort oder gehen Sie zu MS-DOS.
2. Nach Verlassen der Programmiersoftware erstellen Sie den gewünschten Text mit einer MS-DOS-kompatiblen Textverarbeitungssoftware. Weisen Sie der Textdatei den im Kommentar eingetragenen Namen zu und speichern Sie ihn auf der im Kommentar angegebenen Platte ab.

## SVCREQ

Mit der SVCREQ-Funktion können Sie einen der folgenden SPS-Spezialdienste anfordern:

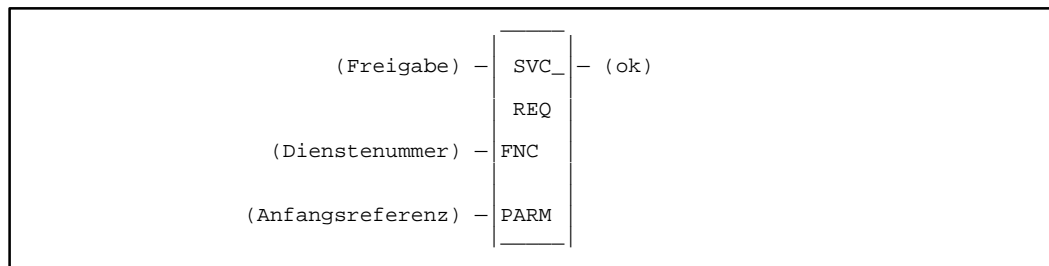
**Tabelle 4-3 SVCREQ-Funktion**

Funktion	Beschreibung
6	Taskzustand und Anzahl Worte in Prüfsumme verändern/lesen
7	Echtzeituhrlesen/stellen
13	SPS abschalten
14	Fehlertabellenlöschen
15	Letzten Fehlertableneintrag lesen
16	Betriebszeituhrlesen
18	E/A-Override-Zustandlesen
23	Master-Prüfsumme lesen
26/30	E/A abfragen
29	Aufgelaufene Netzausfallzeit lesen

Die SVCREQ-Funktion besitzt drei Ein- und einen Ausgangsparameter. Erhält die Funktion Stromfluß, dann wird die SPS aufgefordert, die angegebenen Funktion (FNC) auszuführen. Die Parameter für die Funktion beginnen bei der für PARM angegebenen Referenz. Die SVCREQ-Funktion gibt Stromfluß weiter, sofern nicht eine falsche Funktionsnummer, falsche Parameter oder außerhalb des Bereichs liegende Referenzen angegeben wurden. Weitere Fehlerursachen werden auf den nächsten Seiten beschrieben.

Die für PARM angegebene Referenz muß in einem Wortspeicher (%R, %AI, oder %AQ) liegen. Sie ist die erste einer Gruppe, die den Parameterblock der Funktion bilden. In aufeinanderfolgenden 16-Bit-Speicherplätzen werden weitere Parameter abgelegt. Die Gesamtzahl der benötigten Referenzen hängt von dem eingesetzten SVCREQ-Funktionsstyp ab.

Parameterblöcke können sowohl als Eingänge für die Funktion als auch als die Stelle, an der die Daten nach Funktionsausführung abgelegt werden, verwendet werden. Auf die von der Funktion ausgegebenen Daten kann daher auch an der durch PARM angegebenen Stelle zugegriffen werden.



**Parameter:**

Parameter	Beschreibung
Freigabe	Der gewünschte Bedienaufruf wird ausgeführt, wenn der Freigabeeingang aktiviert wird.
FNC	In FNC steht die Konstante oder Referenz für den gewünschten Dienst.
PARM	PARM enthält die Anfangsreferenz des Parameterblocks für den gewünschten Dienst.
ok	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion normal ausgeführt wird.

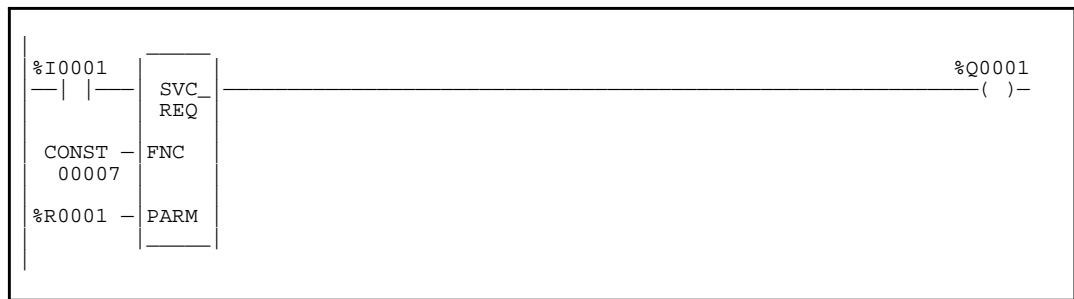
**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
FNC		•	•	•	•		•	•	•	•	•	
PARM		•	•	•	•		•	•	•	•		
ok	•											•

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

**Beispiel:**

Wird im folgenden Beispiel der Freigabeeingang %I0001 aktiviert, dann wird die SVCREQ-Funktion Nummer 7 mit dem bei %R0001 beginnenden Parameterblock aufgerufen. Die Ausgangsspule %Q0001 wird durchgeschaltet, wenn die Operation erfolgreich abläuft.





## SVCREQ #6: Anzahl Worte in Prüfsumme ändern/lesen

Mit der SVCREQ-Funktion mit der Funktionsnummer 6 können Sie

- die aktuelle Anzahl Worte lesen;
- eine neue Wortanzahl einstellen.

Sofern bei der angeforderten Operation (siehe unten) keine von 0 oder 1 verschiedene Zahl eingegeben wird, wird die Operation erfolgreich ausgeführt.

Für die Prüfsummenfunktion hat der Parameterblock eine Länge von 2 Worten.

### Aktuelle Anzahl Worte lesen:

Geben Sie SVCREQ-Funktion 6 mit dem nachstehenden Parameterblock ein:

0	Adresse
ignoriert	Adresse + 1

Nachdem die Funktion ausgeführt wurde, gibt sie die aktuelle Prüfsumme im zweiten Wort des Parameterblocks aus. Für die Lesefunktion wird kein Bereich angegeben, der zurückgegebene Wert gibt an, wieviel Worte derzeit durch die Prüfsummenfunktion erfaßt werden.

0	Adresse
aktuelle Anzahl Worte	Adresse + 1

### Neue Wortanzahl einstellen:

Geben Sie SVCREQ-Funktion 6 mit dem nachstehenden Parameterblock ein:

1	Adresse
neue Anzahl Worte	Adresse + 1

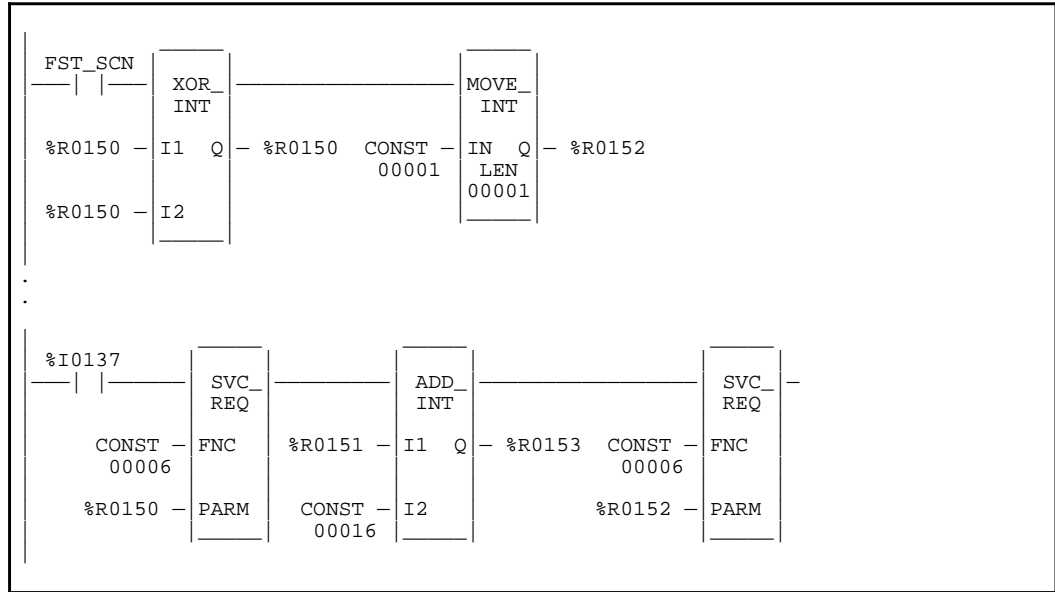
Durch Eingabe einer 1 veranlassen Sie, daß die SPS die Anzahl der durch die Prüfsummenfunktion erfaßten Wörter auf den im zweiten Wort des Parameterblocks eingegebenen Wert einstellt. Bei CPU 331 oder 311 kann dieser Wert 0 oder 8 sein, bei der CPU 211 ist 0 oder 4 möglich.

### Hinweis

Diese SVCREQ-Funktion ist bei der Micro-SPS nicht verfügbar.

**Beispiel:**

Wird im nachstehenden Beispiel der Freigabekontakt FST\_SCN gesetzt, werden die Parameterblöcke für die Prüfsummenfunktion aufgebaut. Später im Programm, wenn der Eingang %I0137 aktiviert wird, wird die Anzahl der in der Prüfsumme berücksichtigten Wörter aus dem SPS-Betriebssystem gelesen. Diese Anzahl wird um 16 erhöht; die Ergebnisse der Funktion ADD\_INT werden dabei in den Parameter "neuer Wert für Einstellung" eingetragen. Der zweite Bedienanforderungsblock fordert die SPS auf, den neuen Wert einzustellen.



Die Parameterblöcke im Beispiel liegen bei Adresse %R0150 und besitzen folgenden Inhalt:

0 = aktuellen Wert lesen	%R0150
aktuellen Wert halten	%R0151
1 = aktuellen Werteinstellen	%R0152
neuer Wert für Einstellung	%R0153

## SVCREQ #7: Echtzeituhr lesen/stellen

Mit der SVCREQ-Funktion #7 können Sie die Echtzeituhr in der SPS lesen oder einstellen.

### Hinweis

Diese Funktion ist nur in der CPU Modell 331 implementiert.

Die Ausführung ist erfolgreich, mit Ausnahme von:

1. Eine von 0 oder 1 verschiedene Zahl wurde als Operationskennung eingegeben (siehe unten).
2. Es wurde ein unzulässiges Datenformat angegeben.
3. Die angebotenen Daten sind nicht im erwarteten Format.

Bei den Datum- und Zeitfunktionen hängt die Länge des Parameterblocks vom Datenformat ab. BCD-Format benötigt 6 Worte, verdichtetes ASCII-Format 12 Worte.

0 = Zeit und Datum lesen	Adresse
1 = Zeit und Datum einstellen	Adresse + 1
1 = BCD-Format	Adresse + 1
3 = verdichtetes ASCII-Format	Adresse + 1
Daten	Adresse + 2 bis Ende

In Wort 1 wird festgelegt, ob die Funktion die Werte lesen oder verändern soll.

0	=	<b>lesen</b>
1	=	<b>verändern</b>

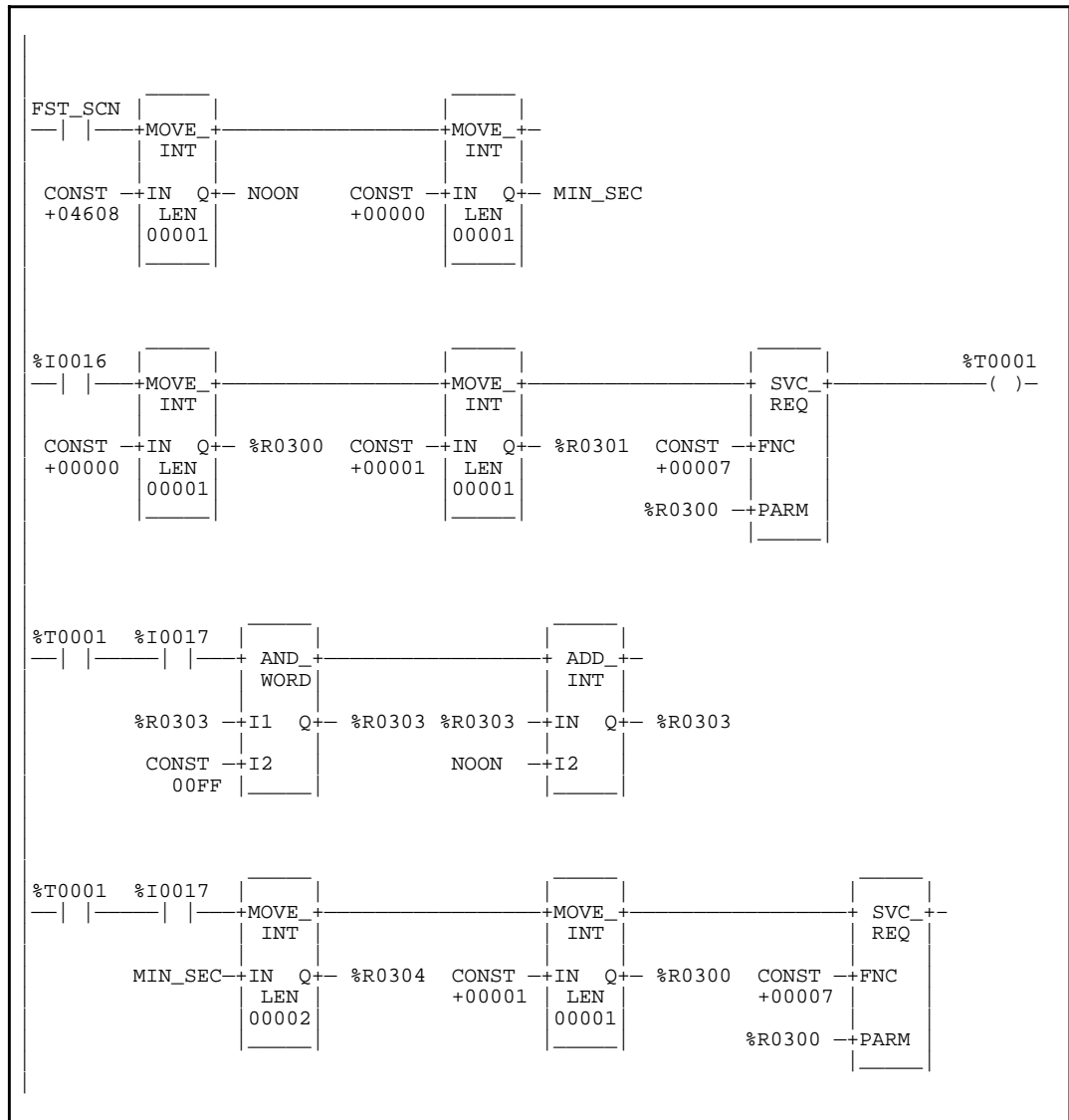
In Wort 2 wird das Datenformat angegeben:

1	=	<b>BCD</b>
3	=	<b>verdichtetes ASCII-Format mit eingebetteten Leerzeichen und Doppelpunkten</b>

Ab Wort 3 bis zum Ende des Parameterblocks werden die von der Lesefunktion ausgegebenen Daten oder neue, von der Veränderungsfunktion gelieferte Daten eingetragen. In beiden Fällen ist das Format dieser Datenworte gleich. Werden Datum und Uhrzeit gelesen, dann werden die Worte (Adresse+2) bis (Adresse+8) des Parameterblocks bei der Eingabe ignoriert.

**Beispiel:**

Im folgenden Beispiel wird beim Aufruf von der vorhergehenden Logik ein Parameterblock für die Echtzeituhr aufgebaut, der zunächst die aktuellen Werte von Datum und Uhrzeit ausliest und dann die Uhr im BCD-Format auf 12 Uhr Mittags einstellt. Der Parameterblock liegt an der Globaladresse %R0300. Das Feld NOON wurde an beliebiger Stelle im Programm eingerichtet und enthält die Werte 12, 0 und 0 (das Feld NOON muß auch die Daten bei %R0300 enthalten). Im BCD-Format werden sechs aufeinanderfolgende Speicheradressen für den Parameterblock benötigt.



### Inhalt des Parameterblocks

Auf den folgenden Seiten wird der Inhalt des Parameterblocks für die verschiedenen Datenformate dargestellt. Bei beiden Datenformaten

- wird die Zeit im 24-Stunden-Format gespeichert;
- sind die Wochentage numerische Werte:

Wert	Wochentag
1	Sonntag
2	Montag
3	Dienstag
4	Mittwoch
5	Donnerstag
6	Freitag
7	Samstag

### Datum und Zeit im BCD-Format lesen/einstellen:

Im BCD-Format belegt jedes Zeit- und Datumselement ein einzelnes Byte. Dieses Format benötigt sechs Worte. Das letzte Byte des sechsten Wortes wird dabei nicht verwendet. Werden Datum und Zeit eingestellt, wird dieses Byte ignoriert; werden Datum und Zeit gelesen, gibt diese Funktion ein Nullzeichen zurück (00).

oberes Byte		unteres Byte		Beispiel Ausgangsparameterblock: Datum und Zeit im BCD-Format lesen (Montag 3. Juli 1988, 14:45:30)	
1 = ändern    oder    0 = lesen				0	
1				1	
Monat	Jahr	Adresse		07	88
Stunden	Tag	Adresse + 1		14	03
Sekunden	Minuten	Adresse + 2		30	45
(Null)	Wochentag	Adresse + 3		00	02
		Adresse + 4			
		Adresse + 5			

## Datum und Zeit im verdichteten ASCII-Format mit eingebetteten Doppelpunkten lesen/ändern

Im verdichteten ASCII-Format mit eingebetteten Doppelpunkten ist jede Stelle der Zeit- und Datumswerte ein Byte im ASCII-Format. Darüberhinaus sind Leerzeichen und Doppelpunkte in den Daten eingebettet, um eine unveränderte Übertragung zu einem Drucker oder Sichtgerät zu ermöglichen. Dieses Format belegt 12 Worte.

oberes Byte		unteres Byte		
1 = ändern	oder	0 = lesen		Adresse
3				Adresse + 1
Jahr		Jahr		Adresse + 2
Monat		(Leerzeichen)		Adresse + 3
(Leerzeichen)		Monat		Adresse + 4
Tag		Tag		Adresse + 5
Stunden		(Leerzeichen)		Adresse + 6
:		Stunden		Adresse + 7
Minuten		Minuten		Adresse + 8
Sekunden		:		Adresse + 9
(Leerzeichen)		Sekunden		Adresse + 10
Wochentag		Wochentag		Adresse + 11

BeispielAusgangsparameterblock:  
Datum und Zeit im verdichteten ASCII-Format lesen  
(Dienstag, 2. Oktober 1989, 23:13:00)

0	
3	
39	38
31	20
20	30
32	30
32	20
3A	33
33	31
30	3A
20	30
33	30

### SVCREQ #13: SPS abschalten (anhalten)

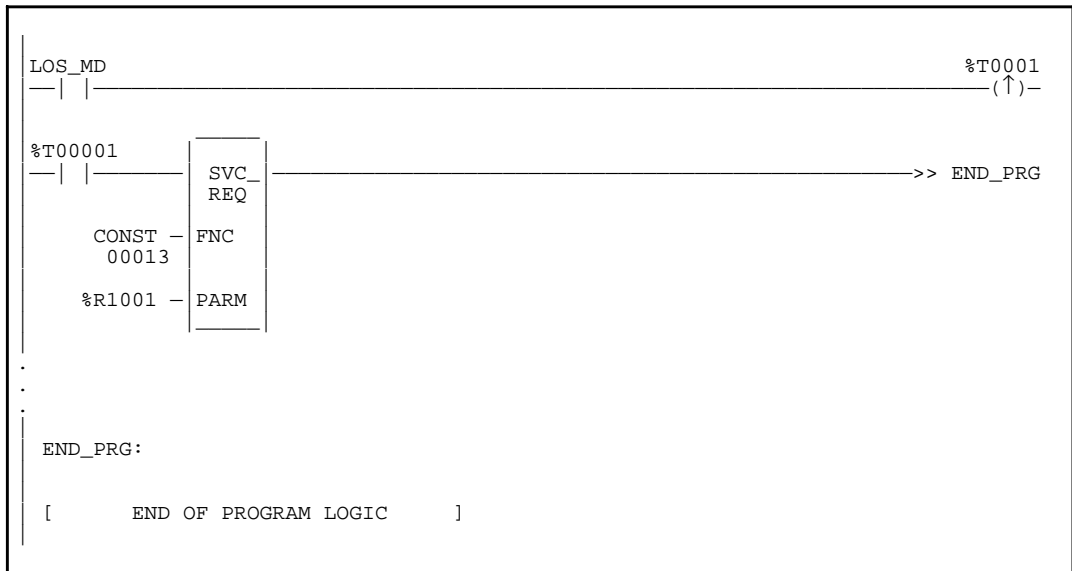
Mit der SVCREQ-Funktion #13 wird die SPS *am Ende des aktuellen Zyklus* abgeschaltet. Alle Ausgänge werden zu Beginn des nächsten SPS-Zyklus auf ihre vorgegebenen Zustände eingestellt. Ein Fehlerhinweis wird in der SPS-Fehlertabelle eingetragen, der darauf hinweist, daß ein Funktionsblock "SHUT DOWN PLC" [SPS abschalten] ausgeführt wurde. Der E/A-Zyklus wird entsprechend der Konfiguration fortgeführt.

Diese Funktion besitzt keinen Parameterblock.

#### Beispiel:

Tritt beim folgenden Beispiel ein Modulverlust auf, wird die SVCREQ-Funktion #13 ausgeführt. Da kein Parameterblock benötigt wird, wird der PARM-Eingang nicht verwendet. Die Programmiersoftware verlangt jedoch, daß für PARM eine Angabe gemacht wird.

Bei diesem Beispiel springt das Programm mit einer JUMP-Funktion auf das Programmende. Hierdurch wird bei erfolgreicher Ausführung der Abschaltfunktion die SPS zwangsweise abgeschaltet. JUMP und LABEL werden benötigt, da der Übergang in den STOP-Modus erst stattfindet, wenn der Zyklus, in dem die Funktion ausgeführt wurde, beendet ist.



#### Hinweis

Um sicherzustellen, daß der Kontakt %S0002 LST\_SCN ordnungsgemäß arbeitet, führt die SPS nach dem Zyklus, in dem die SVCREQ-Funktion #13 ausgeführt wurde, einen zusätzlichen Zyklus aus.

## SVCREQ #14: Fehlertabellen löschen

Mit der SVCREQ-Funktion #14 können SPS- oder E/A-Fehlertabelle gelöscht werden. Der Ausgang dieser SVCREQ-Funktion wird durchgeschaltet, sofern keine andere Zahl als 0 oder 1 für die gewünschte Operation eingegeben wird (siehe unten). Die SVCREQ-Funktion schaltet den Stromfluß nicht durch, wenn die Fehlertabelle leer ist.

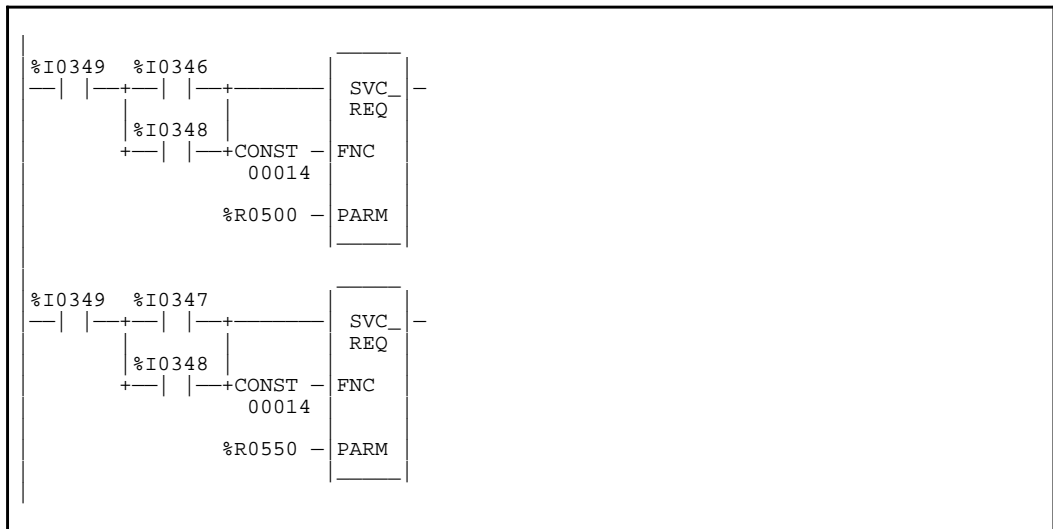
Bei dieser Funktion ist der Parameterblock (nur Eingabe-Parameterblock) ein Wort lang.

0 = SPS-Fehlertabelle löschen	Adresse
1 = E/A-Fehlertabelle löschen	

### Beispiel:

Im folgenden Beispiel wird die SPS-Fehlertabelle gelöscht, wenn die Eingänge %I0346 und %I0349 aktiviert sind. Die E/A-Fehlertabelle wird gelöscht, wenn die Eingänge %I0347 und %I0349 aktiviert sind. Beide Tabellen werden gelöscht, wenn die Eingänge %I0348 und %I0349 aktiviert sind.

Der Parameterblock für die SPS-Tabelle liegt bei %R0500, für die E/A-Fehlertabelle bei %R0550. Beide Parameterblöcke liegen an einer anderen Stelle im Programm.





## SVCREQ #15: Letzten Fehlertableneintrag lesen

Mit der SVCREQ-Funktion #15 können Sie den letzten Eintrag in die SPS- oder E/A-Fehlertabelle lesen. Der Ausgang dieser SVCREQ-Funktion wird durchgeschaltet, sofern keine andere Zahl als 0 oder 1 für die gewünschte Operation eingegeben wird (siehe unten). Die SVCREQ-Funktion schaltet den Stromfluß nicht durch, wenn die Fehlertabelle leer ist. Weitere Informationen über Fehlertableneinträge finden Sie in Kapitel 3.

Bei dieser Funktion ist der Parameterblock 22 Worte lang. Der Eingabe-Parameterblock hat das Format:

0 = SPS-Fehlertabelle lesen	Adresse
1 = E/A-Fehlertabelle lesen	

Das Format des Ausgabe-Parameterblocks hängt davon ab, ob die Funktion Daten aus der SPS- oder der E/A-Fehlertabelle liest.

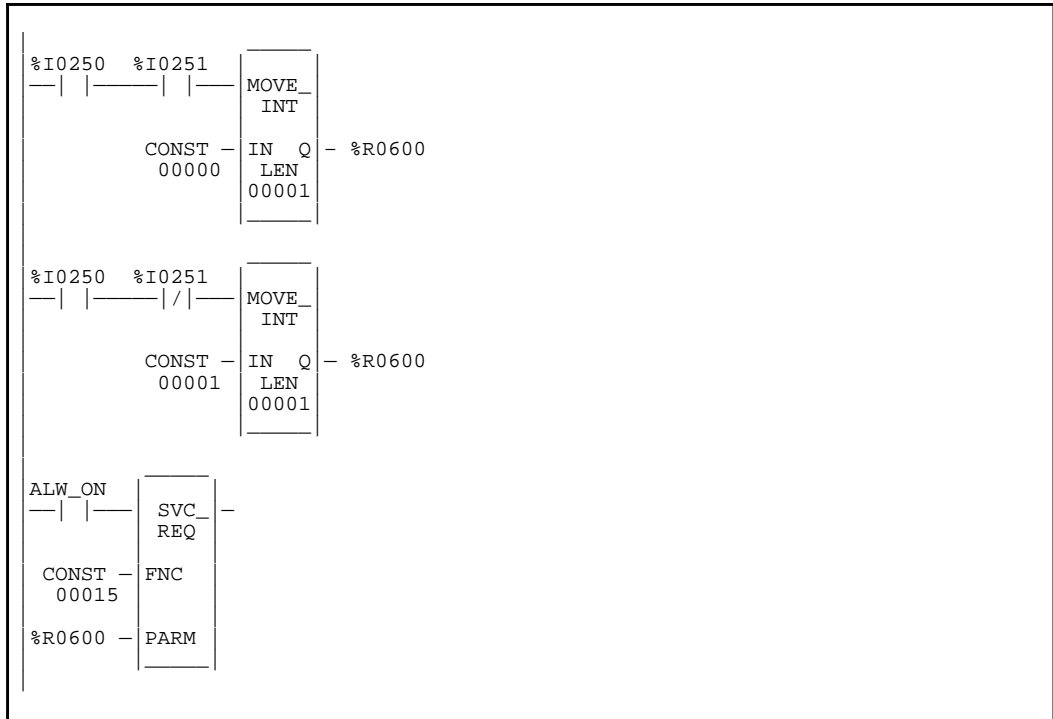
Ausgabeformat SPS-Fehlertabelle		Ausgabeformat E/A-Fehlertabelle	
unteres Byte	oberes Byte	unteres Byte	oberes Byte
0		1	
lang/kurz		lang/kurz	
Reserve		Referenzadresse	
SPS-Fehleradresse		E/A-Fehleradresse	
Fehlergruppe u. Wirkung		Fehlergruppe und Fehlerwirkung	
Fehlercode		Fehlerkategorie	Fehlertyp
		Fehlerbeschreibung	
fehlerspezifischeDaten		fehlerspezifischeDaten	
Zeitstempel		Zeitstempel	

Im ersten Byte von Wort n+1 gibt "kurz/lang" die Anzahl der fehlerspezifischen Daten im Fehlerleintrag an. Es bedeuten:

<b>SPS-Fehlertabelle:</b>	<b>00 = 8 Bytes (kurz)</b>
	<b>01 = 24 Bytes (lang)</b>
<b>E/A-Fehlertabelle:</b>	<b>02 = 5 Bytes (kurz)</b>
	<b>03 = 21 Bytes (lang)</b>

**Beispiel 1:**

Sind im folgenden Beispiel die Eingänge %I0251 und %I0250 aktiviert, wird der letzte Eintrag der SPS-Fehlertabelle in den Parameterblock eingelesen. Ist Eingang %I0251 AUS und Eingang %I0250 EIN, dann wird der letzte Eintrag der E/A-Fehlertabelle in den Parameterblock eingelesen. Der Parameterblock liegt bei der Adresse %R0600.



### Beispiel 2:

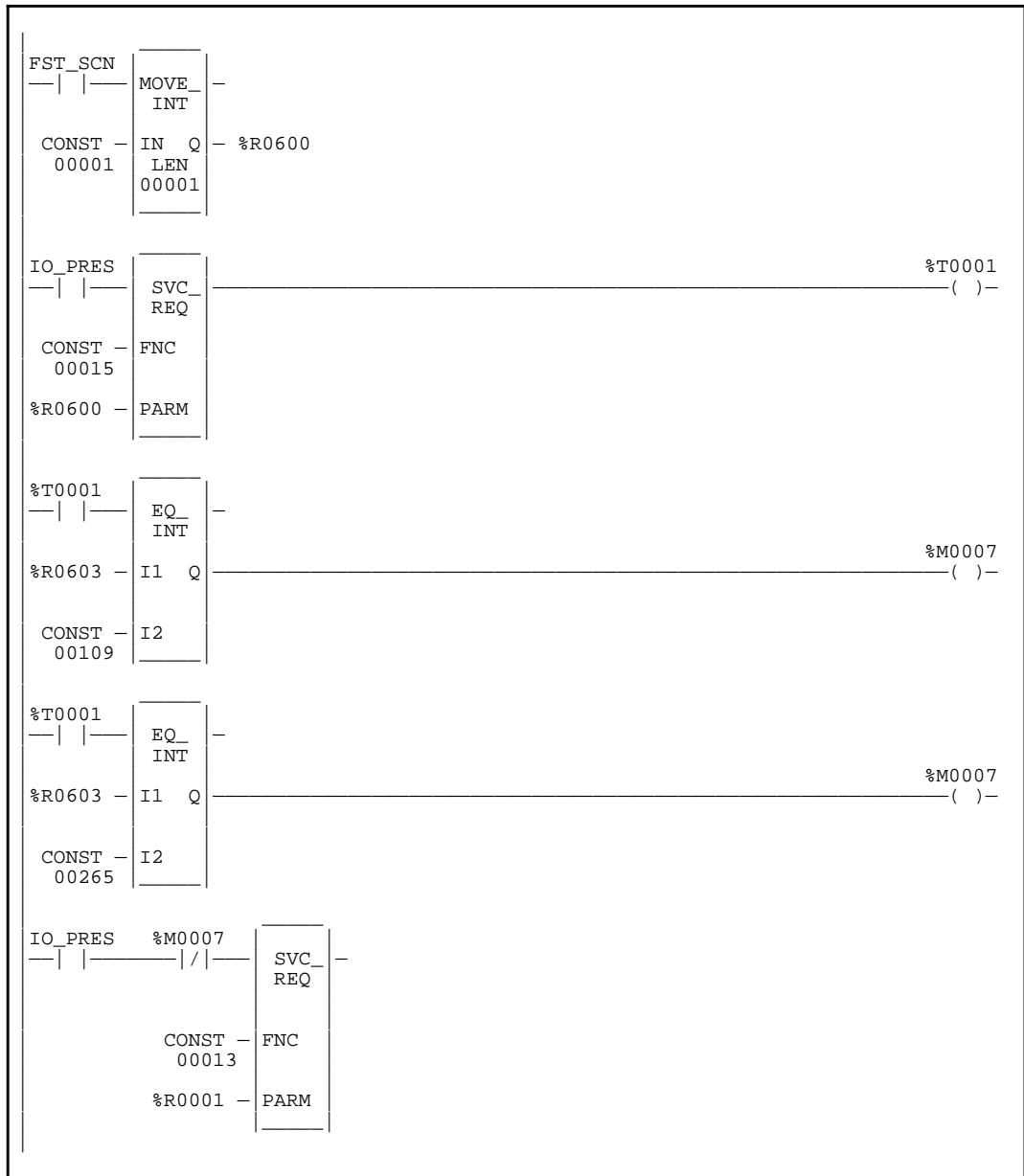
Im folgenden Beispiel wird die SPS abgeschaltet, wenn ein Fehler auf einem E/A-Modul auftritt (Ausnahme: Fehler auf Modulen in Chassis 0, Steckplatz 9 oder Chassis 1, Steckplatz 9). Tritt ein Fehler in einem dieser beiden Module auf, dann läuft das System weiter. Der Parameter für "Tabellentyp" wird im ersten Zyklus eingestellt. Ist der Kontakt IO-PRES gesetzt, enthält die E/A-Fehlertabelle einen Eintrag. Die SPS-CPU setzt den Schließerkontakt in dem auf den Fehlereintrag folgenden Zyklus. Werden bei zwei aufeinanderfolgenden Zyklen Fehler eingetragen, wird der Schließerkontakt bei zwei aufeinanderfolgenden Zyklen gesetzt.

Das Beispiel verwendet einen Parameterblock in %R0600. Nachdem die SVCREQ-Funktion ausgeführt wurde, steht die Moduladresse im 4., 5. und 6. Wort des Parameterblocks:

1		%R0600
lang/kurz		%R0601
Referenzadresse		%R0602
Chassisnummer	Steckplatznr.	%R0603
E/A-Busnummer	Busadresse	%R0604
Punktadresse		%R0605

Fehlerdaten

Im Programm vergleichen die EQ-INT-Blöcke die Chassis-/Steckplatzadresse in der Tabelle mit den hexadezimalen Konstanten. Der interne Merker %M00007 wird durchgeschaltet, wenn Chassis/Steckplatz mit dem fehlerhaften Modul die vorgenannten Kriterien erfüllen. Ist %M00007 durchgeschaltet, dann ist sein Öffnerkontakt AUS und verhindert die Abschaltung. Ist dagegen %M00007 AUS, da der Fehler auf einem anderen Modul aufgetreten ist, dann ist der Öffnerkontakt EIN und die SPS wird abgeschaltet.



## SVCREQ #16: Betriebszeituhr lesen

Mit der SVCREQ-Funktion #16 kann der Wert der Betriebszeituhr ausgelesen werden. Beginnend mit dem Zeitpunkt, zu dem die SPS eingeschaltet wird, erfaßt diese Uhr die Zeit in Sekunden. Der Erfassungszeitraum dieser Uhr beträgt ungefähr 100 Jahre.

Die Funktion besitzt lediglich einen Ausgabe-Parameterblock mit einer Länge von 3 Worten.

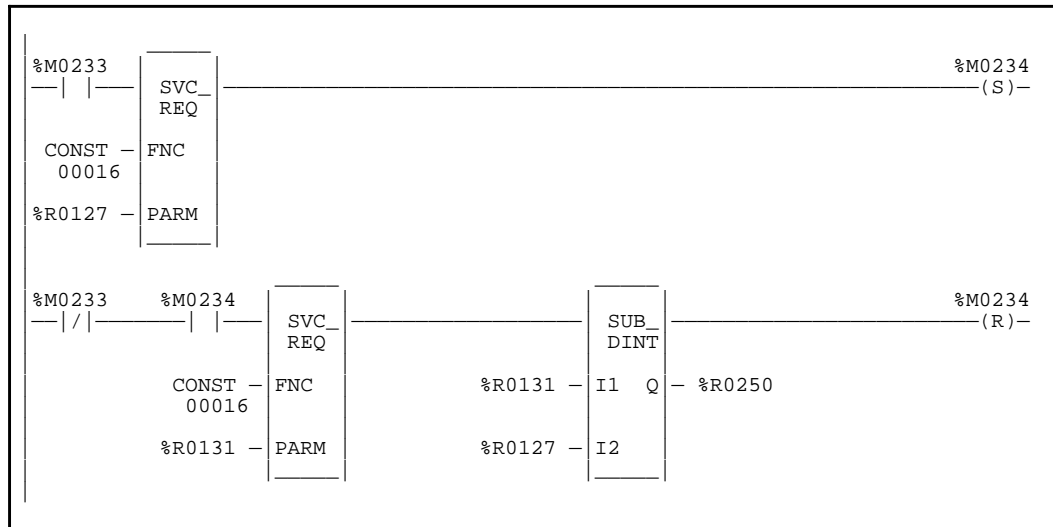
Sekunden seit Einschalten (niedrigstwertig)	Adresse
Sekunden seit Einschalten (höchstwertig)	Adresse + 1
100-Mikrosekunden-Werte	Adresse + 2

Die beiden ersten Werte geben die Betriebszeit in Sekunden an. Das letzte Wort gibt die Anzahl der 100-Mikrosekunden-Anteile der laufenden Sekunde an.

### Beispiel:

Wird beim folgenden Beispiel der interne Merker %M0233 durchgeschaltet, dann wird der Wert der Betriebszeituhr gelesen und die Spule %M0234 gesetzt. Der Wert wird erneut gelesen, wenn der interne Merker abfällt. Der Unterschied zwischen den beiden Werten wird dann berechnet und das Ergebnis im Registerspeicher unter der Adresse %R0250 abgelegt.

Der Parameterblock beim ersten Lesen ist %R0127, beim zweiten Lesen %R0131. Bei der Berechnung werden die 100-Mikrosekunden-Anteile und die Tatsache, daß der DINT-Type eigentlich ein vorzeichenbehafteter Wert ist, vernachlässigt. Die Berechnung ist über einen Zeitraum von etwa 50 Jahren korrekt.



## SVCREQ #18: E/A-Override-Zustand lesen

Mit der SVCREQ-Funktion #18 kann der aktuelle Override-Zustand in der CPU gelesen werden.

### Hinweis

Diese Funktion ist nur ab CPU 331 aufwärts verfügbar.

Bei dieser Funktion besitzt der Parameterblock (nur Ausgabe-Parameterblock) eine Länge von einem Wort.

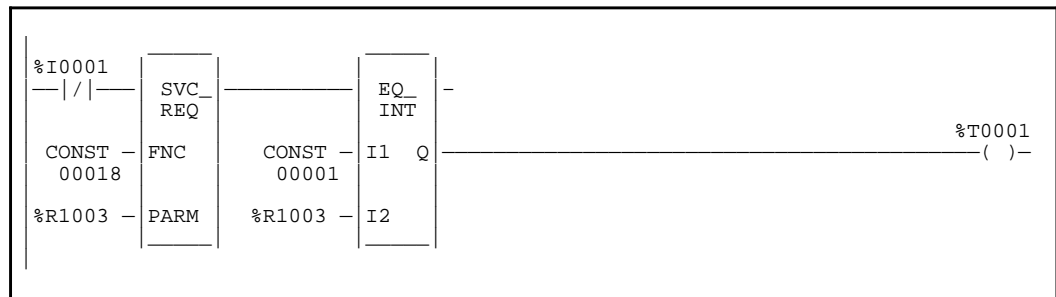
0 = keine Overrides gesetzt	Adresse
1 = Overrides gesetzt	

### Hinweis

SVCREQ #18 meldet nur Override-Zustände der %I- und %Q-Referenzen

### Beispiel:

Im folgenden Beispiel wird der Zustand der E/A-Overrides immer in Adresse %R1003 eingelesen. Ausgang %T0001 wird durchgeschaltet, wenn Override-Zustände bestehen.



## SVCREQ #23: Master-Prüfsumme lesen

Mit der SVCREQ-Funktion #23 können Sie die Master-Prüfsummen von Anwenderprogrammen und Konfiguration lesen. Solange die Funktion freigegeben ist, ist der SVCREQ-Ausgang immer EIN; der Ausgangsdatenblock (siehe unten) beginnt an der in Parameter 3 (PARM) der SVCREQ-Funktion angegebenen Adresse.

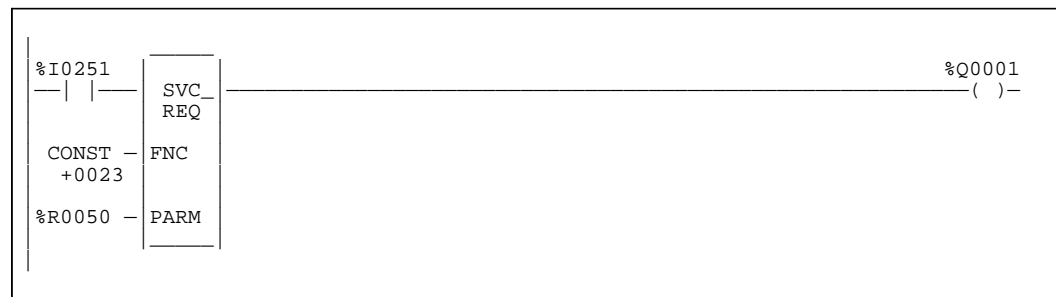
Während eines Speichervorgangs im RUN-Modus können die Programm-Prüfsummen bis zum Ende des Speicherns ungültig sein. Zwei Merker am Anfang des Ausgangsparameterblocks zeigen daher an, ob Programm- und Konfigurations-Prüfsummen gültig sind.

Der Ausgangsparameterblock dieser Funktion ist 12 Worte lang und hat folgendes Format:

Programm-Masterprüfsummegültig (0 = ungültig, 1 = gültig)	Adresse
Konfigurations-Masterprüfsumme gültig (0 = ungültig, 1 = gültig)	Adresse + 1
Anzahl Programmblöcke (einschließlich Hauptblock)	Adresse + 2
Länge des Anwenderprogramms in Bytes (Datentyp DWORD)	Adresse + 3
Additive Programm-Prüfsumme	Adresse + 5
Programm-CRC-Prüfsumme (Datentyp DWORD)	Adresse + 6
Länge der Konfigurationsdaten in Bytes	Adresse + 8
Additive Konfigurations-Prüfsumme	Adresse + 9
Konfigurations-CRC-Prüfsumme (Datentyp DWORD)	Adresse + 10

### Beispiel:

Wird im nachstehenden Beispiel Eingang %I0251 aktiviert, dann werden die Daten zur Master-Prüfsumme in den Parameterblock eingetragen und die Ausgangsspule %Q0001 wird durchgeschaltet. Der Parameterblock liegt bei %R0050.



## SVCREQ #26/30: E/A abfragen

Mit SVCREQ-Funktion #26 (oder #30; beide Funktionen sind identisch und liefern das gleiche Ergebnis) können Sie die vorhandenen Module abfragen und mit der Chassis/Steckplatz-Konfiguration vergleichen und die gleichen Alarme für hinzugefügte, verlorengangene oder falsch konfigurierte Module erzeugen, wie sie bei einer Konfigurationsspeicherung hervorgerufen werden. Je nach Fehler erzeugt diese SVCREQ-Funktion Fehlermeldungen für die SPS- oder die E/A-Fehlertabelle.

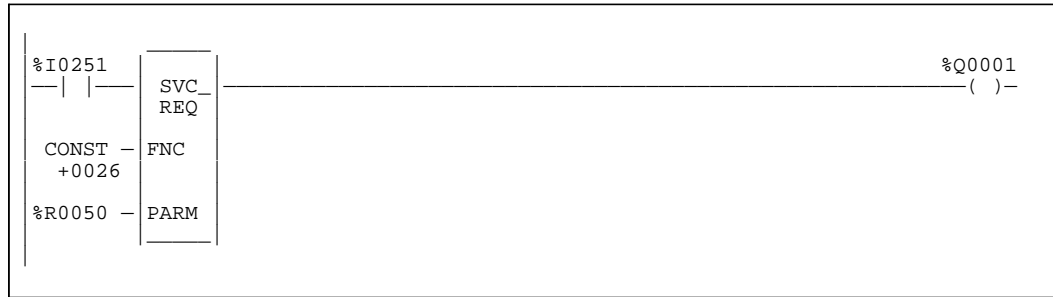
Diese Funktion besitzt keinen Parameterblock und liefert immer Stromfluß am Ausgang.

### Hinweis

Die von dieser SVCREQ-Funktion benötigte Ausführungszeit hängt von der Anzahl der vorliegenden Fehler ab und steigt mit der Anzahl fehlerhafter Module.

### Beispiel:

Wird im nachstehenden Beispiel Eingang %I0251 aktiviert, werden die vorhandenen Module abgefragt und mit der Chassis-/Steckplatz-Konfiguration verglichen. Ausgang %Q0001 wird durchgeschaltet, wenn die SVCREQ beendet ist.



### Hinweis

Diese SVCREQ-Funktion gibt es nicht bei der Micro-SPS.



## SVCREQ #29: Aufgelaufene Netzausfallzeit lesen

Mit der SVCREQ-Funktion #29 können Sie die Zeit auslesen, die zwischen dem letzten Abschalten und dem letzten Einschalten der Versorgungsspannung verstrichen ist. Der SVCREQ-Ausgang ist immer durchgeschaltet und der Ausgangsdatenblock (siehe unten) beginnt an der in Parameter 3 (PARM) der SVCREQ-Funktion eingestellten Adresse.

### Hinweis

Diese Funktion ist nur ab CPU 331 aufwärts verfügbar.

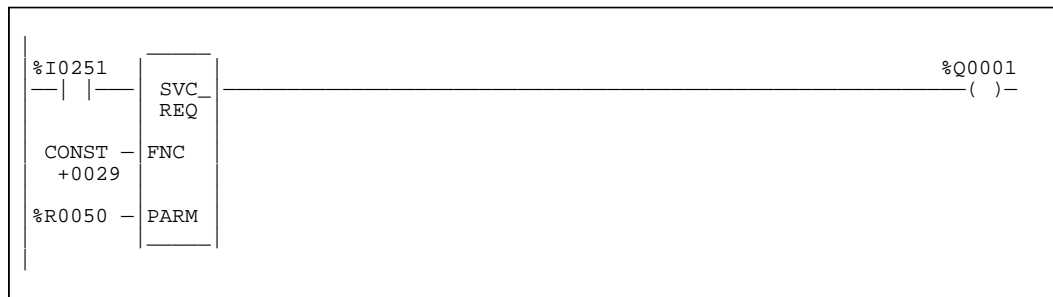
Diese Funktion besitzt nur einen Ausgangsparameterblock mit einer Länge von 3 Worten.

Netzausfallzeit,Sekunden (niedrigstwertig)	Adresse
Netzausfallzeit,Sekunden(höchstwertig)	Adresse + 1
100-Mikrosekundenanteile	Adresse + 2

Die ersten beiden Worte geben die aufgelaufene Netzausfallzeit in Sekunden an. Das letzte Wort gibt die restliche Netzausfallzeit in 100-Mikrosekunden-Anteilen an (ist immer Null). Kann die SPS die aufgelaufene Netzausfallzeit nicht richtig berechnen, wird die Zeit auf 0 gesetzt. Dies ist der Fall, wenn beim Einschalten der SPS eingeschaltet am HHP CLR M/T gedrückt wird oder wenn die Zeitüberwachung (Watchdog) vor dem Abschalten anspricht.

### Beispiel:

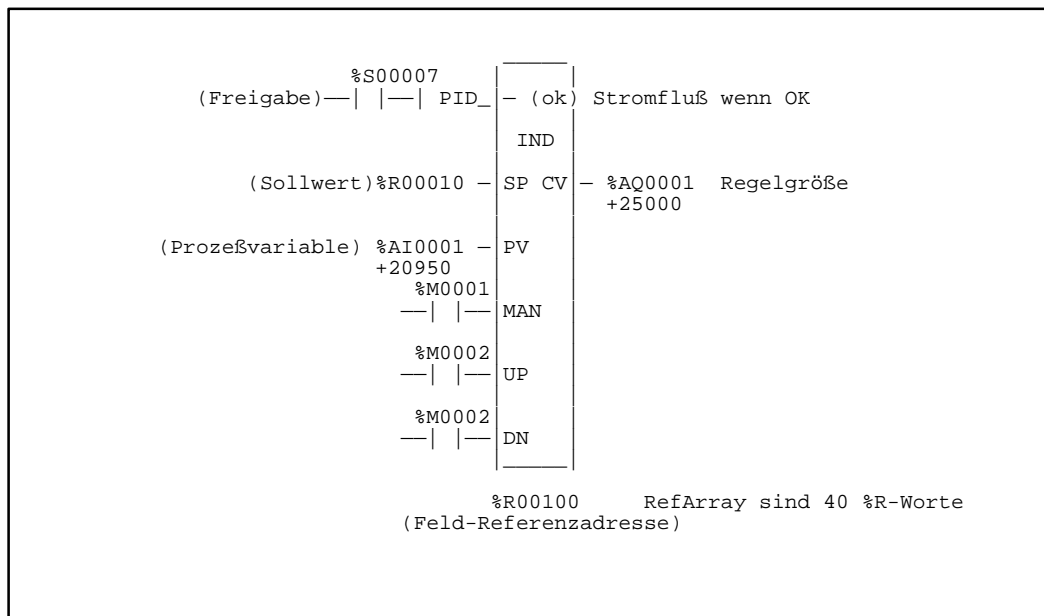
Wird im folgenden Beispiel Eingang %I0251 aktiviert, wird die aufgelaufene netzausfallzeit in den Parameterblock eingetragen und die Ausgangsspule %Q0001 wird durchgeschaltet. Der Parameterblock liegt bei %R0050.



## PID

Die PID-Funktion ist der bekannteste allgemeine Algorithmus, der zur Prozeßregelung benutzt wird. Der PID-Funktionsblock der Series 90 vergleicht den rückgekoppelten Wert einer Prozeßvariablen mit einem Sollwert und aktualisiert den Ausgang der Stellgröße als Funktion des Fehlers.

Zur Lösung des PID-Algorithmus im gewünschten Zeitabschnitt verwendet der Block die PID-Regelverstärkung und weitere Parameter, die in einem Feld von 40 16-Bit-Worten (siehe Seite 4-143) gespeichert sind. Um kompatibel zu den 16-Bit-Analogprozeßvariablen zu sein, sind alle Parameter ganzzahlige 16-Bit-Worte. Hierdurch kann der %AI-Speicher zur Eingabe von Prozeßvariablen und der %AQ-Speicher zur Ausgabe von Regelgrößen verwendet werden. Das nachstehende Beispiel zeigt mögliche Eingangsbelegungen.



In diesem Zusammenhang werden folgende Abkürzungen verwendet: SP = Sollwert, PV = Prozeßvariable und CV = Regelgröße. Als skalierte ganzzahlige 16-Bit-Zahlen müssen zahlreiche Parameter in PV-Zählwerten oder in CV-Zählwerten oder Einheiten definiert werden. Der SP-Eingang muß zum Beispiel über den gleichen Bereich wie PV skaliert werden, da der PID-Block die Abweichung durch eine Subtraktion dieser beiden Eingänge berechnet. In Übereinstimmung mit der Analogskalierung können die PV- und CV-Zählwerte  $-32000$  oder  $0$  bis  $32000$  sein, oder, zur Anzeige von Variablen als  $0,00\%$  bis  $100,00\%$ , von  $0$  bis  $10000$ . Die PV- und CV-Zählwerte brauchen nicht die gleiche Skalierung zu besitzen. In diesem Fall werden die Skalierungsfaktoren in den PID-Verstärkungen eingerechnet.

### Hinweis

Die PID-Funktion kann höchstens einmal alle 10 Millisekunden ausgeführt werden. Dies kann zu einer Veränderung der gewünschten Ergebnisse führen, wenn Sie die Funktion in jedem Zyklus ausführen lassen und die Zyklusdauer kürzer als 10 Millisekunden ist. In einem solchen Fall wird die PID-Funktion erst ausgeführt, wenn genügend Zyklen durchlaufen wurden, daß eine Zeit von 10 Millisekunden aufgelaufen ist. In anderen Worten: Bei einer Zykluszeit von 9 Millisekunden wird die PID-Funktion in jedem zweiten Zyklus mit einer aufgelaufenen Zeit von 18 Millisekunden pro Ausführung bearbeitet.

**Parameter:**

Parameter	Beschreibung
Freigabe	Die PID-Funktion wird durchgeführt, wenn dieser Eingang über einen Kontakt aktiviert wird.
SP	SP ist der Regelsollwert. Wird er mit PV-Zählwerten eingestellt, paßt PID den Ausgangs-CV so an, daß PV zu SP paßt (Nullfehler).
PV	Prozeßvariable vom geregelten Prozeß (häufig ein %AI-Eingang).
MAN	Ist MAN aktiv, dann ist die PID-Funktion in Handbetrieb. Ist Handbetrieb abgeschaltet, ist der PID-Block in Automatikbetrieb.
UP	Der CV-Ausgang wird bei jeder Ausführung um 1 nach oben verändert, wenn dieser Eingang in Handbetrieb aktiviert wird.*
DN	Der CV-Ausgang wird bei jeder Ausführung um 1 nach unten verändert, wenn dieser Eingang in Handbetrieb aktiviert wird.*
Adresse	Die Adresse der PID-Steuerblockinformation (Anwenderparameter und interne Parameter). Benutzt 40 %R-Worte, die nicht gemeinsam genutzt werden können.
ok	Dieser Ausgang wird durchgeschaltet, wenn die Funktion fehlerfrei ausgeführt wurde. Er ist abgeschaltet, wenn ein Fehler vorliegt.
CV	CV ist der Ausgang der Regelgröße zum Prozeß (häufig ein analoger %AQ-Ausgang).

\* Wird pro Zugriff auf die PID-Funktion um 1 erhöht (UP-Parameter) oder erniedrigt (DN-Parameter)

**Zulässige Speichertypen:**

Parameter	Fluß	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	keine
Freigabe	•											
SP		•	•	•	•		•	•	•	•	•	
PV		•	•	•	•		•	•	•	•		
MAN	•											
UP	•											
DN	•											
Adresse								•				
ok	•											•
CV		•	•	•	•		•	•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

## PID-Parameterblock:

Neben den beiden Eingangsworten und den 3 Handsteuerkontakten benutzt der PID-Block 13 der Parameter im Adressfeld RefArray. Diese Parameter müssen eingestellt werden, ehe der Block aufgerufen wird. Die restlichen Parameter werden von der SPS benutzt und sind nicht konfigurierbar. Die in nachstehender Tabelle aufgeführten %Ref ist die gleiche RefArray-Adresse wie die unten am PID-Block. Die Zahl nach dem Pluszeichen gibt den Offset im Feld an. Beginnt RefArray zum Beispiel bei %R100, enthält %R113 den Handbefehl, mit dem die Regelgröße und der Integrator im Handbetrieb eingestellt werden.

**Tabelle 4-4 Übersicht PID-Parameter**

Register	Parameter	Einheit unteres Bit	Wertebereich
%Ref+0000	Kreisnummer	Ganze Zahl	0 bis 25 (nur Anwenderanzeige)
%Ref+0001	Algorithmus	—; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar
%Ref+0002	Ausführungsintervall	10 ms	0 (jeder Zyklus) bis 65535 (10,9 min). Für SPS 90-30 mind. auf 10 einstellen (s. Hinweis auf Seite 4-141)
%Ref+0003	Totzone +	PV-Zählwerte	0 bis 32000 (nie negativ)
%Ref+0004	Totzone –	PV-Zählwerte	–32000 bis 0 (nie positiv)
%Ref+0005	P-Verstärkung –Kp	0,01 CV%/PV%	0 bis 327,67 %/%
%Ref+0006	D-Verstärkung –Kd	0,01 s	0 bis 327,67 s
%Ref+0007	I-Zeit Ki	Wiederholungen/1000 s	0 bis 32.767 Wiederholungen/s
%Ref+0008	CV-Anhebung/Ausgangs-Offset	CV-Zählwerte	–32000 bis 32000 (zu Integratorausgang hinzuzählen)
%Ref+0009	Oberer Begrenzung	CV-Zählwerte	–32000 bis 32000 (>%Ref+09) Ausgangsgrenze
%Ref+0010	Untere Begrenzung	CV-Zählwerte	–32000 bis 32000 (>%Ref+10) Ausgangsgrenze
%Ref+0011	Nachführgrenze	Sekunden/Vollausschlag	0 (keine) bis 32000 s zum Kopieren von 32000 CV
%Ref+0012	Konfigurationswort	Untere 5 Bits verwendet	Bit 0 bis 2 für Vorzeichenfehler; Ausgangspolarität, Diff.
%Ref+0013	Handbefehl	CV-Zählwerte	Nachführung von CV in Automatik- oder Einstellung von CV in Handbetrieb
%Ref+0014	Steuerwort	Von SPS verwaltet, wenn Bit 1 nicht gesetzt ist.	Von SPS verwaltet, sofern nicht anders eingestellt: Unteres Bit = 1 setzt Override (s. Beschreibung in Tabelle auf Seite 4-146)
%Ref+0015	SP intern	—; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar
%Ref+0016	CV intern	—; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar
%Ref+0017	PV intern	—; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar

Register	Parameter	Einheit unteres Bit	Wertebereich
%Ref+0018	Ausgangswert	—; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar
%Ref+0019	Speicherung D-Größe	—; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar
%Ref+0020 und %Ref+0021	Speicherung I-Größe	—; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar
%Ref+0022	Speicherung Änderungsgröße	—; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar
%Ref+0023	Takt	—; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar
%Ref+0024	(letzte Ausführungszeit)		
%Ref+0025			
%Ref+0026	Speicherung Y Rest	—; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar
%Ref+0027	Untergrenze für SP, PV	PV-Zählwerte	-32000 bis 32000 (>%Ref+28) für Anzeige
%Ref+0028	Obergrenze für SP, PV	PV-Zählwerte	-32000 bis 32000 (>%Ref+27) für Anzeige
%Ref+0029 bis %Ref+0034	Reserviert für interne Verwendung	—	Nicht konfigurierbar
%Ref+0035 bis %Ref+0039	Reserviert für interne Verwendung	—	Nicht konfigurierbar

Für das Adreßfeld RefArray müssen %R-Register in der SPS 90-30 verwendet werden. Beachten Sie, daß Sie für jeden Aufruf des PID-Blocks ein eigenes 40-Wort-Feld verwenden müssen, selbst wenn alle 13 Parameter jeweils gleich sind. Dies rührt daher, daß die anderen Worte in dem Feld für PID-interne Speichervorgänge verwendet werden. Stellen Sie sicher, daß das Feld nicht über das Speicherende hinausgeht.

Zur Konfiguration der Anwenderparameter wählen Sie die PID-Funktion und drücken dann **F10**, um ein Menü aufzublenen, in dem die Anwenderparameter angezeigt werden. Wählen Sie dann mit den Pfeiltasten die Felder aus und geben die entsprechenden Werte ein. Für die meisten Standardwerte können Sie 0 eingeben. Eine Ausnahme ist die obere CV-Begrenzung. Dieser Wert muß größer sein als die untere CV-Begrenzung, sonst kann der PID-Block nicht ablaufen. Beachten Sie, daß der PID-Block keinen Stromfluß weitergibt, wenn die Anwenderparameter fehlerhaft sind. Verwenden Sie daher einen temporären Merker zur Überwachung, während Sie die Daten verändern.

Nachdem geeignete PID-Werte eingestellt wurden, sollten sie im BLKMOV als Konstanten definiert werden, so daß sie bei Bedarf zum erneuten Laden der PID-Standard-Anwenderparameter verwendet werden können.

### Arbeitsweise der PID-Anweisung

Im normalen automatischen Betrieb wird der PID-Block in jedem Zyklus aufgerufen. Hierbei liegt am Freigabeeingang Stromfluß an und an den Handeingabekontakten liegt kein Stromfluß. Der Block vergleicht den aktuellen Wert der SPS-Betriebszeituhr mit der im internen Adreßblock gespeicherten Zeit, zu der der PID-Block zuletzt bearbeitet wurde. Ist diese Zeitdifferenz größer als das im dritten Wort (%Ref+0002) definierte Ausführungsintervall, wird der PID-Bl-

gorithmus mit der Differenzzeit bearbeitet. Hierbei werden die letzte Ausführungszeit und der Regelgrößen-Ausgangswert aktualisiert. Im Automatikbetrieb wird der Regelgrößen-Ausgangswert im Handbefehlparameter %Ref+0013 eingetragen.

Liegt an den Eingangskontakten für Freigabe und Handbetrieb Stromfluß an, wird der PID-Block in Handbetrieb versetzt und der Regelgrößen-Ausgangswert wird aus dem Handbefehl-Parameter %Ref+0013 eingestellt. Liegt an den Eingängen UP bzw. DN Stromfluß, wird bei jeder Bearbeitung des PID-Blocks das Handbetriebs-Befehlswort um einen CV-Zählwert erhöht oder erniedrigt. Um schnellere manuelle Änderungen der Regelgröße zu erzielen, kann auch jeder beliebige CV-Zählwert direkt zum Handbetriebs-Befehlswort addiert bzw. von ihm abgezogen werden.

Mit der oberen und unteren CV-Begrenzung grenzt der PID-Block den CV-Ausgangswert ein. Wird eine positive Nachführgrenze definiert, beschränkt dies die Geschwindigkeit, mit der sich der CV-Ausgangswert ändern kann. Werden entweder die CV-Amplitude oder die Änderungsgeschwindigkeit überschritten, wird der im Integrator gespeicherte Wert so eingestellt, daß CV an der Grenze liegt. Dieser Überschwingschutz (definiert auf Seite 4-147) bedeutet, daß sich der CV-Ausgangswert von der Begrenzung wegbewegt, sobald sich das Vorzeichen der Störgröße ändert, selbst wenn die Störgröße über einen längeren Zeitraum versucht hat, CV über (oder unter) die Begrenzung zu zwingen.

Durch diese Arbeitsweise, zusammen mit der Handbefehl-Nachführung von CV im Automatikbetrieb und der Einstellung von CV im Handbetrieb, wird ein stoßfreier Übergang zwischen Hand- und Automatikbetrieb gewährleistet. Die obere und untere CV-Begrenzung sowie die Nachführgrenze gelten für den CV auch im Handbetrieb, und der im Integrator gespeicherte interne Wert wird aktualisiert. Das heißt, daß bei einer Änderung des Handbefehls im Handbetrieb die CV-Ausgabe sich maximal mit der in der Nachführgrenze angegebenen Geschwindigkeit ändert und die obere bzw. untere Begrenzung nicht über- bzw. unterschreitet.

## Hinweis

Jede PID-Funktion sollte in einem Zyklus nur einmal aufgerufen werden.

In der nachstehenden Tabelle finden Sie weitere Einzelheiten zu den in Tabelle 4-4 aufgeführten Parametern. Die bei den einzelnen Parametern in Klammern angegebenen Zahlen stellen den Offset im Datenfeld RefArray dar.

**Tabelle 4-5 PID-Funktionsblockdaten**

Parameter	Beschreibung
Kreisnummer (00)	Die Verwendung der Kreisnummer zur Identifizierung eines PID-Blocks erfolgt wahlweise. Die Kreisnummer, eine ganze Zahl ohne Vorzeichen, die über eine Bedienschnittstelle definiert wird, bildet eine gemeinsame Identifizierung in der SPS. Die Kreisnummer wird unterhalb der Blockadresse angezeigt, wenn das Programm von der Logimaster 90-30/20/Micro Software aus überwacht wird.
Algorithmus (01)	Dieser Wert ist eine vorzeichenlose ganze Zahl, die von der SPS gesetzt wird und angibt, welche Algorithmus von dem Funktionsblock verwendet wird. Der ISA-Algorithmus wird mit Algorithmus 1 bezeichnet, der unabhängige Algorithmus mit Algorithmus 2.
Ausführungsintervall (02)	Die Mindestzeit (in Schritten von 10 ms), die zwischen zwei Ausführungen des PID-Algorithmus liegen muß. Beispiel: Eingabe von 10 für ein Intervall von 100 ms. Der UINT-Wert kann maximal 65535 betragen. Dies ergibt ein Zeitintervall von 10,9 Minuten. Wird 0 eingegeben, wird der Algorithmus bei jedem Aufruf des Blocks bearbeitet (siehe unten: Zeitlicher Ablauf der PID-Blöcke). Der PID-Algorithmus wird nur bearbeitet, wenn sich der Wert der Betriebszeituhr seit der letzten Bearbeitung mindestens um den hier angegebenen Zeitraum erhöht hat. Beachten Sie, daß die 90-30 keine Ausführungszeiten von weniger als 10 ms benutzt (siehe Hinweis auf Seite 4-141); bei kürzeren Zykluszeiten werden somit Zyklen übersprungen. Die Funktion gleicht die seit der letzten Ausführung verstrichene Zeit mit einer Genauigkeit von 100 Mikrosekunden aus. Wird dieser Wert auf 0 gesetzt, wird die Funktion jedesmal ausgeführt, wenn sie freigegeben wird. Die weiter oben erwähnte 10-Millisekunden-Grenze gilt aber auch hier.
Totzone (+/—) (03/04)	INT-Werte, die die obere (+) und untere (–) Grenze der Totzone in PV-Zählwerten festlegen. Wird keine Totzone benötigt, dann sollte dieser Wert auf 0 gesetzt werden. Liegt die PID-Regelabweichung (SP – PV) bzw.(PV – SP) zwischen den Werten (+) und (–) der Totzone, wird die Funktion mit dem Fehlerwert 0 gelöst. Ist der Wert von Null verschieden, muß der Wert (+) größer als 0 und der Wert (–) kleiner als 0 sein, damit der PID-Block funktioniert. Lassen Sie diese Werte auf 0, bis die PID-Regelverstärkung eingestellt oder verändert werden. Danach kann eine Totzone erforderlich werden zur Vermeidung kleiner Änderungen der CV-Ausgabe, die durch kleine Schwankungen der Regelabweichung (z.B. durch mechanischen Verschleiß) verursacht werden.
P-Verstärkung – Kp (05)	In der ISA-Version wird mit dieser INT-Zahl, dem Verstärkungsfaktor Kc, die Änderung von CV in CV-Zählwerten bei eine Änderung von 100 PV-Zählwerte in der Regelabweichung festgelegt. Der Wert wird angezeigt als 0,00 %/%, mit implizit zwei Dezimalstellen. Beispiel: Wird für Kp ein Wert 450 eingegeben, wird dieser Wert als 4,50 angezeigt und ergibt einen Beitrag zur PID-Ausgabe von $K_p \cdot \Delta \text{Fehler} / 100$ oder $450 \cdot \Delta \text{Fehler} / 100$ . Bei der Einstellung eines PID-Regelkreises wird immer zuerst Kp eingestellt.
D-Verstärkung – Kd (06)	Mit dieser INT-Zahl wird festgelegt, wie sich CV in CV-Zählwerten ändert, wenn der Fehler oder PV sich um 1 PV-Zählwert pro 10 ms ändert. Als Zeitwert eingegeben, bei dem das untere Bit 10 ms angibt, wird der Wert angezeigt als 0,00 s, mit implizit zwei Dezimalstellen. Beispiel: Wird für Kd ein Wert 120 eingegeben, wird dieser Wert als 1,20 s angezeigt und ergibt einen Beitrag zur PID-Ausgabe von $K_d \cdot \Delta \text{Fehler} / \Delta t$ bzw., wenn sich der Fehler alle 30 ms um 4 PV-Zählwerte änderte, von $120 \cdot 4 / 30$ . Mit Kd kann eine langsame Reaktion des Regelkreises beschleunigt werden; es ist allerdings sehr anfällig gegenüber Störungen des PV-Eingangssignals.
I-Zeit – Ki (07)	Mit dieser INT-Zahl wird festgelegt, wie sich CV in CV-Zählwerten ändert, wenn der Fehler konstant 1 PV-Zählwert wäre. Er wird angezeigt als 0,000 Wiederholungen/s mit implizit 3 Dezimalstellen. Beispiel: Wird für Ki ein Wert 1200 eingegeben, wird dieser Wert als 1,400 Wiederholungen/s angezeigt und ergibt einen Beitrag zur PID-Ausgabe von $K_i \cdot \text{Fehler} \cdot \Delta t$ oder, bei einem Fehler von 20 Zählwerten und einer SPS-Zykluszeit von 50 ms (Ausführungsintervall 0), $1400 \cdot 20 \cdot 50 / 1000$ . Ki ist normalerweise nach Kp der zweite Parameter, der eingestellt wird.
CV-Anhebung/ Ausgangs-Offset (07)	Dieser INT-Wert in CV-Zählwerten wird vor den Geschwindigkeits- und Amplitudenbegrenzungen zum PID-Ausgang addiert. Mit ihm können von Null verschiedene CV-Werte eingestellt werden, wenn nur Kp-Proportionalverstärkungen benutzt werden, bzw. zur Vorwärtsregelung dieses PID-Regelausgangs von einem anderen Regelkreis.
Obere und untere Begrenzung (09/10)	INT-Werte in CV-Zählwerten, die die höchsten und niedrigsten Werte von CV festlegen. Diese Werte müssen eingegeben werden; die obere Begrenzung muß positiver als die untere Begrenzung sein, da sonst der PID-Block nicht funktioniert. Mit diesen Werten werden normalerweise die auf physikalischen Grenzen beruhenden Grenzwerte des CV-Ausgangs festgelegt. Mit ihnen wird auch die Balkenanzeige von CV für LM90 oder ADS PID-Anzeige skaliert.

Parameter	Beschreibung
Nachführgrenze (11)	<p>Ein positiver UINT-Wert, mit dem angegeben wird, wie lange der CV-Ausgang mindestens brauchen muß, um von 0 auf 100% Vollausschlag oder 32000 CV-Zählwerte zu gelangen. Dies ist die umgekehrte Grenzrate, mit der sich der CV-Ausgangswert ändern kann. Ist der Wert positiv, kann CV nicht schneller als mit <math>32000 \text{ CV-Zählwerten} \cdot \Delta \text{Zeit}(\text{s}) / \text{Nachführgrenze}</math> verändert werden. Beispiel: Bei einem Ausführungsintervall von 2,5 s und einer Nachführgrenze von 500 s, kann CV sich pro PID-Bearbeitung um maximal <math>32000 \cdot 2,5 / 500 = 160</math> CV-Zählwerte ändern. Wie bei den CV-Begrenzungen gibt es einen Überschwingenschutz, der den Integratorwert einstellt, wenn die CV-Grenzgeschwindigkeit überschritten wird. Bei einer Nachführgrenze von 0 gibt es keine CV-Grenzgeschwindigkeit. Beim Einstellen der PID-Schleifenverstärkung müssen Sie die Nachführgrenze auf 0 einstellen.</p>
Konfigurationswort	<p>Mit den unteren 5 Bits dieses Wortes werden drei Standard-PID-Einstellungen verändert. Die übrigen Bits müssen auf 0 gesetzt werden. Setzen Sie das unterste Bit auf 1, um die Standard-PID-Fehlergröße von normalerweise (SP – PV) auf (PV – SP) abzuändern und dadurch das Vorzeichen der Rückführungsgröße zu ändern. Diese Funktion benötigen Sie für Abwärtsregelungen, bei denen CV bei ansteigendem PV reduziert werden muß. Setzen Sie das zweite Bit auf 1, um die Ausgangspolarität zu verändern, so daß CV am PID-Ausgang negativ und nicht wie normal positiv ist. Setzen Sie das vierte Bit auf 1, um den Differenzanteil so zu verändern, daß er die Änderung in der PV-Rückkopplungsgröße und nicht die normale Änderung in der Fehlergröße benutzt. Die unteren 5 Bits im Konfigurationswort werden nachstehend detailliert beschrieben:</p> <p><b>Bit 0</b> = Fehlergröße. Ist dieses Bit auf 0 gesetzt, dann ist die Fehlergröße SP-PV. Ist das Bit 1, dann ist die Fehlergröße PV-SP.</p> <p><b>Bit 1</b> = Ausgangspolarität. Ist dieses Bit 0, dann wird der CV-Ausgang auf den Ausgangswert der PID-Berechnung eingestellt. Ist dieses Bit 1, dann wird der CV-Ausgang auf den negativen Ausgangswert der PID-Berechnung eingestellt.</p> <p><b>Bit 2</b> = Differentialwirkung auf PV. Ist dieses Bit 0, dann wirkt das Differentialverhalten auf die Fehlergröße. Ist dieses Bit 1, dann wirkt das Differentialverhalten auf PV. Alle anderen Bits sollten auf Null gesetzt werden.</p> <p><b>Bit 3</b> = Wirkung der Totzone. Wird das Totzonenbit auf 0 gesetzt, ist die Totzone ohne Wirkung. Liegt der Fehler innerhalb der Grenzen der Totzone, wird er auf Null gesetzt. Im andern Fall wird er durch die Grenzen der Totzone nicht beeinflußt. Wird das Totzonenbit auf 1 gesetzt, zeigt die Totzone Wirkung. Liegt der Fehler innerhalb der Grenzen der Totzone, wird er auf Null gesetzt. Liegt der Fehler außerhalb dieser Grenzen, wird er um die Grenzen reduziert (Fehler = Fehler – Totzonen-grenze).</p> <p><b>Bit 4</b> = Überschwingenschutz. Ist dieses Bit 0, verwendet der Überschwingenschutz eine Rückwärtsberechnung. Ist der Ausgang begrenzt, ersetzt dieses den akkumulativen Restwert von Y (definiert auf Seite 4-148) mit dem zur exakten Erzeugung des begrenzten Ausgangs erforderlichen Wert. Ist das Bit auf 1 gesetzt, wird der akkumulative Y-Wert mit dem Wert ersetzt, den Y zu Beginn der Berechnung hatte. In diesem Fall wird der Y-Wert vor der Begrenzung solange gehalten, wie der Ausgang begrenzt ist.</p> <p>Hinweis: Der Überschwingenschutz steht nur zur Verfügung bei 90-30 CPUs ab Ausgabestand 6.50.</p> <p>Denken Sie daran, daß die Bits in Potenzen von 2 gesetzt werden. Um das Konfigurationswort auf Standard-PID-Konfiguration einzustellen, addieren Sie 1, um die Fehlergröße von SP–PV auf PV–SP zu verändern. Sie addieren 2, um die Ausgangspolarität von CV= PID auf CV = –PID zu ändern. Sie addieren 4, um die Differentialwirkung von Änderung Fehlergröße auf PV-Änderung umzuschalten. Usw.</p>
Handbefehl (13)	<p>Dieser INT-Wert wird im Automatikbetrieb auf den aktuellen CV-Ausgang gesetzt. Wird der Block auf Handbetrieb umgeschaltet, werden mit diesem Wert der CV-Wert und der interne Wert des Integrators innerhalb der oberen und unteren Begrenzung und der Nachführgrenze eingestellt.</p>



Parameter	Beschreibung																								
Steuerwort (14)	<p>Dieser interne Parameter wird normalerweise auf 0 belassen. Wird das untere Overridebit auf 1 gesetzt, müssen dieses Wort und die anderen internen SP-, PV- und CV-Parameter für dezentralen Betrieb dieses PID-Blocks verwendet werden (siehe unten). Hierdurch können dezentrale Bediengeräte, wie z.B. Computer, die Kontrolle vom SPS-Programm wegnehmen. Achtung: Wollen Sie dies verhindern, müssen Sie das Steuerwort auf 0 setzen. Ist das untere Bit 0, können die nächsten vier Bits gelesen werden, um die Zustände der PID-Eingangskontakte solange zu verfolgen, wie am PID-Freigabeeingang Stromfluß anliegt. Diskrete Datenstruktur mit den ersten fünf Bitstellen im folgenden Format:</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bit</th> <th>Wert</th> <th>Funktion</th> <th>Status oder ext. Aktion wenn Overridebit = 1</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>Override</td> <td>0: Blockkontakte überwachen. 1: Blockkontakte extern einstellen.</td> </tr> <tr> <td>1</td> <td>2</td> <td>Hand/Auto</td> <td>1: Block ist in Handbetrieb; andere Werte: Block ist im Automatikbetrieb</td> </tr> <tr> <td>2</td> <td>4</td> <td>Freigabe</td> <td>Muß normalerweise 1 sein, sonst wird der Block nie aufgerufen.</td> </tr> <tr> <td>3</td> <td>8</td> <td>UP/höher</td> <td>1 und Handbetrieb (Bit 1=1): CV wird bei jedem Durchlauf erhöht.</td> </tr> <tr> <td>4</td> <td>16</td> <td>DN/niedr.</td> <td>1 und Handbetrieb (Bit 1=1): CV wird bei jedem Durchlauf erniedrigt.</td> </tr> </tbody> </table>	Bit	Wert	Funktion	Status oder ext. Aktion wenn Overridebit = 1	0	1	Override	0: Blockkontakte überwachen. 1: Blockkontakte extern einstellen.	1	2	Hand/Auto	1: Block ist in Handbetrieb; andere Werte: Block ist im Automatikbetrieb	2	4	Freigabe	Muß normalerweise 1 sein, sonst wird der Block nie aufgerufen.	3	8	UP/höher	1 und Handbetrieb (Bit 1=1): CV wird bei jedem Durchlauf erhöht.	4	16	DN/niedr.	1 und Handbetrieb (Bit 1=1): CV wird bei jedem Durchlauf erniedrigt.
Bit	Wert	Funktion	Status oder ext. Aktion wenn Overridebit = 1																						
0	1	Override	0: Blockkontakte überwachen. 1: Blockkontakte extern einstellen.																						
1	2	Hand/Auto	1: Block ist in Handbetrieb; andere Werte: Block ist im Automatikbetrieb																						
2	4	Freigabe	Muß normalerweise 1 sein, sonst wird der Block nie aufgerufen.																						
3	8	UP/höher	1 und Handbetrieb (Bit 1=1): CV wird bei jedem Durchlauf erhöht.																						
4	16	DN/niedr.	1 und Handbetrieb (Bit 1=1): CV wird bei jedem Durchlauf erniedrigt.																						
SP (15)	(Nicht konfigurierbar und von SPS verwaltet) Verfolgt SP-Eingang. Muß extern gesetzt werden, wenn Override = 1.																								
CV (16)	(Nicht konfigurierbar und von SPS verwaltet) Verfolgt CV-Ausgang.																								
PV (17)	(Nicht konfigurierbar und von SPS verwaltet) Verfolgt PV-Eingang. Muß extern gesetzt werden, wenn Override = 1.																								
Ausgangswert (18)	(Nicht konfigurierbar und von SPS verwaltet) Dieses vorzeichenbehaftete Wort stellt den Ausgang des Funktionsblocks vor wahlweiser Invertierung dar. Wurde keine Invertierung konfiguriert und wurde das Ausgangspolaritätsbit im Steuerwort auf 0 gesetzt, ist dieser Wert gleich dem CV-Ausgang. Wurde Invertierung konfiguriert und wurde das Ausgangspolaritätsbit im Steuerwort auf 1 gesetzt, ist dieser Wert gleich dem negierten CV-Ausgang.																								
Speicherung D-Größe (19)	Intern zur Speicherung von Zwischenwerten verwendet. <i>Schreiben Sie nicht in diese Adresse.</i>																								
Speicherung I-Größe (20/21)	Intern zur Speicherung von Zwischenwerten verwendet. <i>Schreiben Sie nicht in diese Adresse.</i>																								
Speicherung Änderungsgröße (22)	Intern zur Speicherung von Zwischenwerten verwendet. <i>Schreiben Sie nicht in diese Adresse.</i>																								
Takt (23–25)	Interne Speicherung der Betriebszeit (letzte Ausführung des PID-Blocks). <i>Schreiben Sie nicht in diese Adresse.</i>																								
Y-Rest (26)	Enthält den Rest von Integrator-Teilungsskalierung für stationären Fehler 0.																								
Unterer und oberer Bereich (27/28)	Wahlweise INT-Werte in PV-Zählwerten, die den kleinsten und größten Anzeigewert bei horizontaler SP- und PV-Balkenanzeige über Logicmaster Zoom und ADS PID Frontplattenanzeige angeben.																								
Reserviert (29–34 und 35–39)	Reserviert für Verwendung durch GE Fanuc (19–34 für internen Gebrauch; 35–39 für externen Gebrauch). Darf nicht für andere Zwecke verwendet werden.																								

## Interne Parameter im Adreßfeld (RefArray)

Wie in Tabelle 4-5 auf den vorherigen Seiten beschrieben wird, liest der PID-Block 13 Anwenderparameter und verwendet den Rest des 40-Worte-Adreßfeldes zur internen PID-Speicherung. Normalerweise brauchen Sie keinen dieser Werte zu verändern. Wenn Sie den PID-Block nach einer langen Verzögerung in Automatikbetrieb aufrufen, können Sie mit SVC\_REQ #16 die aktuelle SPS-Betriebszeit in %Ref+23 laden, um die letzte PID-Bearbeitungszeit zu aktualisieren und damit einen Schrittwechsel am Integrator zu vermeiden. Haben Sie das Overridebit im Steuerwort (5Ref+14) auf 1 gesetzt, müssen die nächsten vier Bits des Steuerwortes zur Steuerung der Eingangskontakte des PID-Blocks gesetzt werden (siehe Tabelle 4-5 auf den vorherigen Seite). Die internen SP und PV müssen gesetzt werden, da Sie die Kontrolle über den PID-Block vom Kontaktplanprogramm weggenommen haben. Die internen Parameterwerte sind:

## Einstellung von PID-Algorithmus (PIDISA oder PIDIND) und Verstärkung

Bei der Programmierung des PID-Blocks kann entweder die unabhängige (PID\_IND) oder die Standardversion (PID\_ISA) des PID-Algorithmus gewählt werden. Der einzige Unterschied zwischen diesen Algorithmen besteht in der Art, in der die I- und D-Verstärkungen definiert sind. Um diesen Unterschied zu verstehen, müssen Sie folgende Punkte kennen:

Beide PID-Typen berechnen die Fehlergröße als  $SP - PV$  bzw. bei Rückwärtsregelung als  $PV - SP$  (wenn die Fehlergröße, d.h. das untere Bit 0 im Konfigurationswort %Ref+12, auf 1 gesetzt ist). Rückwärtsregelung kann eingesetzt werden, wenn sich die Änderungen des CV-Ausgangssignals und die des PV-Eingangssignals im Gegensatz zum Normalbetrieb (CV steigt wenn PV steigt) in unterschiedliche Richtungen bewegen sollen (CV fällt bei steigendem PV).

**Fehler** =  $(SP - PV)$  oder  $(PV - SP)$ , wenn das untere Bit des Konfigurationswortes 1 ist.

Die Differentiation basiert normalerweise auf der Änderung der Fehlergröße seit der letzten PID-Bearbeitung. Hierdurch kann sich bei einer Änderung des SP-Wertes eine große Änderung der Ausgangsgröße ergeben. Ist dies nicht erwünscht, können Sie das dritte Bit des Konfigurationswortes auf 1 setzen, um den auf PV basierenden Differentialanteil zu berechnen. Das dt (Zeitdifferenz) wird bestimmt, indem der Zeitwert der letzten PID-Bearbeitung dieses Blocks von der aktuellen SPS-Betriebszeit abgezogen wird.

**dt** = aktuelle SPS-Betriebszeit – SPS-Betriebszeit bei der letzten Bearbeitung des PID-Blocks

**Differentialanteil** =  $(\text{Fehler} - \text{vorheriger Fehler}) / dt$  oder  $(PV - \text{vorheriger PV}) / dt$ , wenn 3. Bit des Konfigurationswortes = 1

Der unabhängige PID-Algorithmus (PID\_IND) berechnet die Ausgangsgröße als:

**PID-Ausgang** =  $K_p \cdot \text{Fehler} + K_i \cdot \text{Fehler} + K_d \cdot \text{Differentialanteil} + \text{CV-Anhebung}$

Der Standard-ISA-Algorithmus (PID\_ISA) sieht anders aus:

**PID-Ausgang** =  $K_c \cdot (\text{Fehler} + \text{Fehler} \cdot dt/T_i + T_d \cdot \text{Differentialanteil}) + \text{CV-Anhebung}$

$K_c$  ist der Verstärkungsfaktor der Steuerung,  $T_i$  ist die Nachstellzeit und  $T_d$  die Vorhaltezeit. Der Vorteil von ISA liegt darin, daß eine Veränderung von  $K_c$  den Anteil der I-, D- und P-Größen verändert, wodurch der Regelkreisabgleich einfacher wird. Haben Sie PID-Verstärkungen als Ausdrücke von  $T_i$  und  $T_d$ , verwenden Sie

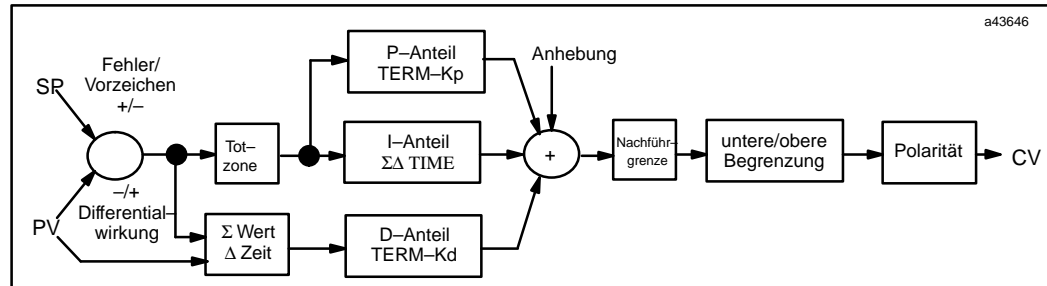
$K_p = K_c$        $K_i = K_c/T_i$       und       $K_d = K_c/T_d$ ,

um sie für die Verwendung als PID-Anwenderparametereingabe umzuwandeln.

Der Ausdruck CV-Anhebung ist eine additive Größe, die von den PID-Komponenten getrennt ist. Diese Größe kann benötigt werden, wenn Sie nur die Proportionalverstärkung  $K_p$  verwenden.

den und wünschen, daß CV bei  $PV = SP$  und Fehler = 0 von Null verschieden ist. In diesem Fall setzen Sie CV-Anhebung auf den gewünschten CV-Wert, wenn PV am SP ist. CV-Anhebung können Sie auch zur Vorwärtsregelung einsetzen, wenn der CV-Ausgang dieses PID-Kreises über einen anderen PID-Kreis oder Steueralgorithmus eingestellt wird.

Wird eine Integrationsverstärkung  $K_i$  verwendet, wäre CV-Anhebung normalerweise 0, da der Integrator als automatische Anhebung arbeitet. Starten Sie lediglich im Handbetrieb, stellen den Integrator über das Hand-Befehlswort (%Ref+13) auf den gewünschten CV ein, und schalten dann um auf Automatikbetrieb. Dies funktioniert auch, wenn  $K_i=0$ , nur wird hierbei nach dem Umschalten in Automatikbetrieb der Integrator nicht auf der Basis des Fehlers eingestellt.



**Abbildung 4-1 Unabhängiger PID-Algorithmus (PIDIND)**

Der ISA-Algorithmus (PIDISA) ist ähnlich. Lediglich der Verstärkungsfaktor  $K_p$  hängt mit  $K_i$  und  $K_d$  zusammen, so daß sich die Integralverstärkung zu  $K_p \cdot K_i$  und die Differentialverstärkung zu  $K_p \cdot K_d$  ergibt. Fehler-Vorzeichen, Differentialwirkung und Polarität werden über Bits im Anwenderparameter "Konfigurationswort" eingestellt.,

### CV-Amplitude und Änderungsgrenzen

Der Block sendet den berechneten PID-Ausgangswert nicht direkt an CV. Beide PID-Algorithmen können der Ausgangs-Regelgröße Grenzen bezüglich Amplitude und Änderungsgeschwindigkeit auferlegen. Die maximale Änderungsgeschwindigkeit wird bestimmt, indem der maximale 100% CV-Wert (32000) durch die minimale Nachführzeit (falls >0 angegeben wurde) dividiert wird. Beispiel: Die minimale Nachführzeit beträgt 100 Sekunden, die Änderungsgeschwindigkeit ist 320 CV-Zählwerte pro Sekunden. Mit einer dt-Bearbeitungszeit von 50 ms darf sich der neue CV-Ausgangswert um nicht mehr als  $320 \cdot 50 / 1000$  oder 16 CV-Zählwerte vom vorherigen CV-Ausgangswert unterscheiden.

Der CV-Ausgangswert wird dann mit den oberen und unteren CV-Begrenzungen verglichen. Wird eine der Grenzen über- bzw. unterschritten, dann wird der CV-Ausgang auf den begrenzten Wert gesetzt. Werden bei der Veränderung von CV entweder Änderungs- oder Amplitudengrenzwert überschritten, wird der interne Integratorwert so angepaßt, daß er dem Grenzwert entspricht, wodurch Überschwingen vermieden wird.

Zuletzt überprüft der Block die Ausgangspolarität (2. Bit des Konfigurationswortes %Ref+12) und ändert das Vorzeichen des Ausgangs, wenn das Bit 1 ist.

$$CV = \begin{matrix} \text{begrenzter PID-Ausgang} & \text{oder} \\ - \text{begrenzter PID-Ausgang, wenn das Polaritätsbit gesetzt ist.} \end{matrix}$$

Ist der Block im Automatikmodus, wird der letzte CV in Handbefehl %Ref+13 eingetragen. Ist der Block im Handmodus, wird die PID-Gleichung übersprungen, da CV durch den Handbefehl eingestellt wird. Es werden jedoch alle Grenzwerte für Änderungsgeschwindigkeit und Amplitude überprüft. Das bedeutet, daß mit dem Handbefehl den Ausgangswert nicht über den obere

ren Grenzwert bzw. unter den unteren Grenzwert hinaus verändert werden kann. Außerdem kann sich der Ausgangswert nur mit der maximalen Änderungsgeschwindigkeit ändern.

## Ausführungsintervall und zeitlicher Ablauf der PID-Blöcke

Der PID-Block ist eine digitale Realisierung einer analogen Steuerungsfunktion. Aus diesem Grund ist die Abtastzeit  $dt$  in der PID-Ausgangsgleichung nicht die bei Analogsteuerungen verfügbare unendlich kleine Zeitspanne. Die meisten der geregelten Prozesse können als eine Verstärkung mit einer Verzögerung erster oder zweiter Ordnung angenähert werden, möglicherweise mit einer reinen Zeitverzögerung. Der PID-Block stellt einen CV-Ausgangswert zum Prozeß ein und benutzt den Rückkopplungs-PV vom Prozeß, um den Fehler zu bestimmen, mit dem die nächste CV-Ausgabe angepaßt wird. Die Gesamtzeitkonstante ist ein Schlüssel-Prozeßparameter, der festlegt, wie schnell der PV reagiert, wenn sich der CV ändert. Wie im Abschnitt "Schleifenverstärkungs-Einstellung" weiter unten besprochen wird, ist die Gesamtzeitkonstante  $T_p+T_c$  bei einem System erster Ordnung die von PV benötigte Zeit, um bei einer sprunghaften Änderung von CV 63% des Endwertes zu erreichen. Der PID-Block kann einen Prozeß nur steuern, wenn das Ausführungsintervall deutlich unter der halben Gesamtzeitkonstante liegt. Größere Ausführungsintervalle machen den Vorgang instabil.

Das Ausführungsintervall sollte nicht größer sein als die Gesamtzeitkonstante geteilt durch 10 (oder, im schlimmsten Fall, herunter bis 5). Erreicht zum Beispiel PV 2/3 seines Endwertes in 2 Sekunden, sollte das Ausführungsintervall kleiner als 0,2 Sekunden (oder maximal 0,4 Sekunden) sein. Auf der anderen Seite sollte das Ausführungsintervall nicht zu klein sein (wie zum Beispiel kleiner als die Gesamtzeitkonstante dividiert durch 1000), da sonst der Ausdruck  $K_i \Delta t$  für den PID-Integrator auf Null gerundet wird. Beispiel: Bei einem sehr langsamen Prozeß, der 10 Stunden bzw. 36000 Sekunden benötigt, um den 63%-Wert zu erreichen, sollte das Ausführungsintervall auf 40 Sekunden oder mehr eingestellt werden.

Sofern der Prozeß nicht gerade sehr schnell abläuft, braucht man kein Ausführungsintervall von 0 einzustellen, bei dem der PID-Algorithmus in jedem PID-Zyklus gelöst wird. Werden viele PID-Kreise verwendet, bei denen das Ausführungsintervall größer als die Zykluszeit ist, können sich bei der SPS-Zykluszeit große Schwankungen ergeben, wenn eine große Anzahl Kreise den Algorithmus gleichzeitig löst. Eine einfache Lösung besteht darin, ein oder mehrere auf 1 gesetzte Bits durch ein Bitfeld von Nullen zu schieben, so daß dadurch Stromfluß zu einzelnen PID-Blöcken durchgeschaltet wird.

## Festlegen der Prozeßdaten

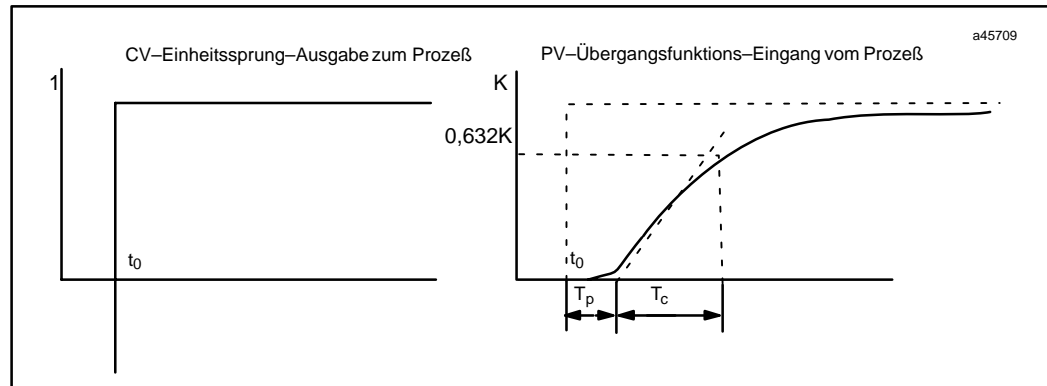
Die PID-Schleifenverstärkungen  $K_p$ ,  $K_i$  und  $K_d$  werden von den Daten des gesteuerten Prozesses bestimmt. Beim Einstellen einer PID-Schleife stellen sich zwei Fragen:

1. Wie groß ist die Änderung von PV, wenn wird CV um einen festen Betrag ändern, bzw. wie groß ist die Leerlaufverstärkung ?
2. Wie schnell reagiert das System, oder wie schnell ändert sich PV nach einer sprunghaften Änderung von CV ?

Zahlreiche Prozesse können angenähert werden durch eine Prozeßverstärkung, eine Verzögerung erster oder zweiter Ordnung und eine reine Zeitverzögerung. Im Frequenzbereich ist die Übertragungsfunktion eines Verzögerungssystems zweiter Ordnung mit reiner Zeitverzögerung gegeben durch:

$$PV(s) / CV(s) = G(s) = K \cdot e^{-T_p s} / (1 + T_c s)$$

Mit einer sprunghaften Änderung zum Zeitpunkt  $t_0$  im Zeitbereich ergibt sich die folgende Änderung der Regelgröße:



Aus der PV-Übergangsfunktion können folgende Prozeßmodell-Parameter bestimmt werden:

K	Leerlaufverstärkung = letzte Änderung von PV / Änderung von CV zum Zeitpunkt $t_0$ (beachten Sie, daß K keinen Index hat)
$T_p$	Prozeß- oder Pipeline-Zeitverzögerung oder Totzeit nach $t_0$ , bis sich der Prozeßausgang PV anfängt zu bewegen.
$T_c$	Prozeßzeitkonstante erster Ordnung; nach $T_p$ von PV benötigte Zeit, um 63,2% des PV-Endwerts zu erreichen.

Normalerweise können Sie diese Parameter am schnellsten messen, indem Sie den PID-Block in Handmodus schalten und den CV-Ausgang um einen kleinen Schritt verändern. Hierzu verändern Sie den Handbefehl %Ref+13 und zeigen die PV-Antwort als Funktion der Zeit an. Bei langsamen Prozessen kann dies von Hand durchgeführt werden; bei schnelleren Prozessen sollten Sie einen Schreiber oder ein Computergraphik-Datenerfassungspaket verwenden. Die Sprunghöhe von CV sollte zwar groß genug sein, um eine merkliche Änderung von PV zu bewirken, aber nicht so groß, daß sie den gemessenen Prozeß unterbrechen würde. Als Richtwert können Sie 2 bis 10% des Unterschieds zwischen oberer und unterer Begrenzung annehmen.

### Einstellen der Anwenderparameter und Abgleich der Schleifenverstärkung

Da alle PID-Parameter vollständig vom geregelten Prozeß abhängen, gibt es keine funktionierenden Standardwerte. Eine brauchbare Schleifenverstärkung können Sie aber in folgenden Schritten iterativ ermitteln:

1. Setzen Sie zunächst alle Anwenderparameter auf 0. Setzen Sie dann die obere und untere CV-Begrenzung auf den höchsten bzw. niedrigsten CV-wert, der zu erwarten ist. Stellen Sie das Ausführungsintervall auf die geschätzte Prozeßzeitkonstante (oben) / 10 bis 100.
2. Schalten Sie den Block auf Handbetrieb und stellen Sie Handbefehl (%Rwef+13) auf unterschiedliche Werte ein. Prüfen Sie dabei, ob CV dadurch auf die obere und untere Begrenzung gebracht werden kann. Schreiben Sie PV an einigen PV-Punkten auf und laden Sie ihn in SP.
3. Stellen Sie  $K_p$  auf eine kleine Verstärkung (z.B. 100  $\dot{A}$  max. CV/max. PV) ein und schalten Sie den Handmodus ab. Verändern Sie SP sprunghaft auf 2 bis 100% des maximalen PV-Bereichs und beobachten Sie die Reaktion von PV. Erhöhen Sie  $K_p$ , wenn die PV-Sprungantwort zu langsam ausfällt bzw. verkleinern Sie  $K_p$ , wenn PV überschwingt und oszilliert, ohne einen stationären Wert anzunehmen.

4. Haben Sie  $K_p$  ermittelt, erhöhen Sie  $K_i$  so, daß Sie ein Überschwingen erhalten, das in 2 oder 3 Zyklen auf einen stationären Wert gedämpft wird. Es kann erforderlich werden,  $K_p$  zu reduzieren. Probieren Sie es mit unterschiedlichen Sprunghöhen und CV-Arbeitspunkten.
5. Nachdem Sie passende Werte für  $K_p$  und  $K_i$  gefunden haben, addieren Sie probierhalber  $K_d$ , um eine schnellere Reaktion auf Eingangsänderungen zu erhalten (vorausgesetzt, das System beginnt nicht zu schwingen).  $K_d$  wird häufig nicht benötigt und funktioniert nicht bei verdrauschem PV.
6. Überprüfen Sie die Verstärkungswerte über verschiedenen SP-Arbeitspunkten. Stellen Sie bei Bedarf zusätzlich Totzone und Nachführgrenze ein. Bei einigen Prozessen mit umgekehrter Wirkungsrichtung kann es erforderlich sein, die Vorzeichen- oder Polaritätsbits im Konfigurationswort zu setzen.

### Ableich nach Ziegler und Nichols

Nachdem Sie die drei Parameter  $K$ ,  $T_p$  und  $T_c$  des Prozeßmodells ermittelt haben, können Sie sie zur Abschätzung der PID-Anfangs-Schleifenverstärkungen benutzen. Mit dem folgenden Verfahren, das Ziegler und Nichols in den 40ern entwickelt haben, kann eine gute Antwort auf Systemstörungen mit Verstärkungsfaktoren erzielt werden, die ein Amplitudenverhältnis von 1/4 erzeugen. Das Amplitudenverhältnis ist das Verhältnis des zweiten zum ersten Maximum im Regelkreis-Übertragungsverhalten.

1. Berechnen Sie die Reaktionsgeschwindigkeit  $R$ :

$$R = K/T_c$$

2. Berechnen Sie bei Proportionalregelung die Proportionalverstärkung  $K_p$ :

$$K_p = 1 / (R \cdot T_p) = T_c / (K \cdot T_p)$$

3. Berechnen Sie bei PI-Regelung:

$$K_p = 0,9 / (R \cdot T_p) = 0,9 \cdot T_c / (K \cdot T_p)$$

$$K_i = 0,3 \cdot K_p / T_p$$

4. Berechnen Sie bei PID-Regelung:

$$K_p = G / (R \cdot T_p) \quad \text{mit } G \text{ von } 1,2 \text{ bis } 2,0$$

$$K_i = 0,5 \cdot K_p / T_p$$

$$K_d = 0,5 \cdot K_p \cdot T_p$$

5. Überprüfen Sie, ob das Ausführungsintervall im Bereich  $(T_p + T_c)/10$  bis  $(T_p + T_c)/1000$  liegt.

Bei einem anderen Verfahren, der "idealen Abstimmung", erzielen Sie die besten Reaktionen auf Änderungen von SP, die nur von der Prozeßverzögerung  $T_p$  oder der Totzeit verzögert werden.

$$K_p = 2 \cdot T_c / (3 \cdot K \cdot T_p)$$

$$K_i = T_c$$

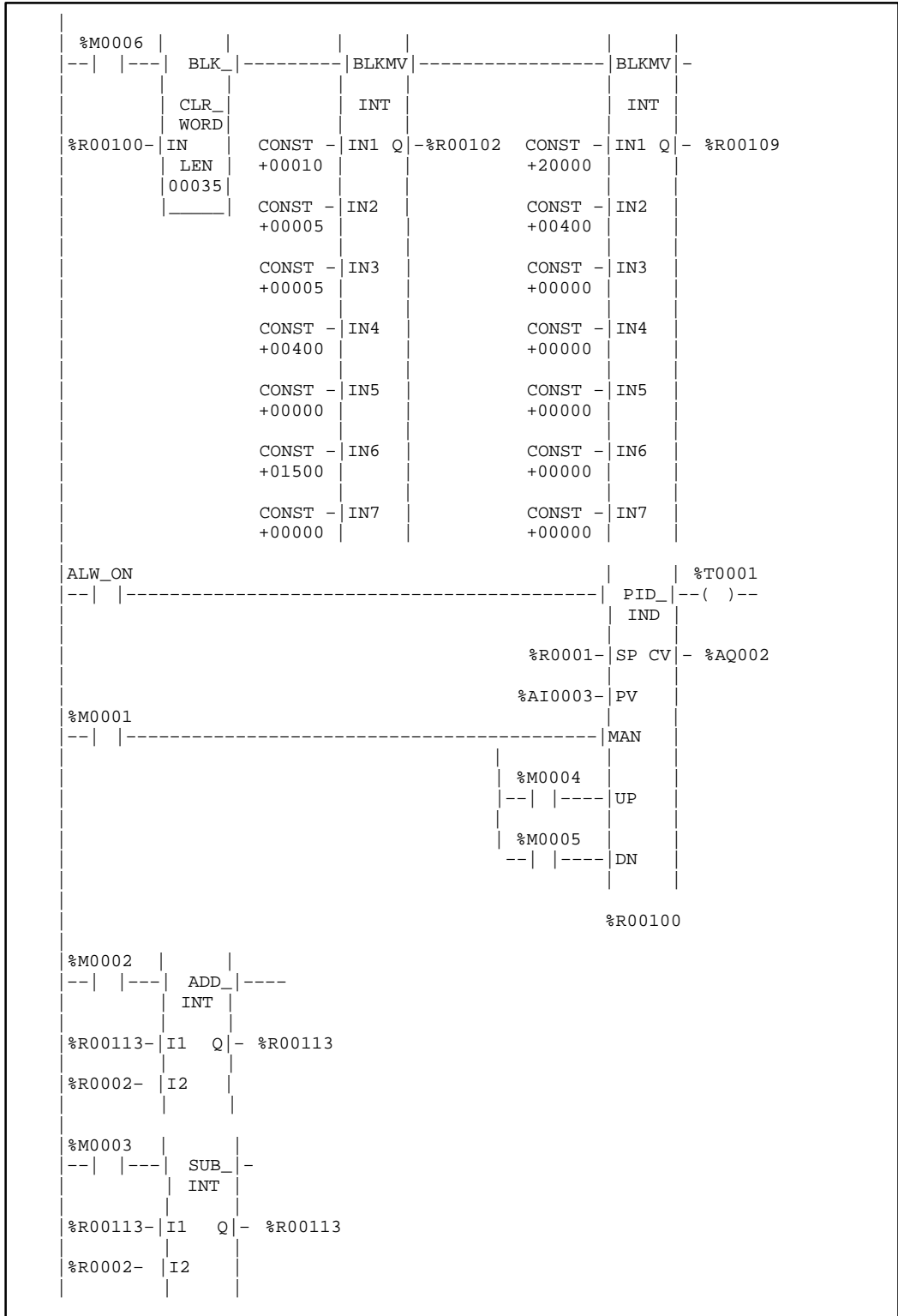
$$K_d = K_i / 4 \quad \text{wenn der Differentialteil verwendet wird.}$$

Nachdem die Anfangsverstärkungsfaktoren ermittelt wurden, müssen sie in ganzzahlige Anwenderparameter umgewandelt werden. Zur Vermeidung von Skalierungsproblemen sollte die

Prozeßverstärkung  $K$  als eine Änderung der Eingangs-PV-Zählwerte dividiert durch die sprungförmige Ausgangsänderung in CV-Zählwerten und nicht in technischen Prozeß-PV- oder CV-Einheiten. Alle Werte sollten in Sekunden angegeben werden. Nachdem  $K_p$ ,  $K_i$  und  $K_d$  bestimmt wurden, können  $K_p$  und  $K_d$  mit 100 multipliziert und als ganze Zahlen eingegeben werden.  $K_i$  kann mit 1000 multipliziert werden und in den Anwenderparameter %RefArray eingetragen werden.

### Beispiel eines PID-Aufrufs

Das nachstehende Beispiel hat ein Ausführungsintervall von 100 ms, eine  $K_p$ -Verstärkung von 4,00 und eine  $K_i$ -Verstärkung von 1,500. Der Sollwert SP wird in %R1 gespeichert, während die Regelgröße CV in %AQ2 ausgegeben und die Prozeßvariable in %AI3 zurückgeführt wird. Die oberen und unteren Begrenzungen müssen in diesem Fall auf 20000 und 400 gesetzt werden; eine schmale Totzone von +5 und -5 wurde wahlweise eingesetzt. Das aus 40 Worten bestehende Adreßfeld beginnt ab %R100. Normalerweise werden die Anwenderparameter im Adreßfeld mit der PID-Zoomtaste **F10** eingestellt. Es kann jedoch auch %M6 gesetzt werden, um die 14 Worte ab %R102 (%Ref+2) aus den im Programm gespeicherten Konstanten neu zu initialisieren.





---

Der Block kann mit %M1 auf Handmodus umgeschaltet werden, so daß der Handbefehl %R113 eingestellt werden kann. Mit den Bits %M4 und %M5 können %R113, PID-CV und Integrator in jeder 100-Millisekunden-Bearbeitung um 1 erhöht bzw. erniedrigt werden. Für schnelleren Handbetrieb kann mit den Bits %M2 und %M3 in jedem SPS-Zyklus der Wert aus %R2 zu %R113 addiert bzw. davon abgezogen werden. Der Ausgang %T1 ist durchgeschaltet, wenn die PID-Funktion in Ordnung ist.

Die speicherprogrammierbaren Steuerungen Series90-30, 90-20 und Micro unterstützen zahlreiche unterschiedliche Funktionen und Funktionsblöcke. Die Tabellen in diesem Anhang zeigen für die einzelnen Funktionen die Ausführungszeiten und den benötigte Speicherplatz. Unter Speicherplatz wird angegeben, wieviele Bytes die Funktion in einem Kontaktplanprogramm belegt.

Für jede Funktion werden zwei Ausführungszeiten angegeben:

<b>Ausführungszeit</b>	<b>Beschreibung</b>
Freigegeben	Von einer Funktion bzw. einem Funktionsblock benötigte Zeit, wenn Stromfluß in die Funktion hinein und aus der Funktion heraus möglich ist. Die kürzesten Zeiten treten normalerweise auf, wenn der Block im Anwender-RAM steht (wortstrukturierter Speicher) und nicht im ISCP-Cache (bitorientierter Speicher).
Gesperrt	Für die Ausführung einer Funktion benötigte Zeit, wenn zwar Stromfluß in die Funktion bzw. den Funktionsblock möglich ist, diese aber in einem inaktiven Zustand sind (z.B. ein Timer, der im Rücksetzzustand gehalten wird).

### Hinweis

Timer und Zähler werden immer aktualisiert, wenn sie in der Logik angetroffen werden; Timer um den seit dem letzten Zyklus aufgelaufenen Zeitwert und Zähler um einen Zählwert.

Tabelle A-1 Befehlsausführungszeiten

Funktionsgruppe	Funktion	Freigegeben				Gesperrt				Inkrement				Größe
		311	313	331	340/41	311	313	331	340/41	311	313	331	340/41	
Timer	Einschaltverzögerung	146	81	101	42	105	39	46	21	–	–	–	–	15
	Ausschaltverzögerung	98	47	54	23	116	63	73	32	–	–	–	–	9
	Timer	122	76	95	40	103	54	66	30	–	–	–	–	15
Zähler	Aufwärtszähler	137	70	87	36	130	63	78	33	–	–	–	–	11
	Abwärtszähler	136	70	86	37	127	61	75	31	–	–	–	–	11
Arithmetikfunktionen	Addition (INT)	76	47	56	24	41	0	0	0	–	–	–	–	13
	Addition (DINT)	90	60	76	34	41	1	0	0	–	–	–	–	13
	Subtraktion (INT)	75	46	57	25	41	0	0	0	–	–	–	–	13
	Subtraktion (DINT)	92	62	78	34	41	1	0	0	–	–	–	–	13
	Multiplikation (INT)	79	49	62	28	41	0	0	0	–	–	–	–	13
	Multiplikation (DINT)	108	80	100	43	41	1	0	0	–	–	–	–	13
	Division (INT)	79	51	61	27	41	0	0	0	–	–	–	–	13
	Division (DINT)	375	346	434	175	41	1	0	0	–	–	–	–	13
	Modulo Division (INT)	78	51	61	27	41	0	0	0	–	–	–	–	13
	Modulo Div (DINT)	134	103	130	54	41	1	0	0	–	–	–	–	13
	Quadratwurzel (INT)	153	124	155	65	42	0	0	0	–	–	–	–	9
Quadratwurzel (DINT)	268	239	299	120	42	0	1	1	–	–	–	–	9	
Relationale Funktionen	Gleich (INT)	66	35	44	19	41	1	1	0	–	–	–	–	9
	Gleich (DINT)	86	56	68	29	41	1	1	0	–	–	–	–	9
	Ungleich (INT)	67	39	48	22	41	1	1	0	–	–	–	–	9
	Ungleich (DINT)	81	51	65	28	41	1	1	0	–	–	–	–	9
	Größer als (INT)	64	33	42	20	41	1	1	0	–	–	–	–	9
	Größer als (DINT)	89	59	73	32	41	1	1	0	–	–	–	–	9
	Größer als/gleich (INT)	64	36	42	19	41	1	1	0	–	–	–	–	9
	Größer als/gleich (DINT)	87	58	73	30	41	1	1	0	–	–	–	–	9
	Kleiner als (INT)	66	35	44	19	41	1	1	0	–	–	–	–	9
	Kleiner als (DINT)	87	57	70	30	41	1	1	0	–	–	–	–	9
	Kleiner als/gleich (INT)	66	36	44	21	41	1	1	0	–	–	–	–	9
	Kleiner als/gleich (DINT)	86	57	70	31	41	1	1	0	–	–	–	–	9
	Bereich (INT)	92	58	66	29	46	1	0	1	–	–	–	–	15
	Bereich (DINT)	106	75	84	37	45	0	0	0	–	–	–	–	15
Bereich (WORD)	93	60	67	29	0	0	0	0	–	–	–	–	15	
Bitoperationsfunktionen	Logisch UND	67	37	48	22	42	0	0	1	–	–	–	–	13
	Logisch ODER	68	38	48	21	42	0	0	1	–	–	–	–	13
	Logische Antivalenz	66	38	47	20	42	0	0	1	–	–	–	–	13
	Logische Invertierung	62	32	40	17	42	0	0	1	–	–	–	–	9
	Bit nach links verschieben	139	89	111	47	74	26	30	13	11,61	11,61	15,05	6,29	15
	Bit nach rechts verschieben	135	87	107	45	75	26	30	13	11,63	11,62	15,07	6,33	15
	Bit nach links rotieren	156	127	158	65	42	1	0	0	11,70	11,78	15,18	6,33	15
	Bit nach rechts rotieren	146	116	147	62	42	1	0	0	11,74	11,74	15,23	6,27	15
	Bitposition	102	72	126	38	42	1	153	0	–	–	–	–	13
	Bit auf 0 setzen	68	38	34	21	42	1	33	1	–	–	–	–	13
	Bit testen	79	49	132	28	41	0	126	1	–	–	–	–	13
	Bit auf 1 setzen	67	37	0	20	42	0	36	0	–	–	–	–	13
	Bitvergl. m. Maske (WORD)	217	154	177	74	107	44	50	21	–	–	–	–	25
	Bitvergl. m. Maske (DWORD)	232	169	195	83	108	44	49	22	–	–	–	–	25

**Hinweise:**

- Die Zeit (in Mikrosekunden) ist angegeben auf der Grundlage von Ausgabestand 5.0 der Logicmaster 90-30/20 Software für CPU-Modelle 311, 313, 331, 340 und 341.
- Bei den Tabellenfunktionen ist das Inkrement in Einheiten der angegebenen Länge; bei Bitoperationen in Mikrosekunden/Bit; bei Datenverschiebefunktionen in Mikrosekunden/Anzahl Bits oder Worte.
- Freigegebene Zeit für Einheiten einfacher Länge der Typen %R, %AI und %AW.
- Die COMMREQ-Zeit wurde gemessen zwischen CPU und HSC.
- DOIO ist die Zeit für die Ausgabe von Werten zu diskreten Ausgangsmodulen.
- Gibt es mehr als einen möglichen Fall, gibt die in der Tabelle angegebene Zeit den ungünstigsten Fall an.

**Ausführungszeiten für die Micro-SPS:** Angaben hierzu in GFK-1065B (oder später), *SPS Series90 Micro, Anwenderhandbuch*.

**Ausführungszeiten für die SPS 351:** Siehe ab Seite A-5

**Tabelle A-1 Befehlsausführungszeiten – Fortsetzung**

Funktionsgruppe	Funktion	Freigegeben				Gespart				Inkrement				Größe
		311	313	331	340/41	311	313	331	340/41	311	313	331	340/41	
Datenverschiebefunktionen	Kopieren (INT)	68	37	49	20	43	0	1	0	1,62	1,62	6,60	1,31	13
	Kopieren (BIT)	94	62	77	35	42	0	0	0	12,61	12,64	15,78	6,33	13
	Kopieren (WORD)	67	37	49	20	41	0	1	0	1,62	1,63	6,60	1,31	13
	Block kopieren (INT)	76	48	61	28	59	30	34	16	–	–	–	–	27
	Block kopieren (WORD)	76	48	62	29	59	29	35	15	–	–	–	–	27
	Block löschen	56	28	33	14	43	0	0	0	1,35	1,29	1,78	0,78	9
	Schieberegister (BIT)	201	153	192	79	85	36	43	18	0,69	0,68	0,88	0,37	15
	Schieberegister (WORD)	103	53	67	29	73	25	29	12	1,62	1,62	2,54	1,31	15
Bitfolgesteuerung	165	101	127	53	96	31	37	16	0,07	0,07	0,10	0,05	15	
	COMM_REQ	1317	1272	1577	884	41	2		0	–	–	–	–	13
Tabellenfunktionen	Feld kopieren													
	INT	230	201	254	104	72	41	49	20	1,29	1,15	7,16	2,06	21
	DINT	231	202	260	105	74	44	53	23	3,24	3,24	13,20	2,61	21
	BIT	290	261	329	135	74	43	51	23	–,03	–,03	0,39	0,79	21
	BYTE	228	198	252	104	74	42	51	23	0,81	0,82	5,58	1,25	21
	WORD	230	201	254	104	72	41	49	20	1,29	1,15	7,16	2,06	21
	Suche gleiche													
	INT	197	158	199	82	78	39	46	20	1,93	1,97	3,17	1,55	19
	DINT	206	166	209	87	79	38	47	21	4,33	4,34	5,72	2,44	19
	BYTE	179	141	177	74	78	38	45	21	1,53	1,49	2,29	1,03	19
	WORD	197	158	199	82	78	39	46	20	1,93	1,97	3,17	1,55	19
	Suche ungleiche													
	INT	198	159	200	83	79	39	46	21	1,93	1,93	3,17	1,52	19
	DINT	201	163	204	84	79	37	46	21	6,49	6,47	8,63	3,82	19
	BYTE	179	141	178	73	79	38	47	19	1,54	1,51	2,29	1,05	19
	WORD	198	159	200	83	79	39	46	21	1,93	1,93	3,17	1,52	19
	Suche größer als													
	INT	198	160	200	82	79	37	47	19	3,83	3,83	5,62	2,59	19
	DINT	206	167	210	88	78	38	46	20	8,61	8,61	11,29	4,88	19
	BYTE	181	143	178	73	79	37	45	19	3,44	3,44	4,69	2,03	19
	WORD	198	160	200	82	79	37	47	19	3,83	3,83	5,62	2,59	19
	Suche größer als/gleich													
	INT	197	160	200	83	77	38	46	20	3,86	3,83	5,62	2,52	19
	DINT	205	167	210	87	80	39	46	21	8,62	8,61	11,30	4,87	19
	BYTE	180	142	178	75	79	37	46	20	3,47	3,44	4,69	2,00	19
	WORD	197	160	200	83	77	38	46	20	3,86	3,83	5,62	2,52	19
	Suche kleiner als													
	INT	199	159	201	84	78	38	46	20	3,83	3,86	5,59	2,48	19
	DINT	206	168	210	87	79	38	45	19	8,62	8,60	11,29	4,88	19
	BYTE	181	143	178	75	80	38	46	20	3,44	3,44	4,69	2,00	19
	WORD	199	159	201	84	78	38	46	20	3,83	3,86	5,55	2,48	19
	Suche kleiner als/gleich													
INT	200	158	200	82	79	38	46	21	3,79	3,90	5,59	2,55	19	
DINT	207	167	209	88	78	39	46	19	8,60	8,61	11,30	4,86	19	
BYTE	180	143	178	74	78	40	46	19	3,46	3,44	4,69	2,02	19	
WORD	200	158	200	82	79	38	46	21	3,79	3,90	5,59	2,55	19	
Konvertierung	Umwandlung in INT	74	46	57	25	42	1	0	1	–	–	–	–	9
	Umwandlung in BCD–4	77	50	60	25	42	1	0	1	–	–	–	–	9

**Hinweise:**

1. Die Zeit (in Mikrosekunden) ist angegeben auf der Grundlage von Ausgabestand 5,0 der Logicmaster 90-30/20 Software für CPU-Modelle 311, 313, 331, 340 und 341.
2. Bei den Tabellenfunktionen ist das Inkrement in Einheiten der angegebenen Länge; bei Bitoperationen in Mikrosekunden/Bit; bei Datenverschiebefunktionen in Mikrosekunden/Anzahl Bits oder Worte.
3. Freigegebene Zeit für Einheiten einfacher Länge der Typen %R, %AI und %AW.
4. Die COMMREQ-Zeit wurde gemessen zwischen CPU und HSC.
5. DOIO ist die Zeit für die Ausgabe von Werten zu diskreten Ausgangsmodulen.
6. Gibt es mehr als einen möglichen Fall, gibt die in der Tabelle angegebene Zeit den ungünstigsten Fall an.

**Ausführungszeiten für die Micro-SPS:** Angaben hierzu in GFK-1065B (oder später), *SPS Series90 Micro, Anwenderhandbuch*.  
**Ausführungszeiten für die SPS 351:** Siehe ab Seite A-5

Tabelle A-1 Befehlsausführungszeiten – Fortsetzung

Funktions- gruppe	Funktion	Freigegeben				Gespart				Inkrement				Größe
		311	313	331	340/41	311	313	331	340/41	311	313	331	340/41	
Steuer- funktionen	Unterprogrammaufruf	155	93	116	85	41	0	0	0	–	–	–	–	7
	E/A aktualisieren	309	278	355	177	38	1	0	0	–	–	–	–	12
	PID – ISA-Algorithmus	1870	1827	2311	929	91	56	71	30	–	–	–	–	15
	PID – IND-Algorithmus	2047	2007	2529	1017	91	56	71	30	–	–	–	–	15
	End Instruction	–	–	–	–	–	–	–	–	–	–	–	–	–
	Bedienanforderung													
	# 6	93	54	68	45	41	2	0	0	–	–	–	–	9
	# 7 (lesen)	–	37	363	161	–	2	0	0	–	–	–	–	9
	# 7 (einstellen)	–	37	363	161	–	2	0	0	–	–	–	–	9
	#14	447	418	599	244	41	2	0	0	–	–	–	–	9
	#15	281	243	305	139	41	2	0	0	–	–	–	–	9
	#16	131	104	131	69	41	2	0	0	–	–	–	–	9
	#18	–	56	365	180	–	2	0	0	–	–	–	–	9
	#23	1689	1663	2110	939	43	1	0	0	–	–	–	–	9
	#26//30*	1268	1354	8774	3538	42	0	0	0	–	–	–	–	9
	#29	–	–	58	41	–	–	1	0	–	–	–	–	9
Geschacht. MCR/ENDMCR kombiniert	135	73	88	39	75	25	28	12	–	–	–	–	8	
COMM_REQ				884		2	0	0	–	–	–	–	13	
	1317	1272	1577		41									

\* Bedienanforderung #26/30 wurde mit schnellem Zähler und 16-Punkte-Ausgangsmodul in Chassis mit 5 Steckplätzen gemessen.

#### Hinweise:

1. Die Zeit (in Mikrosekunden) ist angegeben auf der Grundlage von Ausgabestand 5,0 der Logicmaster 90-30/20 Software für CPU-Modelle 311, 313, 331, 340 und 341.
2. Bei den Tabellenfunktionen ist das Inkrement in Einheiten der angegebenen Länge; bei Bitoperationen in Mikrosekunden/Bit; bei Datenverschiebefunktionen in Mikrosekunden/Anzahl Bits oder Worte.
3. Freigegebene Zeit für Einheiten einfacher Länge der Typen %R, %AI und %AW.
4. Die COMMREQ-Zeit wurde gemessen zwischen CPU und HSC.
5. DOIO ist die Zeit für die Ausgabe von Werten zu diskreten Ausgangsmodulen.
6. Gibt es mehr als einen möglichen Fall, gibt die in der Tabelle angegebene Zeit den ungünstigsten Fall an.

**Ausführungszeiten für die Micro-SPS:** Angaben hierzu in GFK-1065B (oder später), *SPS Series90 Micro, Anwenderhandbuch*.

**Ausführungszeiten für die SPS 351:** Siehe ab Seite A-5

**Tabelle A-1 Befehlsausführungszeiten – Fortsetzung**

Funktions- gruppe	Funktion	Freigegeben	Gesperrt	Inkrement	Größe
		351	351	351	
Timer	Einschaltverzögerung	4	4	–	15
	Timer	2	3	–	15
	Ausschaltverzögerung	2	2	–	15
Zähler	Aufwärtszähler	2	2	–	13
	Abwärtszähler	2	2	–	13
Arithmetik- funktionen	Addition (INT)	1	0	–	13
	Addition (DINT)	2	0	–	19
	Addition (REAL) <i>nur 352</i>	33	0	–	17
	Subtraktion (INT)	1	0	–	13
	Subtraktion (DINT)	2	0	–	19
	Subtraktion (DINT)	2	0	–	19
	Subtraktion (REAL) <i>nur 352</i>	34	0	–	17
	Multiplikation (INT)	21	0	–	13
	Multiplikation (DINT)	24	0	–	19
	Multiplik. (REAL) <i>nur 352</i>	38	1	–	17
	Division (INT)	22	0	–	13
	Division (DINT)	25	0	–	19
	Division (REAL) <i>nur 352</i>	36	2	–	17
	Modulo-Division (INT)	21	0	–	13
	Modulo-Division (DINT)	25	0	–	19
	Quadratwurzel (INT)	41	1	–	10
	Quadratwurzel (DINT)	76	0	–	13
Quadratw. (REAL) <i>nur 352</i>	35	0	–	11	
Trigonom. Funktionen	SIN (REAL) <i>nur 352</i>	32	0	–	11
	COS (REAL) <i>nur 352</i>	29	0	–	11
	TAN (REAL) <i>nur 352</i>	32	1	–	11
	ASIN (REAL) <i>nur 352</i>	45	0	–	11
	ACOS (REAL) <i>nur 352</i>	63	0	–	11
	ATAN (REAL) <i>nur 352</i>	33	1	–	11
Logarithmus	LOG (REAL) <i>nur 352</i>	32	0	–	11
	LN (REAL) <i>nur 352</i>	32	0	–	11
Exponential	EXP <i>nur 352</i>	42	0	–	11
	EXPT <i>nur 352</i>	54	1	–	17
Bogenmaß- konvert.	RAD → DEG <i>nur 352</i>	32	1	–	11
	DEG → RAD <i>nur 352</i>	32	0	–	11

**Hinweise:**

1. Die Zeit (in Mikrosekunden) ist angegeben auf der Grundlage von Ausgabestand 7 der Logicmaster 90-30/20 Software für CPU-Modelle 351 und 352.
2. Bei den Tabellenfunktionen ist das Inkrement in Einheiten der angegebenen Länge; bei Bitoperationen in Mikrosekunden/Bit; bei Datenverschiebefunktionen in Mikrosekunden/Anzahl Bits oder Worte.
3. Freigegebene Zeit für Einheiten einfacher Länge der Typen %R, %AI und %AW.
4. Die COMMREQ-Zeit wurde gemessen zwischen CPU und HSC.
5. DOIO ist die Zeit für die Ausgabe von Werten zu diskreten Ausgangsmodulen.
6. Gibt es mehr als einen möglichen Fall, gibt die in der Tabelle angegebene Zeit den ungünstigsten Fall an.

Tabelle A-1 Befehlsausführungszeiten – Fortsetzung

Funktions- gruppe	Funktion	Freigegeben	Gesperrt	Inkrement	Größe
		351	351	351	
Relationale Funktionen	Gleich (INT)	1	0	–	10
	Gleich (DINT)	2	0	–	16
	Gleich (REAL) <i>nur 352</i>	33	0	–	14
	Ungleich (INT)	1	0	–	10
	Ungleich (DINT)	1	0	–	16
	Ungleich (REAL) <i>nur 352</i>	31	0	–	14
	Größer als (INT)	1	0	–	10
	Größer als (DINT)	1	0	–	16
	Größer als (REAL) <i>nur 352</i>	32	0	–	14
	Größer als/gleich (INT)	1	0	–	10
	Größer als/gleich (DINT)	1	0	–	16
	Gr. als/gl. (REAL) <i>nur 352</i>	36	1	–	14
	Kleiner als (INT)	1	0	–	10
	Kleiner als (DINT)	1	0	–	16
	Kleiner als (REAL) <i>nur 352</i>	36	1	–	14
	Kleiner als/gleich (INT)	1	0	–	10
	Kleiner als/gleich (DINT)	3	0	–	16
	Kl. als/gl. (REAL) <i>nur 352</i>	3	0	–	16
	Bereich (INT)	2	1	–	13
	Bereich (DWORD)	2	1	–	22
Bereich (WORD)	1	0	–	13	
Bit- operations- funktionen	Logisch UND	2	0	–	13
	Logisch ODER	2	0	–	13
	Logische Antivalenz	1	0	–	13
	Logische Invertierung, NOT	1	0	–	10
	Nach links verschieben	31	1	1,37	16
	Nach rechts verschieben	28	0	3,03	16
	Nach links rotieren	25	0	3,12	16
	Nach rechts rotieren	25	0	4,14	16
	Bitposition	20	1	–	13
	Bit auf 0 setzen	20	0	–	13
	Bit testen	20	0	–	13
	Bit auf 1 setzen	19	1	–	13
	Vergl. m. Maske (WORD)	46	0	–	25
	Vergl. m. Maske (DWORD)	48	0	–	25
	Kopieren (INT)	0	0	0,41	10
	Kopieren (BIT)	28	0	4,98	13
	Kopieren (WORD)	1	1	0,41	10
	Kopieren (REAL) <i>nur 352</i>	24	1	0,82	13
	Block kopieren (INT)	3	0	–	28
	Block kopieren (WORD)	3	0	–	28
	Block kop. (REAL) <i>nur 352</i>	36	0	–	13
	Block löschen	1	0	0,24	11
	Schieberegister (BIT)	46	0	0,23	16
	Schieberegister (WORD)	27	0	0,41	16
	Bitfolgesteuerung	38	22	0,02	16

**Hinweise:**

1. Die Zeit (in Mikrosekunden) ist angegeben auf der Grundlage von Ausgabestand 7 der Logicmaster 90-30/20 Software für CPU-Modelle 351 und 352.
2. Bei den Tabellenfunktionen ist das Inkrement in Einheiten der angegebenen Länge; bei Bitoperationen in Mikrosekunden/Bit; bei Datenverschiebefunktionen in Mikrosekunden/Anzahl Bits oder Worte.
3. Freigegebene Zeit für Einheiten einfacher Länge der Typen %R, %AI und %AW.
4. Die COMMREQ-Zeit wurde gemessen zwischen CPU und HSC.
5. DOIO ist die Zeit für die Ausgabe von Werten zu diskreten Ausgangsmodulen.
6. Gibt es mehr als einen möglichen Fall, gibt die in der Tabelle angegebene Zeit den ungünstigsten Fall an.
7. Bei Anweisungen mit Inkrementalwert Inkrement mit (Länge –1) multiplizieren und diesen Wert zur Grundzeit addieren.

**Tabelle A-1 Befehlsausführungszeiten – Fortsetzung**

Funktions- gruppe	Funktion	Freigegeben	Gesperrt	Inkrement	Größe	
		351	351	351		
Tabellen- funktionen	Feld kopieren					
	INT	54	0	0,97	22	
	DINT	54	0	0,81	22	
	BIT	69	0	0,36	22	
	BYTE	54	1	0,64	22	
	WORD	54	0	0,97	22	
	Suche gleiche					
	INT	37	0	0,62	19	
	DINT	41	1	1,38	22	
	BYTE	35	0	0,46	19	
	WORD	37	0	0,62	19	
	Suche ungleiche					
	INT	37	0	0,62	19	
	DINT	38	0	2,14	22	
	BYTE	37	0	0,47	19	
	WORD	37	0	0,62	19	
	Suche größer als					
	INT	37	0	1,52	19	
	DINT	39	0	2,26	22	
	BYTE	36	1	1,24	19	
	WORD	37	0	1,52	19	
	Suche größer als/gleich					
	INT	37	0	1,48	19	
	DINT	39	0	2,33	22	
	BYTE	37	1	1,34	19	
	WORD	37	0	1,48	19	
	Suche kleiner als					
	INT	37	0	1,52	19	
	DINT	41	1	2,27	22	
	BYTE	37	0	1,41	19	
	WORD	37	0	1,52	19	
	Suche kleiner als/gleich					
	INT	38	0	1,48	19	
	DINT	40	1	2,30	22	
	BYTE	37	0	1,24	19	
	WORD	38	0	1,48	19	

**Hinweise:**

1. Die Zeit (in Mikrosekunden) ist angegeben auf der Grundlage von Ausgabestand 7 der Logicmaster 90-30/20 Software für CPU-Modelle 351 und 352.
2. Bei den Tabellenfunktionen ist das Inkrement in Einheiten der angegebenen Länge; bei Bitoperationen in Mikrosekunden/Bit; bei Datenverschiebefunktionen in Mikrosekunden/Anzahl Bits oder Worte.
3. Freigegebene Zeit für Einheiten einfacher Länge der Typen %R, %AI und %AW.
4. Die COMMREQ-Zeit wurde gemessen zwischen CPU und HSC.
5. DOIO ist die Zeit für die Ausgabe von Werten zu diskreten Ausgangsmodulen.
6. Gibt es mehr als einen möglichen Fall, gibt die in der Tabelle angegebene Zeit den ungünstigsten Fall an.
7. Bei Anweisungen mit Inkrementalwert Inkrement mit (Länge -1) multiplizieren und diesen Wert zur Grundzeit addieren.



Tabelle A-1 Befehlsausführungszeiten – Fortsetzung

Funktions- gruppe	Funktion	Freigegeben	Gesperrt	Inkrement	Größe
		351	351	351	
Konvertie- rungs-fkt.	Umwandlung in INT	19	1	–	10
	Umwandlung in BCD-4	21	1	–	10
	Umwandl. in REAL <i>nur 352</i>	21	0	–	8
	Umwandl. in WORD <i>nur 352</i>	30	1	–	11
	Kürzen auf INT <i>nur 352</i>	32	0	–	11
	Kürzen auf DINT <i>nur 352</i>	31	0	–	11
Steuer- funktionen	Unterprogrammaufruf	40	1	–	7
	E/A aktualisieren (DOIO)	123	1	–	13
	PID-ISA-Algorithmus*	162	34	–	16
	PID-IND-Algorithmus	146	34	–	16
	Endeanweisung	–	–	–	–
	Bedienanforderung				
	#6	22	1	–	10
	#7 (lesen)	75	1	–	10
	#7 (einstellen)	75	1	–	10
	#14	121	1	–	10
	#15	46	1	–	10
	#16	36	1	–	10
	#18	261	1	–	10
	#23	426	0	–	10
	#26//30**	2910	1	–	10
	#29	20	0	–	10
Geschachtelte MCR/ ENDMCR kombiniert	1	1	–	4	
COMM_REQ	732	0	–	13	

\* Die PID-Zeiten sind angegeben auf der Grundlage von Ausgabestand 6.5 der CPU 351.

\*\* Bedienanforderung #26/30 wurde mit schnellem Zähler und 16-Punkte-Ausgangsmodul in Chassis mit 5 Steckplätzen gemessen.

#### Hinweise:

1. Die Zeit (in Mikrosekunden) ist angegeben auf der Grundlage von Ausgabestand 7 der Logicmaster 90-30/20 Software für CPU-Modelle 351 und 352.
2. Bei den Tabellenfunktionen ist das Inkrement in Einheiten der angegebenen Länge; bei Bitoperationen in Mikrosekunden/Bit; bei Datenverschiebefunktionen in Mikrosekunden/Anzahl Bits oder Worte.
3. Freigegebene Zeit für Einheiten einfacher Länge der Typen %R, %AI und %AW.
4. Die COMMREQ-Zeit wurde gemessen zwischen CPU und HSC.
5. DOIO ist die Zeit für die Ausgabe von Werten zu diskreten Ausgangsmodulen.
6. Gibt es mehr als einen möglichen Fall, gibt die in der Tabelle angegebene Zeit den ungünstigsten Fall an.
7. Bei Anweisungen mit Inkrementalwert Inkrement mit (Länge –1) multiplizieren und diesen Wert zur Grundzeit addieren.

## Befehlsängen bei CPUs 351 und 352

Die Befehlslänge gibt an, wieviele Speicherbytes der Befehl in einem Kontaktplanprogramm benötigt. Bei den CPUs 351 und 352 benötigen die meisten Boolesche Standardfunktionen drei Bytes (siehe Tabelle A-2).

Tabelle A-2 Befehlsängen bei CPU 351 und 352

Funktion	Größe
Keine Operation	1
Nimm von Stack und UNDe an Anfang	1
Nimm von Stack und ODERe an Anfang	1
Kopiere Stackanfang	1
Nimm von Stack	1
Initialisiere Stack	1
Sprungziel (LABEL)	5
Sprung (JUMP)	5
Alle anderen Anweisungen	3
Funktionsblöcke — siehe Tabelle A-1	–



# Anhang B

## Fehlertabellen-Interpretation

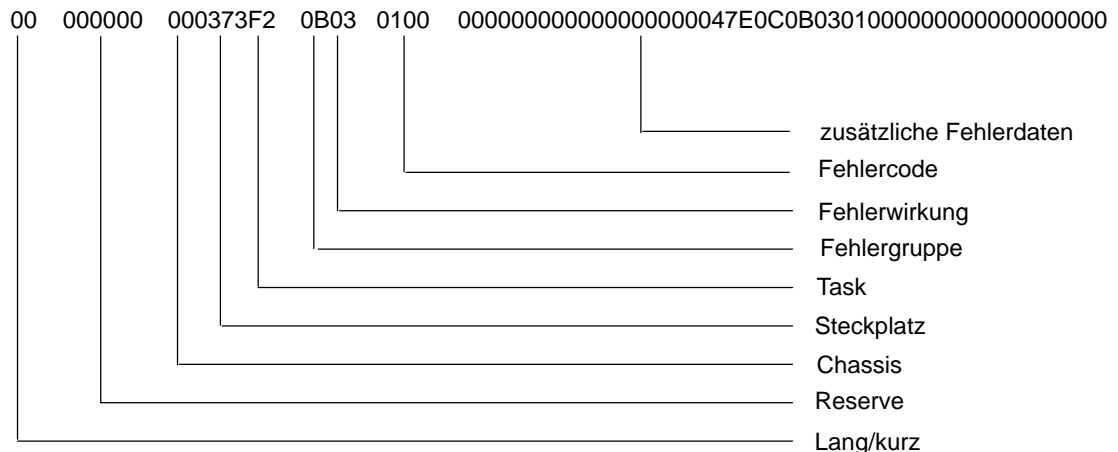
Die speicherprogrammierbaren Steuerungen der Series 90-30, 90-20 und Micro verwalten zwei Fehlertabellen, die E/A-Fehlertabelle für Fehler, die von E/A-Geräten erzeugt wurden (einschließlich E/A-Controller) und die SPS-Fehlertabelle für interne SPS-Fehler. Die Informationen in diesem Anhang sollen Ihnen helfen, das Format der Meldungsstruktur beim Lesen dieser Fehlertabellen zu interpretieren. In beiden Tabellen finden Sie ähnliche Informationen.

- Die SPS-Fehlertabelle enthält:
  - Fehlerort
  - Fehlerbeschreibung
  - Datum und Zeit des Fehlers
- Die E/A-Fehlertabelle enthält:
  - Fehlerort
  - Referenzadresse
  - Fehlerkategorie
  - Fehlertyp
  - Datum und Zeit des Fehlers

### SPS-Fehlertabelle

Rufen Sie die SPS-Fehlertabelle über Ihre Programmiersoftware auf. Wenn Sie Logicmaster verwenden, finden Sie eine Beschreibung des Zugriffs auf die Fehlertabellen in Kapitel 5 von *Logicmaster 90™ Series 90™-30/20/Micro Programmiersoftware, Anwenderhandbuch* (GFK-0466).

Das nachstehende Diagramm zeigt die Bedeutung der einzelnen Felder in dem oben gezeigten Systemkonfigurations-Diskrepanz-Menü:



Der Fehler "Diskrepanz bei Systemkonfiguration" wird nachstehend erläutert (alle Daten in Hexadezimaldarstellung).

<b>Feld</b>	<b>Wert</b>	<b>Beschreibung</b>
Lang/kurz	00	Dieser Fehler enthält 8 Bytes zusätzliche Fehlerdaten.
Chassis	00	Hauptchassis (Chassis 0).
Steckplatz	03	Steckplatz 3.
Task	44	
Fehlergruppe	0B	Diskrepanz bei Systemkonfiguration
Fehlerwirkung	03	FATALER Fehler
Fehlercode	01	

In den folgenden Abschnitten werden die einzelnen Felder des Fehlereintrags erläutert. Außerdem wird tabellarisch angegeben, welche Werte die einzelnen Felder annehmen können.

### **Anzeige Lang/kurz**

Dieses Byte gibt an, ob der Fehler 8 oder 24 Byte zusätzlicher Fehlerdaten enthält.

<b>Typ</b>	<b>Code</b>	<b>Zusätzliche Fehlerdaten</b>
Kurz	00	8 Bytes
Lang	01	24 Bytes

### **Reserve**

Mit diesen sechs Füllbytes wird der SPS-Fehlertabelleneintrag genauso lang wie der E/A-Fehlertabelleneintrag.

### **Chassis**

Die Chassisnummer liegt zwischen 0 und 7. Null ist das Hauptchassis mit der CPU. Die Chassis 1 bis 7 sind Erweiterungschassis, die mit dem CPU-Chassis über ein Erweiterungskabel verbunden sind.

### **Steckplatz**

Die Steckplatznummer liegt zwischen 0 und 9. Die CPU der SPS belegt immer Steckplatz 1 im Hauptchassis (Chassis 0).

### **Task**

Die Tasknummer liegt zwischen 0 und +65,535. Manchmal liefert die Tasknummer zusätzliche Informationen für SPS-Ingenieure. Normalerweise kann die Tasknummer aber ignoriert werden.

## SPS-Fehlergruppe

Fehlergruppe ist die oberste Klassifizierungsebene eines Fehlers. Sie gibt die allgemeine Fehlerkategorie an. Der von der Logicmaster 90-30/20/Micro Software angezeigte Text basiert auf Fehlergruppe und Fehlercodes.

Tabelle B-1 listet die in der SPS-Fehlertabelle möglichen Fehlergruppen auf.

Die letzte nicht maskierbare Fehlergruppe, *weitere SPS-Fehlercodes*, gilt für die Bearbeitung neuer Fehlerzustände im System, ohne daß die SPS die Alarmcodes besonders kennen muß. Sämtliche nicht identifizierten SPS-Alarmcodes gehören zu dieser Gruppe.

**Tabelle B-1 SPS-Fehlergruppen**

Gruppennummer		Gruppenname	Fehlerwirkung
Dezimal	Hexadezimal		
1	1	Verlorenes oder fehlendes Chassis	Fatal
4	4	Verlorenes oder fehlendes Zusatzmodul	Diagnose
5	5	Hinzugefügtes oder überzähliges Chassis	Diagnose
8	8	Hinzugefügtes oder überzähliges Zusatzmodul	Diagnose
11	B	Diskrepanz bei Systemkonfiguration	Fatal
12	C	Systembusfehler	Diagnose
13	D	Hardwarefehler bei SPS-CPU	Fatal
14	E	Nicht-fataler Modulhardwarefehler	Diagnose
16	10	Zusatzmodul-Softwarefehler	Diagnose
17	11	Programmblock-Prüfsummenfehler	Fatal
18	12	Signal für niedrige Batteriespannung	Diagnose
19	13	Konstante Zykluszeit überschritten	Diagnose
20	14	SPS-Systemfehlertabelle ist voll	Diagnose
21	15	E/A-Fehlertabelle ist voll	Diagnose
22	16	Anwenderfehler	Diagnose
-	-	Weitere SPS-Fehlercodes	Wie angegeben
128	80	Systembusfehler	Fatal
129	81	Kein Anwenderprogramm beim Einschalten	Information
130	82	Verstümmeltes Anwender-RAM erkannt	Fatal
132	84	Paßwortfehler	Information
135	87	Softwarefehler bei SPS-CPU	Fatal
137	89	SPS-Speicherfehler	Fatal

## Fehlerwirkung

Mit jedem Fehler kann eine von drei Fehlerwirkungen verknüpft sein. Bei der SPS Series 90-30 sind die Fehlerwirkungen unveränderlich festgelegt.

**Tabelle B-2 SPS-Fehlerwirkungen**

Fehlerwirkung	von CPU durchgeführte Aktion	Code
Information	Trägt Fehler in Fehlertabelle ein	1
Diagnose	Trägt Fehler in Fehlertabelle ein Setzt Fehlerreferenzen	2
Fatal	Trägt Fehler in Fehlertabelle ein Setzt Fehlerreferenzen Geht auf STOP	3

## Fehlercode

Der Fehlercode liefert eine weitere Beschreibung des Fehlers. Jede Fehlergruppe besitzt ihren eigenen Satz Fehlercodes. Tabelle B-3 zeigt die Fehlercodes für Gruppe 87H (SPS-Softwarefehler).

**Tabelle B-3 Alarm-Fehlercodes für SPS-CPU-Softwarefehler**

Dezimal	Hexadezimal	Name
20	14	VerstümmelterSPS-Programmspeicher
39	27	VerstümmelterSPS-Programmspeicher
82	52	Rückwandplatinen-Kommunikationfehlerhaft
90	5A	Anwender fordert Systemabschaltung
Alle übrigen		Interner SPS-CPU Systemfehler

Tabelle B-4 zeigt die Fehlercodes für alle anderen Fehlergruppen.

**Tabelle B-4 Alarm-Fehlercodes für SPS-CPU-Fehler**

Dezimal	Hexadezimal	Name
<b><i>SPS-Fehlercodes für Gruppe "Verlust von Zusatzmodul"</i></b>		
44	2C	Software-Rücksetzen von Zusatzmodul gescheitert
45	2D	Software-Rücksetzen von Zusatzmodul gescheitert
255	FF	Kommunikation mit Zusatzmodul fehlerhaft
<b><i>Fehlercodes für Gruppe "Rücksetzen von, hinzugefügtes oder überzähliges Zusatzmodul"</i></b>		
2	2	Modul-Neustart abgeschlossen
	Alle übrigen	Rücksetzen von, hinzugefügtes oder überzähliges Zusatzmodul
<b><i>Fehlercodes für Gruppe "Zusatzmodul-Softwarefehler"</i></b>		
1	1	Modultyp nicht unterstützt
2	2	COMREQ-Mailbox voll bei ausgehender Meldung, die COMREQ startet.
3	3	COMREQ-Mailbox voll bei Antwort
5	5	Rückwandplatinenkommunikation mit SPS; verlorene Anforderung
11	B	Betriebsmittelfehler (alloc, tbl ovrflw, usw.)
13	D	Anwenderprogrammfehler
401	191	Verstümmelte Modulsoftware; Neuladen erforderlich
<b><i>Fehlercodes für Gruppe "Diskrepanz bei Systemkonfiguration"</i></b>		
8	8	Diskrepanz bei Analogenerweiterung
10	A	Eigenschaft nicht unterstützt
23	17	Programm überschreitet Speichergrenzen
<b><i>Fehlercodes für Gruppe "Systembusfehler"</i></b>		
	Alle übrigen	Systembusfehler
<b><i>Fehlercodes für Gruppe "Programmblock-Prüfsummenfehler"</i></b>		
3	3	Prüfsummenfehler bei Programm oder Programmblock
<b><i>Fehlercodes für Signal "niedrige Batteriespannung"</i></b>		
0	0	Batterie in CPU- oder anderem Modul ausgefallen
1	1	Batteriespannung in CPU- oder anderem Modul niedrig
<b><i>Fehlercodes für Gruppe "Anwenderprogrammfehler"</i></b>		
2	2	SPS-Zeitüberwachung (Watchdog) abgelaufen
5	5	COMREQ WAIT-Modus für diesen Befehl nicht verfügbar
6	6	COMREQ – falsche Task-ID.
7	7	Überlauf Anwenderstack
<b><i>Fehlercodes für Gruppe "Systembusfehler"</i></b>		
1	1	Betriebssystem.
<b><i>Fehlercodes für Gruppe "Verstümmeltes Anwender-RAM beim Einschalten"</i></b>		
1	1	Verstümmeltes Anwender-RAM beim Einschalten
2	2	Unzulässiger Boolescher Operationscode erkannt
3	3	PLC_ISCP_PC_OVERFLOW.
4	4	PRG_SYNTAX_ERR.
<b><i>Fehlercodes für SPS-CPU-Hardwarefehler</i></b>		
	Alle Codes	SPS-CPU-Hardwarefehler

**Zusätzliche Fehlerdaten**

Dieses Feld enthält Einzelheiten zum Fehlereintrag. Das folgende Beispiel zeigt, welche Daten vorhanden sein können:

**Gruppe "ver-** Vier der Fehlercodes in der Gruppe "Diskrepanz bei Systemkonfiguration"  
**stümmelter** liefern zusätzliche Fehlerdaten:  
**Anwender-**  
**RAM":**

**Tabelle B-5 SPS-Fehlerdaten – Unzulässiger Boolescher Operationscode erkannt**

Zusätzliche Fehlerdaten	Modellnummer-Diskrepanz
[0]	Inhalt ISCP-Fehlerregister
[1]	Fehlerhafter OPCODE
[2,3]	ISCP-Programmzähler
[4,5]	Funktionsnummer

**SPS-CPU** Bei einem RAM-Fehler in der SPS-CPU (einer der als SPS-CPU-Hardware-  
**Hardware-** fehler gemeldeten Fehler) wird die Adresse des Fehlers in den ersten vier By-  
**fehler (RAM-** tes des Feldes gespeichert.  
**Fehler):**

**SPS-Zeitstempel**

Der sechs Byte lange Zeitstempel gibt den Wert an, den die Systemuhr hatte, als der Fehler von der SPS-CPU gemeldet wurde. Die Werte sind im BCD-Format.

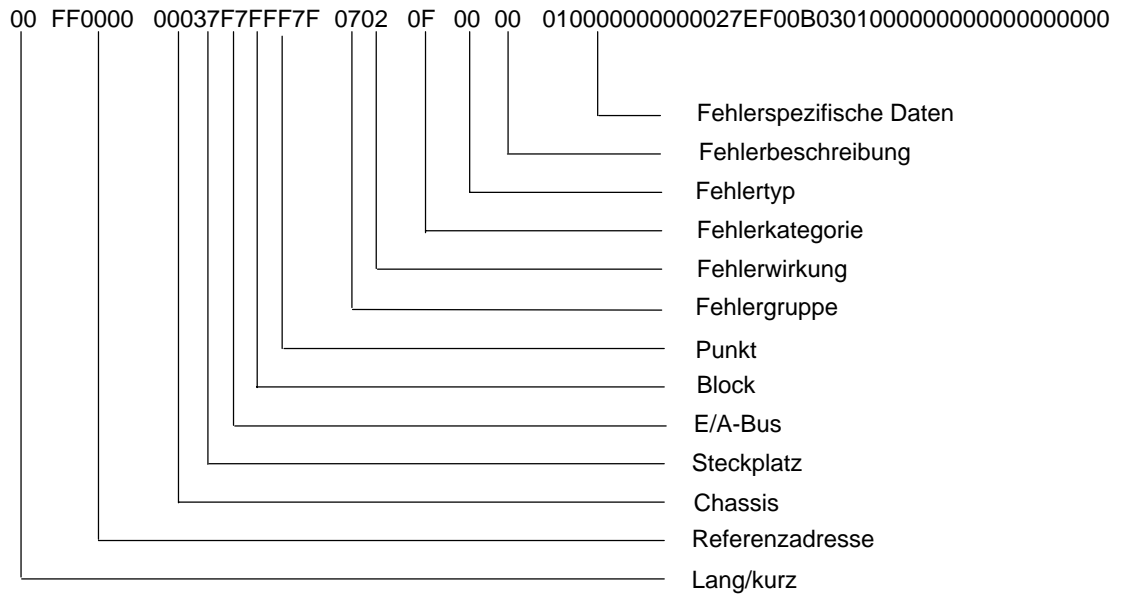
**Tabelle B-6 SPS-Zeitstempel**

Bytenummer	Beschreibung
1	Sekunden
2	Minuten
3	Stunden
4	Tag
5	Monat
6	Jahr



## E/A-Fehlertabelle

Das folgende Diagramm zeigt die in den einzelnen Feldern des Fehlereintrags enthaltenen Hexadezimaldaten



In den folgenden Abschnitten werden die einzelnen Felder der E/A-Fehlertabelle erläutert. Außerdem wird tabellarisch angegeben, welche Werte die einzelnen Felder annehmen können.

### Anzeige Lang/kurz

Dieses Byte gibt an, ob der Fehler 5 oder 21 Byte fehlerspezifische Daten enthält.

**Tabelle B-7 E/A-Fehlertabelle, Formatanzeigebite**

Typ	Code	Fehlerspezifische Daten
Kurz	02	5 Bytes
Lang	03	21 Bytes

### Referenzadresse

Die Referenzadresse besteht aus drei Bytes und enthält den E/A-Speichertyp und die Speicheradresse (oder Offset) in diesem Speicher, die dem fehlerbehafteten Punkt entspricht. Tritt ein Fehler bei einem Genius-Block oder einem Analogmodul auf, zeigt die Referenzadresse auf den ersten Punkt des Blocks, in dem der Fehler aufgetreten ist.

**Tabelle B-8 E/A-Referenzadresse**

Byte	Beschreibung	Bereich
0	Speichertyp	0 – FF
1–2	Offset	0 – 12K (dezimal)

Das Speichertypbyte kann einen der folgenden Werte annehmen:

**Tabelle B-9 Speichertyp der E/A-Referenzadresse**

Name	Wert (Hexadezimal)
Analogeingang	0A
Analogausgang	0C
Analoggruppe	0D
Diskreter Eingang	10 oder 46
Diskreter Ausgang	12 oder 48
Diskrete Gruppe	1F

### E/A-Fehleradresse

Die E/A-Fehleradresse ist sechs Bytes lang und gibt an, bei welchem Chassis, Steckplatz, Bus, Block und Punkt der Fehler aufgetreten ist. Die Punktdressse ist ein Wort, alle anderen Adressen jeweils ein Byte. Ein Fehler muß nicht alle fünf Werte enthalten.

Enthält eine E/A-Fehleradresse nicht alle fünf Adreßwerte, zeigt ein Hexadezimalwert von 7F an, wo die Signifikanz aufhört. Steht zum Beispiel 7F im Busbyte, handelt es sich um einen Modulfehler, bei dem nur die Werte für Chassis und Steckplatz signifikant sind.

## Chassis

Die Chassisnummer liegt zwischen 0 und 7. Null ist das Hauptchassis mit der CPU. Die Chassis 1 bis 7 sind Erweiterungschassis, die mit dem CPU-Chassis über ein Bus-Transmittermodul im Hauptchassis und Bus-Receivermodule in den Erweiterungschassis verbunden sind.

## Steckplatz

Die Steckplatznummer liegt zwischen 0 und 9. Die CPU der SPS belegt immer Steckplatz 1 im Hauptchassis (Chassis 0).

## Punkt

Die Punktnummer liegt zwischen 1 und 1024 (dezimal). Sie gibt bei einem Punktfehler an, welcher Punkt des Blockes fehlerbehaftet ist.

## E/A-Fehlergruppe

Fehlergruppe ist die oberste Klassifizierungsebene eines Fehlers. Sie gibt die allgemeine Fehlerkategorie an. Der von der Logicmaster 90-30/20/Micro Software angezeigte Text basiert auf Fehlergruppe und Fehlercodes.

Tabelle B-10 listet die in der E/A-Fehlertabelle möglichen Fehlergruppen auf. Bei Gruppennummern kleiner als 80 (Hexa) handelt es sich um maskierbare Fehler.

Die letzte nicht maskierbare Fehlergruppe, *weitere E/A-Fehlercodes*, gilt für die Bearbeitung neuer Fehlerzustände im System, ohne daß die SPS die Alarmcodes besonders kennen muß. Sämtliche nicht identifizierten E/A-Alarmcodes gehören zu dieser Gruppe.

**Tabelle B-10 E/A-Fehlergruppen**

Gruppennummer	Gruppenname	Fehlerwirkung
3	Verlorenes oder fehlendes E/A-Modul	Diagnose
7	Hinzugefügtes o. überzähliges E/A-Modul	Diagnose
9	IOC- oder I/O-Bus-Fehler	Diagnose
A	E/A-Modulfehler	Diagnose
–	Weitere E/A-Fehlercodes	Wie angegeben

### E/A-Fehlerwirkung

Die Fehlerwirkung gibt an, welche Maßnahmen die CPU der SPS beim Auftreten eines Fehlers ergreifen soll. Tabelle B-11 gibt die möglichen Fehlerwirkungen an.

**Tabelle B-11 E/A-Fehlerwirkungen**

Fehlerwirkung	von CPU durchgeführte Aktion	Code
Information	Trägt Fehler in Fehlertabelle ein	1
Diagnose	Trägt Fehler in Fehlertabelle ein Setzt Fehlerreferenzen	2
Fatal	Trägt Fehler in Fehlertabelle ein Setzt Fehlerreferenzen Geht auf STOP	3

### E/A-Fehlerdaten

Ein Eintrag in der E/A-Fehlertabelle kann bis zu 5 Bytes fehlerspezifische Daten enthalten.

### Symbolische fehlerspezifische Daten

Tabelle B-12 gibt die für die Konfiguration der Blockschaltkreise benötigten Daten an.

**Tabelle B-12 Fehlerspezifische Daten**

Dezimalzahl	Hexa-Code	Beschreibung
<i>Schaltkreiskonfiguration</i>		
	1	Kreis ist ein Eingang – Tristate.
	2	Kreis ist ein Eingang
	3	Kreis ist ein Ausgang

### Fehlerwirkungen für bestimmte Fehler

Fixierte/unfixierte Schaltkreisfehler werden als informatorische Fehler gemeldet. Alle übrigen sind diagnostisch oder fatal.

Diskrepanz bei Modellnummer oder E/A-Typ sowie nicht vorhandene E/A-Module werden in der SPS-Fehlertabelle unter der Gruppe "Diskrepanz bei Systemkonfiguration" gemeldet. Sie werden nicht in der E/A-Fehlertabelle gemeldet.

### E/A-Zeitstempel

Der sechs Byte lange Zeitstempel gibt den Wert an, den die Systemuhr hatte, als der Fehler von der SPS-CPU gemeldet wurde. Die Werte sind im BCD-Format.

**Tabelle B-13 E/A-Zeitstempel**

Bytenummer	Beschreibung
1	Sekunden
2	Minuten
3	Stunden
4	Tag
5	Monat
6	Jahr

# Anhang C

## Mnemonische Programmieranweisungen

Im Programmanzeige-/Editiermodus können Sie auf eine schnelle Art Programmieranweisungen eingeben oder suchen, indem Sie ein "&"-Zeichen und die mnemonische Bezeichnung der Anweisung eingeben. Bei einigen Anweisungen können Sie auch eine Referenzadresse oder eine symbolische Adresse, einen Kennsatz (Label) oder eine Speicher-Referenzadresse eingeben.

Die Tabellen in diesem Anhang enthalten die mnemonischen Bezeichnungen sämtlicher Programmieranweisungen der Logicmaster 90-30/20/Micro Programmiersoftware. Die komplette Mnemonik steht in Spalte 3 der Tabelle, die kürzestmögliche Eingabe in Spalte 4.

Während der Programmierung können Sie jederzeit ein Hilfsmenü zu diesen mnemonischen Bezeichnungen aufrufen, indem Sie die Tastenkombination ALT-I drücken.

Funktionsgruppe	Anweisung	Mnemonik						
		All	INT	DINT	BIT	BYTE	WORD	REAL
Kontakte	Beliebiger Kontakt	&CON	&CON					
	Schließerkontakt	&NOCON	&NOCON					
	Öffnerkontakt	&NCCON	&NCCON					
	Fortsetzkontakt	&CONC	&CONC					
Spulen	Beliebige Spule	&COI	&COI					
	Schließer	&NOCOI	&NOCOI					
	Negiert	&NCCOI	&NCCOI					
	Positiver Übergang	&PCOI	&PCOI					
	Negativer Übergang	&NCOI	&NCOI					
	SET	&SL	&SL					
	RESET	&RL	&RL					
	SET remanent	&SM	&SM					
	RESET remanent	&RM	&RM					
	Remanent	&NOMC	&NOMC					
	Negiert remanent	&NCM	&NCM					
Fortsetzspule	&COILC	&COILC						
Verbindungen	Horizontal	&HO	&HO					
	Vertikal	&VE	&VE					
Timer	Einschaltverzögerung	&ON	&ON					
	Zeiterfassung	&TM	&TM					
	Ausschaltverzögerung	&OF	&OF					
Zähler	Aufwärtszähler	&UP	&UP					
	Abwärtszähler	&DN	&DN					

Funktions- gruppe	Anweisung	Mnemonic						
		All	INT	DINT	BIT	BYTE	WORD	REAL
Arithmetische Funktionen	Addition	&AD	&AD_I	&AD_DI				&AD_R
	Subtraktion	&SUB	&SUB_I	&SUB_DI				&SUB_R
	Multiplikation	&MUL	&MUL_I	&MUL_DI				&MUL_R
	Division	&DIV	&DIV_I	&DIV_DI				&DIV_R
	Modulo	&MOD	&MOD_I	&MOD_DI				&MOD_R
	Quadratwurzel	&SQ	&SQ_I	&SQ_DI				&SQ_R
	Sinus	&SIN						
	Kosinus	&COS						
	Tangens	&TAN						
	Arkussinus	&ASIN						
	Arkuskosinus	&ACOS						
	Arkustangens	&ATAN						
	Zehnerlogarithmus	&LOG						
	Natürlicher Logarithm.	&LN						
	Exponentialfunktion	&EXP						
Potenz von x	&EXPT							
Relationale Funktionen	Gleich	&EQ	&EQ_I	&EQ_DI				&EQ_R
	Ungleich	&NE	&NE_I	&NE_DI				&NE_R
	Größer als	&GT	&GT_I	&GT_DI				&GT_R
	Größer als/gleich	&GE	&GE_I	&GE_DI				&GE_R
	Kleiner als	&LT	&LT_I	&LT_DI				&LT_R
Kleiner als/gleich	&LE	&LE_I	&LE_DI				&LE_R	
Bitoperationen	UND	&AN					&AN_W	
	ODER	&OR					&OR_W	
	Antivalenz	&XO					&XO_W	
	NICHT	&NOT					&NOT_W	
	Nach links schieben	&SHL					&SHL_W	
	Nach rechts schieben	&SHR					&SHR_W	
	Nach links rotieren	&ROL					&ROL_W	
	Nach rechts rotieren	&ROR					&ROR_W	
	Bit testen	&BT					&BT_W	
	Bit auf 1 setzen	&BS					&BS_W	
	Bit auf 0 setzen	&BCL					&BCL_W	
Bitposition	&BP					&BP_W		
Vergleich mit Maske	&MCM					&MCM_W		
Daten kopieren	Kopieren	&MOV	&MOV_I		&MOV_BI		&MOV_W	&MOV_R
	Block kopieren	&BLKM	&BLKM_I				&BLKM_W	&BLKM_R
	Block löschen	&BLKC						
	Schieberegister	&SHF			&SHF_BI		&SHF_W	
	Bitfolgesteuerung	&BI						
Kommunikationsanfor- derung	&COMMR							
Tabellen- funktionen	Feld kopieren	&AR	&AR_I	&AR_DI	&AR_BI	&AR_BY	&AR_W	
	Suche gleiche	&SRCHE	&SRCHE_I	&SRCHE_DI		&SRCHE_BY	&SRCH_W	
	Suche ungleiche	&SRCHN	&SRCHN_I	&SRCHN_DI		&SRCHN_BY	&SRCHN_W	
	Suche größer als	&SRCHGT	&SRCHGT_I	&SRCHGT_DI		&SRCHGT_BY	&SRCHGT_W	
	Suche größer als/gleich	&SRCHGE	&SRCHGE_I	&SRCHGE_DI		&SRCHGE_BY	&SRCHGE_W	
	Suche kleiner als	&SRCHLT	&SRCHLT_I	&SRCHLT_DI		&SRCHLT_BY	&SRCHLT_W	
Suche kleiner als/gleich	&SRCHLE	&SRCHLE_I	&SRCHLE_DI		&SRCHLE_BY	&SRCHLE_W		

Funktions- gruppe	Anweisung	Mnemonic						
		All	INT	DINT	BIT	BYTE	WORD	REAL
Steuer- funktionen	Unterprogrammaufruf	&CA						
	E/A aktualisieren (DOIO)	&DO						
	PID-ISA-Algorithmus	&PIDIS						
	PID-IND-Algorithmus	&PIDIN						
	Ende	&END						
	Strompfadkommentar	&COMME						
	System-Bedienanforderung	&SV						
	Hauptsteuerrelais	&MCR						
	Ende Hauptsteuerrelais	&ENDMCR						
	Geschachteltes Hauptsteuerrelais	&MCRN						
	Ende geschachteltes Hauptsteuerrelais	&ENDMCRN						
	Sprung	&JUMP	&JUMP					
	Geschachtelter Sprung	&JUMPN	&JUMPN					
	Sprungmarke	&LABEL	&LABEL					
Geschachtelte Sprungmarke	&LABELN	&LABELN						

Funktions- gruppe	Anweisung	Mnemonic							
		All	BCD-4	INT	DINT	BIT	BYTE	WORD	REAL
Konvertierung	Wandle in ganze Zahl	&TO_INT	&TO_INT_BCD4	&MOV					
	Wandle in doppelgenaue ganze Zahl	&TO_DINT		&BLKM					&BCD4_R
	Wandle in BCD-4	&BCD4		&BLKC	&TO_REAL_DI				
	Wandle in REAL	&TO_REAL		&SHF				&TO_REAL_W	
	Wandle in WORD	&TO_W		&BI					
	Kürze auf ganze Zahl	&TRINT		&COMMR					
	Kürze auf doppelgenaue ganze Zahl	&TRDINT							





# Anhang

## D

# Tastenfunktionen

In diesem Anhang werden die in der Softwareumgebung aktiven Tastaturfunktionen beschrieben. Diese Informationen können Sie auch mit der Tastenkombination ALT-K auf dem Bildschirm des Programmiergerätes anzeigen.

Tastenfolge	Beschreibung	Tastenfolge	Beschreibung
<i>Diese Tasten sind im gesamten Softwarepaket verfügbar</i>			
ALT-A	Abbruch	CTRL-Break	Logicmaster90 verlassen
ALT-C	Feld löschen	Esc	Zu höherer Funktionsebene umschalten
ALT-M	Programmiergerätemodus umschalten	CTRL-Home	Vorherige Befehlszeile anzeigen
ALT-R	SPS-Modus (RUN/STOP) umschalten	CTRL-End	Nächste Befehlszeile anzeigen
ALT-E	Statusbereichumschalten	CTRL- ←	Cursor nach links im Feld bewegen
ALT-J	Befehlszeileumschalten	CTRL- →	Cursor nach rechts im Feld bewegen
ALT-L	Plattendirectory anzeigen	CTRL-D	Referenzadressedekrementieren
ALT-P	Bildschirminhalt drucken	CTRL-U	Referenzadresseinkrementieren
ALT-H	Hilfe	Tab	Eingabefeldändern/inkrementieren
ALT-K	Liste der Tastaturfunktionen anzeigen	Shift-Tab	Eingabefeldändern/dekrementieren
ALT-I	Hilfe, mnemon. Anweisungen	Eingabetaste	Feldinhalt übernehmen
ALT-N	Anzeigeoptionen umschalten	CTRL-E	Anzeige des letzten Systemfehlers
ALT-T	Lernmodus Start	F12 od. Zehnertast. -	Diskrete Referenz umschalten
ALT-Q	Lernmodus Stop	F11 od. Zehnertast. *	Diskrete Referenz überschreiben
ALT-n	Playback-Datei n (n = 0 bis 9).		
<i>Diese Tasten sind nur im Programmierer verfügbar</i>			
ALT-B	Glocke beim Texteditieren ein/aus	Zehnertastatur +	Strompfad übernehmen
ALT-D	Strompfadelement/Strompfad löschen	Eingabetaste	Strompfad übernehmen
ALT-S	Block in SPS und auf Platte speichern	CTRL-PgUp	Vorherigen Strompfad anzeigen
ALT-X	Anzeige Zoomebene	CTRL-PgDn	Nächsten Strompfad anzeigen
ALT-U	Platteaktualisieren	~	Horizontalverbindung
ALT-V	Variablentabellen-Fenster		Vertikalverbindung
ALT-F2	Gehe zu Operandenreferenztafel	Tab	Gehe zum nächsten Operandenfeld
<i>Spezialtasten</i>			
ALT-O	Paßwort überschreiben. Nur im Paßwortmenü der Konfigurationssoftware verfügbar.		

Die Übersicht auf der nächsten Seite enthält eine Liste der Hilfstasten und der Mnemonik-Hilfstele für die Logicmaster 90-30/20/Micro Software.

Logicmaster 90–30 Mnemonische Anweisungen (Hilfe: ALT-I)			
Anweisung	Mnemonic	Anweisung	Mnemonic
Beliebiger Kontakt (Suche)	&CON	Bitoperationen	
-] [-	&NOCON	Logisch UND	&AN
-]/[-	&NCCON	Logisch ODER	&OR
<+>————	&CONC	Exklusiv ODER	&XO
Beliebige Spule (Suche)	&COI	Logische Invertierung	&NOT
-( )-	&NOCOI	Nach links verschieben	&SHL
-( / )-	&NCCOI	Nach rechts verschieben	&SHR
-( ↑ )-	&PCOI	Nach links rotieren	&ROL
-( ↓ )-	&NCOI	Nach rechts rotieren	&ROR
-( S )-	&SL	Bit testen	&BT
-( R )-	&RL	Bit auf 1 setzen	&BS
-( SM )-	&SM	Bit auf 0 setzen	&BCL
-( RM )-	&RM	Suche Bit gleich 1	&BP
-( M )-	&NOMC	Datenverschiebefunktionen	
-( / M )-	&NCM	Verschieben	&MOV
————<+>	&COILC	Block verschieben	&BLKM
Verbindungen		Block löschen	&BLKC
————	&HO	Schieberegister	&SHF
À	&VE	Bitfolgesteuerung	&BI
Timer		Kommunikationsanforderung	&COMMR
Einschaltverzögerung	&ON	Tabellenfunktionen (Suche)	
Zeitmessung	&TM	Feld kopieren	&AR
Ausschaltverzögerung	&OF	Suche gleiche	&SRCHE
Zähler		Suche ungleiche	&SRCHN
Aufwärtszähler	&UP	Suche größer als	&SRCHGT
Abwärtszähler	&DN	Suche größer als oder gleich	&SRCHGE
Arithmetische Funktionen		Suche kleiner als	&SRCHLT
Addition	&AD	Suche kleiner als oder gleich	&SRCHLE
Subtraktion	&SUB	Konvertierungsfunktionen	
Multiplikation	&MUL	Umwandlung in ganze Zahl	&TO_IND
Division	&DIV	Umwandlung in BCD-4	&TO_BCD4
Modulo-Division	&MOD	Umwandlung in WORD	TO_WORD
Quadratwurzel	&SQ	Umwandlung in REAL	TO_REAL
Sinus	&SIN	Abschneiden	&TRINT
Kosinus	&COS	Abschneidenauf DINT	&TRDINT
Tangens	&TAN	Steuerfunktionen	
Arkussinus	&ASIN	Aufruf	&CA
Arkuskosinus	&ACOS	E/A aktualisieren	&DO
Arkustangens	&ATAN	PID-IND Algorithmus	&PIDIS
Zehnerlogarithmus	&LOG	PID-ISA Algorithmus	&PIDIN
Natürlicher Logarithmus	&LN	Programmende	&END
Exponentialfunktion	&EXP	Kommentar	&COMME
Potenz von x	&EXPT	Serviceanforderung	&SV
Relationale Funktionen		Ungeschachtelte MCR	&MCR
Gleich	&EQ	Ende ungeschachtelte MCR	&ENDMCR
Ungleich	&NE	Geschachtelte MCR	&MCRN
Größer als	&GT	Ende geschachtelte MCR	&ENDMCRN
Größer als oder gleich	&GE	Sprung	&JUMP
Kleiner als	&LT	Geschachtelter Sprung	&JUMPN
Kleiner als oder gleich	&LE	Ziel von Sprung	&LABEL
		Ziel von geschachteltem Sprung	&LABELN

Bei den nachstehend aufgelisteten Anweisungen mit Typmodifikator kann der Typmodifikator über einen Unterstrich angehängt werden. Zum Beispiel: &ADD\_INT.

<b>Anweisungstyp</b>	<b>Modifikator</b>	<b>Verwendet mit</b>
Ganze Zahl mit Vorzeichen	_I	Arithmetische, relationale, Datenverschiebungs- u. Tabellenfkt.
Doppeltgenaue ganze Zahl	_DI	Arithmetische, relationale und Tabellenfunktionen
Bit	_BI	Datenverschiebungs- u. Tabellenfunktionen
Byte	_BY	Tabellenfunktionen
Wort	_W	Bitoperationen, Datenverschiebungs- und Tabellenfunktionen
REAL	_R	Bitoperationen, Datenverschiebungs- und Tabellenfunktionen
BCD4	_BCD4	Konvertierungsfunktionen
1/10 Sekunde	_TEN	Timer
1/100 Sekunde	_HUN	Timer



# Anhang E

## Verwendung von Gleitpunktzahlen

Bei der Verwendung von Gleitpunktzahlen gibt es einige Punkte zu beachten. Im ersten Abschnitt werden die allgemeinen Überlegungen behandelt. Anweisungen zur Eingabe und Anzeige von Gleitpunktzahlen finden Sie ab Seite E-4.

### Gleitpunktzahlen

Mit der Logicmaster 90 Software können Sie Zahlen mit reellen Werten bearbeiten, anzeigen, speichern und aufrufen. Einige Funktionen verarbeiten Gleitpunktzahlen. Sie können jedoch Gleitpunktzahlen mit der Logicmaster 90-30/20/Micro Software nur im Zusammenhang mit einer CPU 352 verwenden. Gleitpunktzahlen werden in wissenschaftlicher Dezimalschreibweise dargestellt, wobei sechs signifikante Stellen angezeigt werden.

### Hinweis

In diesem Handbuch werden die Begriffe "Gleitpunkt" und "REAL" gleichberechtigt zur Beschreibung der Gleitpunkt-Anzeige- bzw. Eingabefunktion der Logicmaster Software verwendet.

Bei der Logicmaster 90 Software wird folgendes Format benutzt. Bei Zahlen im Bereich zwischen 999999999 und 0,0001 besitzt die Anzeige keinen Exponenten und bis zu sechs oder sieben signifikante Stellen. Zum Beispiel:

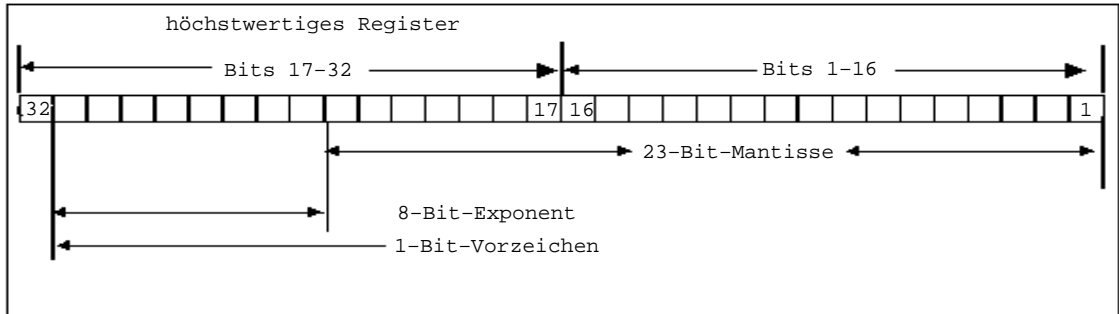
Eingabe	Anzeige	Beschreibung
.000123456789	+.0001234567	Zehn Stellen, sechs oder sieben signifikant
-12.345e-2	-.1234500	Sieben Stellen, sechs oder sieben signifikant
1234	+1234.0000	Sieben Stellen, sechs oder sieben signifikant

Außerhalb des oben aufgeführten Bereichs werden nur sechs Stellen in der folgenden Form angezeigt:

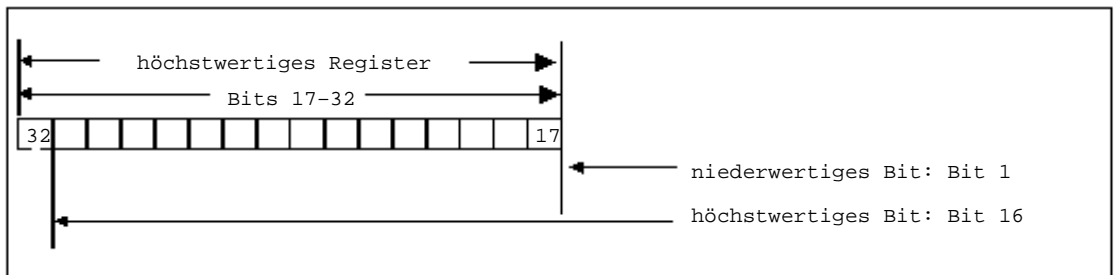
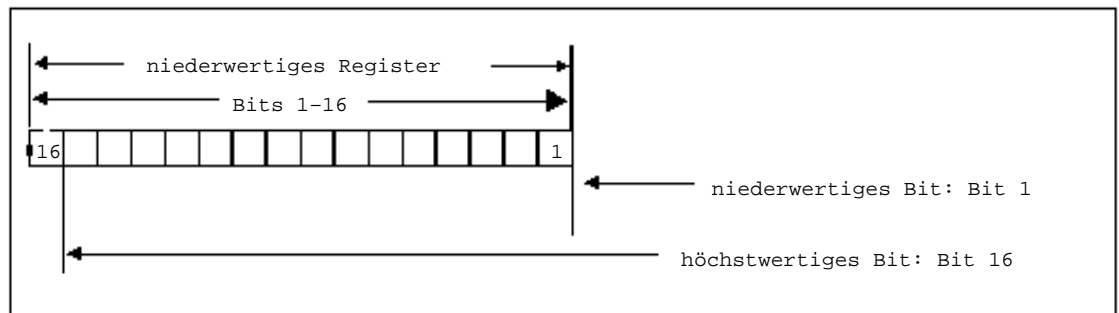
```
+1.23456E+12
||| | |
||| | | +--- Exponent (vorzeichenbehaftete Zehnerpotenz)
||| | |
||| | | +----- Exponentenanzeige und Vorzeichen von Exponent
||| |
||| +----- fünf niedrigwertige Stellen
|||
||+----- Dezimalpunkt
||
|+----- höchstwertige Stelle
+----- Vorzeichen der gesamten Zahl
```

### Internes Format der Gleitpunktzahlen

Gleitpunktzahlen werden im einfachgenauen IEEE-Standardformat gespeichert. Dieses Format benötigt 32 Bits, die in zwei (nebeneinanderliegenden) 16-Bit-SPS-Registern aufgeteilt werden. Die nachstehenden Zeichnungen zeigen die Kodierung der Bits.



Die nachstehenden Zeichnungen zeigen die Registerbelegung durch eine einzelne Gleitpunktzahl. Belegt die Gleitpunktzahl zum Beispiel die Register R5 und R6, ist in diesem Diagramm R5 das niedrigstwertige Register und R6 das höchstwertige Register.



## Werte der Gleitpunktzahlen

Mit der folgenden Tabelle können Sie den Wert einer Gleitpunktzahl aus der in zwei Registern gespeicherten Binärzahl berechnen.

Exponent (e)	Mantisse (f)	Wert der Gleitpunktzahl
255	ungleich 0	keine gültige Zahl
255	beliebiger Wert 0	$-1^S * \infty$
$0 < e < 255$	ungleich Null	$-1^S * 2^{e-127} * 1.f$
0	0	$-1^S * 2^{e-126} * 0.f$
0		0

- f = Die Mantisse. Die Mantisse ist ein binärer Bruch
- e = Der Exponent. Der Exponent ist ein ganzzahliges E, so daß  $E+127$  die Zweierpotenz ist, mit der die Mantisse multipliziert werden muß, um den Gleitpunktwert zu erhalten.
- s = Das Vorzeichenbit
- \* = Der Multiplikationsoperator

Betrachten wir zum Beispiel die Gleitpunktzahl 12.5. Die binäre IEEE-Gleitpunktdarstellung dieser Zahl ist:

01000001 01001000 00000000 00000000

oder, im Hexadezimalformat, 41480000H. Das höchstwertige Bit (das Vorzeichenbit) ist Null (s=0). Die nächsten acht höchstwertigen Bits sind 10000010, oder dezimal 130 (e=130).

Die Mantisse wird als binäre Dezimalzahl gespeichert, bei der der Dezimalpunkt vor dem höchstwertigen der 23 Bits steht. Das höchstwertige Bit in der Mantisse ist daher ein Vielfaches von  $2^{-1}$ , das nächste Bit in der Wertigkeit ein Vielfaches von  $2^{-2}$  usw. bis zum niedrigstwertigen Bit, das ein Vielfaches von  $2^{-23}$  ist. Die letzten 23 Bits (die Mantisse) sind:

1001000 00000000 00000000

Der Wert der Mantisse ist dann .5625 (d.h.  $2^{-1} + 2^{-4}$ ).

Da  $e > 0$  und  $e < 255$ , benutzen wir die dritte Formel in der obenstehenden Tabelle:

$$\begin{aligned}
 \text{Zahl} &= -1^S * 2^{e-127} * 1.f \\
 &= -1^0 * 2^{e130-127} * 1.5625 \\
 &= 1 * 2^3 * 1.5625 \\
 &= 8 * 1.5625 \\
 &= 12.5
 \end{aligned}$$

Wie Sie sehen können, ist die obenstehende binäre Darstellung richtig.

In diesem Format können die Zahl Null sowie der Zahlenbereich von  $\pm 1.401298E-45$  bis  $\pm 3,402823E+38$  gespeichert werden.

## Eingabe und Anzeige von Gleitpunktzahlen

Obwohl in der Mantisse bis zu sechs oder sieben für die Genauigkeit signifikante Stellen gespeichert werden können, zeigt die Logicmaster 90 Software nur die ersten sechs dieser Stellen an. Vor der Mantisse kann ein positives oder ein negatives Vorzeichen stehen. Wird kein Vorzeichen eingegeben, wird die Mantisse als positiv angenommen.

Wird ein Exponent eingegeben, muß er mit dem Buchstaben **E** oder **e** beginnen und die Mantisse muß einen Dezimalpunkt enthalten, damit sie nicht mit einer Hexadezimalzahl verwechselt wird. Vor dem Exponenten kann ein Vorzeichen stehen. Wird kein Vorzeichen eingegeben, wird der Exponent als positiv angenommen. In einer Gleitpunktzahl sind keine Leerzeichen zulässig.

Um die Anwendung einfach zu machen sind bei Einträgen in Befehlszeile und Prozeßdaten mehrere Formate möglich. Hierzu gehören ganze Zahlen, Dezimalzahlen, oder Dezimalzahlen mit Exponent. Nachdem der Anwender diese Zahlen eingegeben und die Taste **Enter** gedrückt hat, werden diese Zahlen in eine Standard-Anzeigeform umgewandelt.

In der nachstehenden Tabelle finden Sie Beispiele, wie Gleitpunktzahlen eingegeben werden und wie ihre genormte Anzeige aussieht.

Eingabe	Anzeige
250	+250.0000
+4	+4.000000
-2383019	-2383019.
34	+34.00000
-.0036209	-.003620900
12.E+9	+1.20000E+10
-.0004E-11	-4.00000E-15
731.0388	+731.0388
99.20003e-29	+9.92000E-28

Die nachstehende Tabelle zeigt Beispiele unzulässiger Eingaben von Gleitpunktzahlen.

Unzulässige Eingabe	Erläuterung
-433E23	Dezimalpunkt fehlt.
10e-19	Dezimalpunkt fehlt.
10.e19	In der Mantisse sind zwischen den Ziffern oder Zeichen keine Leerstellen zulässig. Diese Eingabe wird als 10.e0 angenommen und es wird eine Fehlermeldung angezeigt.
4.1e19	Im Exponenten sind zwischen den Ziffern oder Zeichen keine Leerstellen zulässig. Diese Eingabe wird als 4.1e0 angenommen und es wird eine Fehlermeldung angezeigt.



## Fehler bei Gleitpunktzahlen und Gleitpunkt-Operationen

Ein Überlauf entsteht, wenn eine REAL-Funktion eine Zahl erzeugt, die größer als  $3.402823E+38$  oder kleiner als  $-3.402823E+38$  ist. In diesem Fall wird der OK-Ausgang der Funktion abgeschaltet und das Ergebnis auf plus unendlich (bei einer Zahl  $>3.402823E+38$ ) bzw. auf minus unendlich (bei einer Zahl  $<-3.402823E+38$ ) eingestellt. Sie können dieses Ereignis feststellen, indem Sie den Zustand des OK-Ausgangs testen.

POS\_INF = 7F800000h – positiv unendliche IEEE-Darstellung in Hexadezimalformat  
 NEG\_INF = FF800000h – negativ unendliche IEEE-Darstellung in Hexadezimalformat

Werden die von einem Überlauf erzeugten Unendlichwerte als Operanden für andere REAL-Funktionen benutzt, können Sie ein undefiniertes Ergebnis liefern. Wir nennen solch ein undefiniertes Ergebnis "keine Zahl" (NaN). Die Addition von minus unendlich und plus unendlich liefert zum Beispiel ein undefiniertes Ergebnis. Wird die Funktion ADD\_REAL mit den Operanden minus unendlich und plus unendlich aufgerufen, ist das Ergebnis ein NaN.

Jede REAL-Funktion, die ein NaN hervorbringen kann, erzeugt ein spezielles NaN, das die Funktion kennzeichnet.

NaN_ADD	= 7F81FFFFh	– Hexadezimaler Fehlerwert bei REAL-Addition
NaN_SUB	= 7F81FFFFh	– Hexadezimaler Fehlerwert bei REAL-Subtraktion
NaN_MUL	= 7F82FFFFh	– Hexadezimaler Fehlerwert bei REAL-Multiplikation
NaN_DIV	= 7F83FFFFh	– Hexadezimaler Fehlerwert bei REAL-Division
NaN_SQRT	= 7F84FFFFh	– Hexadezimaler Fehlerwert bei REAL-Wurzelziehen
NaN_LOG	= 7F85FFFFh	– Hexadezimaler Fehlerwert bei REAL-Logarithmus
NaN_POW0	= 7F86FFFFh	– Hexadezimaler Fehlerwert bei REAL-Exponent
NaN_SIN	= 7F87FFFFh	– Hexadezimaler Fehlerwert bei REAL-Sinus
NaN_COS	= 7F88FFFFh	– Hexadezimaler Fehlerwert bei REAL-Kosinus
NaN_TAN	= 7F89FFFFh	– Hexadezimaler Fehlerwert bei REAL-Tangens
NaN_ASIN	= 7F8AFFFFh	– Hexadezimaler Fehlerwert bei REAL-Arkussinus
NaN_ACOS	= 7F8BFFFFh	– Hexadezimaler Fehlerwert bei REAL-Arkuskosinus
NaN_BCD	= 7F8CFFFFh	– Hexadezimaler Fehlerwert bei Umwandlung BCD in REAL
REAL_INDEF	= FFC00000h	– REAL-unendlich; Fehler aus Division 0 durch 0

Wird ein NaN in eine andere Funktion eingespeist, gibt es das Ergebnis weiter. Ist, zum Beispiel, ein NaN\_ADD der erste Operand für die Funktion SUB\_REAL, ist das Ergebnis von SUB\_REAL ein NaN. Sind beide Operationen einer Funktion NaNs, gibt der erste Operand weiter. Durch diese Eigenschaft, NaNs durch Funktionen weiterzugeben, können Sie die Funktion identifizieren, aus der das NaN herrührt.

### Hinweis

Bei einem NaN ist der OK-Ausgang AUS (nicht durchgeschaltet).

**A**

Abschalten, 2-31  
 Abwärtszähler, 4-22  
 ACOS, 4-35  
 ADD, 4-27  
 ADD\_IOM, 2-24  
 ADD\_SIO, 2-24  
 Additionsfunktion, 4-27  
 Adreßfeld (RefArray), interne Parameter, 4-149  
 Alarm-Fehlercodes, B-4  
 Alarmprozessor, 3-2  
 ALT-Tasten, D-1  
 AND, 4-48  
 Anwenderreferenzen, diskrete interne Referenzen, 2-19  
 Anweisungen, relationale Funktionen, 4-41  
 Anwenderprogramm fehlt, 3-12  
 Anwenderprogramm-Bearbeitungszyklus, 2-8  
 Anwenderreferenzen, 2-19  
   Analogausgänge, 2-19  
   Analogeingänge, 2-19  
   diskrete Ausgänge, 2-19  
   diskrete Eingänge, 2-19  
   diskrete Referenzen, 2-19  
   diskrete temporäre Referenzen, 2-20  
   Globaldaten, 2-20  
   Registerreferenzen, 2-19  
   Systemregister, 2-19  
   Systemstatus, 2-23  
 Anwendungsfehler, 3-11  
 ANY\_FLT, 2-25  
 APL\_FLT, 2-24  
 Arithmetische Funktionen, 4-26  
   SQRT, 4-33  
 ARRAY\_MOVE, 4-86  
 ASIN, 4-35  
 ATAN, 4-35  
 Aufwärtszähler, 4-20  
 Ausgabezyklus, 2-8

Ausschaltverzögerungs-Timer, 4-17

**B**

BAD\_PWD, 2-24  
 BAD\_RAM, 2-24  
 Batterieausfallsignal, 3-11  
 BCD-4, 2-22, 4-94  
 BCLR, 4-61  
 Bedienanforderung, 4-122  
   Anzahl Worte in Prüfsumme ändern/lesen, 4-124  
   aufgelaufene Netzausfallzeit lesen, 4-140  
   Betriebszeituhr lesen, 4-136  
   E/AOverride-Zustand lesen, 4-137  
   Echtzeituhr lesen/stellen, 4-126  
   Fehlertabellen löschen, 4-131  
   letzten Fehlertabelleneintrag lesen, 4-132  
   Master-Prüfsumme lesen, 4-138  
 Befehle, Timer und Zähler, 4-9  
 Befehlsausführungszeiten, A-1  
 Befehlssatz, relationale Funktionen, 4-41  
 Befehlsvorrat  
   arithmetische Funktionen, 4-26  
   Bitoperationsfunktionen, 4-46  
   Datenverschiebefunktionen, 4-68  
   Konvertierungsfunktionen, 4-93  
   Relaisfunktionen, 4-2  
   Steuerfunktionen, 4-106  
   Tabellenfunktionen, 4-85  
   Timer und Zähler, 4-9  
 Beitrag zur Zykluszeit bei CPU 351, 2-6  
 Beitrag zur Zykluszeit durch 90-30-Module, 2-5  
 Berechnung der Zykluszeit, 2-7  
 Bereichsfunktion, 4-43  
 Betriebsart KONSTANTEZYKLUSZEIT, 2-12  
 Betriebsfehler, 3-2  
 Betriebszeituhr, 2-32  
 BIT, 2-22  
 Bit auf 0 setzen, 4-61  
 Bit auf 1 setzen, 4-61  
 Bit-Testfunktion, 4-59

Bitfolgesteuerung, 4-79  
Bitoperationsfunktionen, 4-46  
  AND, 4-48  
  BCLR, 4-61  
  BPOS, 4-63  
  BSET, 4-61  
  BTST, 4-59  
  MCMP, 4-65  
  NOT, 4-52  
  OR, 4-48  
  ROL, 4-57  
  ROR, 4-57  
  SHL, 4-54  
  XOR, 4-50  
Bitpositionsfunktion, 4-63  
BITSEQ, 4-79  
  Speicherbedarf, 4-79  
BLKCLR, 4-74  
BLKMOV, 4-72  
Block kopieren, 4-72  
Block löschen, 4-74  
Block-Verriegelungsfunktion  
  EDITLOCK, 2-35  
  permanentes Sperren eines Unterprogramms, 2-35  
  VIEWLOCK, 2-35  
Blockverriegelungsfunktion, 2-35  
Bogenmaß-Umrechnung  
  DEG, 4-39  
  RAD, 4-39  
BPOS, 4-63  
BSET, 4-61  
BTST, 4-59  
BYTE, 2-22

## C

CALL, 4-107  
CFG\_MM, 2-24  
COMMENT, 4-121  
COMMREQ, 4-82  
  Fehlercode, Beschreibung, Beseitigung, 3-9  
COS, 4-35  
CPU-Systemsoftwarefehler, 3-13

CPU-Steuerfunktionen, Befehlsausführungszeiten, A-1  
CPU-Zyklus, 2-2  
CTRL-Tasten, D-1

## D

Datenremanenz, 2-21  
Datentypen, 2-22  
  BCD-4, 2-22  
  BIT, 2-22  
  BYTE, 2-22  
  DINT, 2-22  
  INT, 2-22  
  REAL, 2-22  
  WORD, 2-22  
Datenübertragungsfehler beim Speichern, 3-15  
Datenverschiebefunktionen  
  BITSEQ, 4-79  
  BLKCLR, 4-74  
  BLKMOV, 4-72  
  COMMREQ, 4-82  
  MOVE, 4-69  
  SHFR, 4-76  
Datenverschiebefunktionen, 4-68  
DEG, 4-39  
Diagnosedaten, 2-39  
Diagnosefehler, 3-4  
  Anwendungsfehler, 3-11  
  hinzugefügtes E/A-Modul, 3-17  
  konstante Zykluszeit überschritten, 3-11  
  rückgesetztes, hinzugefügtes oder überzähliges Zusatzmodul, 3-8  
  Signal für niedrige Batteriespannung, 3-11  
  verlorenes E/A-Modul, 3-17  
  verlorenes oder fehlendes Zusatzmodul, 3-8  
DINT, 2-22, 4-98  
Diskrepanz bei Systemkonfiguration, 3-9  
Diskrete Referenzen, 2-19  
  diskrete Ausgänge, 2-19  
  diskrete Eingänge, 2-19  
  diskrete interne Referenzen, 2-19  
  diskrete temporäre Referenzen, 2-20  
  Globaldaten, 2-20

Systemstatus, 2-23  
 Systemzustand, 2-20  
 DIV, 4-27  
 Divisionsfunktion, 4-27  
 DNCTR, 4-22  
 DOIO, 4-108  
 DOIO-Funktion, erweiterte DOIO-Funktion für CPUs ab Modell 331 aufwärts, 4-112  
 Doppeltgenaue ganze Zahl mit Vorzeichen, 2-22

## **E**

E/AAktualisierung, 4-108  
 E/A-Datenformate, 2-39  
 E/A-Ehlerabelle, 3-5, B-7  
   Anzeigedang/kurz, B-8  
   Chassis, B-9  
   Erläuterungen, 3-16  
   Fehleradresse, B-8  
   Fehlergruppe, B-9  
   fehlerspezifische Daten, B-10  
   Fehlerwirkung, B-10  
   Fehlerwirkungen für bestimmte Fehler, B-10  
   Punkt, B-9  
   Referenzadresse, B-8  
   Steckplatz, B-9  
   symbolische fehlerspezifische Daten, B-10  
   Zeitstempel, B-10  
 E/A-Struktur, SPS Serie 90-30, 2-36  
 E/A-System, Serie 90-30, Globaldaten, 2-39  
 E/A-System, SPS Serie 90-Micro, Micro-E/A, 2-40  
 E/A-System, SPS Serie 90-20, 2-36  
   Modell 20 E/A-Module, 2-40  
 E/A-System, SPS Serie 90-30, 2-36  
   Diagnosedaten, 2-39  
   E/A-Datenformate, 2-39  
   Modell 30 E/A-Module, 2-37  
   Standardbedingungen für die Modell 30 Ausgangsmodule, 2-39  
 EDITLOCK, 2-35  
 Ein- und Ausschaltsequenzen, 2-29  
   Ausschalten, 2-31  
   Einschalten, 2-29  
 Eingabezyklus, 2-8  
 Einschalten, 2-29  
 Einschaltverzögerungs-Timer, 4-11, 4-14  
 END, 4-113  
 Ende Hauptsteuerrelaisfunktion, 4-117  
 Endefunktion, 4-113  
 ENDMCR, 4-117  
 EQ, 4-41  
 Erläuterung und BEhebung von Fehlern, verstümmeltes Anwenderprogramm beim Einschalten, 3-12  
 Erläuterung und Behebung von Fehlern  
   Anwenderprogramm fehlt, 3-12  
   CPU-Systemsoftwarefehler, 3-13  
   Datenübertragungsfehler beim Speichern, 3-15  
   Diskrepanz bei Systemkonfiguration, 3-9  
   E/A-Ehlerabelle – Erläuterungen, 3-16  
   E/A-Ehlerabelle, 3-5  
   hinzugefügtes E/A-Modul, 3-17  
   konstante Zykluszeit überschritten, 3-11  
   nicht-konfigurierbare Fehler, 3-8  
   Paßwortfehler, 3-12  
   Programmblock-Prüfsummenfehler, 3-10  
   rückgesetztes, hinzugefügtes oder überzähliges Zusatzmodul, 3-8  
   Signal für niedrige Batteriespannung, 3-11  
   SPS-Fehlergruppe, B-3  
   SPS-Fehlerabelle, 3-5  
   verlorenes E/A-Modul, 3-17  
   verlorenes oder fehlendes Zusatzmodul, 3-8  
 Erläuterung und Beschreibung von Fehlern  
   Fehlerbeschreibung, 3-16  
   Fehlerkategorie, 3-16  
   Fehlertyp, 3-16  
 Erläuterung und Beseitigung von Fehlern, Softwarefehler bei Zusatzmodul, 3-9  
 Erläuterung und Behebung von Fehlern  
   Anwendungsfehler, 3-11

E/A-Fehlergruppe, B-9  
Erweiterte DOIO-Funktion für CPUs ab  
Modell 331 aufwärts, 4-112  
Erweiterter Speicherschutz, bei CPUs ab  
Ausgabestand 8, 2-14  
EXP, 4-37  
EXPT, 4-37  
Externe E/A-Fehler, 3-2

## F

Fatale Fehler, 3-4  
CPU-Systemsoftwarefehler, 3-13  
Datenübertragungsfehler beim Spei-  
cher, 3-15  
Diskrepanz bei Systemkonfiguration,  
3-9  
Programmblock-Prüfsummenfehler,  
3-10  
Softwarefehler bei Zusatzmodul, 3-9  
verstümmeltes Anwenderprogramm  
beim Einschalten, 3-12  
Fehler  
Anwenderprogramm fehlt, 3-12  
Anwendungsfehler, 3-11  
Beschreibung und Behebung, 3-1  
Betriebsfehler, 3-2  
CPU-Systemsoftwarefehler, 3-13  
Datenübertragungsfehler beim Spei-  
chern, 3-15  
Diskrepanz bei Systemkonfiguration,  
3-9  
E/A-Fehlergruppe, B-9  
E/A-Fehlertabelle – Erläuterungen,  
3-16  
E/A-Fehlerwirkung, B-10  
E/A-Fehlertabelle, 3-5  
externe E/A-Fehler, 3-2  
Fehlercodes, B-4  
Fehlerklassen, 3-2  
Fehlerwirkung, 3-4  
hinzugefügtes E/A-Modul, 3-17  
interne Fehler, 3-2  
konstante Zykluszeit überschritten,  
3-11  
Paßwortfehler, 3-12  
Programmblock-Prüfsummenfehler,  
3-10  
Referenzen, 3-4

rückgesetztes, hinzugefügtes oder  
überzähliges Zusatzmodul, 3-8  
Signal für niedrige Batteriespannung,  
3-11  
Softwarefehler bei Zusatzmodul, 3-9  
SPS-Fehlergruppe, B-3  
SPS-Fehlertabelle, 3-5  
SPS-Fehlerwirkung, B-4  
Systemreaktion auf Fehler, 3-3  
verlorenes E/A-Modul, 3-17  
verlorenes oder fehlendes Zusatzmo-  
dul, 3-8  
verstümmeltes Anwenderprogramm  
beim Einschalten, 3-12  
Wirkungen, 3-8  
Fehlerbearbeitung, 3-2  
Alamprozessor, 3-2  
Fehlerwirkung, 3-4  
Fehlerbeschreibung, 3-16  
Fehlerbeschreibungen und Fehlerbehe-  
bung, 3-1  
Fehlercodes, B-4  
Fehlererläuterung und Fehlerbehebung,  
Fehlerbearbeitung, 3-2  
Fehlergruppe, B-3, B-9  
Fehlerkategorie, 3-16  
Fehlerreferenzen, 3-4  
Fehlersuche, 3-1  
E/A-Fehlertabelle – Erläuterungen,  
3-16  
E/A-Fehlertabelle, 3-5  
nicht-konfigurierbare Fehler, 3-8  
SPS-Fehlertabelle, 3-5  
Fehlertyp, 3-16  
Fehlerwirkung, 3-4  
Diagnosefehler, 3-4  
E/A-Fehlerwirkung, B-10  
fatale Fehler, 3-4  
informativische Fehler, 3-4  
SPS-Fehlerwirkung, B-4  
Fehlerwirkungen, 3-8  
Fenster, 2-9  
Programmiergeräte-Kommunikations-  
fenster, 2-9  
System-Kommunikationsfenster, 2-10  
Fortsetzkontakt, 4-8  
Fortsetzspule, 4-8  
Funktionsblockparameter, 2-27

Funktionsblockstruktur, 2-25  
Format der Programm-Funktions-  
blöcke, 2-26  
Funktionsblockparameter, 2-27  
Relaisformat, 2-25  
Spulenverwendungs-Überprüfungs-  
funktion, 2-26  
Stromfluß, 2-28

## G

Ganze Zahl mit Vorzeichen, 2-22  
GE, 4-41  
Gleich, 4-41  
Gleitpunktzahl, 2-22  
Gleitpunktzahlen, E-1  
Eingabe und Anzeige, E-4  
Fehler, E-5  
internes Format, E-2  
Werte, E-3  
Globaldaten, 2-39  
Globaldatenreferenzen, 2-20  
Größer als, 4-41  
Größer als oder gleich, 4-41  
Grundrechenarten  
ADD, 4-27  
MOD, 4-31  
MUL, 4-27  
SUB, 4-27  
GT, 4-41

## H

Hauptsteuerrelaisfunktion, 4-114  
Hexadezimalanzeige der Fehlerdaten,  
3-6  
HinzugefügtesE/A-Modul, 3-17  
Horizontale Verbindung, 4-7  
HRD\_CPU, 2-24  
HRD\_FLT, 2-25  
HRD\_SIO, 2-24

## I

Informatorische Fehler, 3-4  
Anwenderprogramm fehlt, 3-12  
Paßwortfehler, 3-12  
INT, 2-22, 4-96  
Interne Fehler, 3-2  
Interne Parameter im Adreßfeld, 4-149  
IO\_FLT, 2-25  
IO\_PRES, 2-25

## J

JUMP, 4-118

## K

Kleiner als, 4-41  
Kleiner als oder gleich, 4-41  
Kommentarfunktion, 4-121  
Kommunikation mit der SPS, 2-11  
Kommunikationsanforderung, 4-82  
Fehlercode, Beschreibung und Beseiti-  
gung, 3-9  
Kommunikationsfenster-Modus, 2-12  
Konfiguration, 4-1  
Konstante Zykluszeit überschritten, 3-11  
Kontakte, 4-2  
Fortsetzkontakt, 4-8  
Öffnerkontakt, 4-4  
Schließerkontakt, 4-4  
Konvertierungsfunktionen, 4-93  
BCD-4, 4-94  
DINT, 4-98  
INT, 4-96  
REAL, 4-100  
TRUN, 4-104  
WORD, 4-102

Kopieren, 4-69

## L

LABEL, 4-120  
LE, 4-41

LN, 4-37  
LOG, 4-37  
Logarithmus-/Exponentialfunktionen  
  EXP, 4-37  
  EXPT, 4-37  
  LN, 4-37  
  LOG, 4-37  
Logische Invertierung, 4-52  
Logische ODER-Funktion, 4-48  
Logische UND-Funktion, 4-48  
Logische XOR-Funktion, 4-50  
LOS\_IOM, 2-24  
LOS\_SIO, 2-24  
LOW\_BAT, 2-24  
LT, 4-41

## M

MCR, 4-114  
Micro-Modelle, 2-40  
Mnemonicische Programmieranweisungen, C-1  
MOD, 4-31  
Modell 20E/A-Module, 2-40  
Modell 30E/A-Module, 2-37  
Modulo-Funktion, 4-31  
MOVE, 4-69  
MSKCMP, 4-65  
MUL, 4-27  
Multiplikationsfunktion, 4-27

## N

NE, 4-41  
Negierte remanente Spule, 4-5  
Negierte Spule, 4-4  
NOT, 4-52  
NUL\_CFG, 2-24

## O

OFDT, 4-17  
Öffnerkontakt, 4-4  
ONDTR, 4-11  
OR, 4-48  
Organisation, 2-7  
OV\_SWP, 2-24  
Overrides, 2-20

## P

Paßworte, 2-34  
Paßwortfehler, 3-12  
PB\_SUM, 2-24  
PCM-Kommunikation mit der SPS, 2-11  
Periodische Unterprogramme, 2-17  
PID, 4-141  
PID-Algorithmus  
  Einstellung, 4-149  
  PIDIND, 4-149  
  PIDISA, 4-149  
PID-Anweisung, Arbeitsweise, 4-144  
Privilegebene, Anforderung zum Wechsel, 2-35  
Privilegebenen, 2-34  
Programm-Prüfsummenberechnung, 2-8  
Programmbearbeitung, 2-8  
Programmblock  
  Unterprogrammblock, 2-15  
  wie Unterprogrammblöcke aufgerufen werden, 2-17  
Programmblock-Prüfsummenfehler, 3-10  
Programmieranweisungen  
  arithmetische Funktionen, 4-26  
  Bitoperationsfunktionen, 4-46  
  Datenverschiebefunktionen, 4-68  
  Konvertierungsfunktionen, 4-93  
  mnemonicische Programmieranweisungen, C-1  
  Relaisfunktionen, 4-2  
  relationale Funktionen, 4-41  
  Steuerfunktionen, 4-106

Tabellenfunktionen, 4-85  
 Timer und Zähler, 4-9  
 Programmiergeräte-Kommunikations-  
 fenster, 2-9  
 Programmorganisation und Anwender-  
 referenzen/-daten, 2-15  
 Anwenderreferenzen, 2-19  
 Datenremanenz, 2-21  
 Datentypen, 2-22  
 Funktionsblockstruktur, 2-25  
 Systemstatus, 2-23  
 Transitionen und Overrides, 2-20  
 Programmstruktur  
 Unterprogrammblock, 2-15  
 wie Unterprogrammblöcke aufgeru-  
 fen werden, 2-17  
 Prüfsummenberechnung, 2-8  
 Prüfsummenfehler bei Programmblock,  
 3-10

## Q

Quadratwurzelfunktion, 4-33

## R

RAD, 4-39  
 RANGE, 4-43  
 REAL, 2-22, 4-100  
 Referenzen, 2-19  
 Registerreferenzen, 2-19  
 Analogausgänge, 2-19  
 Analogeingänge, 2-19  
 Systemregister, 2-19  
 Relaisfunktionen, 4-2  
 Fortsetzkontakt, 4-8  
 Fortsetzspule, 4-8  
 horizontale und vertikale Verbindun-  
 gen, 4-7  
 Kontakte, 4-2  
 negierte remanente Spule, 4-5  
 negierte Spule, 4-4  
 Öffnerkontakt, 4-4  
 remanente RESET-Spule, 4-7  
 remanente SET-Spule, 4-7  
 remanente Spule, 4-5  
 RESET-Spule, 4-6

Schließerkontakt, 4-4  
 SET-Spule, 4-6  
 Spule für negative Übergänge, 4-5  
 Spule für positive Übergänge, 4-5  
 Spulen, 4-3, 4-4

Relationale Funktionen, 4-41

EQ, 4-41  
 GE, 4-41  
 GT, 4-41  
 LE, 4-41  
 LT, 4-41  
 NE, 4-41

Remanente RESET-Spule, 4-7

Remanente SET-Spule, 4-7

Remanente Spule, 4-5

RESET-Spule, 4-6

ROL, 4-57

ROR, 4-57

Rotiere links, 4-57

Rotiere rechts, 4-57

Rückgesetztes, hinzugefügtes oder  
 überzähliges Zusatzmodul, 3-8

## S

Schieberegisterfunktion, 4-76

Schließerkontakt, 4-4

Schlüsselschalter, CPU 351/352, 2-13

Serviceanforderung, SPS abschalten  
 (anhalten), 4-130

SET-Spule, 4-6

SFT\_CPU, 2-24

SFT\_FLT, 2-25

SFT\_SIO, 2-24

SHFR, 4-76

SHL, 4-54

SHR, 4-54

Sicherheitssystem

Privilegebenen, 2-34

Sperrern/freigeben von Unterprogram-  
 men, 2-35

Wechsel der Privilegeebene, 2-35

Signal für niedrige Batteriespannung,  
 3-11



- SIN, 4-35
- Softwarefehler bei Zusatzmodul, 3-9
- Sperren/freigeben von Unterprogrammen, 2-35
- Sprunganweisung, 4-118
- SPS Serie 90- Micro E/A-System, Micro-CPU und -E/A, 2-40
- SPS Serie 90-20 E/A-System, 2-36
  - Modell 20 E/A-Module, 2-40
- SPS Serie 90-30 E/A-System, 2-36
  - Diagnosedaten, 2-39
  - E/A-Datenformate, 2-39
  - E/A-Struktur, 2-36
  - Globaldaten, 2-39
  - Modell 30 E/A-Module, 2-37
  - Standardbedingungen für die Modell 30 Ausgangsmodule, 2-39
- SPS-Fehlertabelle, 3-5, B-1
  - Anzeigelang/kurz, B-2
  - Chassis, B-2
  - Fehlercodes, B-4
  - Fehlergruppe, B-3
  - Fehlerwirkung, B-4
  - Reserve, B-2
  - Zeitstempel, B-6, B-10
  - zusätzliche Fehlerdaten, B-6
- SPS-Systembetrieb, 2-1
- SPS-Zyklus, 2-2
  - Anwenderprogramm-Bearbeitungszyklus, 2-8
  - Ausgabezyklus, 2-8
  - Beitrag zur Zykluszeit, 2-5
  - Beitrag zur Zykluszeit bei CPU 351, 2-6
  - Beiträge zur Zyklusdauer, 2-4
  - Berechnung der Zykluszeit, 2-7
  - Betriebsart KONSTANTEZYKLUSZEIT, 2-12
  - Eingabezyklus, 2-8
  - Organisation, 2-7
  - PCM-Kommunikation mit der SPS, 2-11
  - Programm-Prüfsummenberechnung, 2-8
  - Programmbearbeitung, 2-8
  - Programmiergeräte-Kommunikationsfenster, 2-9
  - Standard-Programmzyklus, 2-2
  - STOP-Modus, 2-12
  - System-Kommunikationsfenster, 2-10
  - Variationen des Standard-Programmzyklus, 2-12
- Spule, mit Einfach- oder Mehrfachprüfung, 4-6
- Spule für negative Übergänge, 4-5
- Spule für positive Übergänge, 4-5
- Spulen, 4-3, 4-4
  - Fortsetzspule, 4-8
  - negierte remanente Spule, 4-5
  - negierte Spule, 4-4
  - remanente RESET-Spule, 4-7
  - remanente SET-Spule, 4-7
  - remanente Spule, 4-5
  - RESET-Spule, 4-6
  - SET-Spule, 4-6
  - Spule für negative Übergänge, 4-5
  - Spule für positive Übergänge, 4-5
- Spulenverwendungs-Überprüfungsfunktion, 2-26
- SQRT, 4-33
- SRCH\_EQ, 4-90
- SRCH\_GE, 4-90
- SRCH\_GT, 4-90
- SRCH\_LE, 4-90
- SRCH\_LT, 4-90
- SRCH\_NE, 4-90
- Standard-Programmzyklus, 2-2
- Standardbedingungen für die Modell 30 Ausgangsmodule, 2-39
- Steuerfunktionen, 4-106
  - CALL, 4-107
  - COMMENT, 4-121
  - DOIO, 4-108
    - erweiterte DOIO für CPUs ab Modell 331 aufwärts, 4-112
  - END, 4-113
  - ENDMCR, 4-117
  - JUMP, 4-118
  - LABEL, 4-120
  - MCR, 4-114
  - PID, 4-141
  - SVCREQ, 4-122
- STOP-Modus, 2-12
- STOR\_ER, 2-24
- Stromfluß, 2-28
- SUB, 4-27
- Subtraktionsfunktion, 4-27

- Suche gleiche, 4-90  
Suche größer als, 4-90  
Suche größer als oder gleich, 4-90  
Suche kleiner als, 4-90  
Suche kleiner als oder gleich, 4-90  
Suche ungleiche, 4-90  
SVCREQ, 4-122  
    Anzahl Worte in Prüfsumme ändern/  
    lesen, 4-124  
    aufgelaufene Netzausfallzeit lesen,  
    4-140  
    Betriebszeituhr lesen, 4-136  
    E/AOverride-Zustand lesen, 4-137  
    Echtzeituhr lesen/stellen, 4-126  
    Fehlertabellen löschen, 4-131  
    letzten Fehlertabelleneintrag lesen,  
    4-132  
    Master-Prüfsumme lesen, 4-138  
    SPS abschalten (anhalten), 4-130  
SY\_FLT, 2-25  
SY\_PREC, 2-25  
System-Kommunikationsfenster, 2-10  
System-Registerreferenzen, 2-19  
Systembetrieb, 2-1  
    Ein- und Ausschaltsequenzen, 2-29  
    Programmorganisation und Anwen-  
    derreferenzen/-daten, 2-15  
    SPS-Zyklus – Zusammenfassung, 2-2  
    Systemsicherheit, 2-34  
    Uhren und Timer, 2-32  
Systemoperation  
    E/A-System der SPS Serie 90-20, 2-36  
    E/A-System der SPS Serie 90-30, 2-36  
Systemreferenzen, 2-23  
    ADD\_IOM, 2-24  
    ADD\_SIO, 2-24  
    ANY\_FLT, 2-25  
    APL\_FLT, 2-24  
    BAD\_PWD, 2-24  
    BAD\_RAM, 2-24  
    CFG\_MM, 2-24  
    HRD\_CPU, 2-24  
    HRD\_FLT, 2-25  
    HRD\_SIO, 2-24  
    IO\_FLT, 2-25  
    IO\_PREC, 2-25  
    LOS\_IOM, 2-24  
    LOS\_SIO, 2-24  
    LOW\_BAT, 2-24  
    NUL\_CFG, 2-24  
    OV\_SWP, 2-24  
    PB\_SUM, 2-24  
    SFT\_CPU, 2-24  
    SFT\_FLT, 2-25  
    SFT\_SIO, 2-24  
    STOR\_ER, 2-24  
    SY\_FLT, 2-25  
    SY\_PREC, 2-25  
Systemsicherheit, 2-34  
    Paßworte, 2-34
- ## T
- Tabellenfunktionen, 4-85  
    ARRAY\_MOVE, 4-86  
    SRCH\_GE, 4-90  
    SRCH\_GT, 4-90  
    SRCH\_LT, 4-90  
    SRCH\_NE, 4-90  
    suche kleiner als oder gleich, 4-90  
Tageszeituhr, 2-32  
TAN, 4-35  
Timer, 2-32, 4-9  
    Funktionsblockdaten, 4-9  
    OFDT, 4-17  
    ONDTR, 4-11  
    Timer für konstante Zykluszeit, 2-33  
    TMR, 4-14  
    Zeitkontakte, 2-33  
    Zeitüberwachung (Watchdog), 2-33  
Timer für konstante Zykluszeit, 2-33  
TMR, 4-14  
Transitionen, 2-20  
Trigonometrische Funktionen  
    ACOS, 4-35  
    ASIN, 4-35  
    ATAN, 4-35  
    COS, 4-35  
    SIN, 4-35  
    TAN, 4-35  
TRUN, 4-104
- ## U
- Uhren, 2-32  
    Betriebszeituhr, 2-32  
    Tageszeituhr, 2-32  
Umwandlung in BCD-4, 4-94

Umwandlung in doppeltgenaue ganze Zahl mit Vorzeichen, 4-98

Umwandlung in ganze Zahl mit Vorzeichen, 4-96

Ungleich, 4-41

Unterprogrammaufruf, 4-107

Unterprogrammblöcke, 2-15

Unterprogramme sperren/freigeben, 2-35

UPCTR, 4-20

## V

Variationen des Standard-Programmzyklus, 2-12

Verbindungen, horizontal und vertikal, 4-7

Vergleich mit Maske, 4-65

Verlorenes E/A-Modul, 3-17

Verlorenes oder fehlendes Zusatzmodul, 3-8

Verschiebe nach links, 4-54

Verschiebe nach rechts, 4-54

Verstärkung, Einstellung, 4-149

Verstümmeltes Anwenderprogramm beim Einschalten, 3-12

Vertikale Verbindung, 4-7

VIEWLOCK, 2-35

## W

Wartung, 3-1

Wechsel der Privilegeebene, 2-35

WORD, 2-22, 4-102

## X

XOR, 4-50

## Z

Zähler, 4-9

DNCTR, 4-22

Funktionsblockdaten, 4-9

UPCTR, 4-20

Zeitkontakte, 2-33

Zeitüberwachung (Watchdog), 2-33

Zusatzmodul, Softwarefehler, 3-9

Zyklus, SPS, 2-2