

Prüfung der Daten zweier Speicherbereiche auf Gleichheit

Bedienung:

Starten des Programms: mit run C000 (z.B. je nachdem auf welcher Speicherseite das EPROM liegt)

Es erscheint "ad.... " auf dem Display.

- 1) Eingabe der Anfangsadresse des 1. zu vergleichenden Bereichs.
- 2) Eingabe der Endadresse des 1. zu vergleichenden Bereichs.
- 3) Eingabe der Anfangsadresse des 2. zu vergleichenden Bereichs.

Das Programm meldet sich mit " ELBUG " bei Übereinstimmung der Bereiche.

Bei Nichtübereinstimmung erscheint die Adresse und das Datum des ersten sich unterscheidenden Speicherplatzes auf dem Display.

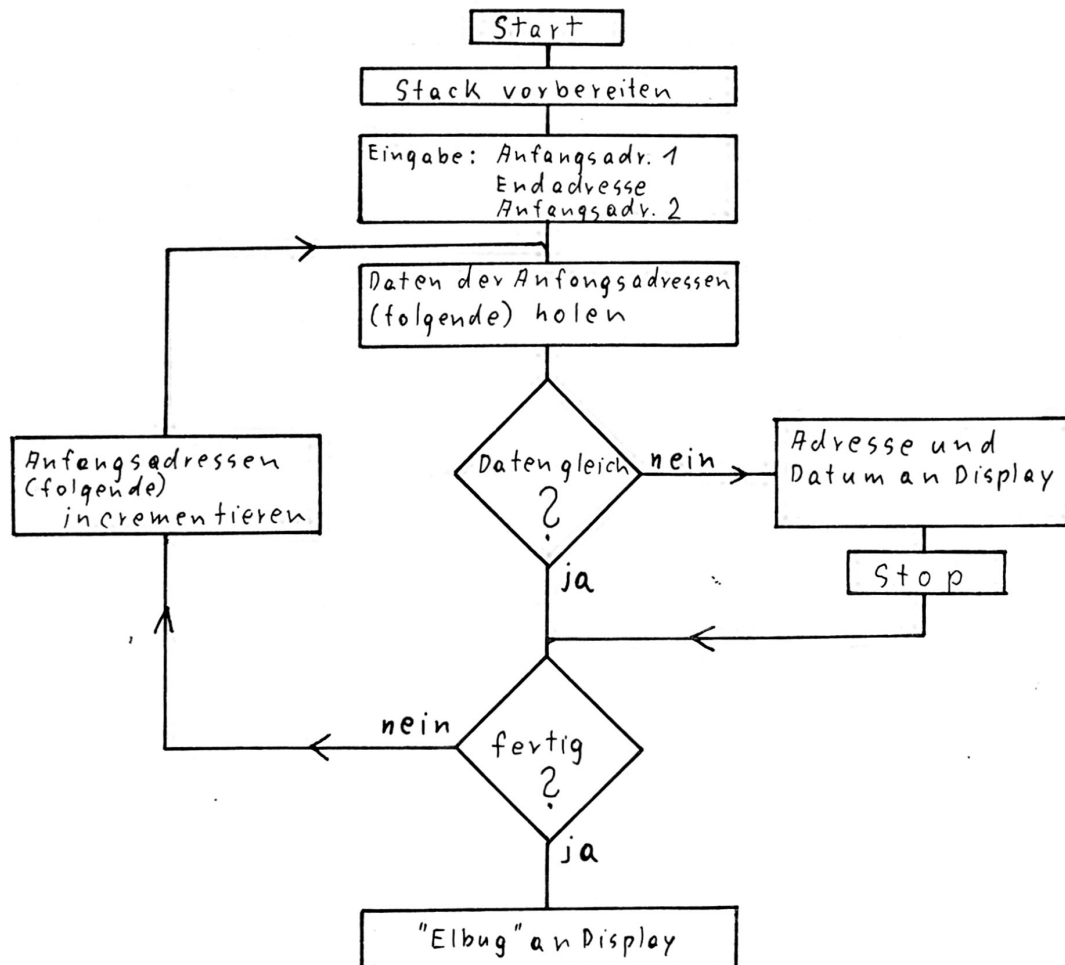
Durch drücken der Run/Stop Taste läuft das Programm weiter und gibt entweder die Adresse des nächsten nicht übereinstimmenden Speicherplatzes aus, oder meldet sich bei Übereinstimmung mit " ELBUG ".

Das Programm belegt auf dem EPROM die Adressen 0000 - 00D3.

Das Programm ist frei verschiebbar (nur relative Sprünge).

Es Verwendet die Monitorroutinen Push, Pull, Puthex, u. Gethex.

Es verwendet die festen Speicheradressen 0FF5-0FF0 als Zwischenspeicher für die Anfangs und Endadressen.



0 0 0 0	0 8	NOP	
	C 4 F F	L D I F F	
	3 1	X P A L (1)	} Pointer I auf Stackpointer (Adr. 0FFF)
	C 4 0 F	L D I 0 F	
	3 5	X P A H (1)	
	C 4 E 0	L D I E 0	
	C 9 0 0	S T O (1) 0 0	} initialize für Stack
	C 4 0 F	L D I 0 F	
	C 9 F F	S T O (1) F F	
	C 4 0 0	L D I 0 0	
0 0 1 1	C 9 F A	S T O (1) F A	
	C 9 F 9	S T O (1) F 9	
0 0 1 5	C 4 0 0	L D I 0 0	} Ptr. II auf Display
	3 2	X P A L (2)	
	C 4 0 7	L D I 0 7	
	3 6	X P A H (2)	
	C 4 5 F	L D I 5 F	} "dd" an Display 7 u. 6
	C A 0 7	S T O (2) 0 7	
	C 4 5 E	L D I 5 E	
2 1	C A 0 6	S T O (2) 0 6	
	C 4 8 0	L D I 8 0	} " " an Displ. 5 - 0
	C A 0 5	S T O (2) 0 5	
	C A 0 4	S T O (2) 0 4	
	C A 0 3	S T O (2) 0 3	
	C A 0 2	S T O (2) 0 2	
	C 4 0 0	L D I 0 0	
	C A 0 1	S T O (2) 0 1	
3 1	C A 0 0	S T O (2) 0 0	
	C 4 3 E	L D I 3 E	} Adr. von "Gethex-1" in "Rout. Adr." (Adr.: 0FFD = lower 0FFC = higher)
	C 9 F E	S T O (1) F E	
	C 4 0 2	L D I 0 2	
	C 9 F D	S T O (1) F D	
	C 4 5 5	L D I 5 5	} Adr. von "Push-1" (= 0055) in PTR III
	3 3	X P A L (3)	
	C 4 0 0	L D I 0 0	
4 0	3 7	X P A H (3)	
	3 F	X P P C (3)	} Anfangsadr. holen (steht in 0FE2 - higher) 0FE1 - Lower
	C 4 E 0	L D I E 0	
	3 1	X P A L (1)	
	C 4 0 F	L D I 0 F	
	3 5	X P A H (1)	} Pointer I auf Adr. 0FE0
	C 1 0 1	L D (1) 0 1	
	C 9 1 5	S T O (1) 1 5	
	C 1 0 2	L D (1) 0 2	
	C 9 1 4	S T O (1) 1 4	} Anfangsadr. des 1. Blocks in Ram Adr. 0FF5 (Lower Byte) u. 0FF4 (higher Byte) sichern
5 0	3 F	X P P C (3)	
1	C 1 0 1	L D (1) 0 1	
3	C 9 1 3	S T O (1) 1 3	
5	C 1 0 2	L D (1) 0 2	} Endadr. des 1. Blocks in Ram Adr. 0FF3 (Lower) u. 0FF2 (higher) sichern
7	C 9 1 2	S T O (1) 1 2	
	3 F	X P P C (3)	} Anfangsadr. des 2. Blocks von keyboard holen (Routine "Gethex")

Prüfung... auf Gleichheit

(Blatt 2)

0 C 5 A	C 1 0 1	LD (1) 01	} Anfangs adr. des 2. Blocks in Ram Adr. 0FF1 (Lower) u. 0FF0 (higher) sichern
	C 9 1 1	STO (1) 11	
	C 1 0 2	LD (1) 02	
	C 9 1 0	STO (1) 10	
	C 1 1 2	} ohne Bedeutung (Programmierfehler)	
	03		
	F 9 1 4		
	08 08	NOP, NOP	
	9 0 1 0	JMP 10	
	C 4 C E	} ohne Bedeutung (Programmierfehler)	
	3 3		
	C 4 0 3		
7 0	3 7		
	3 F		
7 2	C 1 1 3		
7 4	03		
7 5	F 9 1 5		
7 7	9 4 0 2		
7 9	9 0 F 0		
7 B →	C 1 1 5	LD (1) 15	} Anfangs adr. des 1. Blocks (u. folgende) in Ptr. 2
	3 2	XPAL (2)	
	C 1 1 4	LD (1) 14	
8 0	3 6	XPAH (2)	} Anfangs adr. des 2. Blocks (u. folgende) in PTR III
1	C 1 1 1	LD (1) 11	
3	3 3	XPAL (3)	
4	C 1 1 0	LD (1) 10	
6	3 7	XPAH (3)	} Inhalte der Adressen gleich ?
7	C 2 0 0	LD (2) 00	
9	0 1	XAE	
H	C 3 0 0	LD (3) 00	
C	6 0	XRE	} wenn ja: weiter auf "00 A8" JNZ 9C wenn nein:
D	9 8 1 9	JZ 19	
F	C 4 A 0	LDI A 0	
9 1	C 9 1 D	STO (1) 1D	
	C 4 0 2	LDI 0 2	} Adresse u. Datum des nicht gleichen Inhalts zur Ausgabe für Display inden dafür vorgesehenen Ram Adressen ablegen (0FE2 = Adr. high, 0FE1 = Adr. Low 0FE0 = Data
	C 9 1 C	STO (1) 1C	
	3 2	XPAL (2)	
	C 9 0 1	STO (1) 01	
	3 6	XPAH (2)	} "Push-1" in Ptr. III
	C 9 0 2	STO (1) 02	
	0 1	XAE	
	C 9 0 0	STO (1) 00	
A 0	C 4 5 5	LDI 5 5	} Adresse u. Datum an Display HALT
	3 3	XPAH (3)	
	C 4 0 0	LDI 0 0	
	3 7	XPAH (3)	
	3 F	XPC (3)	} Adresse = Endadresse ?
	0 0	HALT	
A 8 →	C 1 1 3	LD (1) 13	
	0 3	SCL	
	F 9 1 5	CAD (1) 15	
	9 C 0 8	JNZ 08	
	C 1 1 2	LD (1) 12	
B 1	F 9 1 4	CAD (1) 14	

00B3	9C02	JNZ02	}
	9130	JMP(1)30	
	02	CCL	
	C115	LD(1)15	
	F401	ADI01	
	C915	STO(1)15	
	C114	LD(1)14	
C0	F400	ADI00	
C2	C914	STO(1)14	
C4	02	CCL	
C5	C111	LD(1)11	
C7	F401	ADI01	
C9	C911	STO(1)11	
CB	C110	LD(1)10	
CD	F400	ADI00	
CF	C910	STO(1)10	
d1	90A8	JMPA8	
d3	00		

ja: "Elbug" anspringen

Anfangsadr. (folgende) des 1. Blocks

u. Anfangsadr. des 2. Blocks (folgende)

für nächsten Durchlauf incrementieren

00

auf Adr. 007B zum nächsten Durchlauf