

Beispiel-Programm: Dezimalzähler

Gesucht ist ein Programm, das im Adreßfeld S5 - S2 der Anzeige einen 4-stelligen Dezimalzähler realisiert.

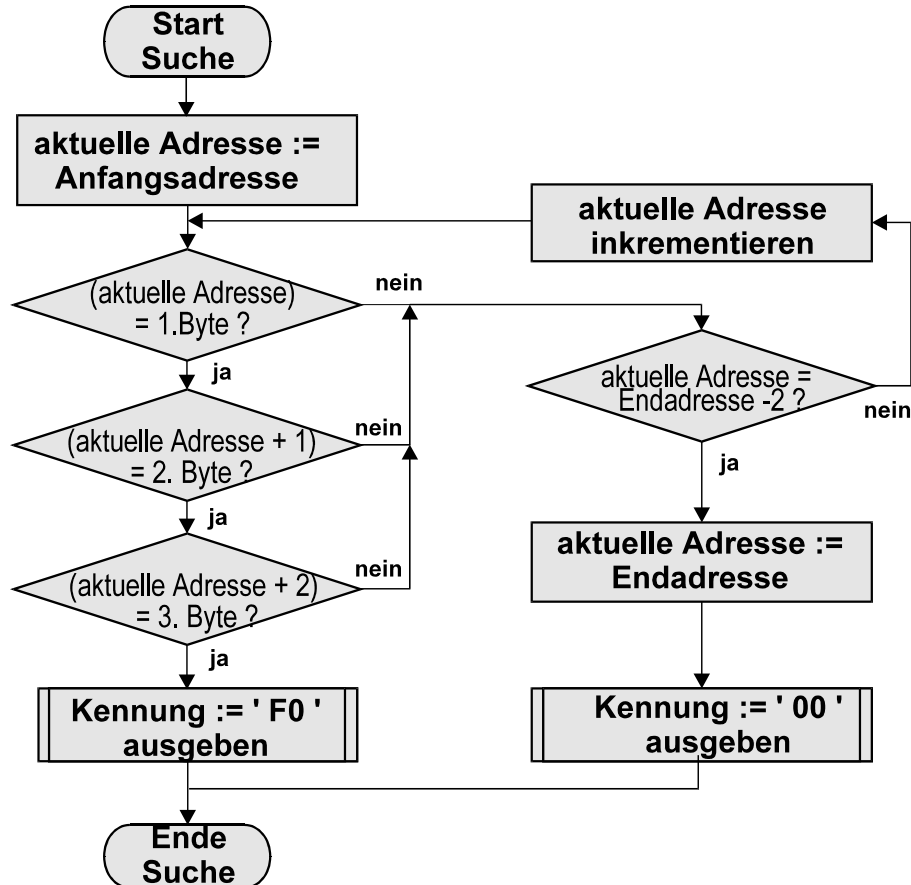
Im einzelnen soll das Programm:

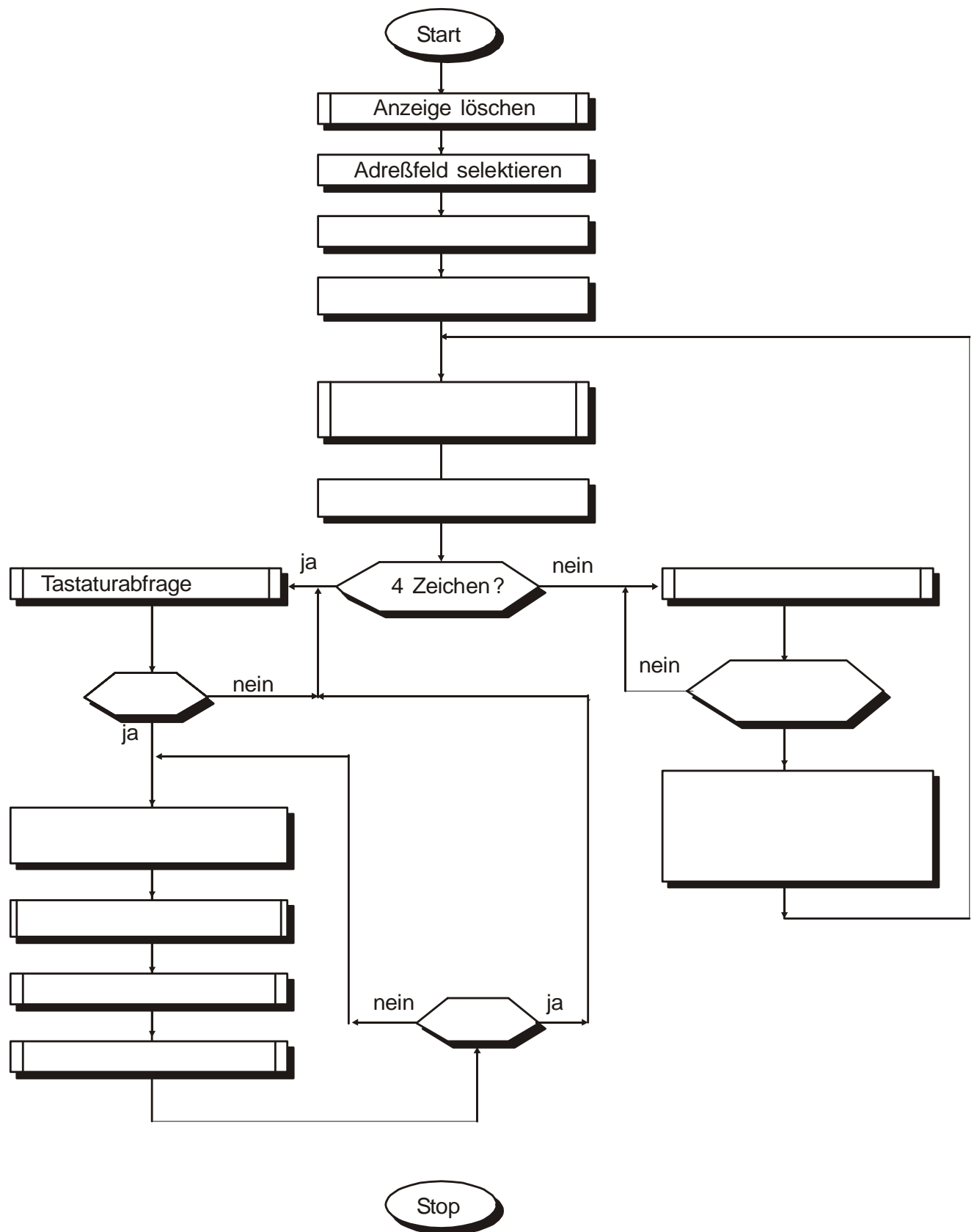
1. die Anzeige löschen,
2. über die Tastatur einen 4-stelligen Startwert einlesen, wobei nur die Ziffern 0, 1, ..., 9 akzeptiert werden sollen, und diesen Startwert im Adreßfeld anzeigen,
3. nach Betätigen der Funktionstaste '+' den Zählvorgang mit einer Zählfrequenz von (ungefähr) 1 Hz beginnen,
4. nach Betätigen der Funktionstaste '-' den Zählvorgang solange stoppen, bis er durch die Taste '+' wieder gestartet wird,
5. bei Erreichen des Endwertes '9999' mit dem Wert '0000' zyklisch weiterzählen.

Alle nicht genannten Funktionstasten sollen keine Wirkung haben. (Natürlich außer 'F4' und 'C'.)

Flußdiagramme

- * Darstellung der Ablauffolge von Programmen (nach DIN 66001)
 - * synonym: Programmablaufpläne oder Ablaufdiagramme
 - * besonders geeignet für Assembler- oder Maschinenprogramme
 - * spezieller Graph aus geometrischen Figuren und gerichteten Kanten
 - Rechtecke: allgemeine Operationen
 - Rechtecke mit Doppelrand: Unterprogramm-Aufruf
 - Parallelogramme: Ein- und Ausgaben
 - Rauten: Verzweigungen
 - Ellipsen: Beginn und Ende des Ablaufs
 - Kreise: Übergänge zwischen mehreren Diagrammen
-
- * **Beispiel:** Suche eines 3-Byte-Datums in einem Speicherbereich
 - Bereich gegeben durch (Anfangsadresse, Endadresse)
 - Ausgabe: 'F0' gefunden, '00' nicht gefunden





6809-Cross-Assembler "AS9"

1. Syntax

Gültige Zeichen:

- Buchstaben A..Z, a..z,
- Ziffern 0..9,
- arithmetische Operatoren +, -, *, /, % (Divisionsrest),
- logische Operatoren &, |, ^ (exklusiv-oder) und
- Sonderzeichen [,], \, _, \$, #, @, . (Punkt), : (Doppelpunkt), ; (Strichpunkt), ' (Hochkomma), , (Komma), Leerzeichen oder Tabulator

Gültige Zeichen für Symbole (Namen):

- Buchstaben, Ziffern, Sonderzeichen: \$, . (Punkt) und _ (Unterstrich)
- maximal 16 Zeichen lang,
- erstes Zeichen: Buchstabe, Unterstrich oder Punkt
- Unterscheidung zwischen Groß- und Kleinschreibung

Befehlszeile:

MARKE	OPERATOR	OPERAND	KOMMENTAR
-------	----------	---------	-----------

Felder durch mindestens ein Leerzeichen getrennt

"KOMMENTAR" durch Strichpunkt getrennt

Marken-Feld

- Stern (*) oder Strichpunkt als erstes Zeichen in der ersten Spalte des Marken-Feldes definiert die ganze Befehlszeile als Kommentar
- Marke muß eindeutig sein
- Marke wird Speicheradresse des ersten Bytes der gerade assemblierten Befehlszeile zugeordnet
- optional mit Doppelpunkt abschließbar

Operator-Feld

- gültiger mnemonischer Operationscode
- oder Assembler-Direktiven
- (spezielle Befehle zur Steuerung des Übersetzungsvorgangs)

Operanden-Feld

- Symbol, Ausdruck oder Symbol und Ausdruck
- Ausdruck: Operand, Speicheradresse, Symbol, Marke oder Register

Tabelle: Operandenformat

Operandenformat *)	Adressierungsart
kein Ausdruck	implizit
Ausdruck	direkt, erweitert oder relativ
#Ausdruck	unmittelbar
<Ausdruck	kennzeichnet 8-bit-Offset
>Ausdruck	kennzeichnet 16-bit-Offset
[Ausdruck]	indirekt
Ausdruck,R	indiziert
<Ausdruck,R	indiziert, erzwingt 8-bit-Offset
>Ausdruck,R	indiziert, erzwingt 16-bit-Offset
[Ausdruck,R]	indirekt-indiziert
<[Ausdruck,R]	indirekt-indiziert, erzwingt 8-bit-Offset
>[Ausdruck,R]	indirekt-indiziert, erzwingt 16-bit-Offset
,Q+	Autoinkrement um 1
,Q++	Autoinkrement um 2
[,Q++]	Autoinkrement um 2, indirekt
,-Q	Autodekrement um 1
,--Q	Autodekrement um 2
[,--Q]	Autodekrement um 2, indirekt
W ₁ , [W ₂ , ..., W _n]	unmittelbar

*) R ist eines der Register PC, S, U, X oder Y. Q ist eines der Register S, U, X oder Y und W_i (i=1 bis n) ist eines der Symbole A, B, CC, D, DP, PC, S, U, X oder Y.

Ausdrücke

- Ausdruck: Kombination von Symbolen, Konstanten, algebraischen und logischen Operatoren
- spezifiziert Wert eines Operanden oder die vom Befehl zu benutzende Adresse
- "<" zu Beginn des Operanden: Maschinencode für 8-bit-Offset
- ">" zu Beginn des Operanden: Maschinencode für 16-bit-Offset

Operatoren

- + Addition
- Subtraktion
- * Multiplikation
- / Division
- % Divisionsrest
- & bitweise UND-Verknüpfung
- | bitweise ODER-Verknüpfung
- ^ bitweise Exklusiv-Oder-Verknüpfung

- von links nach rechts berechnet
- Klammern: als syntaktische Fehler gewertet
- Arithmetische Operationen: im Zweierkomplement
- Ausdruck nur aus "": aktueller Wert des Programmzählers

Symbole

- 16-bit-Integerwert
- innerhalb eines Ausdrucks Repräsentant ihres Integerwertes
- Assembler ersetzt Symbol durch zugehörigen Integerwert

Konstanten

Unveränderliche Zahlenwerte

16-bit-Integerwert gespeichert

fünf verschiedenen Formate:

- \$ Hex
- % Binär
- @ Oktal
- ' ASCII-Zeichen
- Dezimalzahl

Tabelle: Gültige und ungültige Zahlenformate

	Gültiges Format	Ungültiges Format	Begründung
Dezimal	12	123456	Bereichsüberschreitung
	12345	12.3	ungültiges Zeichen
Hexadezimal	\$12	ABCD	kein "\$"-Zeichen
	\$ABCD	\$G2A	ungültiges Zeichen
	\$001F	\$2F018	zu viele Ziffern
Binärformat	%00101	1010101	kein "%" -Zeichen
	%1	%10001100010101011	zu viele Ziffern
	%10100	%210101	ungültiges Zeichen
Oktalformat	@17634	@2317234	zu viele Ziffern
	@377	@277272	Bereichsüberschreitung
	@177600	@23914	ungültiges Zeichen
ASCII-Zeichen	'*'	'XXY'	zu viele Zeichen

2. Erstellen und Assemblieren eines "AS9"-Quellprogramms

- jeder Texteditor möglich
- DOS-Kommandozeile

AS9 Datei1 (Datei2 . . .) (- Option1 Option2 . . .) > Liste

- Optionen, bei Bedarf:

L	Ausgabe des Programmlistings
NOL	kein Programmlisting (Voreinstellung)
CRE	Ausgabe der Cross-Reference-Tabelle
S	Ausgabe der Symboltabelle
C	Ausgabe der Anzahl der Zyklen
NOC	keine Ausgabe der Zyklenanzahl (Voreinstellung)
HEX	Ausgabedatei im Intel Hex-Format (Voreinstellung)
MOT	Ausgabedatei im Motorola S-Format

- auch direkt im Quellprogramm anzugeben: Assembler-Direktive "OPT"
- Dateinamen: vollständige Angabe mit Endung
- erzeugte Datei: Endung "*.HEX"
- Listing: Dateiname "Liste" frei wählbar
- Bewährt: Endung ".LST"

Tabelle: Das Programm "P24-1.ASM"

	ORG	\$0400	;Beginn des Programmbereiches
	JSR	CLRDISP	;Anzeige loeschen
	LDY	#\$0600	;Datenzeiger laden
	CLR	0,Y	;Datenbereich mit Null initialisieren
	CLR	1,Y	;
START:	LDX	#\$0000	;X loeschen
	JSR	HALTKEY	;Zeichen von Tastatur lesen
	CMPB	#\$86	;Vergleich, auf Ende der Eingabe (Taste "S")
	BEQ	ENDE	;
	CMPB	#\$09	;Test der Eingabe auf gueltige Ziffer (0..9)
	BHI	START	;bei ungueltiger Ziffer zurueck
	ADDB	0,Y	;Addition der Eingabe mit LSB der Summe
	TFR	B,A	;
	ANDA	#\$F0	;Test auf Ueberlauf durch vorherige Addition
	BNE	BCDADD	;wenn ja, Korrektur
	TFR	B,A	;
	CMPA	#\$09	;Test auf gueltige BCD-Ziffer
	BLS	CARRY2	;wenn nein, Korrektur
BCDADD:	ADDB	#\$06	;Korrektur, als Ergebnis gueltige BCD-Ziffer
CARRY2:	TFR	B,A	;
	ANDA	#\$F0	;
	BEQ	SHOW	;Test auf Ueberlauf vorh. Addition/Korrektur
	LDA	#\$01	;
	ADDA	1,Y	;naechsthoeherwertige Ziffer um Eins erhoehen
	STA	1,Y	;naechsthoeherwertige Ziffer abspeichern
SHOW:	ANDB	#\$0F	;eventl. Uebertrag der korrig. Ziffer loeschen
	STB	0,Y	;korrigierte Ziffer abspeichern
	JSR	SHOWT7SG	;korrigierte Ziffer in Anzeige
	LDB	1,Y	;naechsthoeherwertige Ziffer laden
	LEAX	1,X	;Anzeigestelle nach links verschieben, X:=X+1
	JSR	SHOWT7SG	;naechsthoeherwertige Ziffer in Anzeige
	BRA	START	;zurueck zur naechsten Eingabe
ENDE:	SWI1		
CLRDISP	EQU	\$F110	;Loeschen der Anzeige, In:-, Out:-
SHOWT7SG	EQU	\$F11C	;unteres Nibble von B in Anzeige, Position in X
HALTKEY	EQU	\$F143	;Lesen der Tastatur mit Warten, In:-, Out:B

Fehlermeldungen

zwei Arten:

Befehlszeile: Beschreibung des Fehlers

Befehlszeile: Warning --- Beschreibung des Fehlers.

4. Assembler-Direktiven

Befehle zur Steuerung des Assemblierungsvorgangs und der Art der Assemblierung

Es reichen:

EQU

Marke EQU Ausdruck (Kommentar)

Mittels der Direktive EQU (*EQUate Symbol to a value*) wird einer Marke der Wert von "Ausdruck" zugewiesen. Dieser Wert ist *nicht* notwendigerweise der aktuelle Wert des Programmzählers. Der Marke kann innerhalb des Programms kein neuer Wert zugewiesen werden. Sollte "Ausdruck" ein undefiniertes Symbol oder ein Symbol enthalten, dem erst später ein Wert zugewiesen wird (*forward reference*), wird ein Fehler generiert.

ORG

ORG Ausdruck (Kommentar)

Die Direktive ORG (*set program counter to ORiGin*) setzt den Programmzähler auf den Wert, der von "Ausdruck" repräsentiert wird. Die folgenden Befehlszeilen werden vom Assembler an die entsprechend nachfolgenden Speicheradressen verschoben und assembliert. Ist in einem Quellprogramm keine "ORG"-Direktive gesetzt, so wird der Programmzähler mit Null initialisiert, und das Maschinenprogramm beginnt entsprechend dem Programmzähler an der Speicheradresse \$0000. Undefinierte Ausdrücke werden vom Assembler als Fehler erkannt.

Programmlisting

Zeilennummer	Speicheradresse	Maschinencode	([#Zyklen])	Quellcode
--------------	-----------------	---------------	-------------	-----------

Symboltabelle

Symbol	Symbolwert
--------	------------

Cross-Reference-Tabelle

Die Cross-Reference-Tabelle hat das folgende Zeilenformat:

Symbol	Symbolwert	* Zeilennummer der Def.	Zeilennummer ...
--------	------------	-------------------------	------------------