

## **2. WORTVERARBEITUNG**

### **2.1 INFORMATIONSDARSTELLUNG BYTE, WORT, DOPPELWORT**

Neben dem Bit als kleinste Informationseinheit mit den beiden Zuständen "1" und "0" werden für die Darstellung aller vorkommenden Informationen das Byte, das Wort und das Doppelwort verwendet.

Damit die einzelnen Bits angesprochen werden können, wird jedem einzelnen Bit innerhalb eines Bytes eine Ziffer, die Bitadresse, zugeordnet. In jedem Byte bekommt das Bit rechts außen die Bitadresse 0 und das links außen stehende Bit die

Bitadresse 7.

### **2.2 DAS BIT.**

Das Bit ist die kleinste Informationseinheit. In ihm sind 2 Informationen 0 und 1. Werden mehrer Bits zusammengefasst so können Zahlen dargestellt werden.

Beispiel:

In 2 Bits sind 4 Informationseinheiten enthalten Es können die Zahlen 0 bis 3 dargestellt werden.

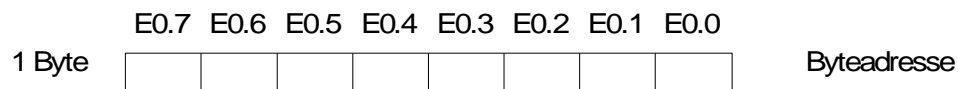
<b>E0.1</b>	<b>E0.0</b>	<b>Wert</b>
<b>2<sup>1</sup></b>	<b>2<sup>0</sup></b>	
<b>2</b>	<b>1</b>	
<b>0</b>	<b>0</b>	<b>0</b>
<b>0</b>	<b>1</b>	<b>1</b>
<b>1</b>	<b>0</b>	<b>2</b>
<b>1</b>	<b>1</b>	<b>3</b>

## **2.3 DAS BYTE**

Das Byte hat eine Länge von 8 Bit, die von rechts nach links durch die Bitadressen von 0...7 gekennzeichnet sind. In der SPS werden insbesondere die Operanden E=Eingang, A=Ausgang und M=Merker zusammengefasst und mit ihrer Byteadresse EB..., AB...,MB.. angesprochen. In einem Byte können insgesamt 256 Informationen zusammengefasst werden.

Auch die einzelnen Bytes erhalten Nummern, die Byteadressen. Zusätzlich wird noch der Operand gekennzeichnet, so dass z. B. EB 2 für Eingangsbyte 2 oder AB 4 für Ausgangsbyte 4 steht.

Einzelne Bits sind eindeutig durch die Kombination von Bit- und Byteadresse adressiert. Dabei wird die Bitadresse von der Byteadresse durch einen Punkt getrennt. Rechts vom Punkt steht die Bitadresse, links davon die Byteadresse.



### **Eingangsbyte 0**

128	64	32	16	8	4	2	1
E0.7	E0.6	E0.5	E0.4	E0.3	E0.2	E0.1	E0.0

## 2.4 DAS WORT

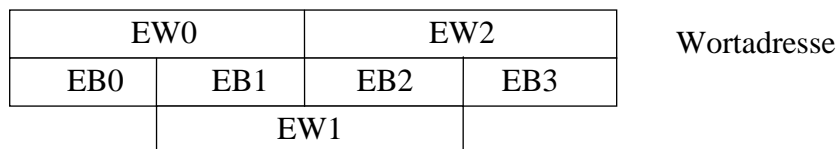
Das Wort ist nach dem Byte die nächst größere Einheit. Es hat eine Länge von 16 Bit, die von rechts nach links durch die Bitadressen 0.....15 gekennzeichnet sind. Insbesondere werden die in den "Datenbausteinen" abgelegten Daten wortweise mit ihrer Wortadresse DW (Datenwort) angesprochen.

In einem Wort können insgesamt 65536 Informationen hinterlegt werden. Es können die Zahlen 0 bis 65535 dargestellt werden.

Auch zwei Bytes aus dem Operandenbereichen Eingang, Ausgang und Merker können zu einem Eingangswort EW, Ausgangswort AW und Merkerwort MW zusammengefasst werden

Beispiel:  $EB\ 8 + EB\ 9 = EW\ 8$                        $AB\ 5 + AB\ 6 = AW\ 5$   
 $MB\ 1 + MB\ 2 = MW\ 1$

Bei der Verwendung von Wörtern, z.B. Eingangswort (EW), Ausgangswort (AW), Merkerwort (MW) usw. ist die Wortadresse immer die kleinere Adresse der zwei dazugehörigen Bytes. z.B. bei einem Wort, bestehend aus EB2 und EB3, ist dies die Adresse EW2.



Bei der Wortverarbeitung ist besonders darauf zu achten, dass sich z.B. das Eingangswort 0 und das Eingangswort 1 in einem Byte überschneiden. Außerdem beginnt man beim Zählen der Bits von rechts.

Zum Beispiel ist das Bit0 vom EW1 der E2.0. Das Bit1 der E2.1. ... Das Bit7 der E 2.7.

Das Bit8 der E1.0. ... Das Bit15 der E1.7. Zwischen den Bits 7 und 8 besteht somit ein Sprung.

Das Eingangswort 0

Eingangsbyte 0

Eingangsbyte 1

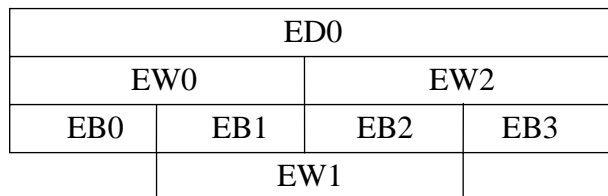
<b>E0.7</b>	<b>E0.6</b>	<b>E0.5</b>	<b>E0.4</b>	<b>E0.3</b>	<b>E0.2</b>	<b>E0.1</b>	<b>E0.0</b>	<b>E1.7</b>	<b>E1.6</b>	<b>E1.5</b>	<b>E1.4</b>	<b>E1.3</b>	<b>E1.2</b>	<b>E1.1</b>	<b>E1.0</b>
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2 <sup>15</sup>	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
High-Byte								Low-Byte							

## 2.5 DAS DOPPELWORT

Die Nummerierung von Doppelwörtern ergibt die Doppelwortadresse.

Bei der Verwendung von Doppelwörtern, z.B. ED, AD, MD usw. ist die

Doppelwortadresse immer die kleinere Wortadresse der zwei dazugehörigen Worte.



Doppelwortadresse

ED0

0000 0000 0000 0000 0000 0000 0000 0000

! !

! !

2 147 483 648

65536

In einem Doppelwort sind 4 294 967 276 Informationen enthalten.

## 2.6 DATENTYPEN

Mit den Datentypen können Sie die Größe und das Format der DATENELEMENTE IN Ihrem Programm festlegen. Damit Ihr Programm richtig bearbeitet werden kann, muss jeder Operation ein Operand mit dem korrekten Datentyp zugeordnet sein.

### Elementare Datentypen

1 Bit	1 Byte	1 Wort	1 Doppelwort
	8 Bit	2 Byte 16 Bit	4 Byte 32 Bit
BOOL	BYTE CHAR	WORD	DWORD
		INT DATE S5-TIME	DINT  REAL TIME

Schlüsselwort	Breite (in Bits)	Beispiel einer Konstanten dieses Typs
BOOL	1	1 oder 0
BYTE	8	B#16#A9
WORD	16	W#16#12AF
DWORD	32	DW#16#ADAC1EF5
CHAR	8	' w '
S5TIME	16	S5T#5s_200ms
INT	16	123
DINT	32	65539
REAL	32	1.2 oder 34.5E-12
TIME	32	T#2D_1H_3M_45S_12MS
DATE	16	D#1993-01-20
TIME_OF_DAY	32	TOD#12:23:45.12

Zusammengesetzte Datentype sind Datentypen die eine Breite von mehr als 32 Bit haben. Zusammengesetzte Datentypen können nur in Verbindung mit Variablen verwendet werden, die in globalen Datenbausteinen vereinbart worden sind. Zusammengesetzte Datentypen können nicht komplett mit Ladeoperationen in den Akku geladen werden. Hierzu stehen Standardbausteine aus der Bibliothek zur Verfügung.

Hierzu gehören die Datentypen

Date and Time

String

Array

Struct

UDT

### Zusammengesetzte Datentypen

Schlüsselwort	Breite (in Bits)	Beispiel	
<b>DATE_AND_TIME</b> (Datum und Uhrzeit)	64	DT#97-09-24-12:14:55.0	
<b>STRING</b> (Zeichenkette mit max. 254 Zeichen)	8 * (Anzahl der Zeichen +2)	'Dies ist ein String' 'SIEMENS'	
<b>ARRAY</b> (Feld, Gruppierung aus Komponenten gleichen Datentyps)	anwenderdefiniert	Messwerte: ARRAY[1..20] INT	
<b>STRUCT</b> (Struktur, Gruppierung aus Komponenten unterschiedlichen Datentyps)	anwenderdefiniert	Motor: STRUCT Drehzahl : INT Strom : REAL END_STRUCT	
<b>UDT</b> (User Defined Data Type = anwenderdefinierter Datentyp, "Schablone" aus elementaren bzw. zusammengesetzten Datentypen)	anwenderdefiniert	UDT als Baustein	UDT als Feldelement
		STRUCT Drehzahl : INT Strom : REAL END_STRUCT	Antrieb: ARRAY[1..4] UDT1

## DATENTYP BOOL

Datentyp	Länge(Bit)	Format	Beispiel für das Format		
Bool	1	Boolesch	True,False		

## DATENTYP BYTE

Byte n

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

Datentyp	Länge (Bit)	Format	Mindestwert	Maxwert
Byte	8	Hexadezimal	B#16#0	B#16#FF
		Binär	2#0	2#11111111

## DATENTYP WORD

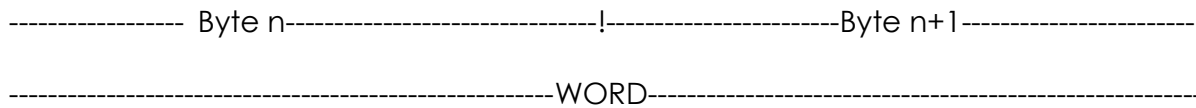
----- Byte n-----!-----Byte n+1-----

-----WORD-----

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

Datentyp	Länge (Bit)	Format	Mindestwert	Maxwert
WORD	16	Hexadezimal	W#16#0	W#16#FFFF
		Binär	2#0	2#1111111111111111
			B#(0,0)	B#(255,255)

### DATENTYP INTEGER

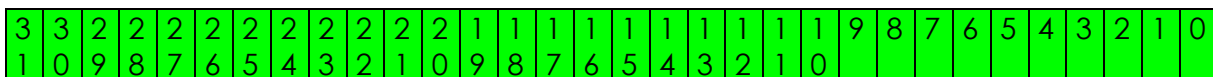
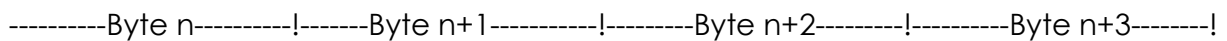


Datentyp	Länge (Bit)	Format	Mindestwert	Maxwert
INT	16	Ganzzahl mit Vorzeichen	-32768	+ 32767

Das Bit 15 ist das Vorzeichenbit

- 0 entspricht dem Vorzeichen „+“
- 1 entspricht dem Vorzeichen „-“

### DATENTYP DWORD



Datentyp	Länge	Format	Mindestwert	Maxwert
DWORD	32	Binär	2#00000000000000000000000000000000 000000000000	2#11111111111111111111111111111111 111111111111
		Hex	DW#16#0	DW#16#FFFFFFFF
		Bytes	B#(0,0,0,0)	B#(255,255,255,255)



## DATENTYP S5TIME

----- Byte n-----!-----Byte n+1-----

-----WORD-----



relevant 00=0,01 s  
 01=0,1 s  
 10=1 s  
 11=10 s

Datentyp	Länge (Bit)	Format	Mindestwert	Maxwert
S5Time	14	S5Time	S5T#0ms	S5T#2h46m30s

Wenn Sie den Datentyp S5TIME mit Zeiten verwenden, müssen Sie die Grenzwerte für den Bereich und die Auflösung beachten. Die folgende Tabelle gibt jeweils den Bereich für die verschiedenen Auflösungen an.

0,01 S	10 ms bis 9s 990 ms
0,1 S	100 ms bis 1m 39s 900 ms
1 S	1s bis 16m 39 s
10 S	10 S bis 2h 46m 30 S

## DATENTYP DINT

-----Byte n-----!-----Byte n+1-----!-----Byte n+2-----!-----Byte n+3-----!



Datentyp	Länge	Format	Mindestwert	Maxwert
DWORD	32	Ganzzahl	L#-2 147 483 648	L#+2 147 483 647

Das Bit 31 entspricht dem Vorzeichenbit

- 0 entspricht dem Vorzeichen „+“
- 1 entspricht dem Vorzeichen „-“

## DATENTYP CHAR

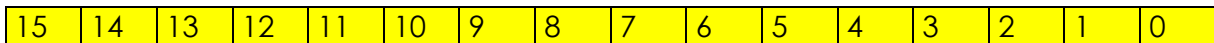
Mit dem Datentyp CHAR können bis zu 4 Buchstaben im ASCII-Zeichen in einem Byte hinterlegt werden.

Datentyp	Länge	Format	Beispiel für das Format
CHAR	8	ASCII-Zeichen	„OTTO“

## DATENTYP DATE

Das Datum wird als vorzeichenlose Ganzzahl in Tagen dargestellt, wobei der erste Tag der 1. Januar 1990 ist.

----- Byte n-----!----- Byte n+1-----  
 -----WORD-----

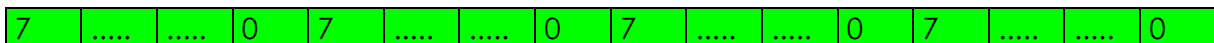


Datentyp	Länge (Bit)	Format	Mindestwert	Maxwert
DATE	16	Jahr-Monat-Tag	D#1990-01-01	D#2168-12-31

## DATENTYP TIME\_OF\_DAY

Der Datentyp TOD wird als vorzeichenlose Ganzzahl in Millisekunden gespeichert, wobei Null gleich Mitternacht ist.

-----Byte n-----!-----Byte n+1-----! -----Byte n+2-----!-----Byte n+3-----



Datentyp	Länge (Bit)	Format
TOD	32	Stunden:Minuten:Sekunden.Millisekunden

Beispiel: TOD#00:00:00.0000 unterer Grenzwert  
 TOD#23:59:59.999 oberer Grenzwert

Die Angabe der Millisekunden ist nicht erforderlich.

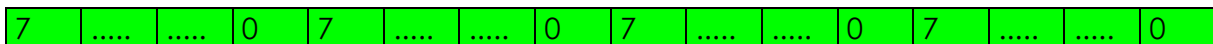
## DATENTYP TIME

Der Datentyp TIME wird in einem Doppelword hinterlegt und wird als Ganzzahl mit Vorzeichen in Millisekunden als Zweierkomplement gespeichert. Es ist nicht erforderlich, alle Zeiteinheiten anzugeben.

Beispiele für das Format:

Max: T#+24d20h31m23s647ms  
 Min: T#-24d20h31m23s647ms

-----Byte n-----!-----Byte n+1-----! -----Byte n+2-----!-----Byte n+3-----



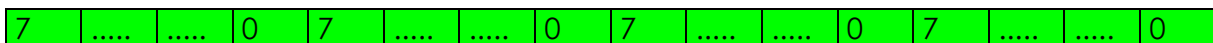
Datentyp	Länge (Bit)	Format
TIME	32	+Tage,Stunden,Minuten,Sekunden

## DATENTYP REAL

Gleitpunktzahlen werden bis auf die sechste Zahl genau gerundet. Die Werte werden exponentiell oder als Bruchzahlen gargestellt Die Länge beträgt 32 Bit.

Beispiele für das Format:           Min positiv: +1,175494e-38  
   Max positiv: +3.402823e+38  
   Min negativ: -1,175494e-38  
   Max negativ:-3.402823e+38

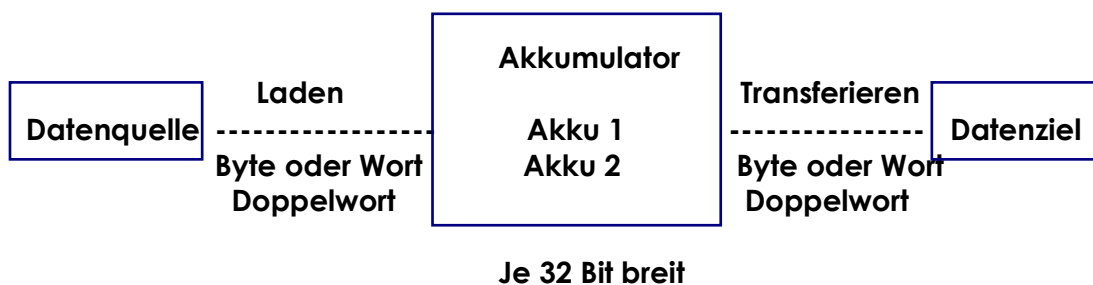
-----Byte n-----!-----Byte n+1-----! -----Byte n+2-----!-----Byte n+3-----



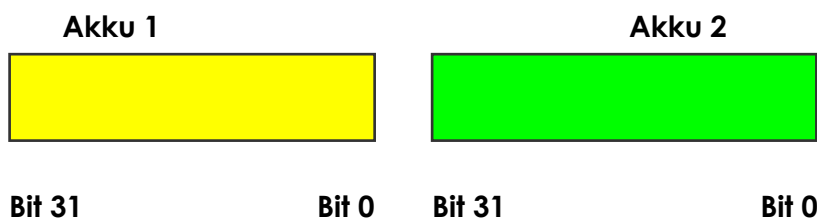
## 2.7 LADE UND TRANSFEROPERATIONEN

Lade- und Transferbefehle dienen zum Verarbeiten von Daten, die größer sind als ein Bit. Sie werden über Akkumulatoren realisiert. Lade- und Transferbefehle sind in KOP/FUP und AWL programmierbar

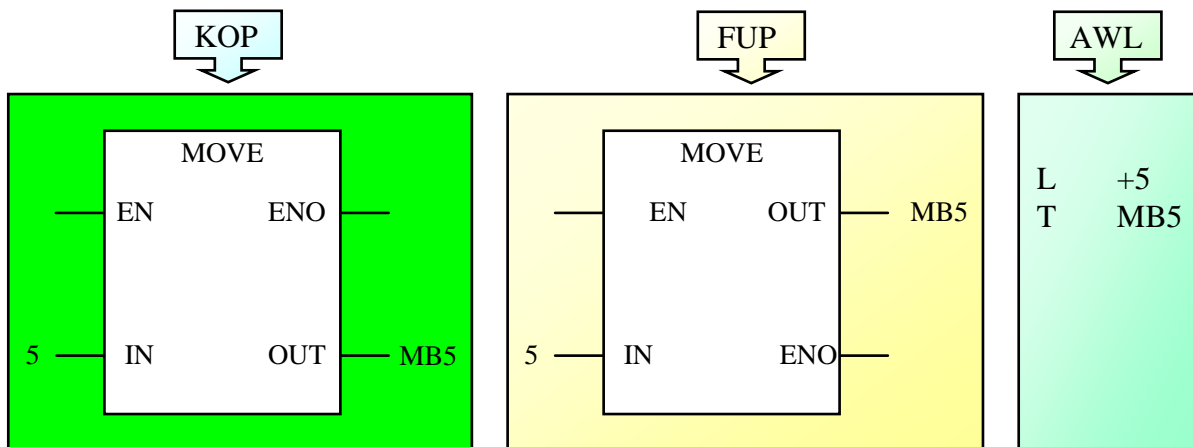
Lade- und Transferoperationen stellen wichtige Programmiermerkmale der Wortverarbeitung dar, da sie einen Informationsaustausch zwischen verschiedenen Baugruppen der Steuerung wie den Ein- und Ausgängen sowie dem Merker und Datenspeicherbereich ermöglichen.



Dieser Informationsaustausch geht jedoch nicht direkt zwischen den genannten Baugruppen vor sich, sondern immer auf dem Umweg über den Akkumulator. Dieser Akkumulator ist ein besonderes Register im Prozessor zur Verarbeitung von 32-Bit-Doppelworten. Er besteht aus einem Hauptregister (Akku 1) und einem Hilfsregister (Akku 2). Bei der Simatic Steuerungen S7-400 beinhaltet die CPU 4 Akkus. Aufbau der Akkumulatoren:



Lade- und Transferbefehle können in allen 3 Programmiersprachen programmiert werden.



L	+156	//Laden einer 16-Bit Konstante INT
L	L#126854	// Laden einer 32-Bit Konstante DINT
L	B#16#FF	// Laden eines Bytes in Hexa-Darstellung
L	2#0010 1010 0000 1111	// Laden eines 16-Bit Binärwertes
L	4.68	// Laden einer 32-Bit Konstante (REAL)

## AKKU1

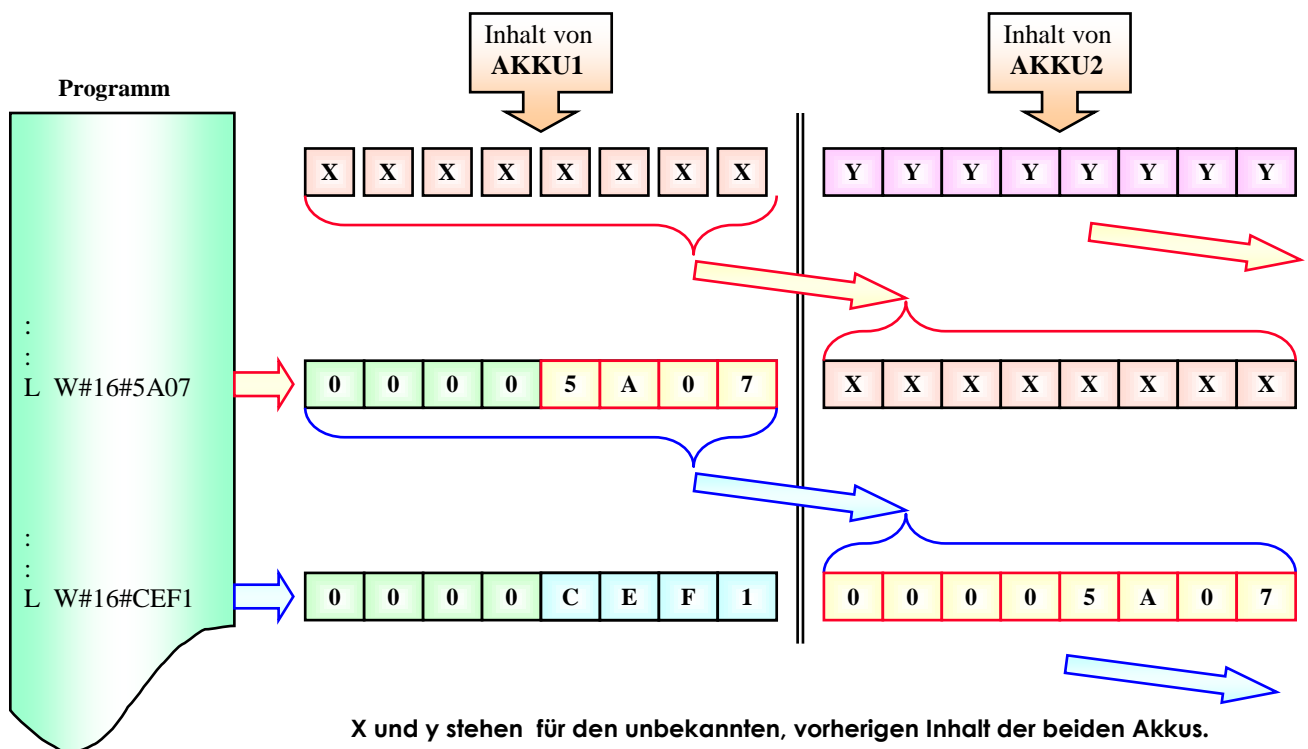
Der Akku 1 ist das zentrale Register in der CPU. Eine Ladeoperation lädt den Inhalt des adressierten Bytes, Wortes oder Doppelwortes rechtsbündig in den Akku 1. Beim Transferieren wird der zu transferierende Wert aus dem Akku1 gelesen, der Inhalt des Akku 1 bleibt erhalten. Somit kann die gleiche Information zu verschiedenen Zielen transferiert werden.

## AKKU2

Beim Ausführen eines Lade-Befehls wird zuerst der alte Inhalt von Akku 1 in den Akku 2 verschoben, der Akku 1 gelöscht (alle Bit auf 0 gesetzt) und erst dann der neue Wert in den Akku1 geschrieben. Der Akku 2 wird auch bei Vergleichen, digitalen Verknüpfungen, Rechen- und Schiebeoperationen verwendet.

### Ladevorgang:

Beim Laden eines Wortes oder einer Zahl wird diese zuerst in den Akku 1 geladen. Der Wert der im Akku 1 steht wird in den Akku 2 verschoben. Der Wert aus dem Akku 2 geht verloren (Bei der S7 400 wird dieser in den Akku 3 geschoben). Aus Akku 1 kann diese Zahl wieder weiterverarbeitet oder transferiert werden. Man bezeichnet den Akku 1 auch als „aktiven“ Akku. Wird eine zweite Zahl geladen so wird der Wert aus Akku 1 in Akku 2 verschoben und der neue Wert in Akku 1 geladen. Anschließend können z.B. beide Zahlen verglichen werden..



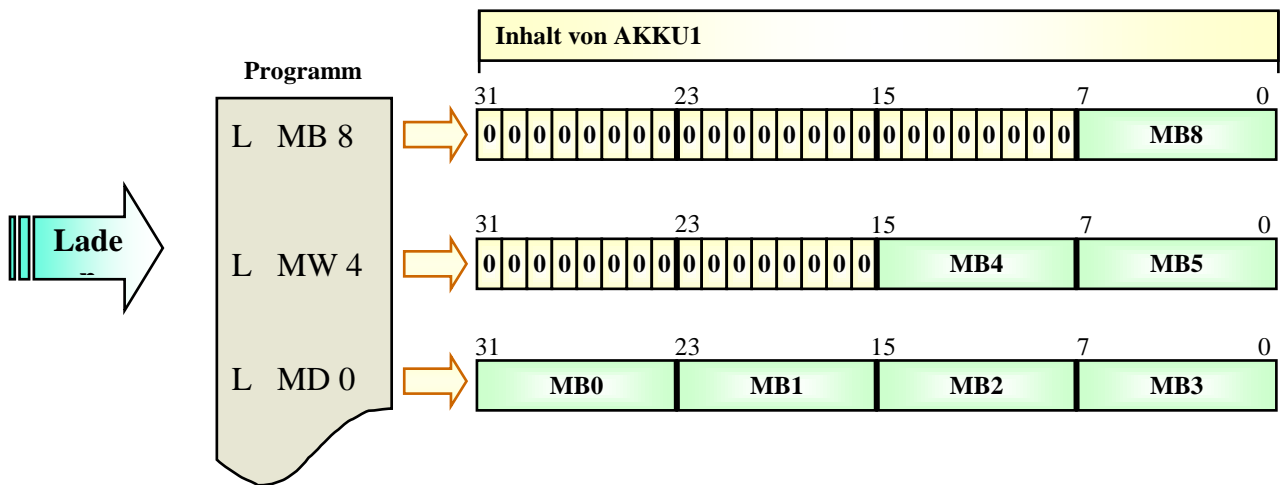
### Übersicht Ladebefehle:

Geladen werden können Zahlen im Format INT, DINT, Real, Bytes Wörter Doppelwörter als Eingänge, Merker, Ausgänge, Lokaldaten oder Datenwörter.

Beispiel:

```
L W#16#1234
L MB 100
L PEW 256
L DB10.DBW12
L AB 100
L#125000
L 32767
```

Die Akkus sind 32 Bit groß, verarbeite also 1 Doppelwort bzw. 2 Wörter oder 4 Byte.  
 Beim Laden von Zahlen werden diese immer rechtsbündig im Akku hinterlegt und fehlende Bits mit Nullen aufgefüllt.



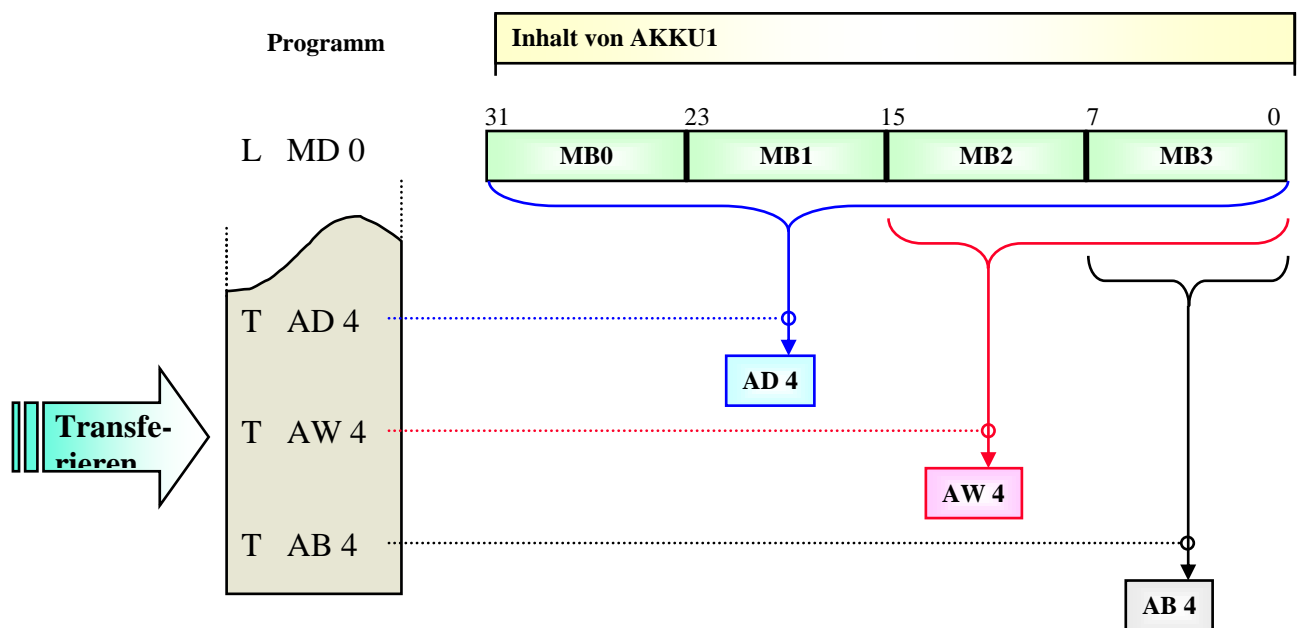


## Transfervorgang:

Transferieren heißt soviel wie Zuweisen. Mit dem Transfervorgang werden Informationen aus dem Akku 1 in die Operandenbereiche E,A,M und D kopiert

Beispiel:

```
L W#16#1234
L MB 100
L PEW 256
L DB10.DBW12
L AB 100
L#125000
I 32767
L 32767
```

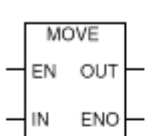


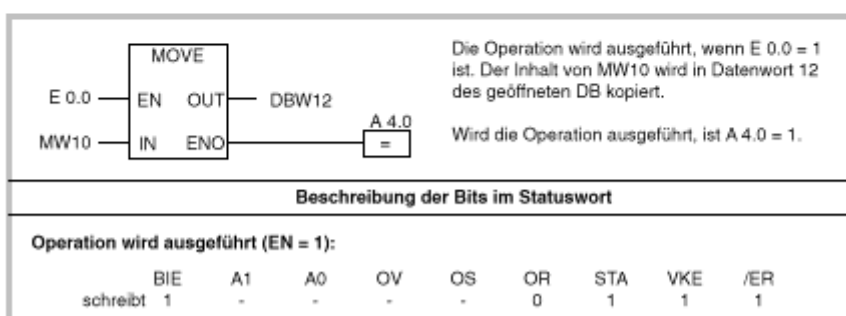
## Wert übertragen

Mit der Operation *Wert Übertragen* können Sie Variablen mit spezifischen Werten vorbelegen.

Der Wert, der an Eingang IN angegeben ist, wird in den Operanden kopiert, der an Ausgang OUT angegeben ist. ENO hat den gleichen Signalzustand wie EN.

Die Operation *Wert Übertragen* kann mit der Box MOVE alle Datentypen kopieren, die eine Länge von 8, 16 oder 32 Bits haben. Anwenderdefinierte Datentypen wie Felder oder Strukturen müssen mit der Systemfunktion SFC 20 "BLKMOV" kopiert werden .

FUP-Box	Parameter	Datentyp	Speicherbereich	Beschreibung
	EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
	IN	Alle Datentypen mit einer Länge von 8, 16 oder 32 Bit	E, A, M, D, L oder Konstante	Quellwert
	OUT	Alle Datentypen mit einer Länge von 8, 16 oder 32 Bit	E, A, M, D, L	Zieladresse
	ENO	BOOL	E, A, M, D, L	Freigabeausgang



Im KOP/FUP besteht die Möglichkeit, durch die Beschaltung des Freigabeeingangs EN der Box "MOVE" das Laden und Transferieren VKE-abhängig durchzuführen.

In AWL sind die Lade- und Transferanweisungen grundsätzlich VKE-unabhängig und werden deshalb immer ausgeführt. Mit VKE-abhängigen Sprüngen können allerdings die Lade-/Transferbefehle übersprungen werden und somit ein VKE-abhängiges Laden und Transferieren ermöglichen.

## 2.78 DAS ZÄHLWORT

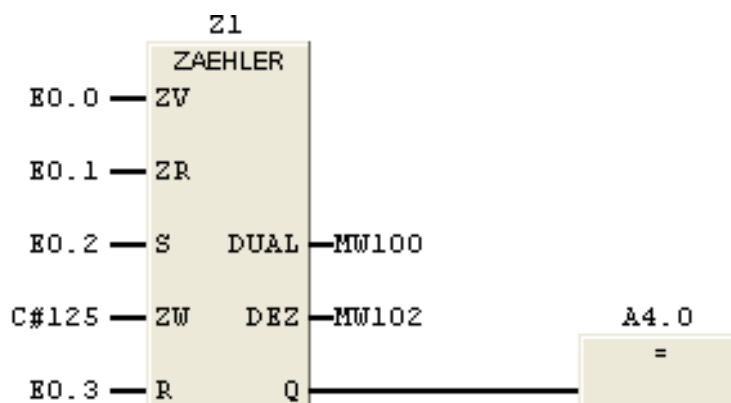
Das Zählwort ist eine 16 Bit BCD codierte Dezimalzahl. Hierzu werden 12 Bits benötigt (0 bis 999) entsprechend für jede Zahl 4 Bits. Die linken 4 Bits werden im Wort nicht benötigt.



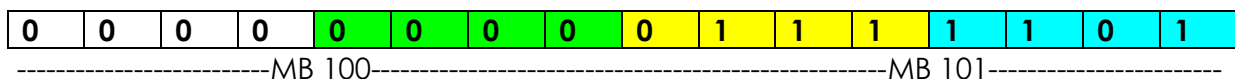
Das Zählwort kann als konstante Zahl oder als Wort (EW, DBW, MW oder AW) vorgegeben werden. Am Zählerausgang kann der Zählwert als Dualzahl (Dual) oder BCD codiert (DEZ) angezeigt und ausgewertet werden.

### Beispiel 1:

Mit dem Eingang E 0.2 soll der Zähler auf den Anfangswert 125 gesetzt werden.



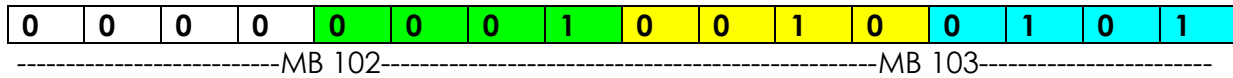
Im MW 100 steht anschließend folgendes Bitmuster



Folgende Merker haben anschließend 1-Signal

M 101.0	entsprechend 1
M 101.2	entsprechend 4
M 101.3	entsprechend 8
M101.4	entsprechend 16
M101.5	entsprechend 32
M101.6	entsprechend 64

Am Ausgang DEZ wird der Wert BCD codiert angezeigt. Im MW 102 erscheint beim Zählwert 125folgendes Bitmuster

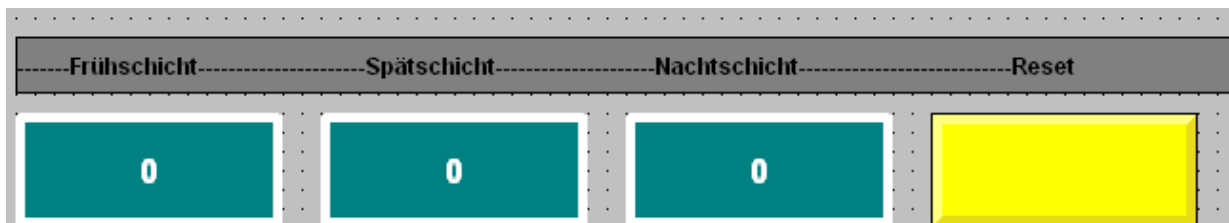


Folgende Merker haben 1-Signal

- M 103.0      entsprechend BCD-1
- M 103.2      entsprechend BCD-4
- M 101.5      entsprechend BCD 20
- M 102.0      entsprechend BCD 100

**Beispiel 2:**

Auf einem Touch Panel oder in einer Prozessvisualisierung soll auf einem Ausgabefeld der Zählwert einer produzierten Stückzahl(z.B. Frühschicht) angezeigt werden. Der Wert ist im als Dezimalzahl im Format INT hinterlegt. In diesem Anwendungsfall muss am Zählerausgang der Dualausgang programmiert werden.



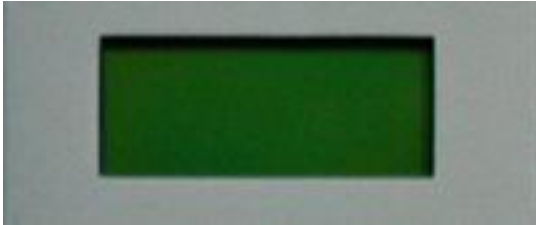
Die SPS verarbeitet in diesem Fall eine Dezimalzahl im Format INT. Im Dualformat müssen diese beiden Zahlen übereinstimmen.

Beispiel:

Es wurden 67 Teile produziert. Diese sollen angezeigt werden.

Dezimalzahl 67      Integerzahl 67      Dualzahl: 2# 0100 0011

## Siebensegmentanzeigen:



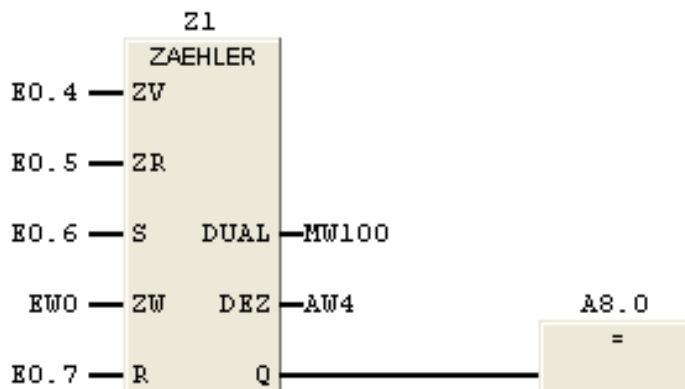
Häufig werden Zahlen mit einer Siebensegmentanzeige angezeigt. Mit einer Siebensegmentanzeige sind Dezimalzahlen 0 bis 9 und Hexadezimalzahlen A bis F darstellbar. Hierfür sind 4 Bits notwendig. Die Siebensegmentanzeige verfügt über 4 Anschlüsse, die mit 8,4,2 und 1 gekennzeichnet sind.

### Nachteil:

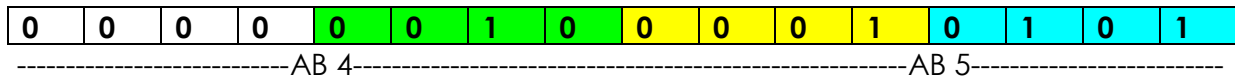
Der Nachteil von Siebensegmentanzeigen ist dass für einen Zählwert 3 eine dreiziffrige Anzeige und zusätzlich eine Ausgangsbaugruppe mindestens mit 16 DO benötigt wird. Dieser Aufwand ist heute durch preiswerte Panels zu aufwendig und zu teuer.

Zahlendarstellung	8	4	2	1
	A5.3	A5.2	A5.1	A5.0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
d	1	1	0	1
E	1	1	1	0
F	1	1	1	1

Auf einer Siebensegmentanzeige soll im AW 4 ein Zählwert 215 angezeigt werden. In diesem Anwendungsfall muss beim Zähler der BCD codierte Ausgang benutzt werden.



Folgende Ausgänge habe anschließend 1-Signal



- A 5.0      BCD    1
- A 5.2      BCD -  4
- A 5.4      BCD - 10
- A 4.1      BCD - 200

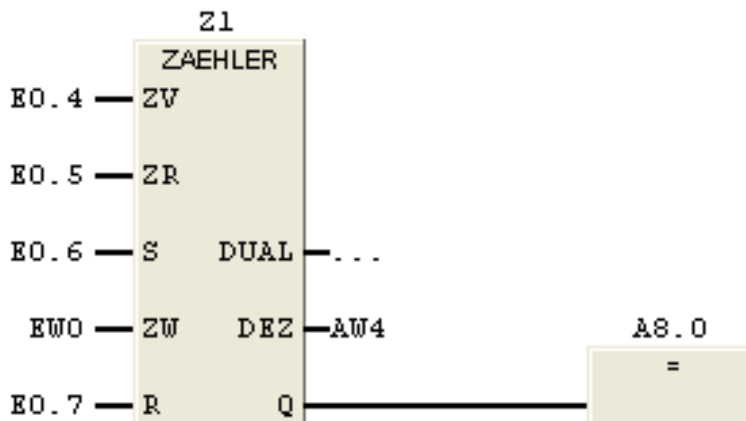
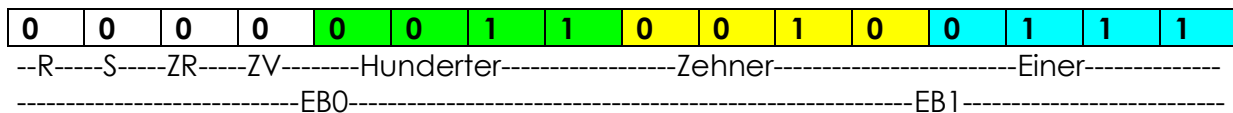
Auch kann über ein Wort dem Zähler am Setzeingang TW ein variabler Wert über ein Wort (EW, MW, DBW) vorgegeben werden. Dieser Wert ist ebenfalls BCD codiert.

**Beachte:**

In diesem Wort darf für die einzelnen BCD codierten Zahlen (Einer, Zehner, Hunderter) niemals der Wert 9 entsprechend 1001 überschritten werden.

**Beispiel:**

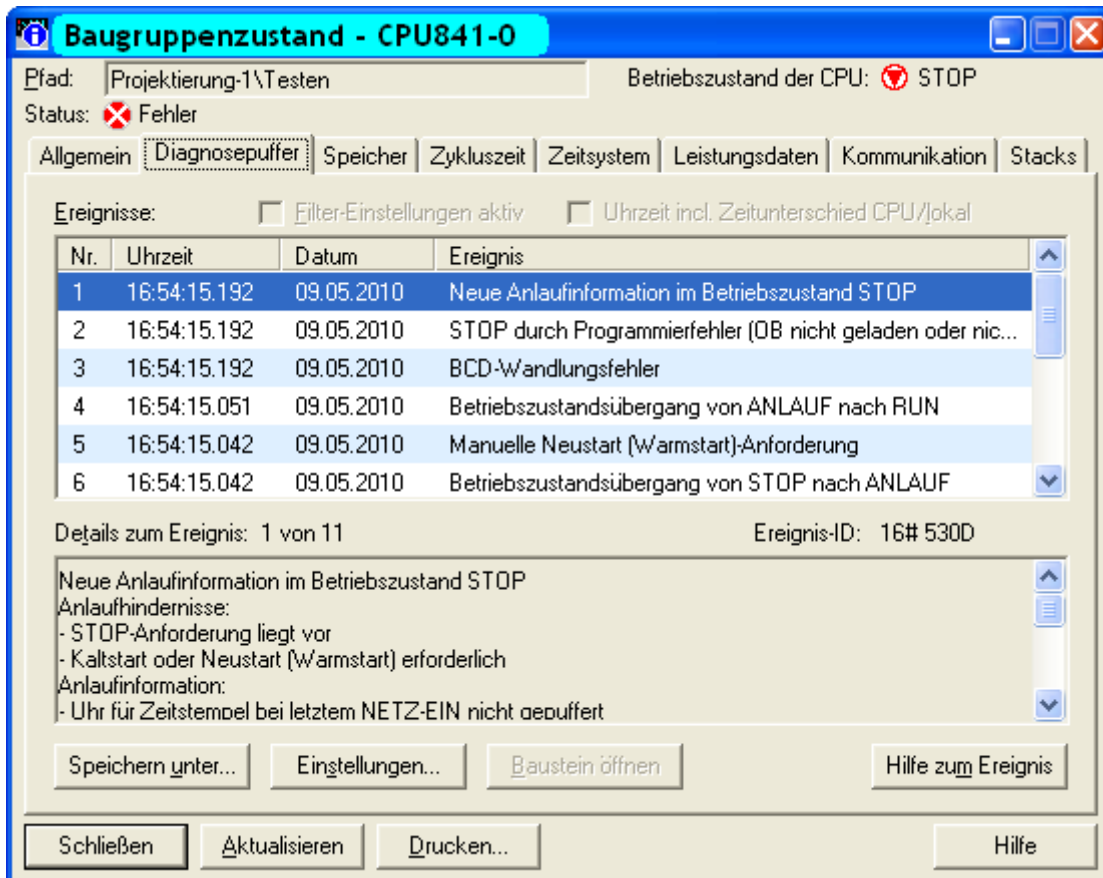
Mit dem Eingangswort soll ein variabler Zählwert vorgegeben werden. Da für den Zählwert nur  $1^2$  Bit benötigt werden können die vier linken Eingänge im EW 0 zur Steuerung des Zählers benutzt werden.. Mit den restlichen Eingängen im EW 0 soll der Zählerwert 327 eingegeben werden. Welche Eingänge müssen betätigt werden.



Vorgabe Eingänge:

- |       |          |
|-------|----------|
| E 1.0 | Wert 1   |
| E 1.1 | Wert 2   |
| E 1.2 | Wert 4   |
| E 1.5 | Wert 20  |
| E 0.0 | Wert 100 |
| E 0.1 | Wert 200 |

Wird ein falscher Wert (nicht BCD codiert z.B. 0000 1111 1111 1111 vorgegeben so entspricht dieser Wert nicht einer BCD codierten Zahl und die SPS geht in Stop und meldet im Diagnosepuffer **BCD Wandlungsfehler**.



**Baugruppenzustand - CPU841-0**

Pfad: Projektierung-1\Testen Betriebszustand der CPU: STOP

Status: Fehler

Allgemein **Diagnosepuffer** Speicher Zykluszeit Zeitsystem Leistungsdaten Kommunikation Stacks

Ereignisse:  Filter-Einstellungen aktiv  Uhrzeit incl. Zeitunterschied CPU/lokal

Nr.	Uhrzeit	Datum	Ereignis
1	16:54:15.192	09.05.2010	Neue Anlaufinformation im Betriebszustand STOP
2	16:54:15.192	09.05.2010	STOP durch Programmierfehler (OB nicht geladen oder nic...
3	16:54:15.192	09.05.2010	BCD-Wandlungsfehler
4	16:54:15.051	09.05.2010	Betriebszustandsübergang von ANLAUF nach RUN
5	16:54:15.042	09.05.2010	Manuelle Neustart (Warmstart)-Anforderung
6	16:54:15.042	09.05.2010	Betriebszustandsübergang von STOP nach ANLAUF

Details zum Ereignis: 1 von 11 Ereignis-ID: 16# 530D

Neue Anlaufinformation im Betriebszustand STOP  
Anlaufhindernisse:  
- STOP-Anforderung liegt vor  
- Kaltstart oder Neustart (Warmstart) erforderlich  
Anlaufinformation:  
- Uhr für Zeitstempel bei letztem NETZ-EIN nicht geupfert

Speichern unter... Einstellungen... Baustein öffnen Hilfe zum Ereignis

Schließen Aktualisieren Drucken... Hilfe



In älteren Anlagen wurde häufig auch der Wert über einen BCD codierten Zahleneinsteller vorgegeben. Dies entspricht nicht mehr dem heutigem Stand der Technik weil hierzu einmal der Zahleneinsteller und eine 16 kanalige Digitaleingangsbaugruppe benötigt wird. Dies übersteigt den Preis eines kleinsten Textpanels.

Ansicht: BCD Zahleneinsteller



In der heutigen Zeit werden Werte durch Touch Panels oder Operator Panels vorgegeben. Diese Werte werden für den Anwender im Format INT vorgegeben oder angezeigt. Hier werden für den Anwender die Zählerwerte im Format INT vorgegeben und angezeigt.



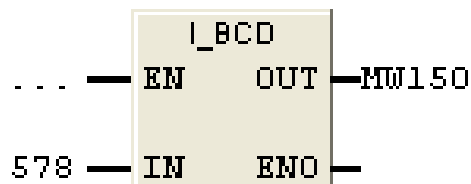
**In diesem Fall muss der vorgegebene Wert vom Panel (INT) von INT in BCD gewandelt werden.**

## WANDLER INT - BCD

Der BCD Wandler I\_BCD wandelt eine 16 Bit Dezimalzahl (Format INT) im Bereich + 999 und -999 in eine BCD Zahl um. Dieser Wandler wird benötigt wenn ein Zählerwert von einem Touch Panel vorgegeben und im SPS Programm weiter verarbeitet werden soll.

Beispiel:

Der Wert + 578 soll als Dezimalzahl in eine BCD codierte Zahl gewandelt und im MW 150 hinterlegt werden.



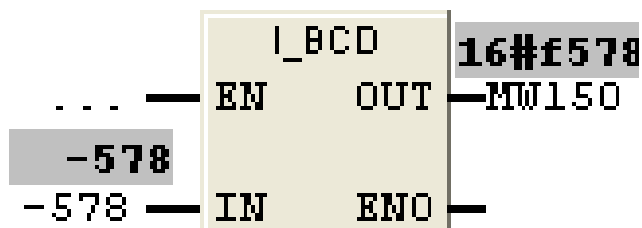
### Wert INT 578



### Wert im MW 150

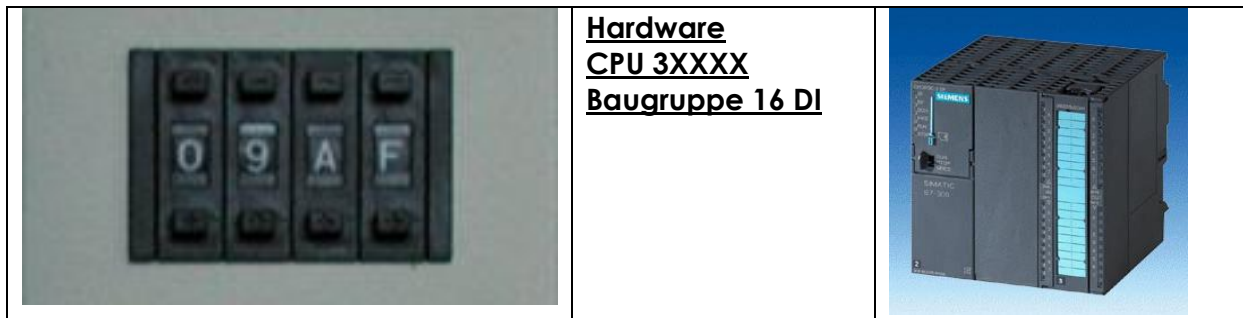


Ist der vorgegebene Wert negativ so werden die vier linken Bits im Zählerwort alle auf 1-Signal gesetzt.

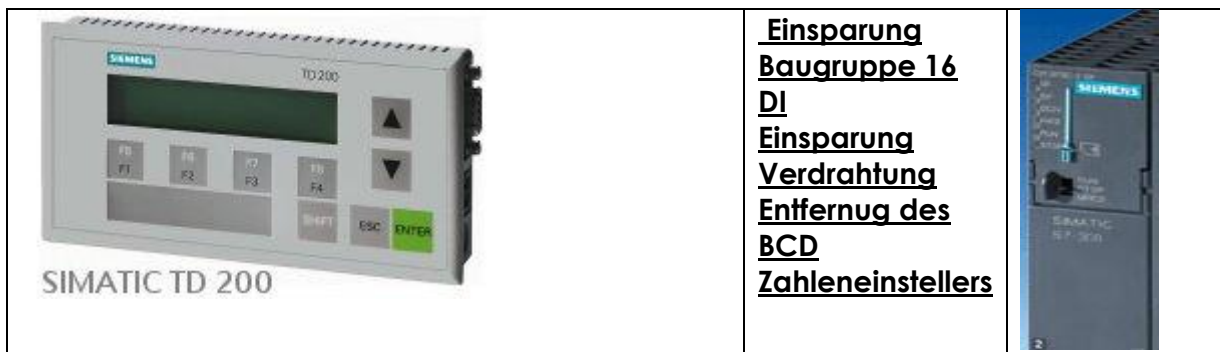


### Umsetzungsbeispiel;:

In einem Unternehmen wird ein Zahlenwert für eine Materialzählung durch einen 4 stelligen BCD Zahleneinsteller vorgegeben



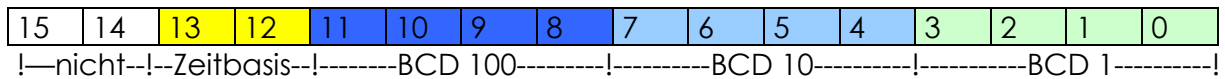
Dieser soll durch ein preiswertes TD 200 Panel über eine MPI Verbindung ersetzt werden. Die Umrüstung würde folgendermaßen aussehen.



Der Zahleneinsteller wurde bisher im EW 0 hinterlegt und soll jetzt durch das Panel ersetzt werden. Im TD 200 ist der vorgegebene Zahlenwert in der Variablen MW 100 im Formt INT hinterlegt.

## 2.9. DAS ZEITWORT

Das Zeitwort ist wie das Zählwort eine BCD codierte 16 Bit Zahl. Allerdings wird hier zusätzlich die Zeitbasis (0 bis 3) hinterlegt.



relevant 00=0,01 s  
 01=0,1 s  
 10=1 s  
 11=10 s

Wie jedes elektronische Gerät hat auch die SPS Zeit eine Toleranz. Die Zeit wann der Timer tatsächlich schaltet liegt +/- in der Zeitbasis. Der Datentyp S5time sucht sich automatisch immer die kleinste Auflösung (Zeitbasis) aus.

Dies hat den Vorteil das die Abweichung immer die kleinste ist, aber auch den Nachteil, wenn man eine kleine Zeit im Sekundenablauf anzeigen will es hier in der Darstellung auf Siebensegmentanzeigen oder Touch Panels zu Problemen kommen kann.

Wenn Sie den Datentyp S5TIME mit Zeiten verwenden, müssen Sie die Grenzwerte für den Bereich und die Auflösung beachten.

Die folgende Tabelle gibt jeweils den Bereich für die verschiedenen Auflösungen an.

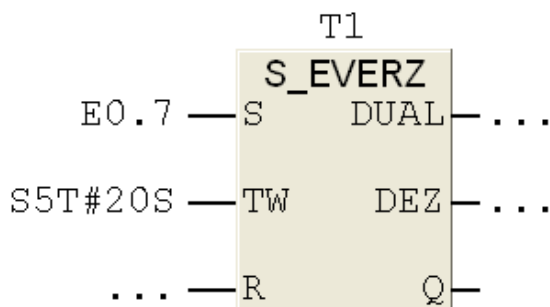
**Beachte:**

**Der Datentyp s5time sucht sich immer die kleinste Zeitbasis heraus.**

Zeibasis	kleinster Wert möglich	größter Wert möglich	kleinster Zeitwert	größter Zeitwert	Zeitbereich in dem die Zeit abläuft
0	1.0	999.0	10ms	9s990ms	10ms – 9s990ms
1	1.1	999.1	100ms	99s900ms	10s – 99s900ms
2	1.2	999.2	1s	999s 16m 239s	100s – 16m 39s
3	1.3	999.3	10s	9990s 2h46m30s	16m40s – 2h46m30s

**Umsetzungsbeispiel:**

Programmieren Sie folgenden Timer



In welchem Zeittakt läuft der Timer T2 ab?

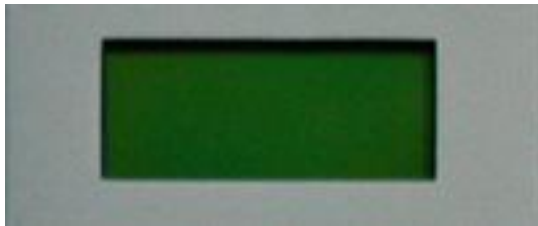
Antwort Zeittakt; \_\_\_\_\_

**Versuch 1**

Ändern Sie die Simatic Zeit folgende Zeiten und tragen Sie die Zeitbasis in die Tabelle ein.

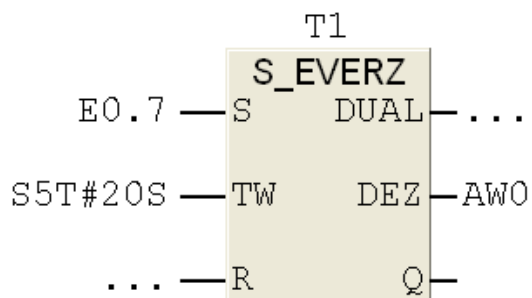
vorgegebene Simatic Zeit	ablaufende Zeitbasis
s5t#8s	
s5t#90s	
s5t#100s	
s5t#16m30s	
s5t#1h	

Die ablaufende Zeit soll im Ausgangswort 0 auf einer vierstelligen Siebensegmentanzeige angezeigt werden



Programm;

Stoppen Sie die Zeit und werten Sie diese an Ausgängen der SPS aus



Das Bitmuster der ablaufenden Zeit sieht im AW 0 folgendermaßen aus. Stoppen Sie die Zeit z. B bei 12 Sekunden und 500 ms- Das Bitmuster im AW0 sieht folgendermaßen aus.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	1

## Versuch 2

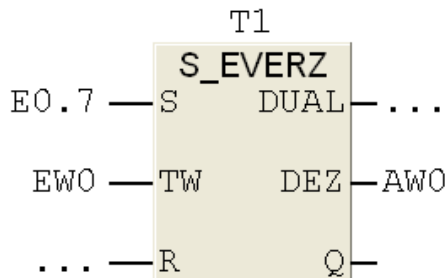
### Vorgabe des Zeitwertes unter Nichtbeachtung der Zeitbasis.

Der Zeitwert 20s soll als feste Konstante im Format 2 (Sekunden) vorgegeben werden. Dies ist möglich wenn der Zeitwert nicht als S5t#20s sondern als BCD codierte vorgegeben wird.

Mit 16 digitalen Eingängen oder einem BCD Zahleneinsteller kann ein variabler Zeitwert vorgegeben werden. **Dies ist nicht praxisgerecht, weil kein Anwender mit 16 digitalen Eingängen einen Zeitwert vorgibt, bzw. bei einem 4stelligem BCD Zahleneinsteller eine 16 digitale Eingangsbaugruppe plus Verdrahtung und Zahleneinsteller benötigt wird. Dies ist nicht mehr Stand der Technik, dient aber dem Verständnis dieses Umsetzungsbeispiels.**

Ihr SPS Schulungsgerät verfügt über 16 digitale Eingänge (z.B. EW 0).

Es soll eine variable Zeit von 20 Sekunden vorgegeben werden. Die Zeit von 20 Sekunden kann auch variabel sein.



Die Zeit kann im EW 0 in folgenden drei Varianten vorgeben werden.

#### Variante 1:

Zeitwert	200	Zeitbasis
15	14	13
12	11	10
9	8	7
6	5	4
3	2	1
0	0	0
Start	0	0
-----E0.4-----E0.1-----		

Diese Einstellung entspricht der gleichen wie s5t#20s (kleinste Auflösung)

Vorteil: Kleinste Auflösung

Nachteil: Probleme bei Anzeigen auf OP oder Siebensegmentanzeigen

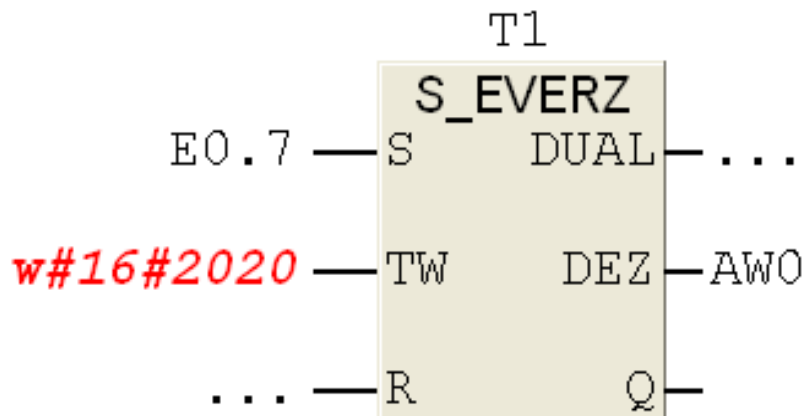




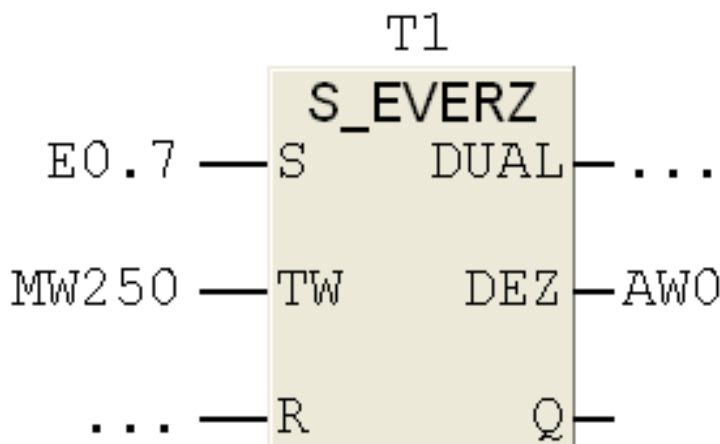
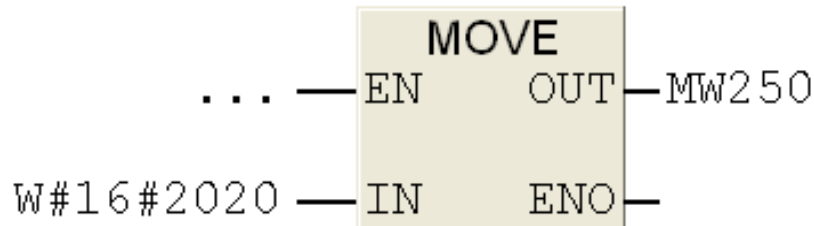
### Versuch 3

Der Kunde möchte auf einer Siebensegmentanzeige eine feste (nicht variable) Zeit von 20 Sekunden im Sekundentakt angezeigt bekommen. Die Siebensegmentanzeige soll im AW 0 angezeigt werden. In diesem Fall muss dem Timer keine Variable über ein EW sondern eine konstante Zahl im BCD Code vorgegeben werden,

Wird der Zeitwert von 20 Sekunden vorgegeben, so erscheint eine Fehlermeldung das der Datentyp w#16#2020 als Konstante nicht zum Datentyp s5t#20s Sekunden passt. Der Timer erwartet am Parameter TW ein Wort oder das Format s5time..



Aus diesem Grund muss der Wert vorher über einen MOVE Befehl neutralisiert werden. Das Ergebnis im Movebefehl kann ein MW oder besser eine temporäre Zahl sein um 16 Merker einzusparen. Im folgenden Beispiel wurde die einfachere Variante mit einem Merkerwort (250) gewählt.



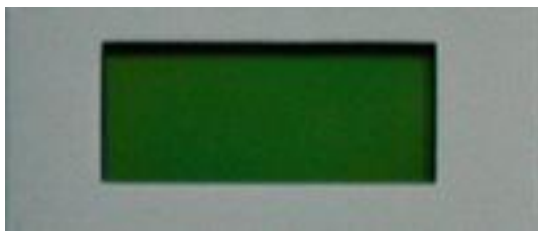
Bei Austesten des Programms stellen Sie fest dass im AW 0 der Ausgang A 0.5 1-Signal hat. Dies liegt dran, das dieser die Zeitbasis anzeigt. Auch dieser ist für die Anzeige irrelevant, und muss noch zurück gesetzt werden

**Wichtig: Dies muss nach der Zuweisung von AW 0 geschehen.**

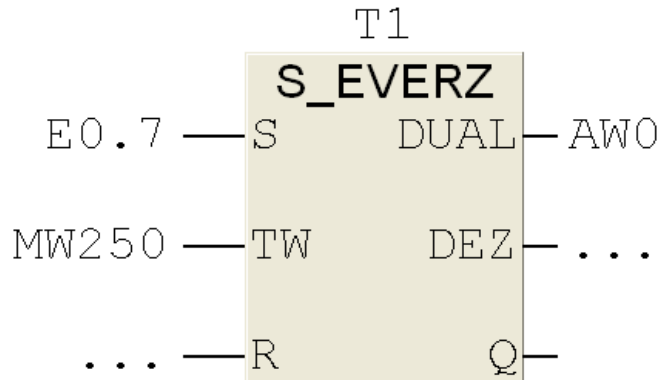


## Versuch 4

Die Zeit soll nicht auf einer Siebensegmentanzeige sondern auf einem Touch Panel angezeigt werden. Hierzu wird jetzt der **Dualausgang** benutzt.



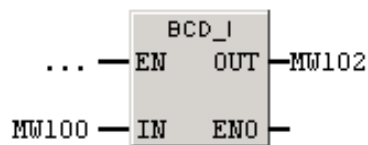
Das Programm sieht dann folgendermaßen aus.





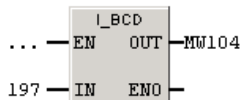
## Lernkontrollfragen

- 1. Im Merkerwort 100 steht der Zahlenwert 87.  
Welches Bitmuster finden Sie im Merkerwort 102 vor?**



- a) 0001 1010
- b) 1001 1011
- c) 1001 0011
- d) 0101 1101
- e) 0101 0111

- 2. In welchem Zahlenbereich kann das MW 104 gewandelt werden**



- a) 0 bis +9999
- b) -9999 bis + 9999
- c) -999 bis + 999
- d) 0 bis 32767
- e) 0 bis 999

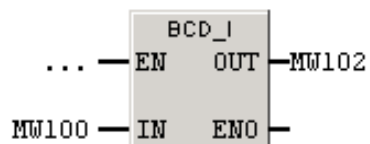
- 3. Welches Bitmuster entspricht der Zahl 8192?**

- a) 0000 1000 1111 0000
- b) 0111 1111 1111 1111
- c) 1001 1000 1111 0000
- d) 0001 1000 1234 0000
- e) 0010 0000 0000 0000

**4. Welches Bitmuster entspricht der Zahl 32767?**

- a) 1111 1000 1111 0000
- b) 0111 1111 1111 1111
- c) 1000 0000 0000 0000
- d) 0100 0000 0000 0000
- e) 0001 0000 0000 0000

**5. Im Merkerwort 100 steht der Zahlenwert 93. Welche Dezimalzahl (INT) steht im MW 102**



- a) 112
- b) 93
- c) 5234
- d) -34
- e) 147