

Unfortunately, only remnants of Section 2 from the Service Manual of TA1000 could be found, until now - but the information present in the [Service Manual of TA1069](#) will be applicable also for the TA1000 in many cases.

Section 0

Section 1

Section 2

Section 3

Section 4

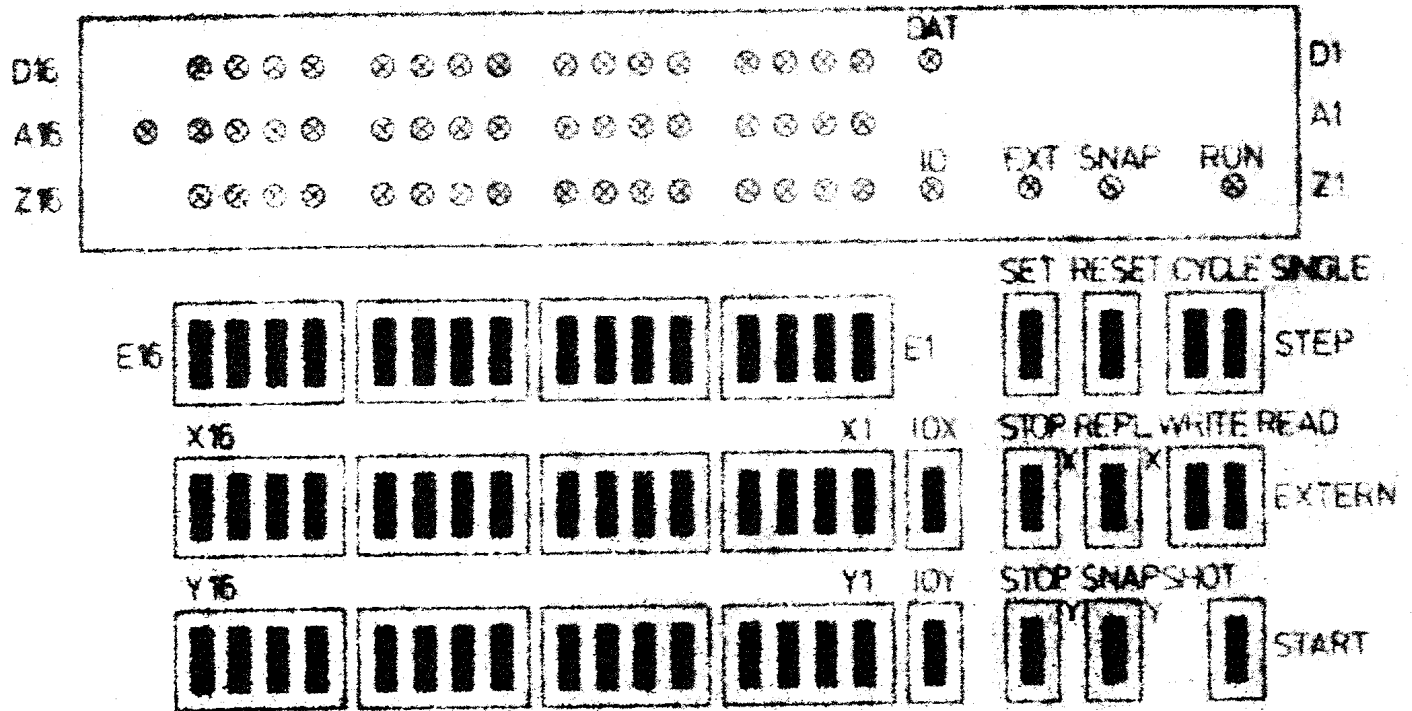
Section 5

Section 6

Section 7

Section 8

Section 9



Bedeutung der Leuchtanzeige:

D1 bis D16	: E-Bus	RUN	: Rechner "LAUFT"
A1 bis A16	: Akkuinhalt	EXT	: Read bzw. Write-Schalter betätigt
Z1 bis Z16	: Adress-Register	SNAP	: Snap Shot wurde ausgeführt
C	: Übertrag Akku	I/O	: Z1 bis Z16 entspricht I/O-Adresse
		DAT	: D1 bis D16 = Datenwort

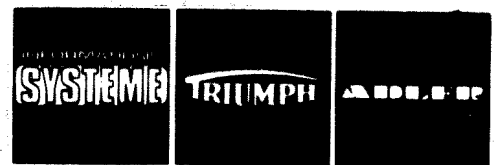
Bedeutung der Schalter:

E1 bis E16	: Schalter für externe Eingaben
X1 bis X16	: Schalter für Adresse X
Y1 bis Y16	: Schalter für Adresse Y
Stop X	: Rechner stoppt wenn Adresse = X-Schalter-Adresse ist keine ungeraden Adressen möglich
Stop Y	: Rechner stoppt wenn Adresse = Y-Schalter-Adresse ist keine ungeraden Adressen möglich
Replace	: Beim Erreichen der X-Adresse wird Datenwort durch E1 bis E16 ersetzt
SET	: E1 bis E16 ist immer Befehlswort
RESET	: Rechner auf Mikroprogrammumfang setzen
STEP SINGLE	: Rechner Stop beim folgenden Mikrobefehl
STEP CYCLE	: Rechner Stop beim folgenden 2 Byte Zyklus
EXTERN READ	: Lesen der Information aus der Speicherzelle der eingestellten X-Adresse
EXTERN WRITE	: Schreiben der Information E1 bis E16 in die Speicherzelle der eingestellten X-Adresse
SNAP SHOT Y	: Schnappschuss von D1 bis D16, A1 bis A16, u. Z1 bis Z16 bei eingestellter Y-Adresse
START	: Programm-Fortsetzung, Start-teste

TA 1000 Mikroprogramm

Bedienungsanleitung

Mikrotesttableau

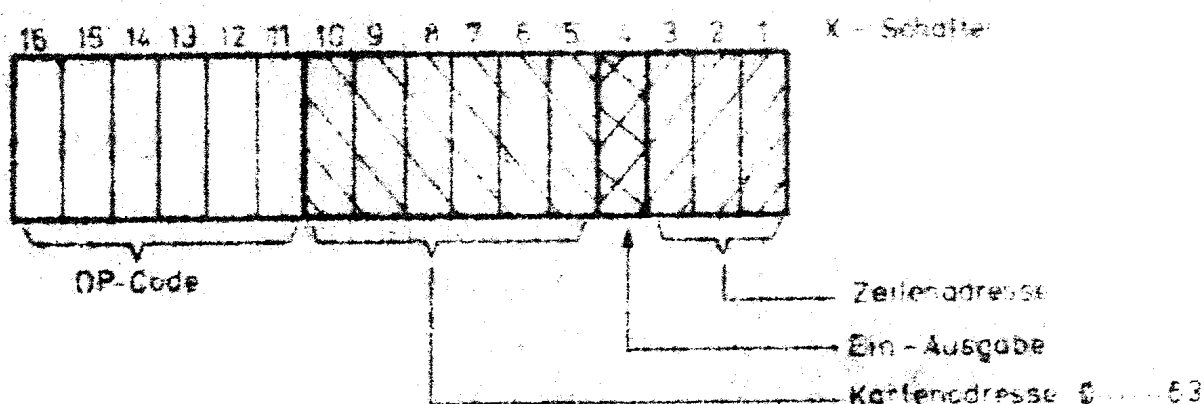


Einzustellende Schalter

Gewünschte Funktion	Adressen	Funktionen	Daten	Bemerkungen
Stop bei X-Adresse	X	Stop X	-	Stop bei Erreichen der X-Adresse
Stop bei Y-Adresse	Y	Stop Y	-	Stop bei Erreichen der Y-Adresse
Befehlserregung	-	SET	E1 bis E16	E1 bis E16 ist Befehlswort
Stop bei folgenden Mikrobefehl	-	STEP-SINGLE	-	
Stop bei folgenden 2 Byte Zyklus (Mikro)	-	STEP-SINGLE, CYCLE	-	
Setzen auf Mikro-Programmstart	-	RESET	-	Programm-Beginn bei #. #. #. #.
Fortsetzung Mikro-Programm	-	START	-	
Lesen extern	X	EXTERN READ	-	Speicherinhalt (D1 bis D16) der eingestellten Absolut-Adresse Auslesen
Schreiben extern	X	EXTERN WRITE	E1-bis E16	Schreiben auf X-Adresse (E1 bis E16)
Replace bei X-Adresse	X	REPLACE	E1 bis E16	E1 bis E16 ersetzt Datenwort bei X-Adresse
Schnappschuss bei Y-Adresse	Y	SNAP SHOT Y	-	Die Inhalte von E-, R-Bus und Akku werden dargestellt. Programm läuft weiter

Einige Beispiele für Prüfung mit Testtableau

J/O - Befehl



1. Prüfung Rechner

Mit funktion "SET" können die Mikrobefehle nachgebildet werden.

2. Prüfung Drucker 2 (Ausgabe ∞ - Zeichen, Ziffern)

x-Schalter: Kartendresse 1 (Schalter x5), IOX-Schalter
 Zeile 4 (Schalter x3), Write Extern-Schalter
 Ausgabe (Schalter x4),
 Gewünschtes Zeichen laut ISO-7 Bit-Code
 (Schalter E9...E14)
 Kopftransport und Nadelmagnet (Schalter E3 und E4)
 Anzeige: EX1, Run und IO leuchten.

3. Prüfung Systemlampen

x-Schalter: Kartendresse 1 (Schalter x5), IOX-Schalter
 Zeile 6 (Schalter x2, x3), Write Extern-Schalter
 Ausgabe (Schalter x4)
 Systemlampen leuchten durch Ansteuerung über E1...E8
 Anzeige: EX1, Run und IO leuchten

4. Prüfung Rastlampen

Wie unter 3.
 Jedoch Zeile 2 (nur Schalter x2)
 Rastlampen leuchten durch Ansteuerung über E1...E8
 Anzeige: EX1, Run und IO leuchten.

5. Prüfung Hauptlampen

Wie unter 3.
 Jedoch Zeile 0.
 Hauptlampen leuchten durch Ansteuerung über E1...E8
 Anzeige: EXT und Run leuchten

6. Prüfung WS (Eingabe: Taktscheibe Drucker 2)

x-Schalter: Kartenadresse 1 (Schalter x5), IOX-Schalter
Zeile 4 (Schalter x3), Extern Read-Schalter
Eingabe (Schalter x4 nicht betätigt)
Lampe E2 verlöscht, wenn der Druckkopf von Hand nach
rechts bewegt wird.

7. Prüfung Testator-Code, Parity Bit

x-Schalter: Kartenadresse 1 (Schalter x5), IOX-Schalter
Zeile 6 (Schalter x2, x3) Extern READ-Schalter
Eingabe (Schalter x4 nicht betätigt)
Beim Betätigen einer Taste wird der Code durch die nicht-
leuchtende Anzeige der Lampe D₁...D₈ angezeigt
(ISO-7 Bit-Code).
Sollte die Codierung ein Parity Bit verlangen, dann er-
löscht Lampe 9.
Anzeige: RUN, IO und EX1 leuchten

8. Prüfung Papiertransport (1 Traktorpaar)

x-Schalter: Kartenadresse 0, IOX-Schalter
Zeile 0, Extern-Write-Schalter
Ausgabe (x4 Schalter betätigt)
Zuerst E₁₁ betätigen (Motor läuft an)
dann E₁₂ betätigen (Kupplung legt ein)
Abschaltung erfolgt in umgekehrter Reihenfolge.

rationscode 11 1					Assem- bler- Code	0 0	Assem- bler- Code	0 1
						Mathematic Code		Mathematic Code
16	15	14	13	12				
0	0	0	0	0	BS	BS→N _Z 1)	BS	BS→P + N 1)
0	0	0	1	0	BR	BR→N _Z	BR	BR→P + N
0	0	1	0	0	BE	BR / (A) = 0→N _Z	BE	BR / (A) = 0→P + N
0	0	1	1	0	BU	BR / (A) ≠ 0→N _Z	BU	BR / (A) ≠ 0→P + N
0	1	0	0	0	BP	BR/(A1...A9) odd Parity→N _Z 2)	BP	BR/(A1...A9) odd Parity→P+N 2)
0	1	0	1	0	BL	BR / (A1) = 1→N _Z	BL	BR / (A1) = 1→P + N
0	1	1	0	0	BZ	BR / (C) = 0→N _Z	BZ	BR / (C) = 0→P + N
0	1	1	1	0	BC	BR / (C) = 1→N _Z	BC	BR / (C) = 1→P + N
1	0	0	0	0	LAD	N→A		
1	0	0	1	0	MAD	(A) ∧ N→A		
1	0	1	0	0	EAD	(A) ⊕ N→A		
1	0	1	1	0	OAD	(A) ∨ N→A		
1	1	0	0	0	SY	Sonderbefehl Special commande ordre spécial 5)		
1	1	0	1	0	AFD	(A) + N + H.B.→A, C 6)		
1	1	1	0	0	CAD	(A, C) + N→A, C		
1	1	1	1	0	AAD	(A) + N→A, C		
0	0	0	0	1	SML	(A)→S + N _L	SMB	(A)→(S + N) _B
0	0	0	1	1	MML	(A) ∧ (S + N) _L →A, S + N _L	MMB	(A) ∧ ((S+N)) _B →A (S+N) _B
0	0	1	0	1	EML	(A) ⊕ (S+N) _L →A, S + N _L	EMB	(A) ⊕ ((S+N)) _B →A (S+N) _B
0	0	1	1	1	OML	(A) ∨ (S+N) _L →A, S + N _L	OMB	(A) ∨ ((S+N)) _B →A (S+N) _B
0	1	0	0	1	IML	(S+N) _L + 1→A, S+N _L	IMB	((S+N)) _B + 1→A (S+N) _B
0	1	0	1	1	DML	(S+N) _L - 1→A, S+N _L	DMB	((S+N)) _B - 1→A (S+N) _B
0	1	1	0	1	CML	(A,C) + (S+N) _L →A,C, S+N _L	CMB	(A,C) + ((S+N)) _B →A,C, (S+N) _B
0	1	1	1	1	AML	(A) + (S+N) _L →A,C, S+N _L	AMB	(A) + ((S+N)) _B →A,C, (S+N) _B
1	0	0	0	1	LAL	(S+N) _L →A	LAB	((S+N)) _B →A
1	0	0	1	1	MAL	(A) ∧ (S+N) _L →A	MAB	(A) ∧ ((S+N)) _B →A
1	0	1	0	1	EAL	(A) ⊕ (S+N) _L →A	EAB	(A) ⊕ ((S+N)) _B →A
1	0	1	1	1	OAL	(A) ∨ (S+N) _L →A	OAB	(A) ∨ ((S+N)) _B →A
1	0	0	1	1	IOL 8)	IO (N) _L ↔A	IOB	IO ((S+N)) _B ↔A
1	1	0	1	1	DAL	(S+N) _L - 1→A	DAB	((S+N)) _B - 1→A
1	1	1	0	1	CAL	(A,C) + (S+N) _L →A, C	CAB	(A,C) + ((S+N)) _B →A,C
1	1	1	1	1	AAL	(A) + (S+N) _L →A, C	AAB	(A) + ((S+N)) _B →A,C



Operationscode Bit 11 1	Assembler- Code	1 0	Assembler- Code	1 1
		Mathematic Code		Mathematic Code
0 0 0 0 0 1	BSX	$BS \rightarrow (S+N)$ 1)	BSI	$BS \rightarrow (P+N)$ 1)
0 0 0 1 0 1	BRX	$BR \rightarrow (S+N)$	BRI	$BR \rightarrow (P+N)$
0 0 1 0 0 1	BEX	$BR / (A) = 0 \rightarrow (S+N)$	BEI	$BR / (A) = 0 \rightarrow (P+N)$
0 0 1 1 0 1	BUX	$BR / (A) \neq 0 \rightarrow (S+N)$	BUI	$BR / (A) \neq 0 \rightarrow (P+N)$
0 1 0 0 0 1	RTX	$RT \rightarrow (S+N)$ 3)	RTI	$RT \rightarrow (P+N)$ 4)
0 1 0 1 0 1	BLX	$BR / (A1) = 1 \rightarrow (S+N)$	BLI	$BR / (A1) = 1 \rightarrow (P+N)$
0 1 1 0 0 1	BZX	$BR / (C) = 0 \rightarrow (S+N)$	BZI	$BR / (C) = 0 \rightarrow (P+N)$
0 1 1 1 0 1	BCX	$BR / (C) = 1 \rightarrow (S+N)$	BCI	$BR / (C) = 1 \rightarrow (P+N)$
1 0 0 0 0 1	LA	$(N)_Z \rightarrow A$	LA	$(P+N) \rightarrow A$
1 0 0 1 0 1	MA	$(A) \wedge (N)_Z \rightarrow A$	MA	$(A) \wedge (P+N) \rightarrow A$
1 0 1 0 0 1	EA2	$(A) \oplus (N)_Z \rightarrow A$	EA	$(A) \oplus (P+N) \rightarrow A$
1 0 1 1 0 1	OA	$(A) \vee (N)_Z \rightarrow A$	OA	$(A) \vee (P+N) \rightarrow A$
1 1 0 0 0 1	IA	$(N)_Z + 1 \rightarrow A$	IA	$(P+N) + 1 \rightarrow A$
1 1 0 1 0 1	DA	$(N)_Z - 1 \rightarrow A$	DA	$(P+N) - 1 \rightarrow A$
1 1 1 0 0 1	CA	$(A, C) + (N)_Z \rightarrow A, C$	CA	$(A, C) + (P+N) \rightarrow A, C$
1 1 1 1 0 1	AA	$(A) + (N)_Z \rightarrow A, C$	AA	$(A) + (P+N) \rightarrow A, C$
0 0 0 0 1 1	SMX	$(A) \rightarrow S+N$	SMI	$(A) \rightarrow (S+N)$
0 0 0 1 1 1	MMX	$(A) \wedge (S+N) \rightarrow A, S+N$	MMI	$(A) \wedge ((S+N)) \rightarrow A, (S+N)$
0 0 1 0 1 1	EMX	$(A) \oplus (S+N) \rightarrow A, S+N$	EMI	$(A) \oplus ((S+N)) \rightarrow A, (S+N)$
0 0 1 1 1 1	OMX	$(A) \vee (S+N) \rightarrow A, S+N$	OMI	$(A) \vee ((S+N)) \rightarrow A, (S+N)$
0 1 0 0 1 1	IMX	$(S+N) + 1 \rightarrow A, S+N$	IMI	$((S+N)) + 1 \rightarrow A, (S+N)$
0 1 0 1 1 1	DMX	$(S+N) - 1 \rightarrow A, S+N$	DMI	$((S+N)) - 1 \rightarrow A, (S+N)$
0 1 1 0 1 1	CMX	$(A, C) + (S+N) \rightarrow A, C, S+N$	CMI	$(A, C) + ((S+N)) \rightarrow A, C, (S+N)$
0 1 1 1 1 1	AMX	$(A) + (S+N) \rightarrow A, C, S+N$	AMI	$(A) + ((S+N)) \rightarrow A, C, (S+N)$
1 0 0 0 1 1	LAX	$(S+N) \rightarrow A$	LAI	$((S+N)) \rightarrow A$
1 0 0 1 1 1	MAX	$(A) \wedge (S+N) \rightarrow A$	MAI	$(A) \wedge ((S+N)) \rightarrow A$
1 0 1 0 1 1	EAX	$(A) \oplus (S+N) \rightarrow A$	EAI	$(A) \oplus ((S+N)) \rightarrow A$
1 0 1 1 1 1	OAX	$(A) \vee (S+N) \rightarrow A$	OAI	$(A) \vee ((S+N)) \rightarrow A$
1 1 0 0 1 1	IOX	$IO (N) \leftrightarrow A$ 7)	IOI	$IO ((S+N)) \leftrightarrow A$ 7)
1 1 0 1 1 1	DAX	$(S+N) - 1 \rightarrow A$	DAI	$((S+N)) - 1 \rightarrow A$
1 1 1 0 1 1	CAX	$(A, C) + (S+N) \rightarrow A, C$	CAI	$(A, C) + ((S+N)) \rightarrow A, C$
1 1 1 1 1 1	AAX	$(A) + (S+N) \rightarrow A, C$	AAI	$(A) + ((S+N)) \rightarrow A, C$

0

Bemerkung:

- 1) Beim Unterprogrammsprung BS wird vor der Ausführung des Sprungs der um +2 erhöhte alte Befehlszähler P in das A-Register A2 bis A16 und das C-Register nach A1 geladen
(P) + 2 → A2 ... A16, (C) → A1
- 2) Dieser Sprungbefehl wird dann ausgeführt, wenn im A-Register in A1 bis A9 eine ungerade Anzahl von "Einsen" stehen.
- 3) Rücksprung vom Unterbrechungsprogramm ins Hauptprogramm:
Der Inhalt von B1 der Rücksprungadresse wird nach C geladen.
Die Generalunterbrechungssperre wird rückgesetzt und wenn die Generalunterbrechungssperre vorher nicht gesetzt war, wird die Uhrunterbrechungssperre gelöscht.
- 4) Rücksprung vom Unterbrechungsprogramm ins Hauptprogramm:
Der Inhalt von B1 der Rücksprungadresse wird nach C geladen.
Die Generalunterbrechungssperre wird rückgesetzt und wenn die Generalunterbrechungssperre vorher gesetzt war, wird die Uhrunterbrechungssperre gesetzt und der Uhrunterbrechungswunsch gelöscht.

5) Sonderbefehl

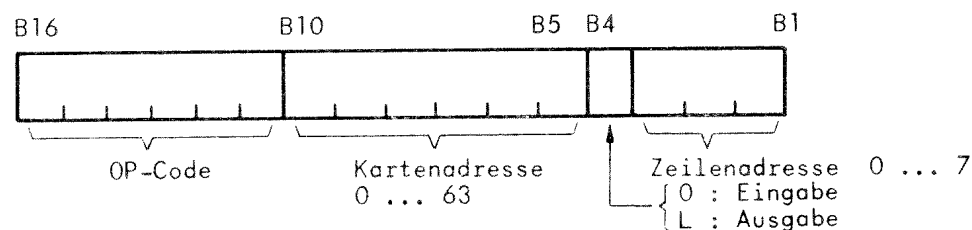
Der Sonderbefehl bezieht sich nur auf das A-Register. Im Adreßteil sind nebenstehende Befehle codiert. Kombinationen beider Befehlsgruppen sind möglich.

*	B10	B9	
RS	1	0	(A) Rechtssshift um 1 Bit; 0 → A16
RC	1	1	(A) Rechtssshift um 1 Bit; (C) → A16
GR	0	1	B1 Löschen Rechner
PR	1	x	B3 Löschen Parityfehler (Option)

*	B3	B2	B1	
CC	0	0	0	: (C) → C
SC	0	0	1	: 1 → C
LC	0	1	0	: (A1) → C
HC	0	1	1	: (A16) → C
TC	1	0	0	: Uhrunterbrechungssperre → C
FC	1	0	1	: Netzausfall → C
PC	1	1	0	: Paritätsfehler → C
DC	1	1	1	: IO Unterbrechungswunsch → C

6) Dem Direktoperanden N werden die "Hohen Bits H.B." 11 ... 16 = 1 hinzugefügt.

7) IO-Befehlsformat:



Programmunterbrechung: 1 ms Uhr und Parityfehler (als Option) bewirkt (A) → S0, BS → 12, IO-Unterbrechung bewirkt (A) → S0, BS → S.