

INTERNSCHRIFT Nr. 27

THEMA:

Simulation der TR 440 auf der TR 4

VERFASSTER:

Heepel

DATUM:

27.5.1969

FORM DER ABFASSUNG

ENTWURF

AUSARBEITUNG

☒ ENDFORM

SACHLICHE VERBINDLICHKEIT

ALLGEMEINE INFORMATION

DISKUSSIONSGRUNDLAGE

ERARBEITETER VORSCHLAG

☒ VERBINDLICHE MITTEILUNG

VERALTET

ÄNDERUNGSZUSTAND

BEZUG AUF BISHERIGE INTERNSCHRIFTEN

Vorkenntnisse aus:

Erweiterung von:

Ersatz für:

BEZUG AUF KÜNFTIGE INTERNSCHRIFTEN

Vorkenntnisse zu:

Erweiterung in:

Ersetzt durch:

ANDERWEITIGE LITERATUR

TR 440-Telefunken-Assembler-Sprache TAS Ausg. 1268.  
Telefunken-Mitteilung-SIMULATOR.

Aktuelles Nr. 25, TAS-TR 4

Alles Bibliothek Betriebssystemgruppe.

## Simulation der TR 440 auf der TR 4

1. Einleitung

Es besteht die Möglichkeit, mit Hilfe eines Simulators auf der TR 4 TAS-Programme zu simulieren und zwar eine TR 4-TAS-Untermenge des TAS-TR 440. (TR 440-Telefunken-Assembler-Sprache TAS Ausg. 1268, Bibliothek, BS-Gruppe.)

Es gibt keine Ein- und Ausgabe von Daten mittels Systembefehle, jedoch können Speicherbereiche mit Hilfe des Simulators ausgedruckt werden.

Die Abweichungen des TR 4-TAS, das simuliert werden kann, zum TR 440-TAS ist im Anhang beschrieben.

Vorliegende Schrift gilt nur für Programme im Normalmodus. Für TAS-Programme in anderen Modi siehe Telefunken-Mitteilung-SIMULATOR (Bibliothek, BS-Gruppe).

2. Aufgabe und Wirkungsweise des Simulators SIMYOP

Der Simulator SIMYOP hat die Aufgabe, die Hardware der TR 440 auf der TR 4 nachzubilden.

Das wichtigste Hilfsmittel für den Simulator ist der TR 4 - Plattenspeicher. Auf ihm wird der Kernspeicher der TR 440 simuliert bis zu einer Größe von 128 K echtem Kernspeicher.

Der TR 440-Kernspeicher (d.h. TR 4-Plattenspeicher) wird durch den LADER "gefüllt", d.h. durch die mit TAS erzeugten TR 440-Materialien (Monatage-Code) belegt. Der Simulator arbeitet nun Befehl für Befehl diese Materialien ab, genau wie später der reale Rechner.

3. Kontrolldrucke des Simulators

Neben der reinen Simulation ist es für den praktischen Betrieb notwendig, in den Simulator Testhilfen einzubauen, damit die simulierten Programme auch "sichtbar" gemacht werden können.

Nach jedem Befehl wird ein Parameter P\*) abgefragt, der angibt, ob zusätzlich zu den dynamischen Überwachungshilfen noch statische Angaben (Versorgungsblock) für einen Kontrolldruck ausgewertet werden sollen.

\*) siehe nächste Kapitel

- P = 2 : Der Simulator wertet einen Versorgungsblock aus. Es ist daher notwendig, daß der Versorgungsblock auch "vorhanden" ist.
- P = 0, 1 : Der Simulator verlangt keinen Versorgungsblock.

Bei P = 0 wird (außer nach dynamischen Angaben) kein Kontrolldruck ausgegeben, bei P = 1 wird jeder Befehl protokolliert, bei P = 2 werden Kontrolldrucke in Abhängigkeit vom Versorgungsblock ausgegeben.

#### 4. Aufbau eines TAS-Operators (Normalmodus)

Das erste Ganzwort (Halbzellen 0, 1) eines TAS Operators muß ein Wort mit TK = 3 enthalten. Von diesem werden die ersten 3 Oktaden als Programmname aufgefaßt.

Zu unterscheiden sind die beiden Programmnamen:

1. Der erste als Benennung auftretende Name (im Beispiel NAME)
2. Der dem Simulator übermittelte Name (1. Ganzwort), der in den Simulatorendruckungen erscheint (im Beispiel: TEST).

In Halbzelle 2 steht der Druckmodus P. Für P = 2 muß in Halbzelle 3 die Adresse des Versorgungsblocks stehen.

Beispiel:

```

----- TESTPROGRAMM TAS -----
      DRUCK 3,
      NAME = SEGM,
      KANF =      "TEST",           Halbzellen 0, 1
              '2'/H,               Halbzelle 2
              VERS/A,               Halbzelle 3
      -- NAME UND MODUS --
      !
      !
      VERS ± ....

```

./ 3

## 5. Statische Kontrolldrucke des Simulators

Für  $P = 2$  verlangt der Simulator einen Versorgungsblock.

### 5.1 Aufbau des Versorgungsblocks

Der Versorgungsblock besteht aus einer Folge von Halbworten mit Typenkennung 2, die bei geeignetem Aufbau den Simulator zum Druck einer Registerzeile oder zum Ausdruck von Speicherinhalten oder Inhalten von Indexzellen veranlassen. Die ersten beiden Bits eines Halbwortes geben dem Simulator an, was er drucken soll (Registerzeile oder Speicherinhalt), die hinteren 22 Bits werden als Adresse aufgefaßt, bei der oder bis zu der gedruckt werden soll.

Für die ersten beiden Bits eines Halbwortes gibt es vier Möglichkeiten:

00, 0L, L0 und LL .

Dementsprechend unterscheiden wir vier "Versorgungseinheiten".

5.1.1 Ein Halbwortpaar der Form ADRESSE1/A, ADRESSE2/A bewirkt den Druck einer Registerzeile, wenn

$ADRESSE1 \leq \langle F \rangle \leq ADRESSE2$  .

5.1.2 Ein Halbwortpaar der Form ADRESSE1/AN, ADRESSE2/AN bewirkt den Druck einer Registerzeile, wenn

$ADRESSE1 \leq adr \leq ADRESSE2$  .

5.1.3 Ein Halbwort der Form ADRESSE/AM bewirkt den Druck einer Registerzeile, wenn  $\langle F \rangle = ADRESSE$  .

- 5.1.4 Ein Halbwort der FORM ADRESSE/AMN leitet eine (evtl. leere) Folge von Halbwortpaaren ein, die unmarkiert oder mit N markiert sind. Nach dem letzten dieser Paare muß ein Halbwort 'C00000' stehen.

Beispiel:

```
      ORT/AMN,  
      SPEICHERANF1/A,  
      SPEICHERENDE1/A,  
      INDEXANF/AN,  
      INDEXENDE/AN,  
      SPEICHERANF2/A,  
      SPEICHERENDE2/A,  
      'C00000',
```

Wirkung:

1. Wenn  $\langle F \rangle = \text{ORT}$  ist, wird eine Registerzeile ausgegeben.
2. Dann wird der Inhalt der Speicherzellen SPEICHERANF1 bis SPEICHERENDE1, dann die Indexzellen INDEXANF bis INDEXENDE, dann die Speicherzellen SPEICHERANF2 bis SPEICHERENDE2 ausgegeben.
3. Das Halbwort 'C00000' bewirkt das Ende dieses Kontroll-drucks.

- 5.1.5 Das letzte Halbwort des gesamten Versorgungsblocks muß 'FFFFFF' sein.

- 5.2 Die unter 5.1.1 bis 5.1.5 beschriebenen "Versorgungseinheiten" dürfen in beliebiger Reihenfolge aufeinanderfolgen, müssen aber lückenlos aufeinanderfolgen. Es soll noch einmal darauf hingewiesen werden, daß als Adresse nur die letzten 22 Bits ausgewertet werden. Bei einem Fehler erscheint der Ausdruck "FEHLER IN VERSORGUNG".

5.3 Der Versorgungsblock darf höchstens 46 Halbworte lang sein, sonst erscheint Druck "VERSORGUNG ZU LANG".

## 6. Dynamische Testhilfen des Simulators

Neben oben beschriebenem Versorgungsblock, der während eines Testlaufes dynamisch nicht mehr geändert werden kann, gibt es im Simulator die Möglichkeit, mit Interncodes, die im realen Rechner Markos sind, (in der ersten Stufe als Nullbefehl interpretiert) die Kontrolldrucke zu steuern.

### 6.1 Befehle zur Änderung von P durch "Pseudobefehle"

Es besteht die Möglichkeit, P durch den Pseudobefehl (im realen Rechner Makro, der als Nullbefehl interpretiert wird) 'C1' zu ändern.

I'C1'(B) 2, : P := 2

I'C1'(B) 1, : P := 1

I'C1'(B) 0, : P := 0

Bemerkung: Der Befehl I'C1'(B) 2 ist im Normalmodus nur sinnvoll, wenn von vornherein P = 2 war, d.h. überhaupt ein Versorgungsblock da ist.

6.2 'C1' adr. Dieser Befehl druckt so viele dynamisch ablaufende Befehle, als der Adreßteil adr angibt, wobei FFFF=adr als Befehlsanzahl 0 aufgefaßt wird. Die Zählung der zu druckenden Befehle wird dabei durch eine Modusumschaltung nicht unterbrochen.

adr > 3

Während dieser Befehl wirksam ist, wird der Versorgungsblock nicht ausgewertet.

### 6.3 'C6' adr

adr weist auf eine Folge von Halbwortpaaren mit TK2, bei denen die ersten beiden Bits 00 oder 0L sein müssen. Sind im Halbwortpaar in beiden Halbworten die ersten beiden Bits 00, wird der Inhalt von Speicherzellen ausgedruckt, beginnend mit der im 1. Halbwort angegebenen Zelle bis zu der im 2. Halbwort angegebenen. Sind im Halbwortpaar die ersten beiden Bits 0L, wird der Inhalt von Indexzellen ausgedruckt, beginnend mit der im 1. Halbwort stehenden bis zu den im 2. Halbwort stehenden.

Es dürfen beliebig viele Halbwortpaare aufeinanderfolgen. Nach dem letzten Halbwortpaar muß ein Halbwort mit den ersten beiden Bits LL stehen oder eine Zelle mit TK+2.

Bemerkung: Dieser Befehl 'C6' bewirkt im wesentlichen dasselbe wie unter 5.1.4 beschriebenen, wobei natürlich das dort notwendige erste Halbwort mit den ersten Bits LL entfällt.

### 6.4 'CO' adr und 'CD' adr

Diese Befehle erlauben den Ausdruck des bei adr beginnenden  $\alpha$ -Text über den Schnelldrucker. Der zu druckende Text muß folgende Bedingungen erfüllen:

Er darf nur 120 Oktaden (20 Worte) lang sein,  
er darf das Zeichen  $\%$  nicht enthalten,  
er darf nur Zeichen enthalten, die auf dem  
TR4-Drucker ausdrückbar sind.

'CO' : vor Ausdruck des  $\alpha$ -Textes wird der Befehl  
'CO' selbst mit einer Registerzeile ausgedruckt, nach dem  $\alpha$ -Text erscheint eine Leerzeile.

'CD' : Registerzeile und Leerzeile entfallen.

Abbruch der Druckausgabe bei TK+3.

### 6.5 'C2' adr

adr besteht aus 2 Tetraden x und 2 Tetraden y  
I'C2' xy

x=1 bedeutet: Einschalten der Sprungbefehle  
x=0 bedeutet: Ausschalten der Sprungbefehle  
y gibt jenen Modus an, in dem der Druck für die Sprung-  
befehle ein- oder ausgeschaltet wird.  
y besteht aus den Bits  $2^0$ ,  $2^1$ ,  $2^2$   
 $2^0$  für Normalmodus  
 $2^1$  für Abwicklermodus  
 $2^2$  für System und Spezialmodus

Es ist damit möglich, von einem beliebigen Modus aus, die Protokollierung der Sprungbefehle für denselben Modus und auch für andere (in welchen das Programm sich später evtl. befindet) entweder ein- oder auszuschalten.

### 6.6 'C3' adr

adr enthält 2 Befehlscodes. Sobald dieser Befehl ausgeführt wird, setzt die Protokollierung aller Befehle mit den angegebenen Codes ein.

Beispiel:

I'C3'(B) '7080',

ab diesem Befehl werden alle B- und C-Befehle protokolliert.

Tritt ein weiterer 'C3'-Befehl auf, so wird dessen Adreßteil aktuell. Enthält dieser Adreßteil jedoch den Code '00' (Nullbefehl), so wird weiterhin jener Befehlscode protokolliert, der auf dieser Position (links oder rechts in adr) im vorangehenden 'C3'-Befehl stand.



Beispiel:

protokolliert werden

I'C3'(B)	'7080',	B , C
.		
.		
.		
I'C3'(B)	'009B'	B , SH
.		
.		
.		
I'C3'(B)	'3800'	SU , SH
.		
.		
.		
I'C3'(B)	'427C' ,	A , AZ

Somit können maximal 2 Befehlscodes angegeben werden  
(Nullbefehl wird nicht protokolliert).

#### 6.7 'FB' adr

Zahl der (noch) zu protokollierenden Seiten := adr. Damit ist es möglich, die ursprünglich eingestellte Seitenschranke von 50 Seiten zu ändern.

#### 6.8 Der Endebefehl '50'

- W1. Der beim "Erststart" des gerade bearbeiteten Materials vorgegebene Wert von P wird wieder eingestellt.
- W2. Nach eventueller Protokollierung des Befehls (von P und Versorgungsblock abhängig), wird die Simulation abgebrochen. Es folgt der Ausdruck

"SIMULATOR ENDE , ZEIT ... " .

Die angegebene Zeit ist die TR4-Zeit vom Beginn der Simulation des letzten Befehls bis nach Ausführung des Endebefehls.

## 7. Fehlermeldungen

7.1 Bei Auftreten eines SSR, MAKRO, ALARM oder EINGRIFF wird als Testhilfe eine Zeile mit der Ablage der 4 Ganzworte ausgedruckt, außerdem davor eine Registerzeile. Bei Eingriffen und den wichtigsten Alarmen erscheint noch vor der Registerzeile eine Zeile mit einer genauen Spezifikation des Ereignisses. (Z.B. von welchem Kanal der Eingriff kommt oder welcher Alarm eingetreten ist). Diese Ausdrücke verstehen sich von selbst und können deshalb hier übergangen werden.

7.2 Wird dem Simulator eine Kacheladresse größer als 128 K angeboten, wird nach dem Ausdruck

"KACHELADRESSE ZU GRØSS , ABS ... REL ... "

die Simulierung abgebrochen.

7.3 Neben diesen Kontrollldrucken gibt es Meldungen, die mit dem Aufbau des Simulators als TR4-Operator zusammenhängen.

Es sind dies:

"SIMULATOR, KEINE PLATTE FREI" oder

"SIMULATOR, FEHLER VØN EINSØP"

Diese Meldungen kommen, wenn der LADER oder der EINSØP entweder keine Platte zur Verfügung hat oder ein Ereignis an den SIM melden, das einen Simulatorlauf nicht sinnvoll erscheinen läßt.

"SIMULATOR, KASIM NICHT LADBAR"

Wenn aufgrund der TR4\_Speicherverteilung ein Zuladen des KASIM den freien Speicher unter die zur TR440-Simulation nötige Mindestgröße reduziert.

"PLUP ALARM... VADR... HR... BLØCKNR...  
TR4ADR... GRUPPE..."

Das den Verkehr TR4 - TR4-Plattenspeicher regelnde Unterprogramm PLUP meldet einen technischen Fehler. Dieser kann rein technischer Art sein, aber auch ein Folgefehler eines Programmierfehlers (z.B. eines Transportes mit zu hoher TR440 Adresse). Die nach der Alarmart folgenden Angaben können evtl. eine Fehlererkennung ermöglichen.

"SIMULATOR ALARM ... ADRESSE ..."

Diese Meldung weist auf einen Alarm, den das Betriebssystem gemeldet hat (z.B. Gerätefehler).

Bei Alarm 0,1 oder 2 ist der Alarm meist im Simulator aufgetreten, etwa als Folge einer nicht genügend abgesicherten TR440 Fehlprogrammierung oder als ein echter Simulatorfehler.

Wenn diese Fehlermeldung auftritt, bitte mit Protokoll zu Herrn Heupel.

Bei allen diesen, unter 7.3., beschriebenen Fehlern wird die Simulierung abgebrochen.

## 8. Das Mini-Betriebssystem

Der Ablauf des TAS-Operators wird von einem Minibetriebssystem gesteuert, das seinerseits vom Simulator abgearbeitet wird. Im Betriebssystem sind 2 Systembefehle SSR 0 20 und SSR 4 8 implementiert, die nun im TAS-Quellprogramm aufgerufen werden können.

### 8.1 Anmelden einer Alarmadresse

Bei SSR 0 20 wird über den unmittelbar dahinter stehenden Befehl zurückgekehrt (wie später im Abwickler). Im Alarmfall wird auf die im Versorgungsblock angegebene Adresse gesprungen, wobei der 4er Block des Leitblocks vom Halbwort 32 bis einschließlich 38 im Register <A, Q, D, H> abgelegt wird.

Die Adresse des Versorgungsblocks (gerade Adresse) muß vor dem SSR-Befehl in das BB-Register gebracht werden, wobei die Alarmadresse im rechten Halbwort des Versorgungsblocks steht, während das linke Halbwort Null ist.

Beispiel: VB = O/H, ADR/A---SSR-VERSØRGUNGSBLØCK---  
 XBA VB,  
 SSR 0 20,  
 B ('1'/1),  
 SH AL 47,  
 C ANTØN---Arithmetischer-Alarm---  
 N 1,  
 ADR = I'5C (B) 0 ---Alarm-Adresse---

## 8.2 Abspeichern einer Alarmablage

Zur Alarmbehandlung wurde der SSR 4 8 eingebaut. Der SSR 4 8 speichert bei Alarmen in einem Kernspeicherbereich von 14 Ganzworten mit einem QCR die Register ab und trägt außerdem den Alarm-Viererblock in diesen Kernspeicherbereich ein. Die Rechenwerksregister ändern sich nicht.

Man meldet vor dem Setzen eines SSR 4 8 eine Alarmadresse mit SSR 0 20 an.

Im Alarmfalle wird dann diese Alarmadresse gesprungen. Wenn dann im RA das Bit 2<sup>0</sup> als Alarmart steht, gibt man einen SSR 4 8.

TAS-Beispiel:

.....  
 VB= O/H , ADR/A ,  
 XBA VB,  
 SSR 0 20 ---ALARMADRESSE ANMELDEN---  
 BU ('700000 000000'/0),  
 SH AL 1 ,  
 RT AH ---ARITHMETISCHER ALARM---  
 .....

w:

0	0
EDRex	0

ADR = SRN 4R,  
 VA = O/G , ADR48/AG ,  
 XBA VA--VERSORGBLOCK--  
 SSR 4 8,  
 I'C6'(B) (ADR48/A,  
 ADR48 + 38/A,  
 O/AMN),  
 .....

ADR48:

BB	BU
RH	
RQ	
RD	
RU	
BT	O
Ind.Bis	Q
RS	
BB	BR
BF	Steuer- bit
BC "	BH

## 9. Das TAS-Protokoll

Der Assembler liefert auf Wunsch (siehe Steuerkarte) ein Protokoll, in dem für jede IE eine Zeile gebildet wird, die die laufende Nummer der IE, die Quelle und den Interncode (bei Halbwort-IE mit Lade-Anweisungen) aufnimmt.

Bei Beginn einer neuen Seite erscheint eine Kopfzeile mit dem Programm- und Segmentnamen sowie der zuletzt aufgetretenen Überschrift. Jede neue Überschrift verursacht den Beginn einer neuen Seite (der SEGM-Befehl dagegen nicht).

Die Protokollierung kann durch den Pseudobefehl DRUCK (siehe 2.9.1, TR440-TAS-Beschreibung, Bibliothek Betriebssystemgruppe) teilweise unterbunden werden.

Hat der Assembler einen Fehler erkannt, so gibt er unter der betreffenden IE einen spezifischen Fehlertext aus. Fehleranzeigen erscheinen auch dann, wenn das Protokoll nicht verlangt wird.

Von der ursprünglichen Quelle abweichende IE werden durch das Zeichen § gekennzeichnet (hinter der laufenden Nummer). Dabei handelt es sich um IE, die durch Ersetzungsaufrufe entstehen oder um eingeschobene Halbworte, die dadurch nötig werden, daß für die folgende IE eine gerade oder ungerade Ablageadresse gefordert wird (oder sogar ein Teilseitenanfang).

Zum Interncode von Halbworten:

Die linke Hälfte enthält Information für den Lader und hat die Struktur (in Tetraden):

O TK AE TE O S

AE ist die Ablageebene des IE-Halbworts

(0, 1, 2 : Ebenen des K-Bereichs ,

3, 4, 5 : " " V-Bereichs ,

6, 7, 8 : " " B-Bereichs ,

9, A, B : " " D-Bereichs ,

C : Ebene der Einwortliterale (im K-Bereich) ).

Der Lader ermittelt die Ablageadresse des Halbworts als Summe der Ebenenanfangsadresse und der angegebenen ebenenrelativen Adresse.

Ist die Ablageadresse gerade, so erhält das betreffende Ganzwort die Typenkennung TK.

Der MC-Schlüssel S:

S = 0 (bei HW-Konstanten und Befehlen mit Absolutadresse):  
das IE-Halbwort bleibt unverändert.

S = 1 (bei Befehlen mit ebenenrelativem Adressteil):  
der 16-Bit-Adressteil des IE-Halbworts wird um die großseitenrelative Anfangsadresse der Translations-ebene TE erhöht. (Die Ebene TE enthält die bezogene Information).

S = 2 (bei Adresskonstanten):  
auf das IE-HW wird die Anfangsadresse der Translationsebene TE (auf die sich die Adresskonstante bezieht) addiert.  
(24-Bit-Translation!)

S = 6 (bei ASP und DSP):  
Das rechte HW enthält die Freihaltezahl, um die die aktuelle Adresse in der Ablageebene AE erhöht wird.  
Auch das rechte HW verschwindet beim Laden.

## 10. Der Steuerkartensatz

Die zum Testlauf eines TAS-Quellprogramm benötigten Operatoren befinden sich im Systemdepot unter folgenden Kennzeichen:

K	Q	
2	80	TAS-Assembler
2	505	Lader (lädt auf TR4-Platte)
2	551	Mini-Betriebssystem
2	502	Simulator

### 10.1 Steuerkarte für TAS-Assembler

7 S 2, 80, d1, d2, m, K, Q;

Die 5 Parameter sind dezimal anzugeben und haben folgende Bedeutung:

#### 1. Steuerung des Protokolldrucks durch d1 und d2

d1 = 0 Kein Protokoll (jedoch Angabe fehlerhafter IE)

1 Einzeiliger Druck, schmales Format (69 Zeichen pro Zeile)

2 Einzeilig, breites Format (120 Zeichen)

3 Zweizeilig (jeweils eine Leerzeile), schmal

4 Zweizeilig, breit

d2 = 0 Kommentare und Interncode werden nicht protokolliert

1 Mit Kommentaren

2 Mit Interncode

3 Mit Kommentaren und Interncode

#### 2. TAS-Modus m

m = 7 Adressbuch für 2000 Namen

m = 8 Adressbuch für 4000 Namen, wobei Namen mit mehr als 6 Zeichen nur in beschränktem Umfang aufgenommen werden.

3. K, Q-Kennzeichen für das zu schaffende Montagecode-Objekt.

#### 10.2 Beispiel für einen normalen Steuerkartensatz

7S2,5,"..... rote Abschnittskarte  
7S2,80,2,3,8,K,Q; TAS-Assembler

TAS-Quellprogramm im Normalmodus

7S2,505,0,2,551,2,"OG"; Lader für Mini-BS  
7S2,505,3,K,Q,3,"EEEEEEEEEEEEEG"; Lader für Montagecode  
7S2,502,2,16; Simulator

Der Zeitbedarf für den ganzen Lauf bei kleinen TAS-Programmen beträgt etwa 90 sec.

#### Anhang:

##### Abweichungen des TR4-TAS vom TR440-TAS

Das TR4-TAS ist im wesentlichen eine Untermenge des TR440-TAS. Die folgenden Nummern beziehen sich auf die Nummern in der TR440-TAS-Beschreibung (Bibliothek-Betriebssystemgruppe). Neben diesen Nummern sind die Abweichungen im TR4-TAS aufgeführt.

1.4.8 entfällt

1.5.1 Jede Informationseinheit darf nur 360 Zeichen umfassen (einschl. Leertasten und Abgrenzungsteil, ausschl. führende Z und Z nach = einer Benennung)

1.6.9 entfällt

1.7.3 Die Ganzadresse ist im TR4-TAS anders definiert, doch sind "nicht zu kompliziert aufgebaute" Ganzadressen des



TR4-TAS erlaubte Ganzadressen des TR440-TAS. Näheres siehe Aktuelles Nr. 25, TAS-TR4 Seite 22, 23 (Bibliothek Betriebssystemgruppe).

- 1.7.6 Hier gilt das Analoge wie unter 1.7.3
- 1.8 Pseudobefehle dürfen nicht in der Informationseinheit vorkommen. Kommentare sind in oder nach Literalen nicht zugelassen.
- 1.9 Vor INDEX darf kein Benennungsteil stehen.
- 1.10 Es gibt keine Textkonstanten und Bitfeldkonstanten.
- 1.10.5 Spezifikation F,G,U sind nicht erlaubt.
- 1.10.6 entfällt
- 1.10.8 entfällt
- 2.2.1 Erstes Segment muß durch SEGM-Befehl eingeleitet werden.
- 2.4 entfällt
- 2.5.2 In der Definition des STARR-Befehl ersetze / durch Z
- 2.5.4 - 2.5.9 entfällt
- 2.6 entfällt
- 2.8 entfällt
- 2.9.1 Druckparameter 2 ist irrelevant, da es keine Makros gibt.
- 2.10 entfällt
- 3 entfällt