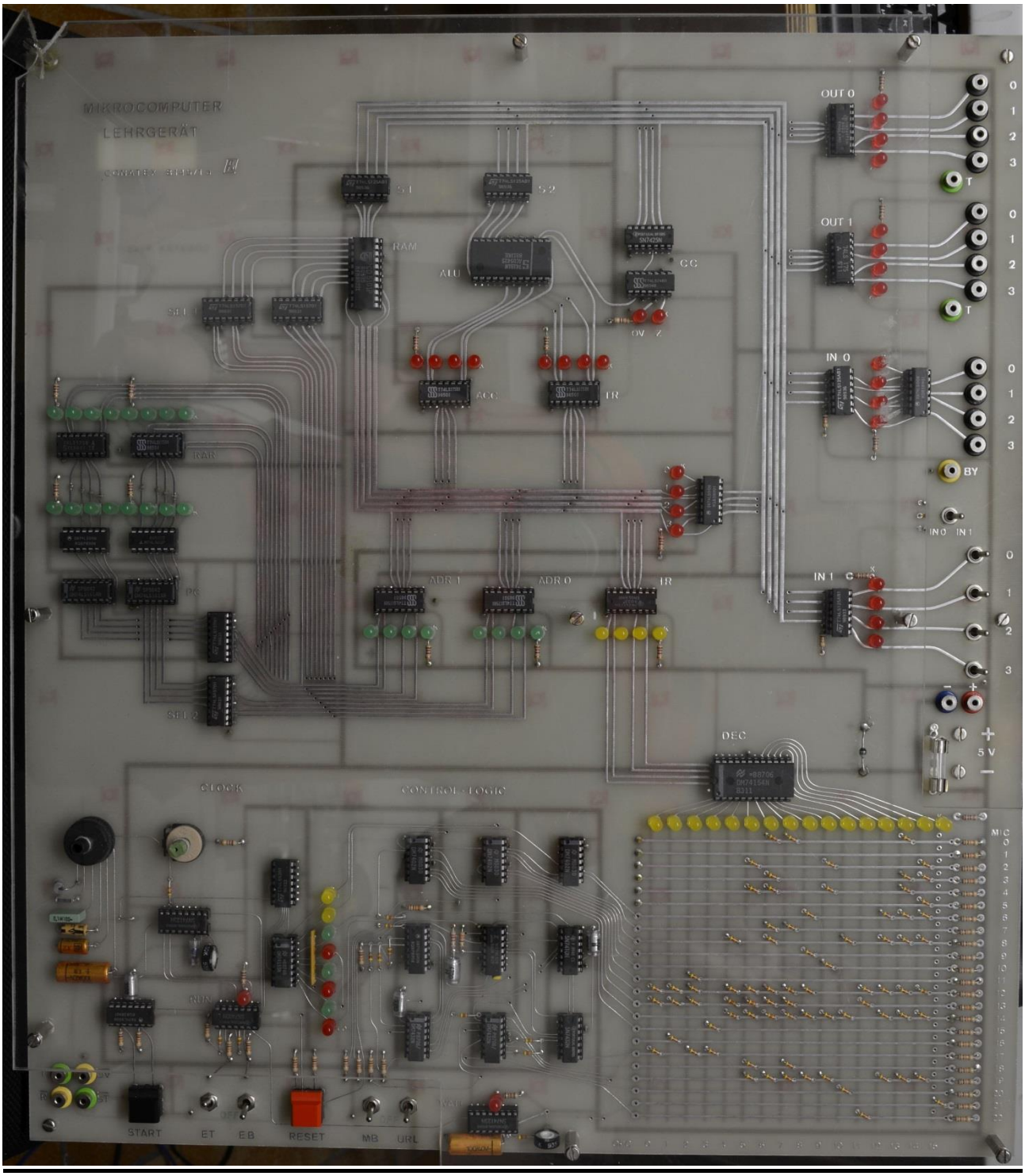
Modellrechner CO 8145



Version 2018

Autor: Dieter Lindenberg

Code	Mnemonic	Bedeutung	Anzahl Worte
0000	JMP adr	**JUMP:** unbedingter Sprung auf Adresse adr	3
0001	JNZ adr	**Jump if not zero:** Sprung auf Adresse adr, falls Zero-Flag ungleich Null	3
0010	Call adr	**Call:** Unterprogrammsprung auf Adresse adr	3
0011	RET	**Return:** Unterprogramm-Rücksprung	1
0100	LDA adr	**Load accumulator:** <acc> := <adr>	3
0101	STA adr	**Store accumulator:** <adr> := <acc>	3
0110	ADDA adr	**Add accumulator:** <acc> := <acc> + <adr>	3
0111	DECM adr	**Decrement memory:** <adr> := <adr> - 1	3
1000	OUT0 n	**Output 0:** <OUT 0> := n	2
1001	OUT1 -	**Output 1:** <OUT 1> := <acc>	1
1010	INA -	**Input accumulator:** <acc> := <IN0 / IN1>	1
1011	JBY adr	**Jump if busy:** Sprung auf Adresse adr, falls Signal auf BY-Buchse low	3
1100	ENTA n	**Enter accumulator:** <acc> := n	2
1101	INCA n	**Increment accumulator:** <acc> := <acc> + n	2
1110	NANDA n	**NAND accumulator:** <acc> := <acc> NAND n	2
1111	WAIT -	**Wait:** Stop des Rechners für eine einstellbare Zeitdauer	1

Bemerkungen:

- das sog. Zero-Flag wird u.a. immer dann gesetzt, wenn der Inhalt des Akkus gleich Null ist.

- das Zero-Flag wird auch dann gesetzt, wenn z.B. durch den Befehl **DECM adr** der Inhalt einer Speicherzelle den Wert Null erhält, obwohl dies keinen Einfluss auf den Akku-Inhalt hat.

  das Folgende ist weniger wichtig, aber es kann vielleicht manchmal ein Programmierproblem erklären:
  Bei den Befehlen *ENTA*, *OUT0* und *NANDA*, bei denen kein Überlauf möglich ist, wird das ***Overflow-Flag*** gesetzt, wenn der Operand bei diesen Befehlen ungleich Null ist. Dies liegt an der Konstruktion des Modellrechners.

- Durch eine Kabelverbindung zwischen der Buchse BY und der Buchse $\overline{OV}$ erhält der Befehl **JBY adr** die Bedeutung: „Springe, wenn Überlauf erfolgt ist!“

- Bei Betätigung der Reset-Taste am Modellrechner werden der Befehlszähler und alle Register auf Null gesetzt. Das Overflow-Flag wird ebenfalls auf Null gesetzt, während das Zero-Flag auf 1 gesetzt wird, weil alle Register ja Null sind.

Urladen

Um ein Programm in den Modellrechner zu laden mache Folgendes:

- Drehschalter Taktfrequenz Stufe 2, Poti ganz rechts.
- Einzeltakt-Schalter aus.
- Einzelbefehl-Schalter ein.
- falls Taktgenerator noch nicht in Grundstellung, Drücken der Start-Taste.
- Schalter für manuelle Befehlsausführung aus.
- Schalter Urladen ein.
- den Eingang ***IN1*** wählen (Schalterstellung!)
- Reset-Taste drücken.

Am Eingang ***IN1*** stellt man nun das erste 4-Bit-Wort ein und betätigt anschließend die Start-Taste. Danach steht dieses 4-Bit-Wort in der nullten Zelle. Anschließend gibt man genauso alle weiteren 4-Bit-Worte ein.

Danach steht das Programm im Speicher und kann ausgeführt werden.

Mache dazu Folgendes:
- Einzeltakt-Schalter aus.
- Einzelbefehl-Schalter aus.
- Schalter Manuelle Befehlsausführung aus.
- Schalter Urladen aus.
- Reset-Taste drücken.
- Start-Taste drücken.

Das Programm kann durch Einschalten des Einzelbefehls-Schalters gestoppt werden.

Übungsaufgaben

1. Gib (unter Benutzung einer Schleife) an ***OUT 1*** nacheinander die Zahlen 5 bis 15 aus.

2. *(schwierig)* Gib (unter Benutzung einer Schleife) an ***OUT 0*** nacheinander die Zahlen 5 bis 15 aus.

3. Gib (unter Benutzung einer Schleife) am Ausgang ***OUT 1*** nacheinander (rückwärts) die Zahlen von 10 bis 0 aus. Hinweis: Man kann die Zahl in einer Speicherzelle (auch in einer Zelle, deren Inhalt zum Maschinenprogramm gehört) dekrementieren.

4. Gib unendlich oft hintereinander abwechselnd die Zahlen 0 und 1 am Ausgang ***OUT 0*** aus!

5. Gib (unter Benutzung einer Schleife) 10 mal hintereinander abwechselnd die Zahlen 0 bzw. 1 am Ausgang ***OUT 0*** aus!

6. Bringe zunächst in die Zelle 80 die Zahl 7 und in die Zelle 81 die Zahl 5, indem du in die Zelle 79 einen (relativ sinnlosen) JMP-Befehl mit den oben genannten Zahlen (als Zieladresse) eingibst!
Schreibe nun ab Zelle 0 ein entsprechendes Maschinenprogramm, welches diese beiden Zahlen aus den Zellen 80 und 81 herausliest, im Akku addiert und die Summe im Ausgang ***OUT 1*** ausgibt! Im Ausgang OUT1 muss also die Zahl 12 erscheinen

7. Bringe zunächst in die Zellen 80 und 81 zwei beliebige Zahlen, indem du in die Zelle 79 einen JMP-Befehl mit den entsprechenden Zahlen eingibst!
Schreibe nun ein entsprechendes Maschinenprogramm, welches diese Zahlen aus den Zellen 80 und 81 herausliest und im Akku addiert!
Falls diese Summe größer als 15 sein sollte, erscheint in ***OUT 0*** die Zahl 1, ansonsten die Zahl 0. Hinweis: Verbinde dazu die Busy-Buchse mit dem $\overline{Overflow}$-Flag!

8. Wie Aufgabe 7, nur soll die Ausgabe jetzt in ***OUT 1*** erfolgen.

9. Es soll untersucht werden, ob der Inhalt der Speicherzelle 80 gerade oder ungerade ist. Bei geradem Inhalt wird in ***OUT 0*** eine 0 ausgegeben, ansonsten eine 1.
Tip: Hochzählen bis Null.

10. Am Ausgang ***OUT 0*** soll periodisch abwechselnd 0 oder 1 ausgegeben werden. Ein Lautsprecher wird an diesem Ausgang angeschlossen, sodass ein Ton erzeugt wird. Die Tonfrequenz soll abhängig von der Zahl im Eingang sein. Realisiert wird das Ganze mit folgendem Algorithmus:

Der Inhalt des Einganges wird in den Akku geschrieben. Anschließend wird der Akku solange erhöht, bis ein Übertrag stattfindet bzw. im Akku die Zahl 0 steht. Dann wird an ***OUT 0*** eine 1 ausgegeben. Nun wird wieder dasselbe gemacht. Beim nächsten Übertrag wird eine 0 ausgegeben. Das Ganze läuft endlos lange, sodass abwechselnd Null und Eins ausgegeben werden.

11. Schreibe zunächst in die beiden Ausgänge ***OUT 0*** und ***OUT 1*** und in den Akku jeweils die Zahl 0. Anschließend soll dauernd der Akkuinhalt hochgezählt werden und in ***OUT 0*** ausgegeben werden. Bei einem Übertrag wird dieser zu dem Inhalt von ***OUT 1*** addiert, sodass sich die Zahl in ***OUT 1*** erhöht.

12. Gib mit einer Schleife alle geraden Zahlen 0, 2, 4, …, 14 in ***OUT 1*** aus!

13. Gib mit einer Schleife alle ungeraden Zahlen 1, 3, 5, … ,15 in ***OUT 1*** aus!

14. Der Eingang (***IN 0*** bzw. ***IN 1***) soll dauernd überprüft werden. Sobald dort der Wert 0 erscheint, soll ein Warnsignal am Ausgang ***OUT 0*** auf den dort angeschlossenen Lautsprecher gegeben werden.

Lösungen

Aufgabe 1

```
 0  ENTA
 1     5
 2  OUT1
 3  INCA
 4     1
 5   JNZ
 6     2
 7     0
 8   JMP
 9     8
10     0
```

Aufgabe 2

```
 0  ENTA
 1     5
 2  OUT0
 3     5
 4  INCA
 5     1
 6   JNZ
 7    12
 8     0
 9   JMP
10     9
11     0
12   STA
13     3
14     0
15   JMP
16     2
17     0
```

Dieser Befehl bewirkt, dass es sich hier um einen sog. ***selbstmodifizierenden Programm-Code*** *handelt: Das eigene Programm wird durch diesen Befehl geändert!*

Bemerkung: Manche Computerviren enthalten selbstmodifizierenden Programmcode. Harmlos erscheinende Befehle werden damit durch andere Befehle ersetzt.

Aufgabe 3

```
 0  ENTA
 1    10
 2  Out1
 3  DECM
 4     1
 5     0
 6   JNZ
 7     0
 8     0
 9   JMP
10     9
11     0
```

Aufgabe 4

```
 0  OUT0
 1     0
 2  OUT0
 3     1
 4   JMP
 5     0
 6     0
```

Aufgabe 5

```
 0  ENTA
 1    10
 2  OUT0
 3     0
 4  OUT0
 5     1
 6  DECM
 7     1
 8     0
 9   JNZ
10     0
11     0
12   JMP
13    12
14     0
```

Aufgabe 6

```
 0   LDA
 1     0
 2     5
 3  ADDA
 4     1
 5     5
 6  OUT1
 7   JMP
 8     7
 9     0
……..….
……..….
79   JMP
80     7
81     5
……..
```

Aufgabe 7

0	LDA
1	0
2	5
3	ADDA
4	1
5	5
6	JBY
7	14
8	0
9	OUT0
10	1
11	JMP
12	11
13	0
14	OUT0
15	0
16	JMP
17	0
18	1
…….	….
…….	….
79	JMP
80	?
81	?

Aufgabe 8

```
 0    LDA
 1      0
 2      5
 3   ADDA
 4      1
 5      5
 6    JBY
 7     15
 8      0
 9   ENTA
10      1
11   OUT1
12    JMP
13     12
14      0
15   ENTA
16      0
17   OUT1
18    JMP
19      2
20      1
……..….
……..….
79    JMP
80      ?
81      ?
```

Aufgabe 9

0	LDA
1	0
2	5
3	INCA
4	1
5	JNZ
6	13
7	0
8	OUT0
9	1
10	JMP
11	10
12	0
13	INCA
14	1
15	JNZ
16	3
17	0
18	OUT0
19	0
20	JMP
21	4
22	1

Aufgabe 10

0	OUT0
1	0
2	INA
3	INCA
4	1
5	JNZ
6	3
7	0
8	OUT0
9	1
10	INA
11	INCA
12	1
13	JNZ
14	11
15	0
16	OUT0
17	0
18	JMP
19	2
20	0

Aufgabe 11

```
 0  OUT0
 1     0
 2  ENTA
 3     0
 4  OUT1
 5   STA
 6     8
 7     2
 8  INCA
 9     1
10   STA
11    14
12     0
13  OUT0
14     9      Dummy-Zahl
15   JNZ
16     8
17     0
18   LDA
19     8
20     2
21  INCA
22     1
23   STA
24     8
25     2
26  OUT1
27   JMP
28     8
29     0
```

Aufgabe 12

```
 0  ENTA
 1     0
 2  OUT1
 3  INCA
 4     2
 5   JNZ
 6     2
 7     0
 8   JMP
 9     8
10     0
```

Aufgabe 13

Die Busy-Buchse muss mit der $\overline{Overflow}$-Buchse verbunden werden!

```
 0  ENTA
 1     1
 2  OUT1
 3  INCA
 4     2
 5   JBY
 6     2
 7     0
 8   JMP
 9     8
10     0
```

Logische Bit-Operatoren

```
      1001
OR    1100
_____________
=     1101
```

Der Operator NOT negiert jedes einzelne Bit:
NOT 1001 = 0110

```
      1001
AND   1100
_____________
=     1000
```

Mit dem **AND**-Operator kann man bestimmte Bitstellen einer Zahl ermitteln.

Beispiel: Sei x irgendeine beliebige 4-stellige Binärzahl. Dann hat x AND 0001 entweder den Wert 1 oder den Wert 0.
Der Term x AND 1000 könnte entscheiden, ob x größer oder gleich dez 8 ist.

x = 1001, x **AND** 1100 = 1000

x **NAND** 1100 = **NOT** (x **AND** 1100) = **NOT** 1000 = 0111

Berechne folgende Binärzahlen:

a) 1010 AND 1100 =

b) 1110 **AND** 0001 =

c) 1101 **AND** 1111 =

d) 1101 **AND** 0000 =

e) 1101 **OR** 0011 =

f) 0101 **OR** 1010 =

g) 1101 **OR** 1110 =

h) 0110 **OR** 1111 =

i) 0110 **OR** 0000 =

j) **NOT** 0110 =

k) **NOT** 0000 =

l) **NOT** 1111 =

m) 1010 **NAND** 1010 =

n) 1010 NAND 0101 =

o) 1010 NAND 0000 =

p) 1010 NAND 1111

q) **NOT** {[(1100 **OR** 0101) **AND** 0001] **NAND** 1101}

Subtraktion

Im Folgenden wird beschrieben, wie man eine Subtraktion berechnen kann, indem man diese Subtraktion auf eine passende Addition zurückführt. Weil diese Methode (für uns Menschen zwar völlig ungewohnt) sehr schnell ist, genügt es, den Computern nur die Addition beizubringen. Die Subtraktion führen Computer dann mit dieser Methode aus.

Das sog. **Neunerkomplement** einer Zahl im Dezimalsystem wird gebildet, indem man ziffernweise die Differenz zu 9 bildet.
Beispiel: Das Neunerkomplement von 76 ist 23.

Durch Addition von 1 zum Neunerkomplement erhält man das sog. **Zehnerkomplement**.
Beispiel: Das Zehnerkomplement von 76 ist 24. Mit anderen Worten: Die Zehnerpotenz einer Dezimalzahl ist die Differenz dieser Zahl zur nächsthöheren Zehnerpotenz (hier: 100).
Also: Zehnerkomplement von 76 = 100 – 76 = 24

Eine Subtraktion im Dezimalsystem lässt sich auch durchführen, indem man das entsprechende Zehnerkomplement des Subtrahenden addiert und den dabei zwangsläufig entstehenden Übertrag ignoriert.
Beispiel: Angenommen ein bestimmter Rechnertyp kann grundsätzlich nur mit 4-stelligen Dezimalzahlen rechnen. Dieser Rechner soll nun die Subtraktion 7 – 4 ausführen. Er kann aber nicht subtrahieren, sondern nur addieren.
Deshalb addiert er nun zur Zahl 7 das Zehnerkomplement von 4,
also: 7 + (10 – 4) = (7 – 4) + 10. Offensichtlich ist dieses Ergebnis um 10 zu groß.

Beachte, dass der Rechner grundsätzlich nur mit vierstelligen Dezimalzahlen rechnet!
Es soll also gerechnet werden: 0007 – 0004
Das Neunerkomplement von 0004 ist 9995, demzufolge ist das Zehnerkomplement von 0004 die Zahl 9995 + 0001 = 9996
Addiert man nun zur Zahl 0007 die Zahl 9996, so erhält man 10003. Weil der Rechner nur mit vierstelligen Dezimalzahlen rechnen kann, zeigt er das Ergebnis 0003 an. Natürlich wird er bemerken, dass bei seiner „Addition“ ein Übertrag stattgefunden hat, und er wird dies im sog. Overflow-Flag speichern.

Völlig analog wird im Dualsystem eine Subtraktion durchgeführt. Als Beispiel soll wieder 7 – 4 berechnet werden. Anstatt 4 zu subtrahieren, wird das 2-Komplement von 4 (in Binärdarstellung) addiert und der zwangsläufig stattfindende Übertrag wird ignoriert bzw. kann nicht mehr dargestellt werden.

Das 2-Komplement wird analog wie oben ermittelt, indem man zunächst das 1-Komplement berechnet und anschließend noch die Zahl 1 hinzu addiert.

Also: dez 7 = % 0111, dez 4 = 0100, 1-Komplement von 0100 = 1011

Das 1-Komplement einer Dualzahl erhält man sehr einfach, indem man in der Dualdarstellung jede Ziffer 1 durch die Ziffer 0 ersetzt und umgekehrt. Das 1-Komplement der Zahl im Akku erhält man durch den Befehl ***NANDA 15***. Nach diesem Befehl steht das entsprechende 1-Komplement im Akku.

Das 2-Komplement von dez 4 = % 0100 erhält man dann so:

```
  1011
+ 0001
= 1100
```

Addiert man nun die Dezimalzahl 7 mit diesem 2-Komplement, so folgt:

```
    0111
+   1100
= 1 0011
```

Wenn man den Übertrag ignoriert, was Computer, die nur mit 4-stelligen Dualzahlen rechnen können, zwangsläufig machen müssen, so erhält man das richtige Ergebnis % 0011 = dez 3.

Anwendungsaufgabe:

Der Inhalt der Speicherstelle 31 soll vom Inhalt der Zelle 30 subtrahiert werden. Das Ergebnis soll in der Zelle 33 gespeichert werden. Das zugehörige Programm lautet:

```
LDA 31
NANDA 15  Das 1-Komplement wird gebildet
INCA 1    Das 2-Komplement wurde berechnet
ADDA 30
STA 33
```

Aufgabe: Gib am Ausgang 1 die Zahlen von 15 bis 0 rückwärts aus, indem du den Inhalt des Akkus jeweils um 1 subtrahierst.

Lösung: Das 2-Komplement von 0001 ist 1111.

```
0   ENTA  15
1
2   OUT1
3   ADDA  15   Der Akkuinhalt wird um 1 verringert
4
5   JNZ  2
6
7
8   JMP  8
9
10
```

Tongenerator

Mit dem folgenden Programm kann man 16 unterschiedliche Tonfrequenzen erzeugen. Man stellt am Eingang 0 mit den 4 Schaltern eine Zahl zwischen 0 und 15 ein. Der Wert dieser Zahl bestimmt die Tonhöhe. Je größer die Zahl ist, desto höher ist die Tonfrequenz. Die Zahlen entsprechen mit etwas Phantasie einer Tonreihe. Leider entsprechen die Frequenzabstände natürlich nicht den von der Musik her gewohnten Ganz- oder Halbtonschritten. Aber mit etwas gutem Willen lassen sich damit gespielte Melodien hinreichend genau erkennen. Melodien lassen sich erzeugen, indem man jedem Ton entsprechend die Zahl am Eingang 0 einstellt.
Die Anschlüsse des Lautsprechers müssen zwischen dem 0ten Bit von OUT0 und einem beliebigen Pol der Spannungsversorgung liegen.

```
0   OUT0  0   1000
1             0000
2   INA       1010
3   INCA  1   1101
4             0001
5   JNZ   3   0001
6             0011
7             0000
8   OUT0  1   1000
9             0001
10  INA       1010
11  INCA  1   1101
12            0001
13  JNZ   11  0001
14            1011
15            0000
16  JMP 0     0000
17            0000
18            0000
```