

**Ausbildungsunterlage für die durchgängige
Automatisierungslösung
Totally Integrated Automation (T I A)**

ANHANG III

Grundlegende Programmierbefehle

KOP/FUP/AWL in STEP 7

Deutsche Mnemonik

Diese Unterlage wurde von der Siemens AG, für das Projekt Siemens Automation Cooperates with Education (SCE) zu Ausbildungszwecken erstellt.

Die Siemens AG übernimmt bezüglich des Inhalts keine Gewähr.

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts ist innerhalb öffentlicher Aus- und Weiterbildungsstätten gestattet. Ausnahmen bedürfen der schriftlichen Genehmigung durch die Siemens AG (Herr Michael Knust michael.knust@siemens.com).

Zuwiderhandlungen verpflichten zu Schadensersatz. Alle Rechte auch der Übersetzung sind vorbehalten, insbesondere für den Fall der Patentierung oder GM-Eintragung.

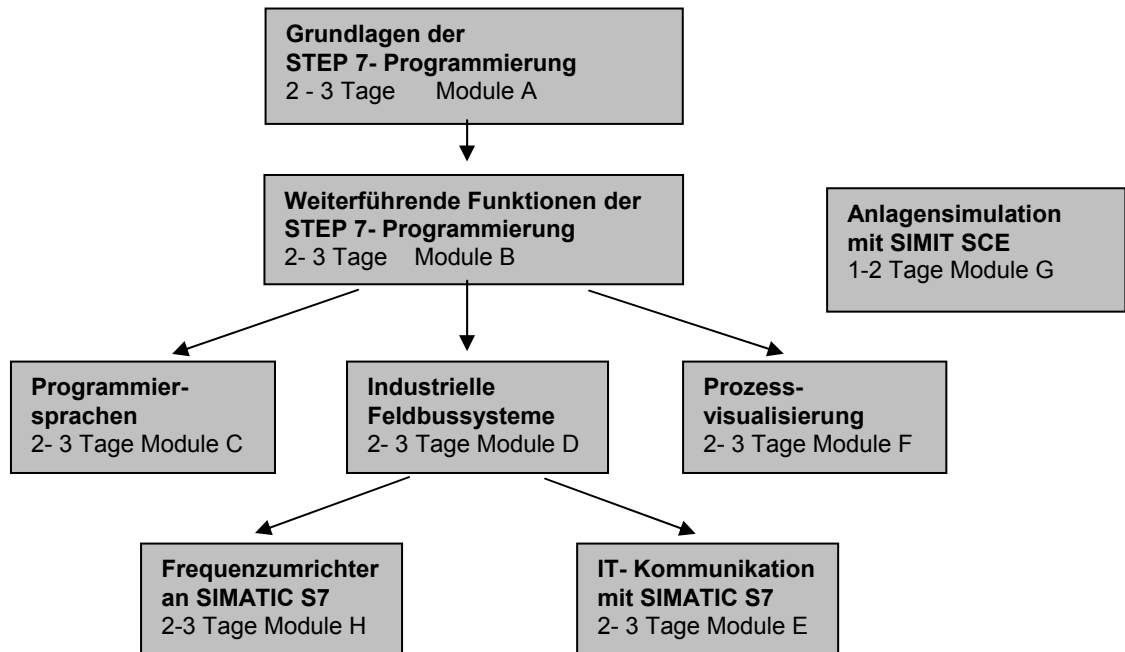
Wir danken der Fa. Michael Dziallas Engineering und den Lehrkräften von beruflichen Schulen sowie weiteren Personen für die Unterstützung bei der Erstellung der Unterlage

		SEITE:
1.	Vorwort	5
2.	Grundlegende Programmierbefehle	6
2.1	Zuweisung.....	7
2.2	UND - Verknüpfung	7
2.3	ODER - Verknüpfung.....	8
2.4	UND - vor ODER - Verknüpfung.....	8
2.5	ODER - vor UND - Verknüpfung.....	9
2.6	Abfrage auf Signalzustand 0.....	10
2.7	Exklusiv - ODER - Verknüpfung	10
2.8	Abfrage von Ausgängen	11
2.9	R- S - Speicherfunktionen.....	11
2.9.1	Vorrangiges Zurücksetzen.....	12
2.9.2	Vorrangiges Setzen	12
2.10	Flankenoperationen	13
2.10.1	Steigende Flanke (FP).....	13
2.10.2	Fallende Flanke (FN).....	14
2.11	Zeitfunktionen	15
2.11.1	Zeit freigeben (FR) nur in AWL.....	15
2.11.2	Zeit starten (SI/SV/SE/SS/SA).....	15
2.11.3	Zeitwert - Vorgabe (TW)	16
2.11.4	Zeit rücksetzen (R)	16
2.11.5	Zeitwert abfragen (L/LC).....	16
2.11.6	Signalzustand der Zeit binär abfragen (Q)	17
2.11.7	Zeit als Impuls (SI).....	17
2.11.8	Verlängerter Impuls (SV)	18
2.11.9	Einschaltverzögerung (SE).....	19
2.11.10	Speichernde Einschaltverzögerung (SS).....	20
2.11.11	Ausschaltverzögerung (SA).....	21
2.12	Taktgeber	22

		SEITE:
2.13	Zähloperationen	23
2.13.1	Zähler freigeben (FR) nur in AWL	23
2.13.2	Vorwärtszählen (ZV)	23
2.13.3	Rückwärtszählen (ZR)	23
2.13.4	Zähler setzen (S)	24
2.13.5	Zählwert - Vorgabe (ZW)	24
2.13.6	Zähler rücksetzen (R)	24
2.13.7	Zählwert abfragen (L/LC).....	24
2.13.8	Signalzustand des Zählers binär abfragen (Q).....	25
2.14	Lade- und Transferoperationen (L/T) nur in AWL	26
2.15	Vergleichsfunktionen	27
2.16	Programmorganisation	28
2.16.1	Bausteinaufruf (CALL)	28
2.16.2	Konditionierter Bausteinaufruf (CC).....	28
2.16.3	Unkonditionierter Bausteinaufruf (UC).....	29
2.16.4	Datenbaustein aufschlagen (AUF)	29
2.16.5	Bausteinende bedingt (BEB) nur in AWL	29
2.16.6	Bausteinende absolut (BEA) nur in AWL.....	30
2.17	Sprungoperationen	31
2.17.1	Springe absolut (SPA)	31
2.17.2	Springe bedingt (SPB/SPBN)	31
2.17.3	Programmschleife (LOOP) nur in AWL	32
2.18	Nulloperationen	32
2.18.1	Nulloperation 0/1 (NOP0/NOP1) nur in AWL.....	32
2.19	Bearbeitung des VKE	33
2.19.1	Negieren des VKE (NOT) nur in AW72	33
2.19.2	Setzen des VKE (SET) nur in A72.....	33
2.19.3	Rücksetzen des VKE (CLR) nur 72 AWL	33
2.19.4	Sichern des VKE (SAVE) nur in 72L	33

1. VORWORT

Anhang III wird bei der Bearbeitung sämtlicher Module benötigt.



Lernziel:

Der Leser erhält mit diesem Anhang eine Sammlung der wichtigsten Programmierbefehle, die zur Lösung der Programmieraufgaben in den Modulen benötigt werden. Dabei wird in diesem Anhang III die deutsche Mnemonik verwendet.

Voraussetzungen:

Damit die Befehle und die Programmierweise verstanden werden kann wird folgendes Wissen vorausgesetzt:

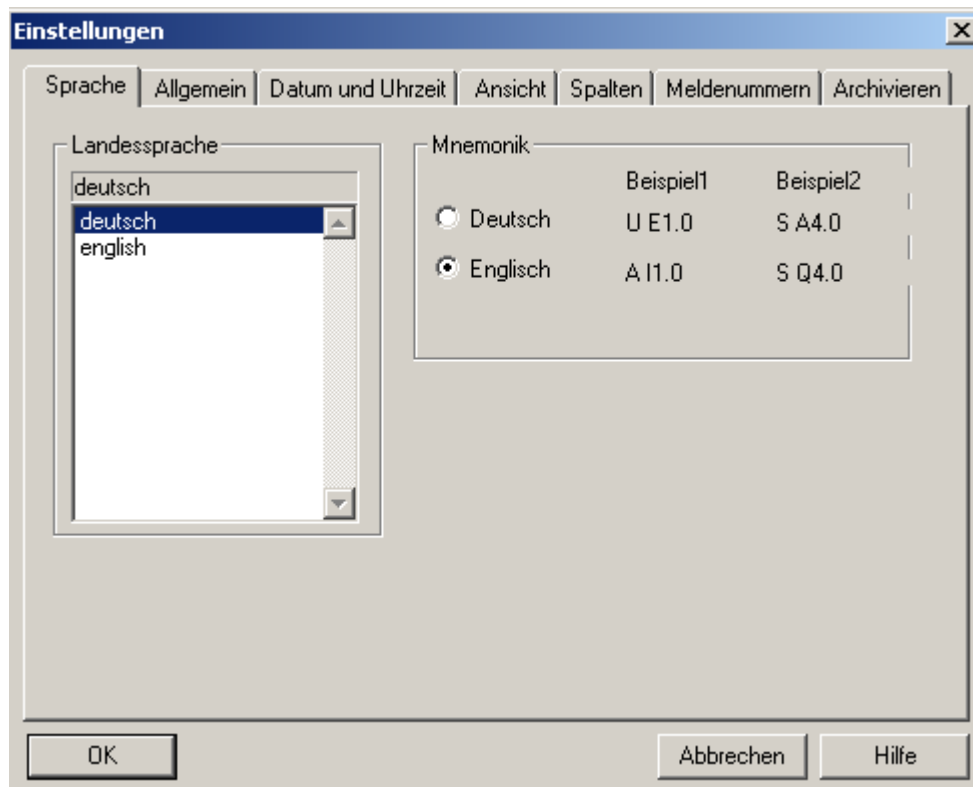
- Grundlagen der SPS- Programmierung (z.B. Anhang I – Grundlagen zur SPS - Programmierung mit SIMATIC S7-300)

2. GRUNDLEGENDE PROGRAMMIERBEFEHLE

Die folgenden Programmierbefehle sind für die Grundlagen der Programmierung ausreichend. Dies ist jedoch keine vollständige Auflistung aller Befehle.

Information zu weiteren Befehlen in KOP/FUP/AWL finden Sie in den Handbüchern oder am besten in der **Online-Hilfe** unter dem Stichpunkt **Sprachbeschreibung KOP, FUP bzw. AWL**.

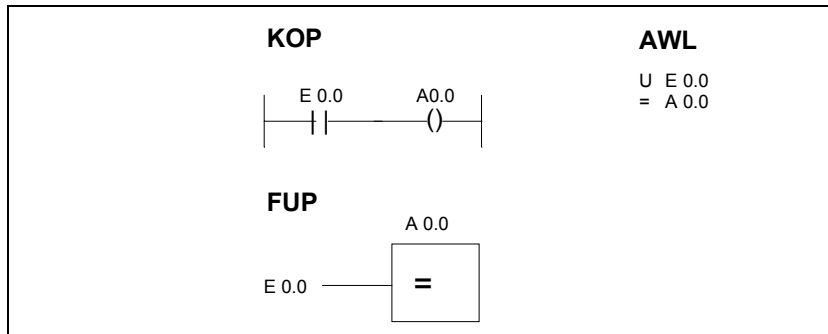
Die Umstellung auf die deutsche Mnemonik (deutsche Programmierbefehle) erfolgt im **SIMATIC-Manager** im Menü unter **Extras**, **Einstellungen**. Dann wählen Sie bei **Sprache** die **Mnemonik** in **Englisch**. (→ SIMATIC- Manager → Extras → Einstellungen → Sprache → Mnemonik → Deutsch → OK).



2.1 ZUWEISUNG

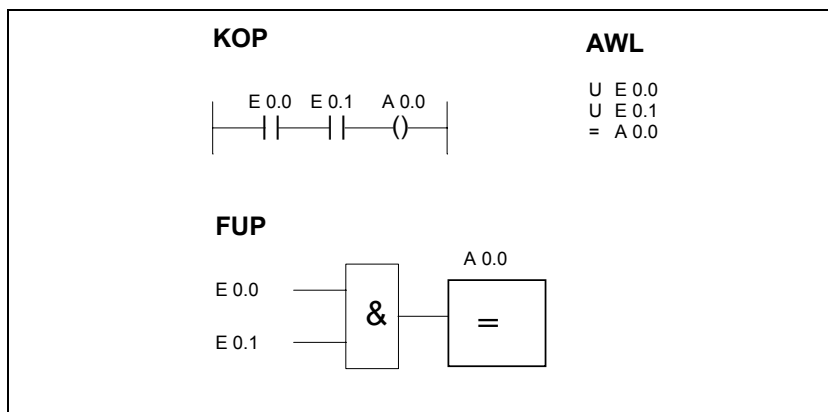
Die Zuweisung (=) kopiert das Verknüpfungsergebnis (VKE) der vorhergehenden Operation und weist es dem nachfolgenden Operanden zu.

Eine Verknüpfungskette kann durch eine Zuweisung abgeschlossen werden.



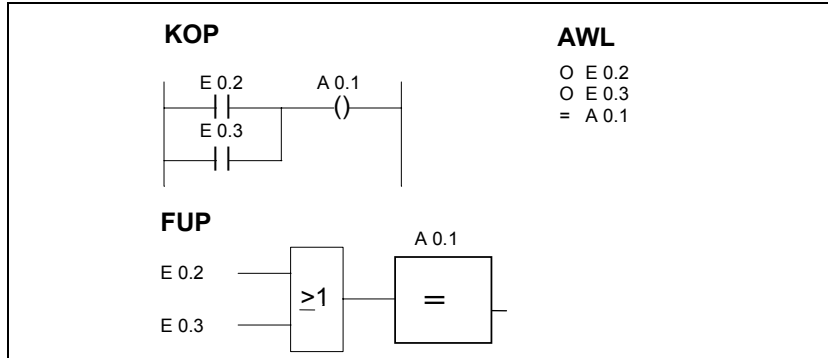
2.2 UND - VERKNÜPFUNG

Die UND - Verknüpfung entspricht einer Reihenschaltung von Kontakten im Stromlaufplan. Am Ausgang A 0.0 erscheint Signalzustand 1, wenn alle Eingänge gleichzeitig den Signalzustand 1 aufweisen. Wenn einer der Eingänge den Signalzustand 0 aufweist, bleibt der Ausgang im Signalzustand 0.



2.3 ODER - VERKNÜPFUNG

Die ODER - Verknüpfung entspricht einer Parallelschaltung von Kontakten im Stromlaufplan. Am Ausgang A 0.1 erscheint der Signalzustand 1, wenn mindestens einer der Eingänge den Signalzustand 1 aufweist. Nur wenn alle Eingänge den Signalzustand 0 aufweisen, bleibt der Signalzustand am Ausgang auf 0.



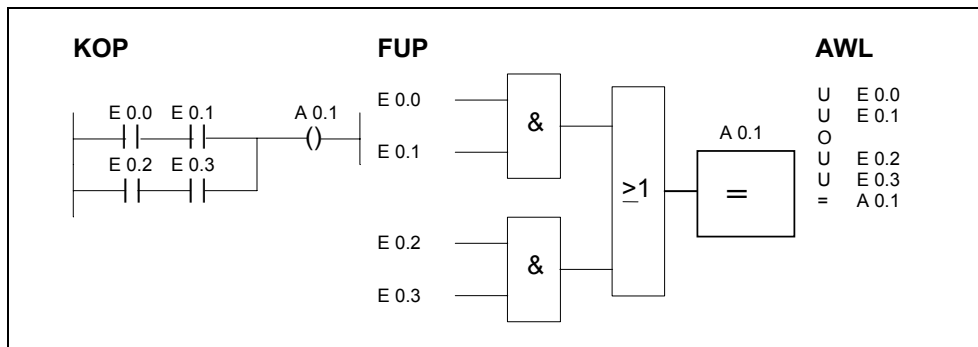
2.4 UND - VOR ODER - VERKNÜPFUNG

Die UND- vor ODER- Verknüpfung entspricht einer Parallelschaltung mehrerer in Reihe geschalteter Kontakte im Stromlaufplan.

Bei diesen aus Reihen- und Parallelschaltung zusammengesetzten Zweigen führt der Ausgang 0.1 den Signalzustand 1, wenn in mindestens einem Zweig alle in Reihe geschalteten Kontakte geschlossen sind (den Signalzustand 1 haben).

UND- vor ODER- Verknüpfungen werden in der Darstellungsart AWL ohne Klammern programmiert, es müssen jedoch die parallelgeschalteten Zweige durch die Eingabe des Zeichens O (ODER-Funktion) voneinander getrennt werden.

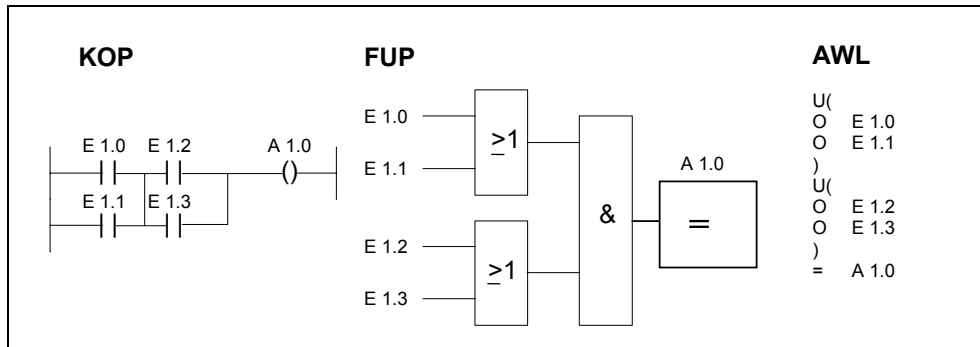
Dabei werden zuerst die UND- Funktionen bearbeitet und aus Ihren Ergebnissen das Ergebnis der ODER- Funktion gebildet. Die erste UND- Funktion (E 0.0, E 0.1) wird von der zweiten UND- Funktion (E 0.2, E 0.3) durch das einzelne O (ODER- Funktion) getrennt.



UND- Verknüpfungen haben Vorrang und werden damit immer vor den ODER- Verknüpfungen bearbeitet.

2.5 ODER - VOR UND - VERKNÜPFUNG

Die ODER- vor UND- Verknüpfung entspricht einer Reihenschaltung mehrerer parallelgeschalteter Kontakte im Stromlaufplan. Bei dieser aus Reihen- und Parallelschaltung zusammengesetzten Verknüpfung hat der Ausgang 1.0 nur dann den Signalzustand 1, wenn in jedem der beiden Parallelzweige mindestens einer der Kontakte den Signalzustand 1 aufweisen.

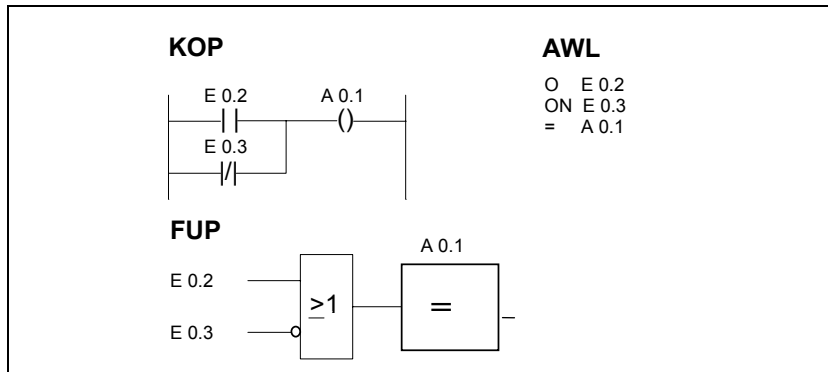


Damit ODER- Verknüpfungen Vorrang vor UND- Verknüpfungen haben müssen sie durch Klammern zusammengefaßt werden.

2.6 ABFRAGE AUF SIGNALZUSTAND 0

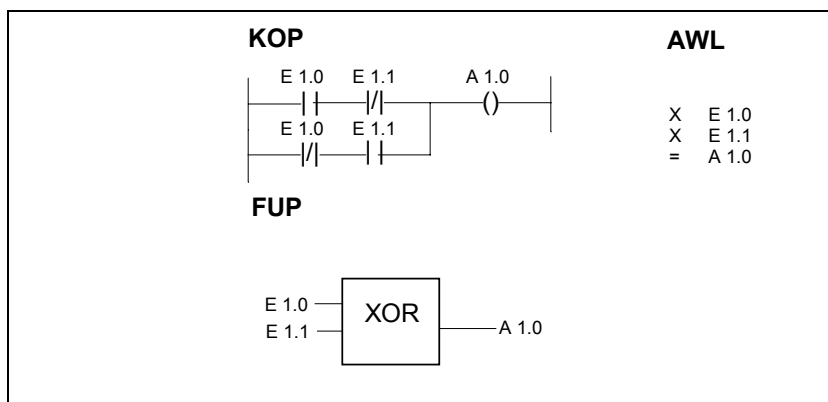
Die Abfrage auf Signalzustand 0 entspricht in einer kontaktbehafteten Schaltung einem Öffner und ist in den Verbindungen UND NICHT (UN), ODER NICHT (ON) und EXKLUSIV ODER NICHT (XN) realisiert.

Beispiel einer ODER NICHT - Verknüpfung:



2.7 EXKLUSIV - ODER - VERKNÜPFUNG

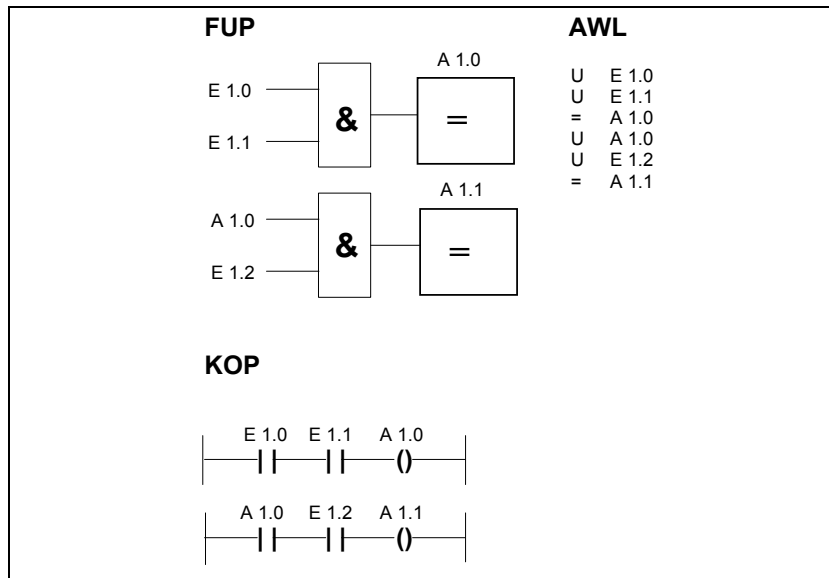
Die Schaltung zeigt eine Exklusiv- ODER- Verknüpfung (X), bei der der Ausgang 1.0 nur dann eingeschaltet ist (Signalzustand 1), wenn *nur einer* der Eingänge den Signalzustand 1 aufweist. In einer kontaktbehafteten Schaltung kann dies nur mit Öffnern und Schließern realisiert werden.



Hinweis: Die Exklusiv- ODER- Verknüpfung darf nur mit genau zwei Eingängen verwendet werden.

2.8 ABFRAGE VON AUSGÄNGEN

Für das Einschalten der Ausgänge A 1.0 und A 1.1 gelten unterschiedliche Bedingungen. In diesen Fällen muss für jeden Ausgang ein eigener Strompfad bzw. ein eigenes Verknüpfungssymbol vorgesehen werden. Da das Automatisierungsgerät nicht nur den Signalzustand von Eingängen, sondern auch den von Ausgängen, Merkern usw. abfragen kann, wird in der UND- Verknüpfung für den Ausgang A 1.1 der Ausgang A 1.0 abgefragt.



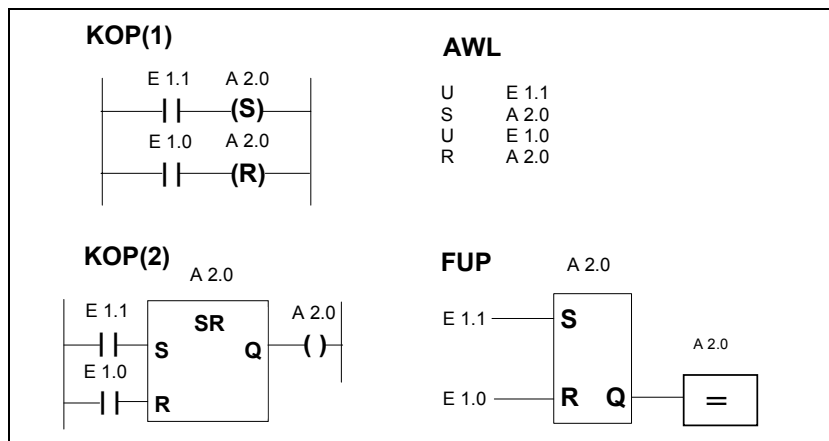
2.9 R - S - SPEICHERFUNKTIONEN

Nach DIN 40900 und DIN 19239 wird eine R-S- Speicherfunktion als Rechteck mit dem Setzeingang S und dem Rücksetzeingang R dargestellt.

Kurzzeitiger Signalzustand 1 am Setzeingang S setzt die Speicherfunktion. Kurzzeitiger Signalzustand 1 am Rücksetzeingang R führt zum Rücksetzen der Speicherfunktion. Signalzustand 0 an den Eingängen R und S verändert den vorher eingestellten Zustand nicht.

Sollte an beiden Eingängen R und S der Signalzustand 1 gleichzeitig anstehen, wird vorrangig gesetzt oder zurückgesetzt. Dieses vorrangige Rücksetzen oder Setzen muß bei der Programmierung berücksichtigt werden.

2.9.1 VORRANGIGES ZURÜCKSETZEN

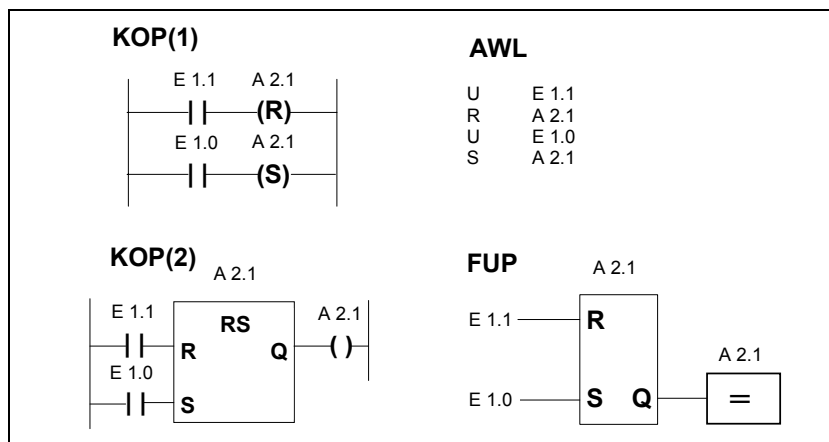


Die zuletzt programmierten Anweisungen werden von der Steuerung mit Vorrang bearbeitet. Im Beispiel wird zunächst die Setzoperation ausgeführt; der Ausgang A 2.0 wird wieder zurückgesetzt und bleibt für den Rest der Programmbearbeitung zurückgesetzt.

Dieses kurzzeitige Setzen des Ausgangs wird nur im Prozessabbild durchgeführt. Der Signalzustand auf der dazugehörigen Peripheriebaugruppe wird während der Programmbearbeitung nicht beeinflusst.

2.9.2 VORRANGIGES SETZEN

Gemäß Abschnitt 4.10.1. wird der Ausgang A 2.1 in diesem Beispiel mit Vorrang gesetzt.



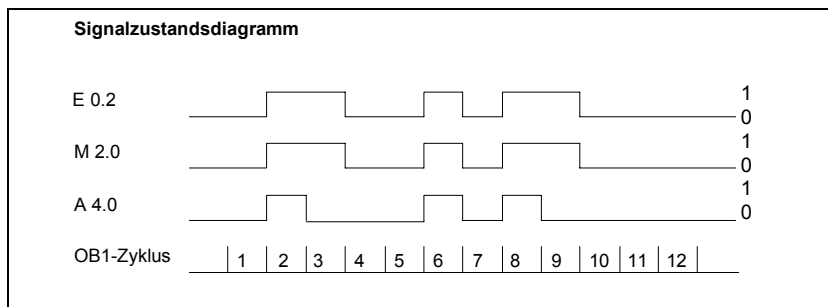
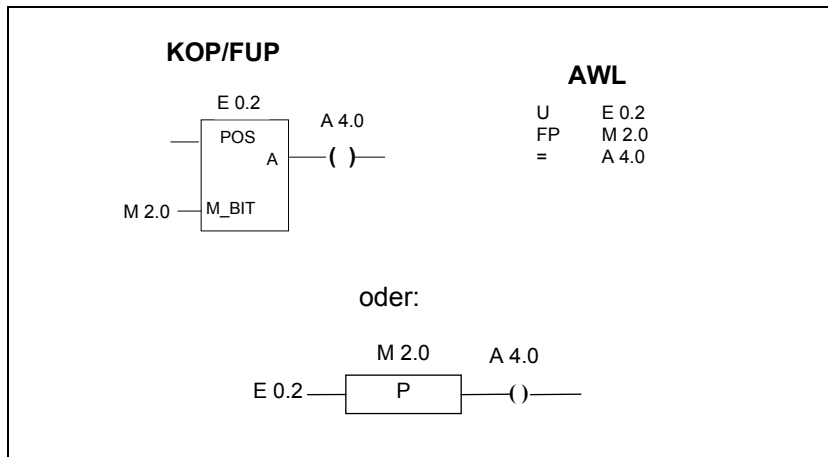
2.10 FLANKENOPERATIONEN

Die Flankenoperationen erfassen im Gegensatz zu einem statischen Signalzustand "0" und "1" die *Signaländerung*, z.B. eines Eingangs. Das Programm einer Flankenoperation entspricht einem flankenerkennenden Kontakt in einem Relaisschaltkreis.

2.10.1 STEIGENDE FLANKE (FP)

Ist eine steigende (positive) Flanke (Wechsel von '0' auf '1') an E 0.2 erkannt, so wird A 4.0 für einen OB1-Zyklus auf '1' gesetzt. Dieser Ausgang kann wiederum verwendet werden um z.B. einen Merker zu setzen. Eine steigende Flanke wird erkannt, indem das Automatisierungssystem das VKE, das die Operation U geliefert hat, im Flankenmerker M 2.0 speichert, und es mit dem VKE des vorhergehenden Zyklus vergleicht.

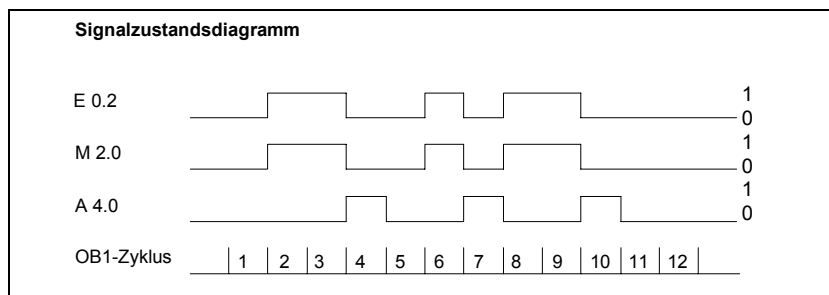
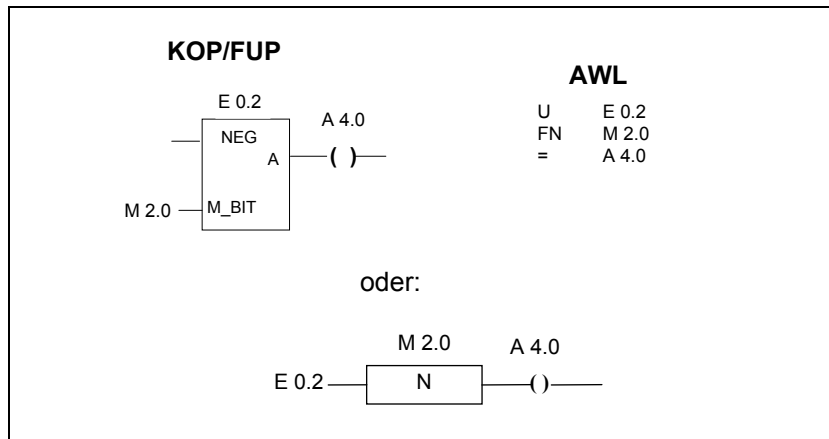
Der Vorteil der zweiten Darstellungsart in KOP/FUP ist, dass hier am Eingang der Flankenoperation auch Verknüpfungen stehen können.



2.10.2 FALLENDE FLANKE (FN)

Ist eine fallende (negative) Flanke (Wechsel von '1' auf '0') an E 0.2 erkannt, so wird A 4.0 für einen OB1-Zyklus auf '1' gesetzt. Dieser Ausgang kann wiederum verwendet werden um z.B. einen Merker zu setzen. Eine fallende Flanke wird erkannt, indem das Automatisierungssystem das VKE, das die Operation U geliefert hat, im Flankenmerker M 2.0 speichert, und es mit dem VKE des vorhergehenden Zyklus vergleicht.

Der Vorteil der zweiten Darstellungsart in KOP/FUP ist, dass hier am Eingang der Flankenoperation auch Verknüpfungen stehen können.



2.11 ZEITFUNKTIONEN

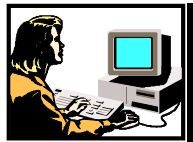
Für die Realisierung von Steuerungsaufgaben müssen sehr häufig verschiedene Zeitfunktionen eingesetzt werden. Die Zeitfunktionen sind in der Zentralbaugruppe des Automatisierungsgeräts integriert. Die Einstellung der gewünschten Laufzeit und das Starten der Zeitfunktion muss über das Anwenderprogramm erfolgen. Die SIMATIC - Automatisierungsgeräte stellen Ihnen eine bestimmte Anzahl von Zeitgliedern (CPU abhängig) mit unterschiedlichen Zeitfunktionen zur Verfügung. Jedem der Zeitglieder ist ein 16-BIT-WORT zugeordnet.

Folgende Funktionen können bei einer Zeit programmiert werden:

2.11.1 ZEIT FREIGEBEN (FR) NUR IN AWL

Ein positiver Flankenwechsel (von '0' auf '1') im Verknüpfungsergebnis der Operation Freigabe (FR) gibt eine Zeit frei.

Zum Starten oder für die normale Funktion einer Zeit wird die Freigabe nicht benötigt. Die Freigabe wird lediglich dazu verwendet, eine laufende Zeit nachzutriggern, d.h. sie wieder anlaufen zu lassen. Dieser Wiederanlauf ist nur dann möglich, wenn die Startoperation weiterhin mit dem VKE '1' bearbeitet wird.



Die Operation Freigabe (FR) existiert nur in der Darstellungsart AWL.

2.11.2 ZEIT STARTEN (SI/SV/SE/SS/SA)

Bei Signalwechsel am Starteingang (positive Flanke) wird das Zeitglied gestartet. Um eine Zeit zu starten, fügen Sie drei Anweisungen in Ihrem AWL- Programm ein:

- *Abfragen eines Signalzustandes*
- *Laden einer Startzeit in AKKU 1*
- *Startoperation (wahlweise SI, SV, SE, SS oder SA)*

z.B.:	
U	E 0.0
L	S5T#2S
SE	T5

2.11.3 ZEITWERT - VORGABE (TW)

Ein Zeitglied soll immer eine bestimmte Zeit ablaufen. Die Zeitdauer TW kann entweder als vordefinierte Konstante im Programm fest vergeben werden oder als Eingangswort EW, als Ausgangswort AW, als Datenwort DBW/DIW, als Lokaldatenwort LW oder als Merkerwort MW vorgegeben werden. Das Aktualisieren der Zeit vermindert den Zeitwert um jeweils eine Einheit in einem Intervall, das von der Zeitbasis festgelegt wurde.

Einen vordefinierten Zeitwert können Sie mit folgender Syntax laden:

- *L W#16#abcd*
 - mit: a = Zeitbasis binär- codiert(d.h. Zeitintervall oder Auflösung; siehe unten)
 - bcd = Zeitwert im BCD- Format
- *L S5T#aH_bbM_ccS_dddMS*
 - mit: a = Stunden, bb = Minuten, cc = Sekunden und ddd = Millisekunden
 - Die Zeitbasis wird automatisch gewählt

Zeitbasis:

Die Zeitbasis definiert das Intervall, in dem der Zeitwert um eine Einheit vermindert wird. Werte die keine genaue Vielfache des Zeitintervalls sind werden abgeschnitten. Werte, deren Auflösung für den gewünschten Bereich zu groß ist, werden abgerundet.

<u>Zeitbasis</u>	<u>Binärcode</u>	<u>Zeitbereich</u>
10ms	00	10MS bis 9S_990MS
100ms	01	100MS bis 1M_39S_900MS
1s	10	1S bis 16M_39S
10s	11	10S bis 2H_46M_30S

2.11.4 ZEIT RÜCKSETZEN (R)

Ein Signal am Rücksetzeingang beendet die Bearbeitung des Zeitgliedes. Der aktuelle Zeitwert wird gelöscht, der Ausgang Q der Zeitzelle zurückgesetzt.

2.11.5 ZEITWERT ABFRAGEN (L/LC)

Ein Zeitwert ist in einem Zeitwort binär- codiert gespeichert. Der im Zeitwort stehende Wert kann als Dualzahl (DUAL) oder als BCD- Zahl (DEZ) in den AKKU geladen und von dort in andere Operanden - bereiche transferiert werden.

Bei der AWL-Programmierung haben Sie die Wahl zwischen *L T1* für die Abfrage der Dualzahl und *LC T1* für die Abfrage der BCD- Zahl.

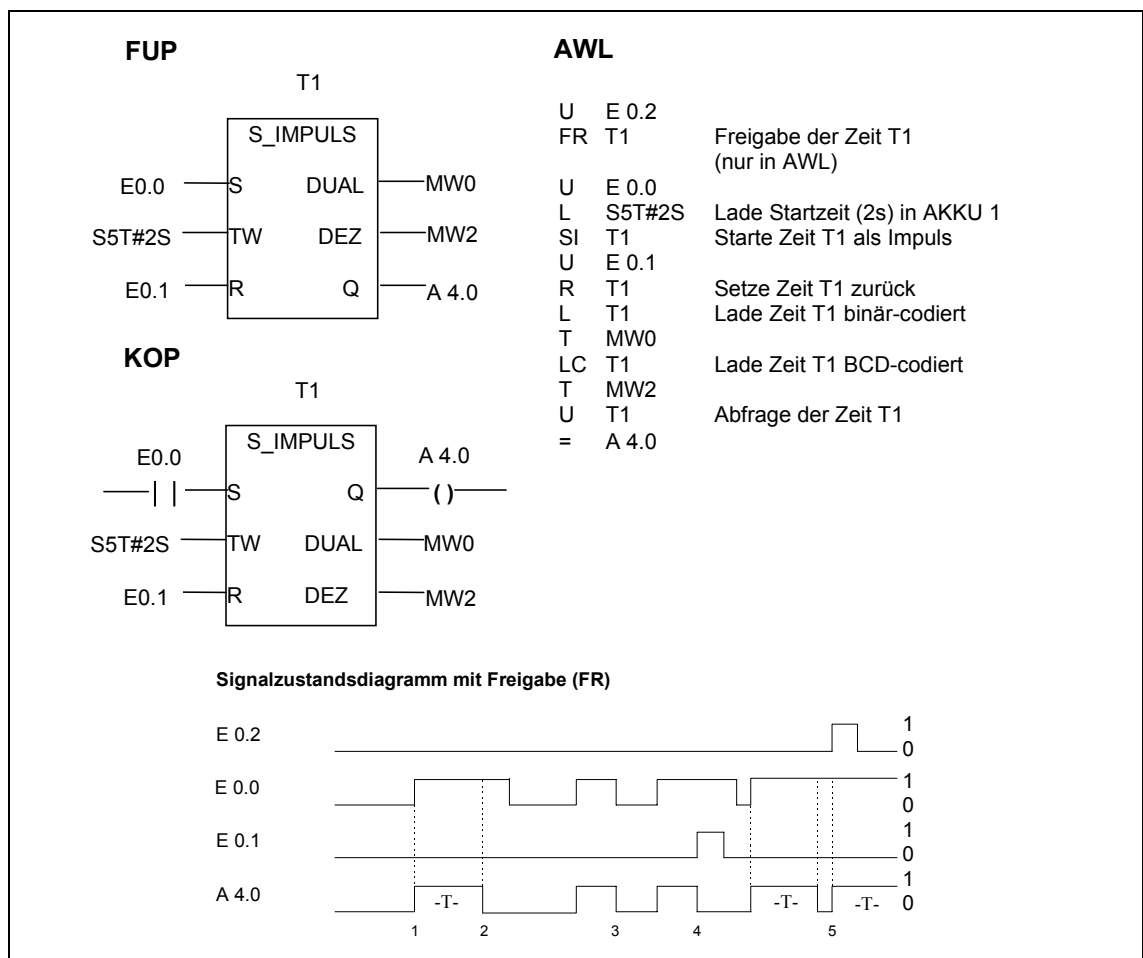
2.11.6 SIGNALZUSTAND DER ZEIT BINÄR ABFRAGEN (Q)

Eine Zeit kann auf ihren Signalzustand ('0' oder '1') abgefragt werden. Die Signalzustände können - wie gewohnt - mit U T1, UN T1, ON T1, etc... abgefragt und für weitere Verknüpfungen verwandt werden. Sie können unter fünf verschiedenen Zeiten auswählen:

2.11.7 ZEIT ALS IMPULS (SI)

Der Ausgang eines Zeitglieds, das als Impulsglied gestartet wird, führt nach dem Starten Signalzustand 1 (1). Der Ausgang wird zurückgesetzt, wenn die programmierte Zeitdauer abgelaufen ist (2), wenn das Startsignal auf Null zurückgesetzt wird (3) oder wenn am Rücksetzeingang des Zeitglieds Signalzustand 1 ansteht (4).

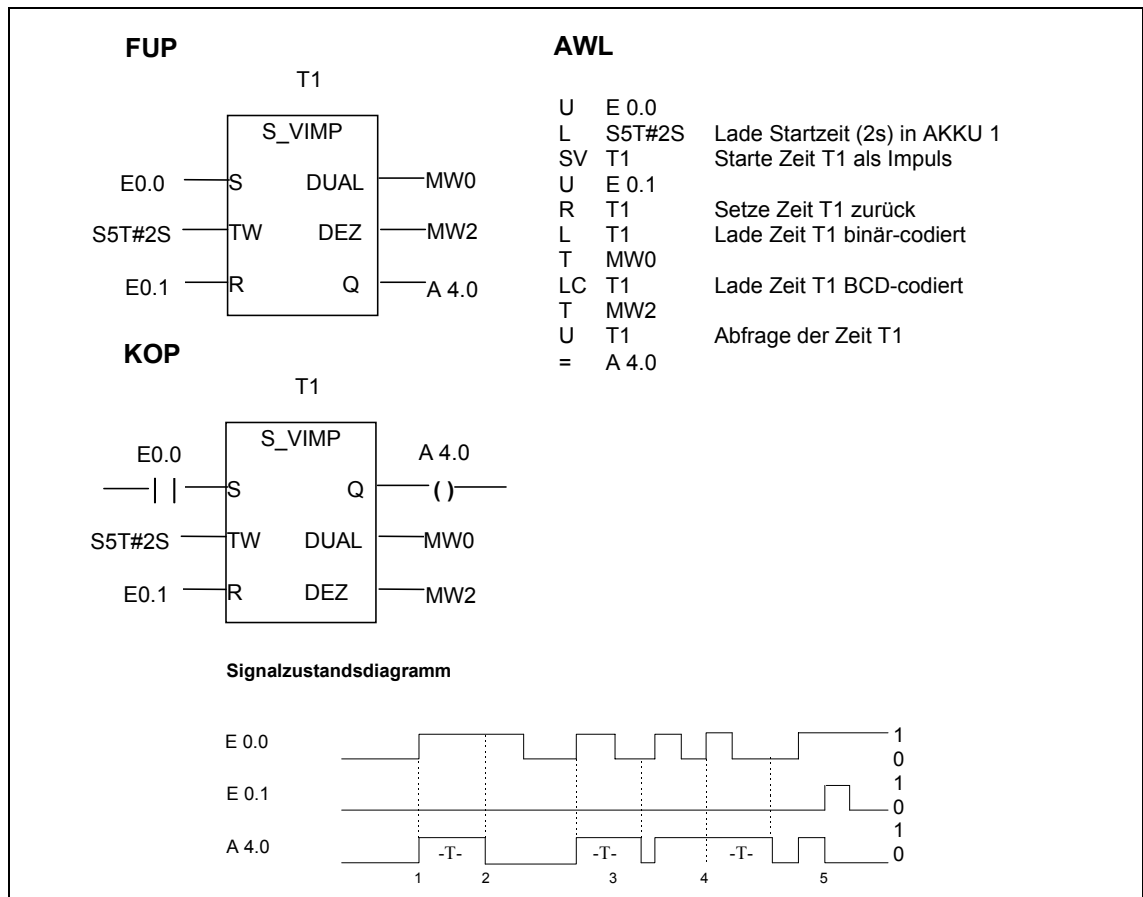
Ein positiver Flankenwechsel (von '0' auf '1') im Verknüpfungsergebnis der Operation Freigabe (FR) startet die Zeit neu (5). Dieser Wiederanlauf ist nur dann möglich, wenn die Startoperation weiterhin mit dem VKE '1' bearbeitet wird.



2.11.8 VERLÄNGERTER IMPULS (SV)

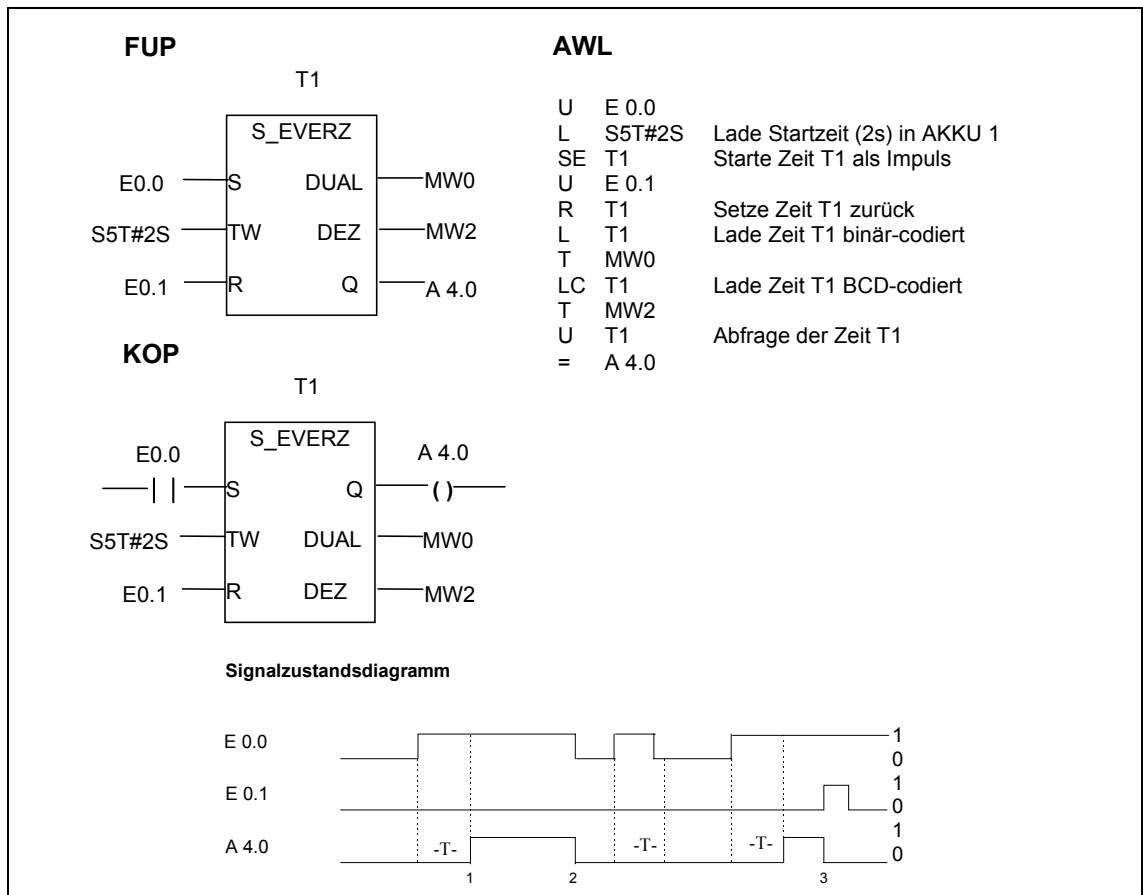
Der Ausgang eines Zeitglieds, das als verlängerter Impuls gestartet wird, führt nach dem Starten Signalzustand 1 (1). der Ausgang wird zurückgesetzt, wenn die vorgegebene Zeitdauer abgelaufen ist (2) oder der Rücksetzeingang der Zeitfunktion eingeschaltet wird (5).

Ein Ausschalten des Starteingangs bewirkt, während die Zeit läuft, kein Zurücksetzen des Ausgangs (Selbsthaltung) (3). Tritt - während die Zeit noch läuft - ein erneuter Signalwechsel auf 1 am Starteingang auf, wird das Zeitglied neu gestartet (nachgetriggert) (4).



2.11.9 EINSCHALTVERZÖGERUNG (SE)

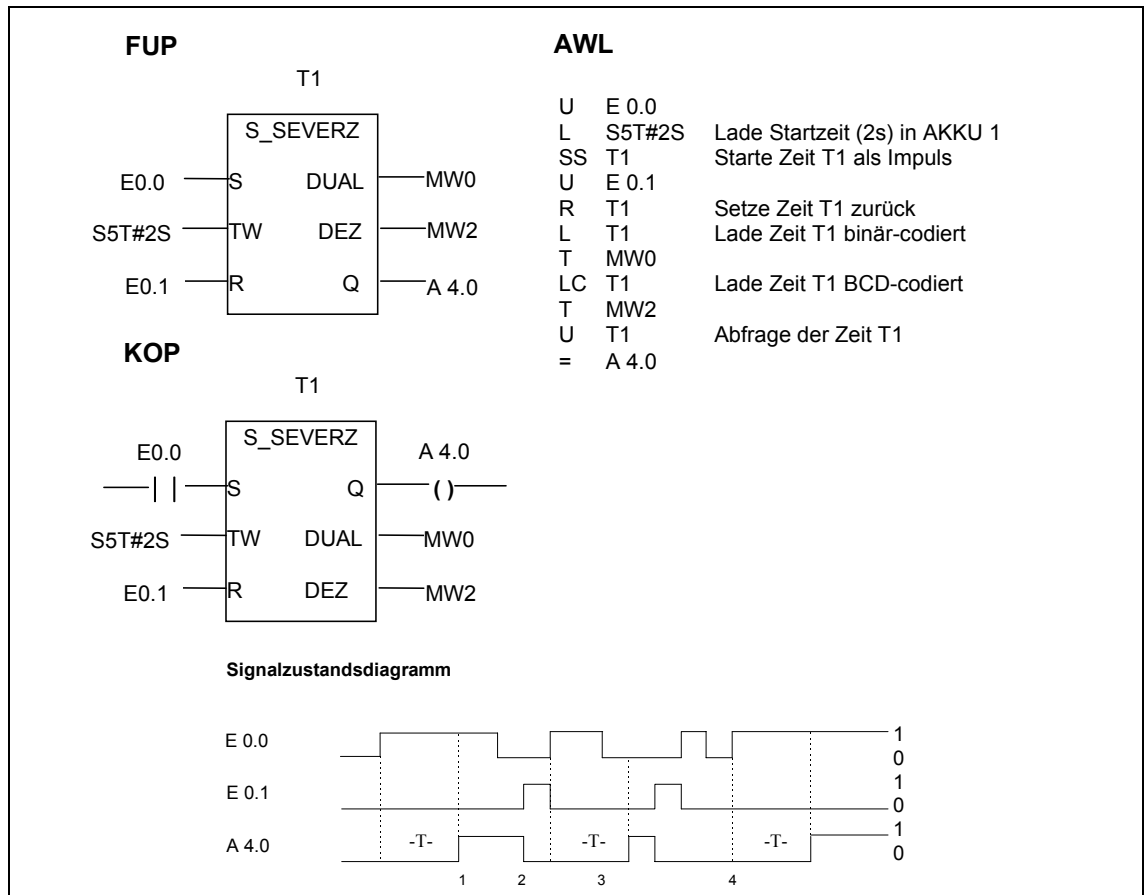
Der Ausgang eines Zeitgliedes, das als Einschaltverzögerung gestartet wird, führt nach dem Starten erst dann Signalzustand 1, wenn die programmierte Zeit abgelaufen ist und das VKE 1 am Starteingang noch ansteht (1). Das Einschalten des Starteingangs bewirkt also ein um die vorgegebene Zeitdauer verzögertes Einschalten des Ausgangs Q. Der Ausgang wird zurückgesetzt, wenn der Starteingang ausgeschaltet wird (2) oder wenn am Rücksetzeingang des Zeitgliedes Signalzustand 1 ansteht (3). Der Ausgang Q wird nicht eingeschaltet, falls noch während die Zeit läuft der Starteingang ausgeschaltet wird, oder Signalzustand 1 am Rücksetzeingang des Zeitglieds anliegt.



2.11.10 SPEICHERNDE EINSCHALTVERZÖGERUNG (SS)

Der Ausgang eines Zeitglieds, das als Speichernde Einschaltverzögerung gestartet wird, führt nach dem Starten erst dann Signalzustand 1, wenn die programmierte Zeit abgelaufen ist (1). Die Funktion benötigt nach dem Starten am Starteingang kein VKE 1 mehr, dieser kann also ausgeschaltet werden (Selbsthaltung) (3).

Der Ausgang wird nur dann zurückgesetzt, wenn der Rücksetzeingang der Zeitfunktion eingeschaltet wird (2). Ein Aus- und erneutes Einschalten des Starteingangs bewirkt, solange die Zeit läuft, dass die Zeitfunktion neu gestartet (nachgetriggert) wird (4).

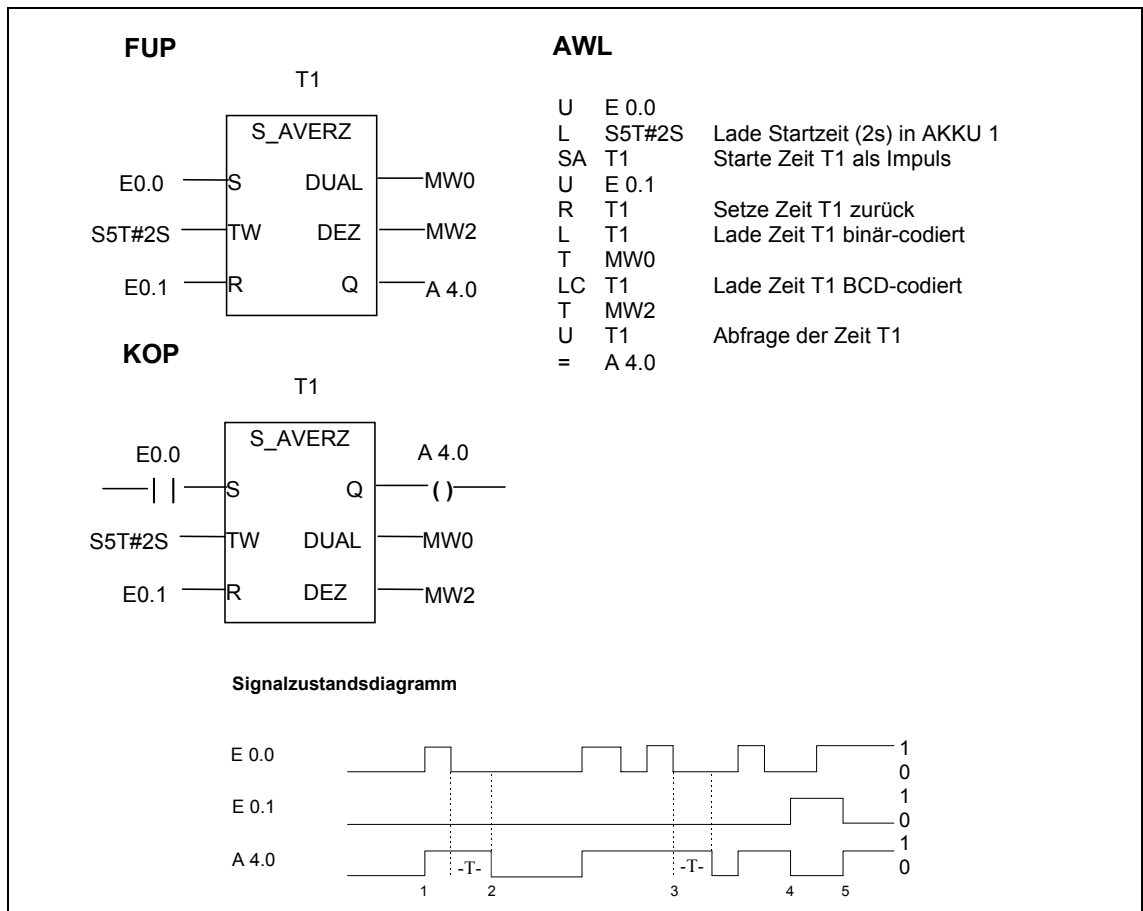


2.11.11 AUSSCHALTVERZÖGERUNG (SA)

Bei einem Signalwechsel (positive Flanke) am Starteingang eines Zeitglieds, das als Ausschaltverzögerung gestartet wird, wird der Ausgang Q der Zeitfunktion eingeschaltet (1). Wird der Starteingang ausgeschaltet, liefert der Ausgang noch solange Signalzustand 1, bis die programmierte Zeit abgelaufen ist (2). Das Ausschalten des Starteingangs (fallende Flanke) bewirkt also ein, um die vorgegebene Zeitdauer, verzögertes Ausschalten des Ausgangs.

Der Ausgang des Zeitglieds wird auch ausgeschaltet, wenn am Rücksetzeingang Signalzustand 1 ansteht (4). Ein erneutes Einschalten der Zeitfunktion bewirkt, während die Zeit läuft, daß die ablaufende Zeit angehalten und erst beim nächsten Ausschalten des Starteingangs neu gestartet wird (3).

Führen sowohl der Start- als auch der Rücksetzeingang der Zeitfunktion Signalzustand 1, wird der Ausgang des Zeitglieds erst gesetzt, wenn das dominierende Rücksetzen ausgeschaltet ist (5).



2.12 TAKTGEBER

Taktgeber werden für unterschiedliche Kontroll-, Überwachungs- und Steuerungsaufgaben eingesetzt. In der Digitaltechnik werden sie als astabile Kippschaltungen bezeichnet. Häufig benötigt man in der Praxis für Betriebs- oder Störmeldungen eine Blinkfrequenz.

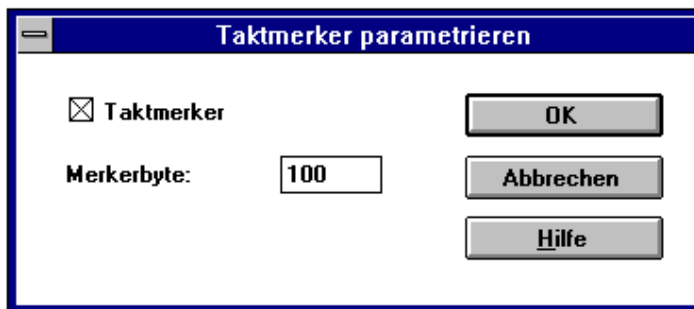


In der CPU S7-300 existiert ein parametrierbarer Taktmerker, der mit dem Tool HW Konfig eingestellt werden kann.

Taktmerker parametrieren:

Taktmerker sind Merker innerhalb eines Taktmerkerbytes. Ein beliebiges Merkerbyte der CPU wird durch Parametrierung (Im Tool Configuration auf die Zeile CPU doppelklicken!) zum Taktmerkerbyte. Ein Taktmerker ändert seinen binären Wert periodisch.

Wenn Sie Taktmerker aktivieren (Kreuz im Kontrollkästchen ist sichtbar), dann müssen Sie auch die Nummer des Merkerbytes bestimmen. Das gewählte Merkerbyte kann nicht für die Zwischenspeicherung von Daten genutzt werden.



Periodendauer von Takten:

Jedem Bit des Taktmerkerbytes ist eine Periodendauer/Frequenz zugeordnet. Es gilt folgende Zuordnung:

Bit:	7	6	5	4	3	2	1	0
Periodendauer (s):	2	1,6	1	0,8	0,5	0,4	0,2	0,1
Frequenz (Hz):	0,5	0,625	1	1,25	2	2,5	5	10

2.13 ZÄHLOPERATIONEN

In der Steuerungstechnik werden für das Erfassen von Stückzahlen oder Impulsen, zur Auswertung von Zeiten und Entfernungen Zählfunktionen benötigt. Bei der SIMATIC S7 sind Zähler bereits in der Zentralbaugruppe integriert. Diese Zähler besitzen einen eigenen reservierten Speicherbereich. Der Bereich des Zählwertes liegt zwischen 0 und 999.

Folgende Funktionen können bei einem Zähler programmiert werden:

2.13.1 ZÄHLER FREIGEBEN (FR) NUR IN AWL

Ein positiver Flankenwechsel (von '0' auf '1') im Verknüpfungsergebnis der Operation Freigabe (FR) gibt einen Zähler frei.

Weder für das Setzen eines Zählers, noch für normale Zähloperationen wird eine Zählerfreigabe benötigt. Will man jedoch ohne steigende Flanke vor der entsprechenden Zähloperation (ZV, ZR oder S) einen Zähler setzen bzw. Vorwärts - oder Rückwärtszählen , so kann dies mit einer Freigabe erfolgen. Dies ist jedoch nur möglich, wenn das VKE- Bit vor der entsprechenden Operation (ZV, ZR oder S) den Signalzustand '1' hat.



Die Operation Freigabe (FR) existiert nur in der Darstellungsart AWL.

2.13.2 VORWÄRTSZÄHLEN (ZV)

Der Wert des adressierten Zählers wird um 1 erhöht. Die Funktion wird nur bei einem positiven Flankenwechsel der vor ZV programmierten Verknüpfung wirksam. Erreicht der Zählwert die obere Grenze 999, wird er nicht mehr erhöht. (*Ein Übertrag wird nicht gebildet!*)

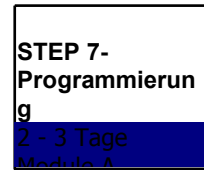
2.13.3 RÜCKWÄRTSZÄHLEN (ZR)

Der Wert des adressierten Zählers wird um 1 verringert. Die Funktion wird nur bei einem positiven Flankenwechsel der vor ZR programmierten Verknüpfung wirksam. Erreicht der Zählwert die untere Grenze 0, wird er nicht mehr verringert. (*Nur positive Zählwerte!*)

2.13.4 ZÄHLER SETZEN (S)

Um einen Zähler zu setzen, fügen Sie drei Anweisungen in Ihrem AWL- Programm ein:

- Abfragen eines Signalzustandes
- Laden eines Zählwertes
- Setzen eines Zählers mit dem geladenen Zählwert
Diese Funktion wird nur bei einem positiven Flankenwechsel der Abfrage bearbeitet.



2.13.5 ZÄHLWERT - VORGABE (ZW)

Wird ein Zähler gesetzt, so wird der Inhalt von AKKU 1 als Zählwert verwendet. Sie haben die Möglichkeit den Zählwert entweder binär- codiert oder BCD- codiert einzuladen.

Als Operanden sind möglich:

- | | | | |
|---|-----------------------|-------------------|------------------|
| - | <i>Eingangswort</i> | <i>EW</i> .. | |
| - | <i>Ausgangswort</i> | <i>AW</i> .. | |
| - | <i>Merkerwort</i> | <i>MW</i> .. | |
| - | <i>Datenwort</i> | <i>DBW/DIW</i> .. | |
| - | <i>Lokaldatenwort</i> | <i>LW</i> .. | |
| - | <i>Konstanten</i> | <i>C#5,</i> | <i>2#...etc.</i> |

2.13.6 ZÄHLER RÜCKSETZEN (R)

Bei VKE 1 wird der Zähler auf Null gesetzt (rücksetzen). Bei VKE 0 bleibt der Zähler unbeeinflusst. Das Rücksetzen eines Zählers wirkt statisch. Bei erfüllter Rücksetzbedingung kann weder gesetzt noch gezählt werden.

2.13.7 ZÄHLWERT ABFRAGEN (L/LC)

Ein Zählwert ist in einem Zählerwort binär- codiert gespeichert. Der im Zähler stehende Wert kann als Dualzahl (DU) oder als BCD- Zahl (DE) in den AKKU geladen und von dort in andere Operandenbereiche transferiert werden.

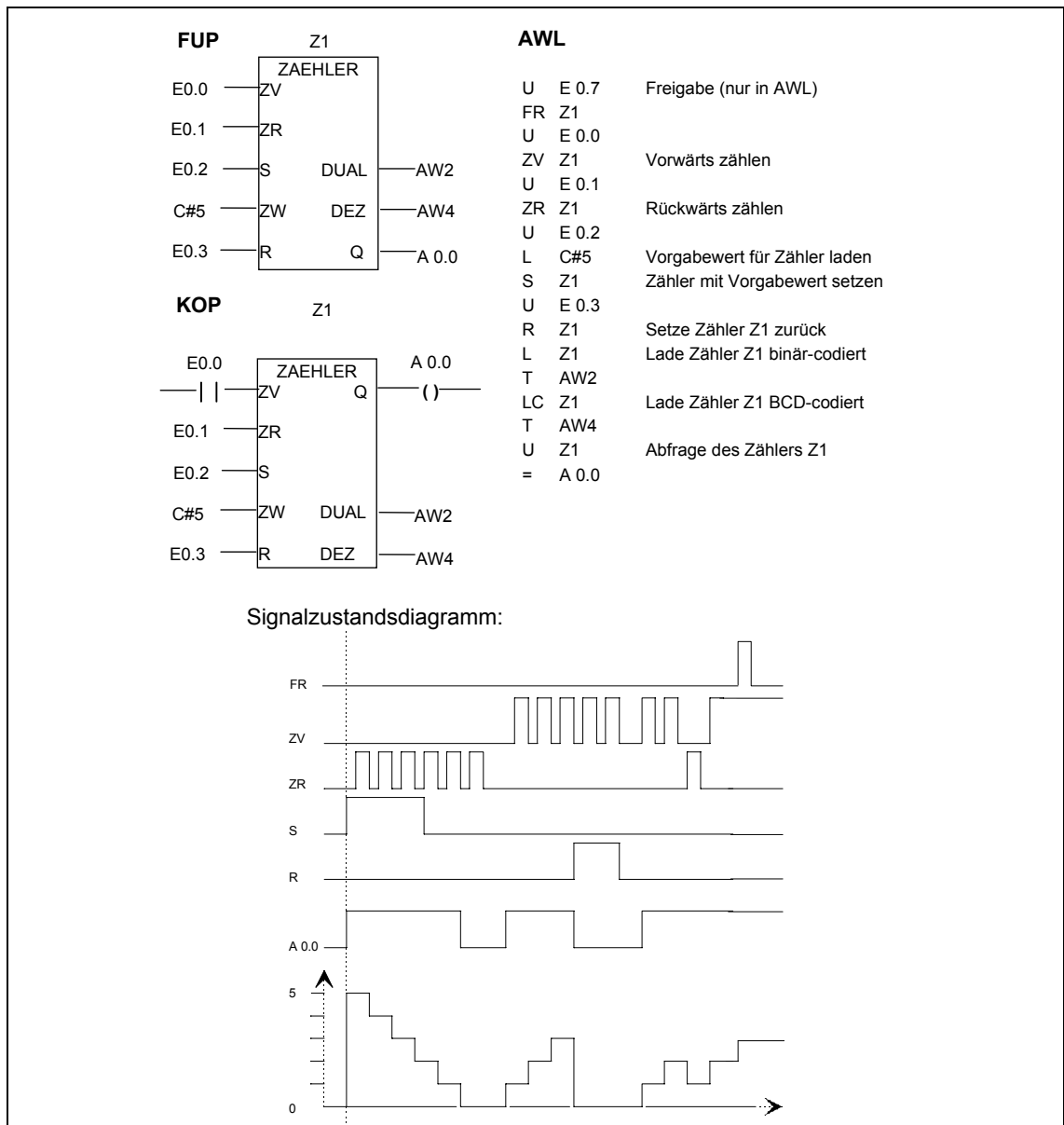
Bei der AWL- Programmierung haben Sie die Wahl zwischen *L Z1* für die Abfrage der Dualzahl und *LC Z1* für die Abfrage der BCD- Zahl.

2.13.8 SIGNALZUSTAND DES ZÄHLERS BINÄR ABFRAGEN (Q)

Der Zähler kann auf seinen Signalzustand abgefragt werden. Hierbei bedeutet:

- Signalzustand 0 = Zähler steht auf Wert 0;
- Signalzustand 1 = Zähler arbeitet, d.h. ist zählbereit.

Die Signalzustände können - wie gewohnt - mit *U Z1*, *UN Z1*, *ON Z1*, etc... abgefragt und für weitere Verknüpfungen verwandt werden.



2.14 LADE - UND TRANSFEROPERATIONEN (L/T)

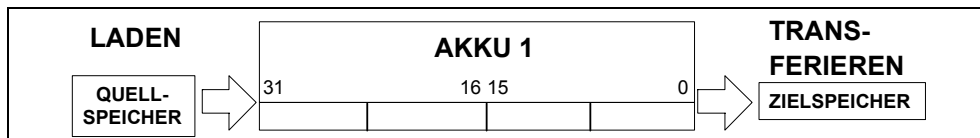
In der Programmiersprache STEP7 ermöglichen Lade- und Transferoperationen den byte-, wort-, bzw. doppelwortweisen Austausch von Informationen zwischen Ein- und Ausgabebaugruppen, dem Prozessabbild der Ein- und Ausgänge, den Zeit-, Zähler - und Merkerspeichern sowie den Datenbausteinen.

Dieser Informationsaustausch erfolgt nicht direkt, sondern immer über den Akkumulator 1 (AKKU 1). Der AKKU 1 ist ein Register im Prozessor und dient als Zwischenspeicher.

Der Informationsfluss ist richtungsgebunden:

LADEN: vom Quellspeicher in den AKKU 1

TRANSFERIEREN: vom AKKU in den Zielspeicher



Beim Laden wird der Inhalt des angesprochenen Quellspeichers kopiert und in den AKKU 1 geschrieben. Der bisherige Akkuinhalt wird dabei in ein AKKU 2 übertragen.

Beim Transferieren wird der Inhalt von AKKU 1 kopiert und in den angesprochenen Zielspeicher geschrieben.

Da der Akkuinhalt nur kopiert wurde, steht er für weitere Transferoperationen zur Verfügung.

AWL:

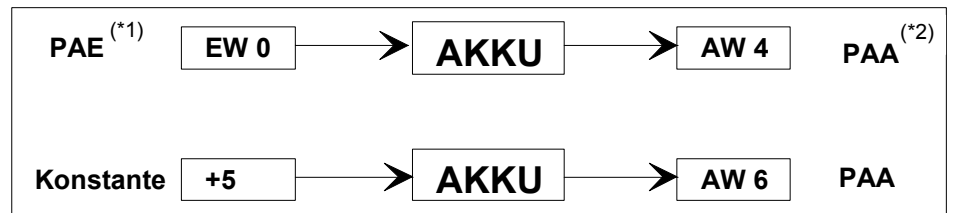
: L EW 0

: T AW 4

: L +5

: T AW 6

: BE



*1: Prozeßabbild der Eingänge

*2: Prozeßabbild der Ausgänge

Laden und Transferieren sind unbedingte Operationen, die unabhängig vom Verknüpfungsergebnis in jedem zyklischen Umlauf ausgeführt werden.

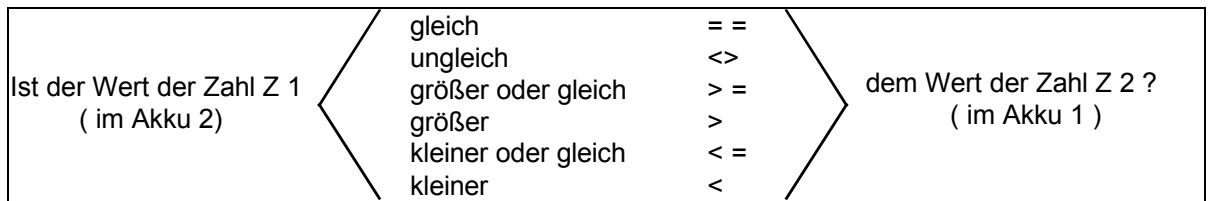
2.15 VERGLEICHSFUNKTIONEN

Die Programmiersprache STEP7 bietet Ihnen die Möglichkeit, zwei Zahlenwerte direkt zu vergleichen und das Ergebnis des Vergleichs (VKE) sofort weiterzuschalten. Voraussetzung dafür ist , dass beide Zahlen dasselbe Zahlenformat haben.

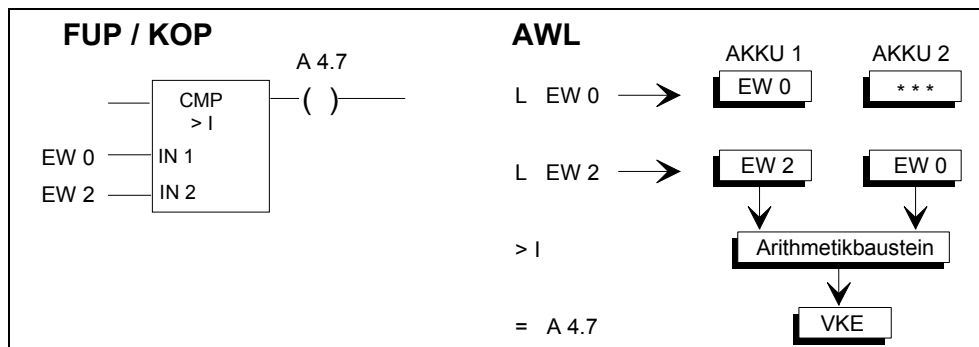
Folgende Paare von Zahlenwerten können verglichen werden:

- zwei Ganzzahlen (16 Bit Symbol: I)
- zwei Ganzzahlen (32 Bit Symbol: D)
- zwei Realzahlen (Gleitpunktzahlen 32 Bit, Symbol: R)

Sie können unter sechs verschiedenen Vergleichen auswählen:



Mit den Vergleichsfunktionen werden zwei Werte, die in den AKKU 's 1 und 2 stehen, direkt miteinander verglichen. Mit der ersten Ladeoperation wird der erste Operand (z.B. EW 0) in den AKKU 1 geladen. Mit der zweiten Ladeoperation wird zunächst der erste Operand vom AKKU 1 in den AKKU 2 umgeladen und dann der zweite Operand (z.B. EW 2) in den AKKU 1 geladen. Danach werden die in beiden Akkumulatoren stehenden Zahlenwerte im Arithmetikbaustein bitweise miteinander verglichen. Das Ergebnis des Vergleichs ist binär. Ist der gewünschte Vergleich erfüllt, wird das Verknüpfungsergebnis 1. Ist der gewünschte Vergleich nicht erfüllt, so wird das VKE 0.

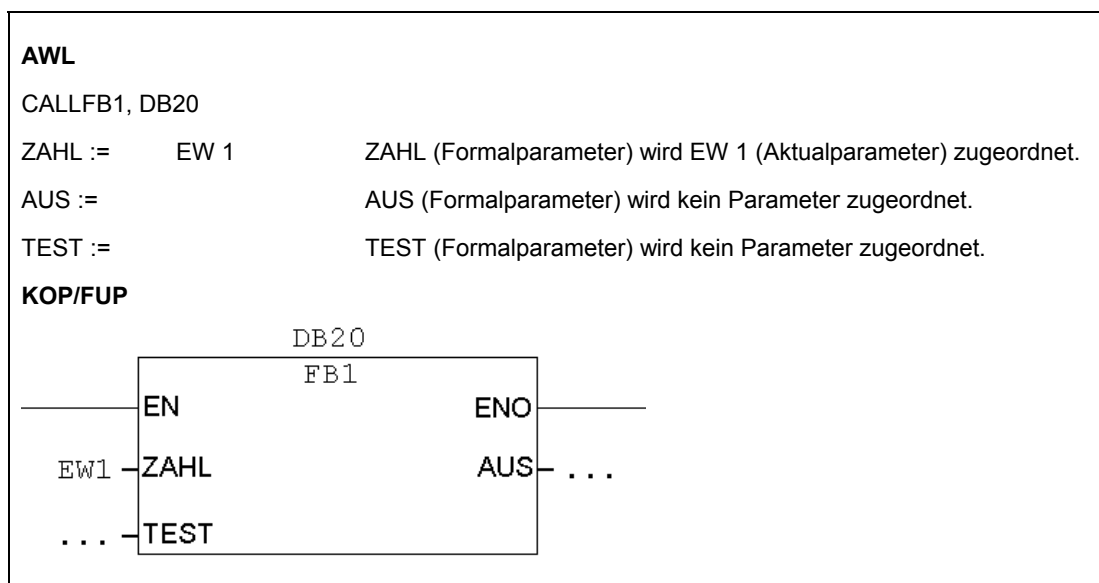


2.16 PROGRAMMORGANISATION

2.16.1 BAUSTEINAUFRUF (CALL)

Mit dem Bausteinaufruf *CALL* können Sie Funktionen (FCs) und Funktionsbausteine (FBs) sowie Systemfunktionen (SFCs) und Systemfunktionsbausteine (SFBs) aufrufen. Gleichzeitig können Parameter transferiert, bzw. Variablen beschrieben sowie der dem FB oder SFB zugehörige lokale Datenbaustein geöffnet werden.

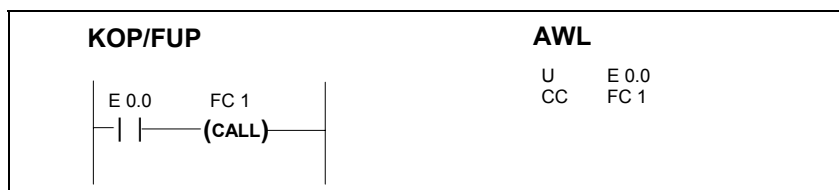
Sind in dem aufgerufenen Baustein keine Variablen deklariert, so entspricht dieser Befehl dem Befehl UC.



2.16.2 KONDITIONIERTER BAUSTEINAUFRUF (CC)

Mit dem Bausteinaufruf *CC* können Sie Funktionen (FCs) und Funktionsbausteine (FBs) sowie Systemfunktionen (SFCs) und Systemfunktionsbausteine (SFBs) aufrufen. Sie können jedoch keine Parameter transferieren bzw. Variablen beschreiben.

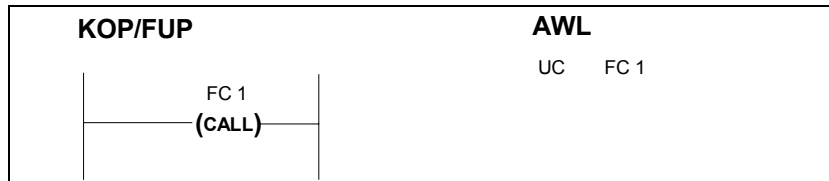
Der Aufruf wird nur ausgeführt, wenn das Verknüpfungsergebnis '1' beträgt.



2.16.3 UNKONDITIONIERTER BAUSTEINAUFRUF (UC)

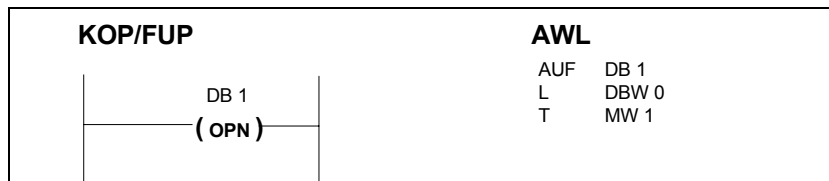
Mit dem Bausteinaufruf *UC* können Sie Funktionen (FCs) und Funktionsbausteine (FBs) sowie Systemfunktionen (SFCs) und Systemfunktionsbausteine (SFBs) aufrufen. Sie können jedoch keine Parameter transferieren bzw. Variablen beschreiben.

Der Aufruf wird unabhängig vom Verknüpfungsergebnis ausgeführt.



2.16.4 DATENBAUSTEIN AUFSCHLAGEN (AUF)

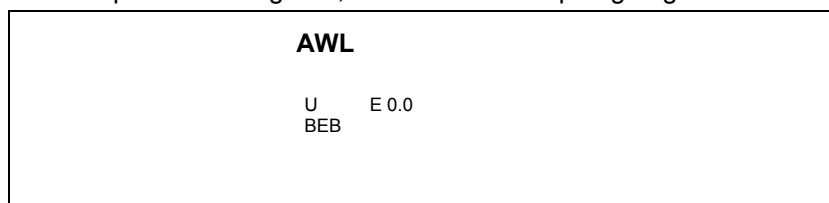
Mit der Operation Aufschlagen Datenbaustein (AUF) können Sie einen Datenbaustein (DB) oder Instanz - Datenbaustein (DI) öffnen, um auf die enthaltenen Daten zugreifen zu können (z.B. mit Lade- und Transferoperationen).



2.16.5 BAUSTEINENDE BEDINGT (BEB) NUR IN AWL

Abhängig vom Verknüpfungsergebnis beendet diese Operation die Bearbeitung des aktuellen Bausteins und springt zurück in den Baustein, der den gerade Beendeten aufgerufen hat.

Diese Operation erfolgt nur, wenn das Verknüpfungsergebnis '1' beträgt.



2.16.6 BAUSTEINENDE ABSOLUT (BEA) NUR IN AWL

Diese Operation beendet die Bearbeitung des aktuellen Bausteins und springt zurück in den Baustein, der den gerade Beendeten aufgerufen hat.

Diese Operation erfolgt unabhängig vom Verknüpfungsergebnis.

AWL

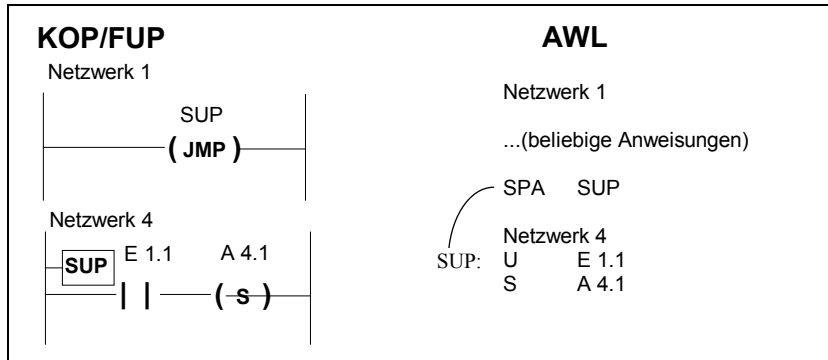
...(beliebige Anweisungen)

BEA

2.17 SPRUNGOOPERATIONEN

2.17.1 SPRINGE ABSOLUT (SPA)

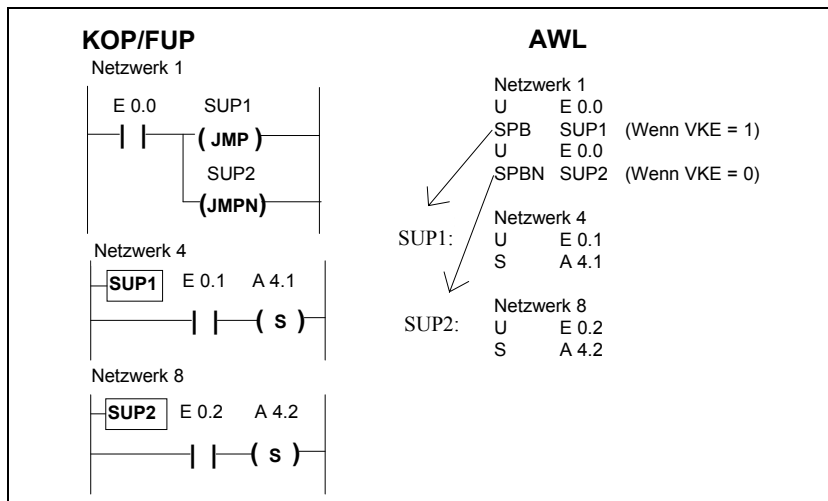
Die Operation SPA unterbricht den normalen Ablauf des Programms und springt zu der, im Operanden angegebenen, Sprungmarke.
Der Sprung erfolgt unabhängig vom Verknüpfungsergebnis.



2.17.2 SPRINGE BEDINGT (SPB/SPBN)

Die bedingten Sprungoperationen unterbrechen den normalen Ablauf des Programms und initiieren einen Sprung zu der, im Operanden angegebenen, Sprungmarke.
Der Sprung erfolgt in Abhängigkeit vom Verknüpfungsergebnis.
Folgende bedingte Sprungoperationen können Sie ausführen:

- **SPB** : Springe, wenn VKE = 1
- **SPBN** : Springe, wenn VKE = 0



2.17.3 PROGRAMMSCHLEIFE (LOOP) NUR IN AWL

Mit einer Programmschleife (LOOP) können Sie einen Programmteil mehrmals hintereinander bearbeiten.

Dazu müssen Sie eine Konstante in das niederwertige Wort von AKKU 1 laden. Diese Zahl wird dann durch die Operation LOOP um '1' vermindert. Danach wird der Wert dieser Zahl auf <>0 überprüft. Beträgt er nicht '0', dann wird ein Sprung zur Marke (hier ,NEXT') der Operation LOOP ausgeführt; andernfalls wird die nächste Operation ausgeführt.

```
      L      5
NEXT:T  MW 10
      .....
      L      MW 10
      LOOP  NEXT
```



Die Programmschleife (LOOP) existiert nur in der Darstellungsart AWL.

2.18 NULLOPERATIONEN

2.18.1 NULLOPERATION 0 / 1 (NOP0/NOP1) NUR IN AWL

Diese Operationen führen keine Funktion aus und beeinflussen den Inhalt des Statuswortes nicht. Der Übersetzer benötigt die Nulloperationen für die Rückübersetzung, z.B. von AWL in den KOP.

2.19 BEARBEITUNG DES VKE

In STEP7 gibt es Operationen mit denen das Verknüpfungsergebnis (VKE) verändert werden kann. Da dabei das VKE direkt beeinflusst wird, besitzen diese Operationen keine Operanden.

2.19.1 NEGIEREN DES VKE (NOT) NUR IN AWL

Sie können mit der Operation NOT in ihrem Programm das aktuelle VKE negieren (umkehren). Ist das aktuelle VKE '0', dann ändert die Operation NOT es in '1'; ist das aktuelle VKE '1', dann ändert NOT es in '0'.

2.19.2 SETZEN DES VKE (SET) NUR IN AWL

Sie können mit der Operation SET in ihrem Programm das VKE- Bit unconditioniert auf '1' setzen.

2.19.3 RÜCKSETZEN DES VKE (CLR) NUR IN AWL

Sie können mit der Operation CLR in ihrem Programm das VKE- Bit unconditioniert auf '0' rücksetzen.

2.19.4 SICHERN DES VKE (SAVE) NUR IN AWL

Sie können mit der Operation SAVE in ihrem Programm das VKE für einen zukünftigen Gebrauch im Statusbit (BIE) des Statuswortes sichern.



Das Statuswort enthält Bits, auf die Sie im Operanden der Bit- und Wortverknüpfungsoperationen zugreifen können.

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
z.B.:	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Anweisungsliste:	Signalzustand:	Verknüpfungsergebnis(VKE):
SET		1
= M 1.0	1	
= E 0.0	1	
CLR		0
= M 1.0	0	
= E 0.0	0	
NOT	1	
SAVE	1	sichern im BIE- Bit im Statuswort