



CHIP-8

... von Volker Pohl



???





COSMAC VIP

COSMAC VIP

\$249* gets the entire family into creating video games, graphics and control functions. For starters.



COSMAC VIP, the completely assembled, ready-to-operate RCA Video Interface Processor, opens up a whole new world of computer excitement. New challenges in graphics, games and control functions. Yet it's just \$249.00.

Easy to buy. And easy to program, thanks to its unique, easy-to-use interpretive language. You get a complete how-to book including programs for 20 games: fun, challenging, and ready to load and record on your cassette.

Simple but powerful.

Built around an RCA COSMAC microprocessor, the VIP is a complete computer system that can grow with you. It has 2K of RAM, expandable on-board to 4K. Plus a ROM monitor, audio tone output to a built-in speaker, power supply, and 8-bit input and output ports for control of relays, sensors, or other peripherals.

Soon RCA will offer options for color graphics and 256 tone sound generation. An optional auxiliary keyboard will open up an exciting world of two-player games.

Take the first step now.

Check your local computer store or electronics distributor for the VIP. Or contact RCA VIP Marketing, New Holland Avenue, Lancaster, PA 17604. Phone (717) 291-5848.

*Suggested retail price. Does not include video monitor or cassette recorder.

The fun way into computers.

CIRCLE 169 ON READER SERVICE CARD

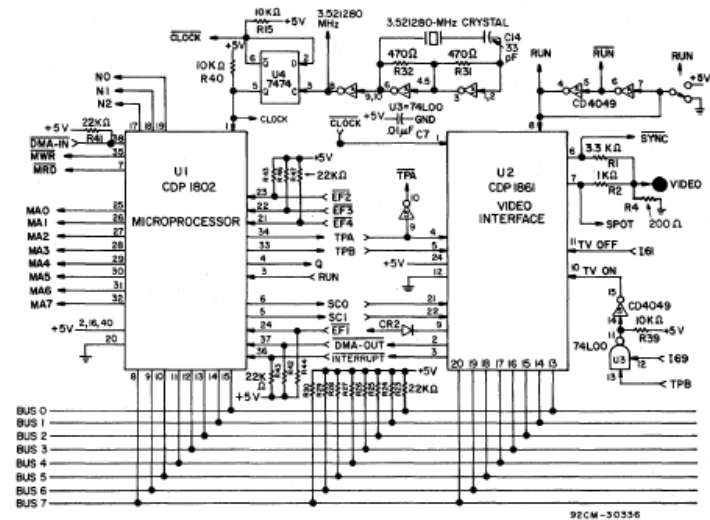
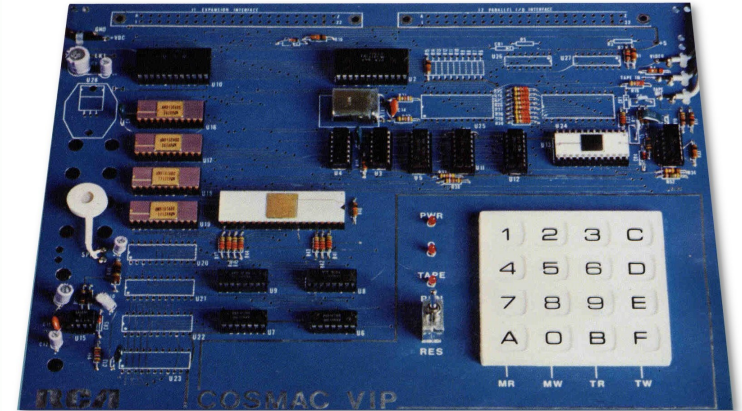


Fig. E-1 - Microprocessor and Display Interface Circuits



COSMAC VIP

Merkmal	Beschreibung
CPU	CDP-1802 (8-Bit)
ROM	0.5 KByte
RAM	2 KByte
Takt	1,76 MHz
Anzeige	TV, 64×32 Pixel, monochrom
Tastatur	16 Tasten (Hex)
Peripherie	Kassettenrekorder
Tastatur	Programmiersprache CHIP-8

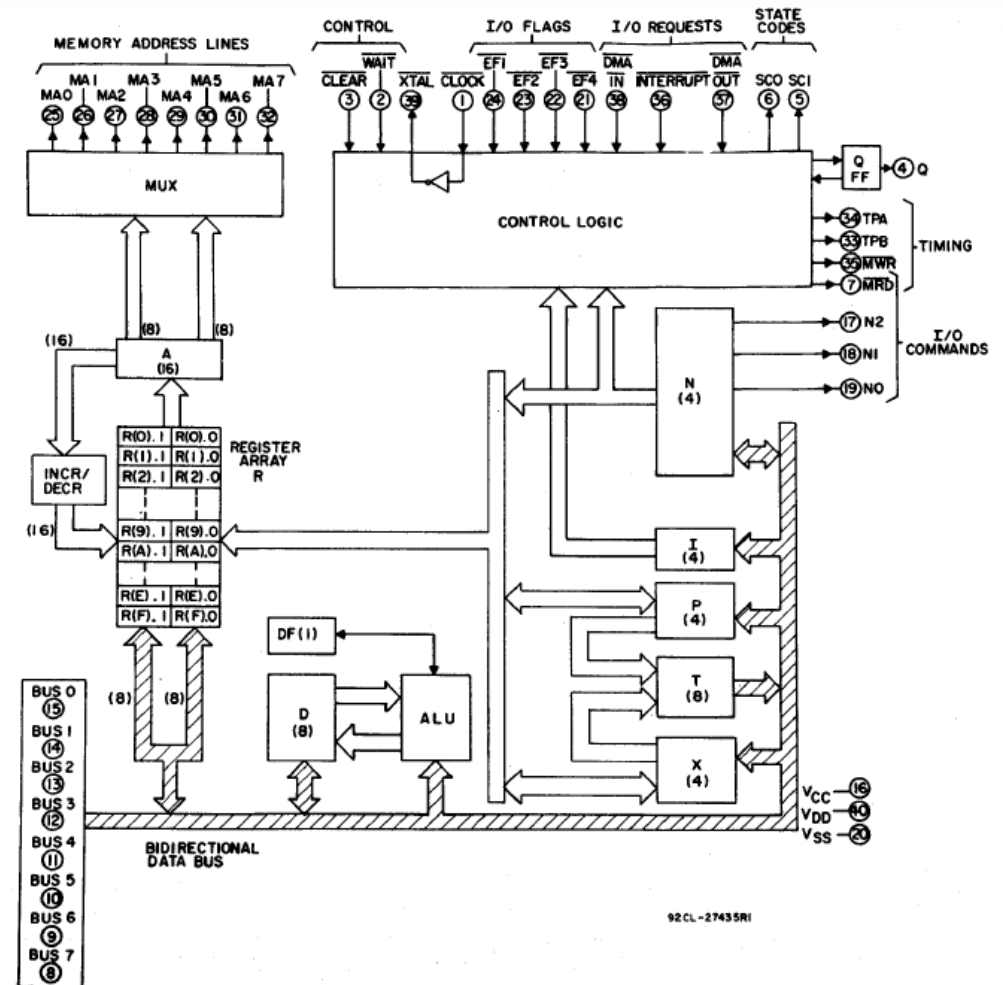


Fig. 14 – CDP1802 block diagram.



Chip-8: virtueller Prozessor

- 16 x 8-Bit-Register V0..VF
- 16-Bit-Index-/Adressregister I
- 8-Bit-Register Delay Timer DT
- 8-Bit-Register Sound Timer ST

- (16-Bit Stackregister SP)
- (16-Bit Program Counter PC)



Befehlssatz 1

Instruction	Operation
1MMM	Go to 0MMM
BMMM	Go to 0MMM + VO
2MMM	Do subroutine at 0MMM (must end with 00EE)
00EE	Return from subroutine
3XKK	Skip next instruction if $VX = KK$
4XKK	Skip next instruction if $VX \neq KK$
5XY0	Skip next instruction if $VX = VY$
9XY0	Skip next instruction if $VX \neq VY$
EX9E	Skip next instruction if $VX = \text{Hex key (LSD)}$
EXA1	Skip next instruction if $VX \neq \text{Hex key (LSD)}$
6XKK	Let $VX = KK$
CXKK	Let $VX = \text{Random Byte (KK = Mask)}$
7XKK	Let $VX = VX + KK$
8XY0	Let $VX = VY$
8XY1	Let $VX = VX/VY$ (VF changed)
8XY2	Let $VX = VX \& VY$ (VF changed)
8XY4	Let $VX = VX + VY$ (VF = 00 if $VX + VY \leq FF$, VF = 01 if $VX + VY > FF$)
8XY5	Let $VX = VX - VY$ (VF = 00 if $VX < VY$, VF = 01 if $VX \geq VY$)



Befehlssatz 2

FX07	Let VX = current timer value
FX0A	Let VX = hex key digit (waits for any key pressed)
FX15	Set timer = VX (01 = 1/60 second)
FX18	Set tone duration = VX (01 = 1/60 second)
AMMM	Let I = 0MMM
FX1E	Let I = I + VX
FX29	Let I = 5-byte display pattern for LSD of VX
FX33	Let MI = 3-decimal digit equivalent of VX (I unchanged)
FX55	Let MI = VO : VX (I = I + X + 1)
FX65	Let VO : VX = MI (I = I + X + 1)
00E0	Erase display (all 0's)
DXYN	Show n-byte MI pattern at VX-VY coordinates. I unchanged. MI pattern is combined with existing display via EXCLUSIVE-OR function. VF = 01 if a 1 in MI pattern matches 1 in existing display.
0MMM	Do machine language subroutine at 0MMM (subroutine must end with D4 byte)



Grafik

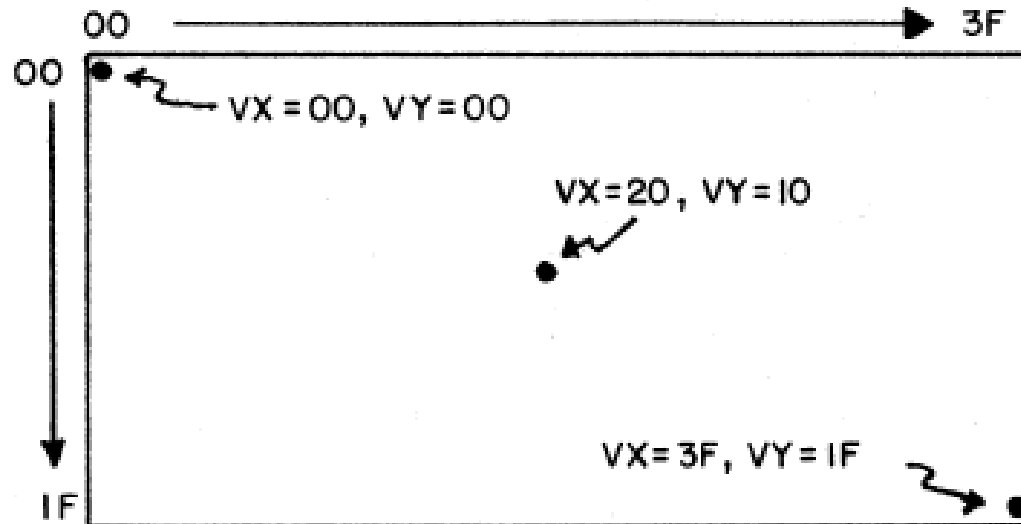
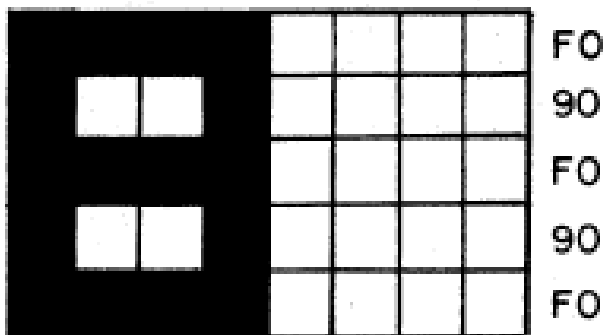


Fig. 1 – Display screen coordinate structure.



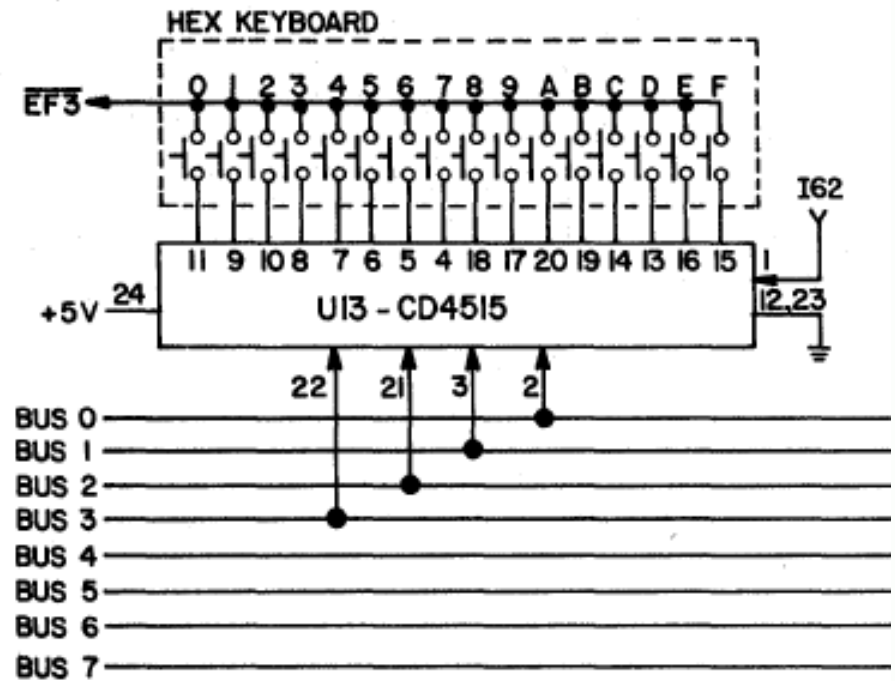
```

0200 A20A I=020A
0202 6100 V1=00
0204 6200 V2=00
0206 D125 SHOW 5MI@V1V2
0208 1208 GO 0208
020A F090
020C F090
020E F000
  
```

Fig. 2 – Pattern of bits forming digit 8.



Tastatur





Chip-8: Interpreter

- coole Software, nur 500 Byte, davon 1/3 für Sprite-Befehl DXYN !
- viele speicherplatzsparende Programmkniffe
 - -> Fx-Befehle
 - -> Bitmuster der Hexziffern



Bitmuster

HEX	ROM	BYTES	BITS
DIGIT	ADDRESS	BYTE	7 6 5 4 3 2 1 0
E-	8110	F0	█
	11	80	█
F-	8112	F0	█
	13	80	█
C-	8114	F0	█
	15	80	█
	16	80	█
	17	80	█
B-	8118	F0	█
	19	50	█
	1A	70	█
	1B	50	█
D-	811C	F0	█
	1D	50	█
	1E	50	█
	1F	50	█
5-	8120	F0	█
	21	80	█
2-	8122	F0	█
	23	10	█
6-	8124	F0	█
	25	80	█
8-	8126	F0	█
	27	90	█
9-	8128	F0	█
	29	90	█
3-	812A	F0	█
	2B	10	█
	2C	F0	█
	2D	10	█
A-	812E	F0	█

A-	812E	F0	█
	2F	90	█
0-	8130	F0	█
	31	90	█
	32	90	█
	33	90	█
7-	8134	F0	█
	35	10	█
	36	10	█
	37	10	█
1-	8139	60	█
	3A	20	█
	3B	20	█
	3C	20	█
4-	813E	A0	█
	3F	A0	█
	40	F0	█
	41	20	█
	42	20	█

- nur 50 Byte
- normalerweise 16x5 = 80 Byte nötig



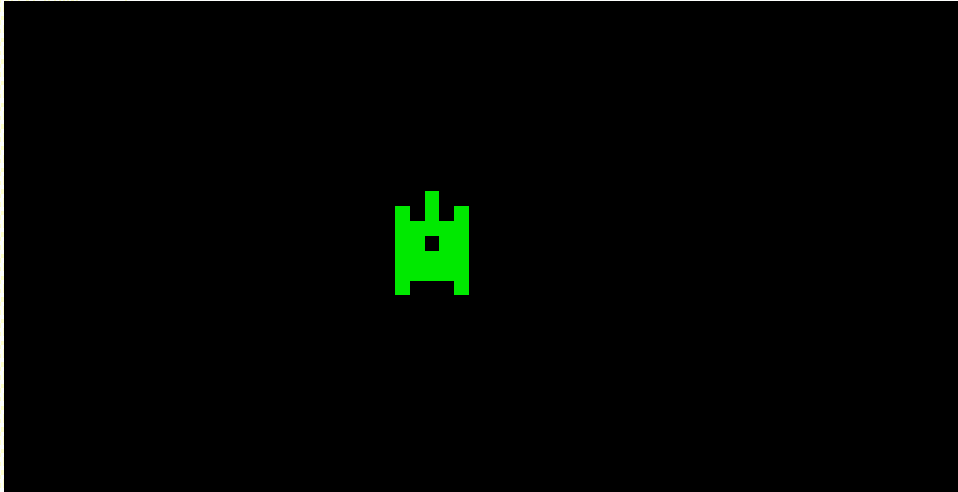
Beispiele

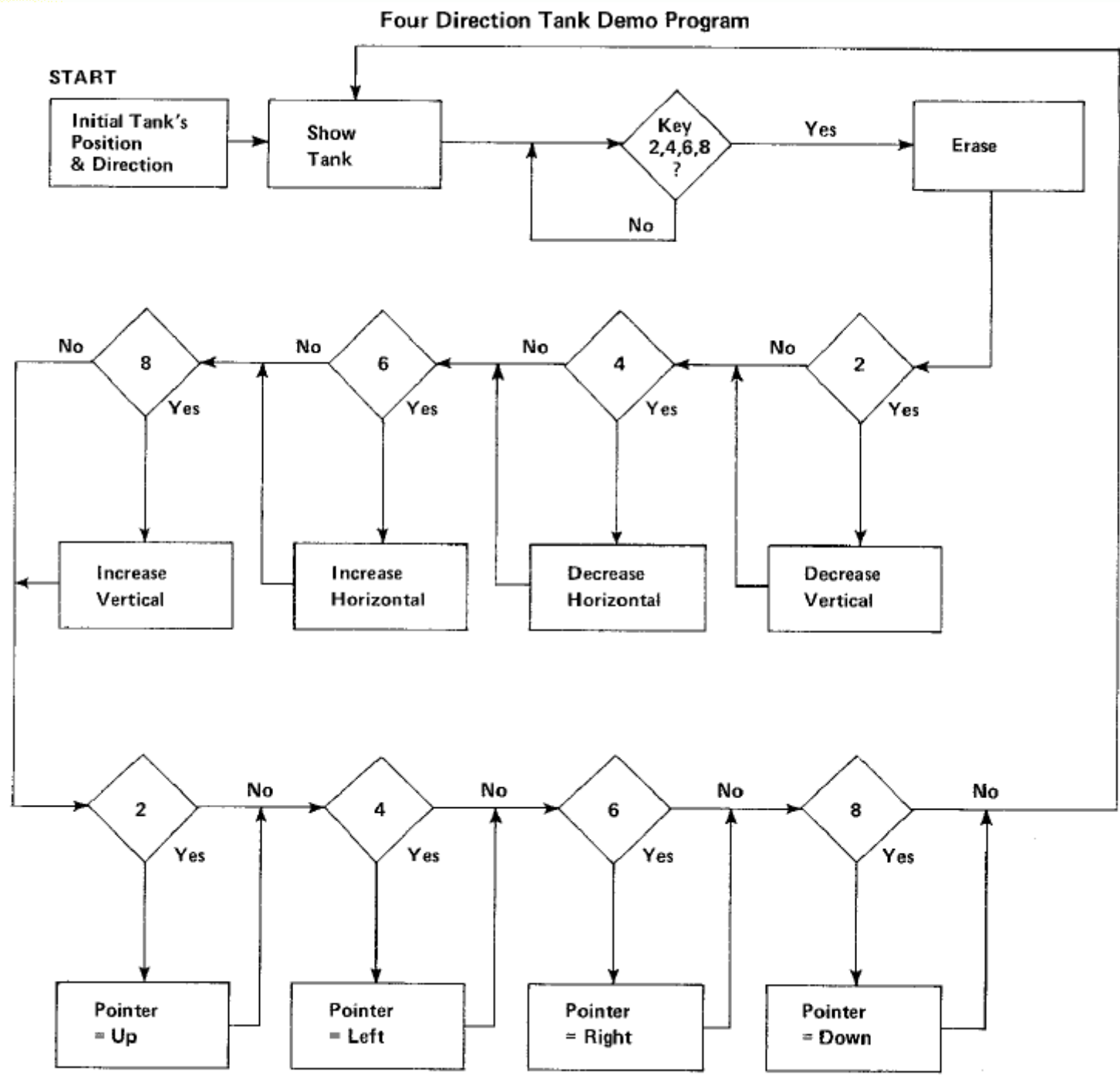
- sehr kompakte Code, z.B.
Pong 246 Bytes,
Breakout 232 Bytes



Panzer fahren

- ein Panzer soll mit dem Tasten 2-4-6-8 gesteuert werden

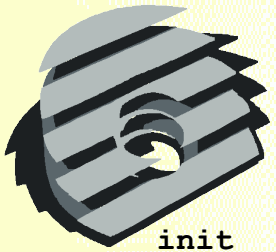






Panzer Sprites

```
51/      240 :                               org  240h
52/      240 :
53/      240 : 10          s_up      db   00010000b      ; ...#....
54/      241 : 54          db   01010100b      ; .#.#.#..
55/      242 : 7C          db   01111100b      ; .#####..
56/      243 : 6C          db   01101100b      ; .##.##..
57/      244 : 7C          db   01111100b      ; .#####..
58/      245 : 7C          db   01111100b      ; .#####..
59/      246 :
60/      246 : 44          s_down     db   01000100b      ; .#...#..
61/      247 : 7C          db   01111100b      ; .#####..
62/      248 : 7C          db   01111100b      ; .#####..
63/      249 : 6C          db   01101100b      ; .##.##..
64/      24A : 7C          db   01111100b      ; .#####..
65/      24B : 54          db   01010100b      ; .#.#.#..
66/      24C : 10          db   00010000b      ; ...#....
67/      24D :
68/      24D : 00          s_right    db   00000000b      ; .....
69/      24E : FC          db   11111100b      ; #####..
70/      24F : 78          db   01111000b      ; .####...
71/      250 : 6E          db   01101110b      ; .##.###.
72/      251 : 78          db   01111000b      ; .####...
73/      252 : FC          db   11111100b      ; #####..
74/      253 :
75/      253 : 00          s_left     db   00000000b      ; .....
76/      254 : 3F          db   00111111b      ; ..#####
77/      255 : 1E          db   00011110b      ; ...####.
78/      256 : 76          db   01110110b      ; .###.##.
79/      257 : 1E          db   00011110b      ; ...####.
80/      258 : 3F          db   00111111b      ; ..#####
81/      259 : 00          db   00000000b      ; .....
```

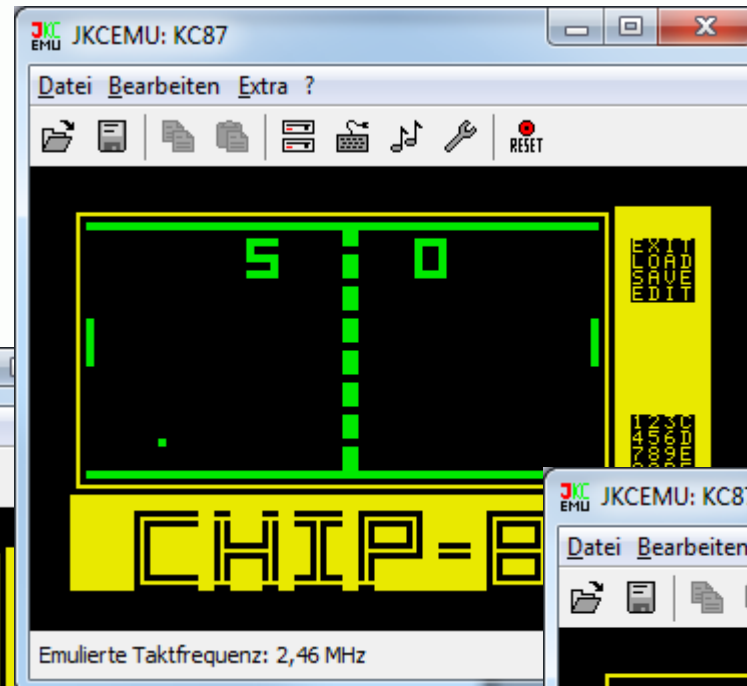
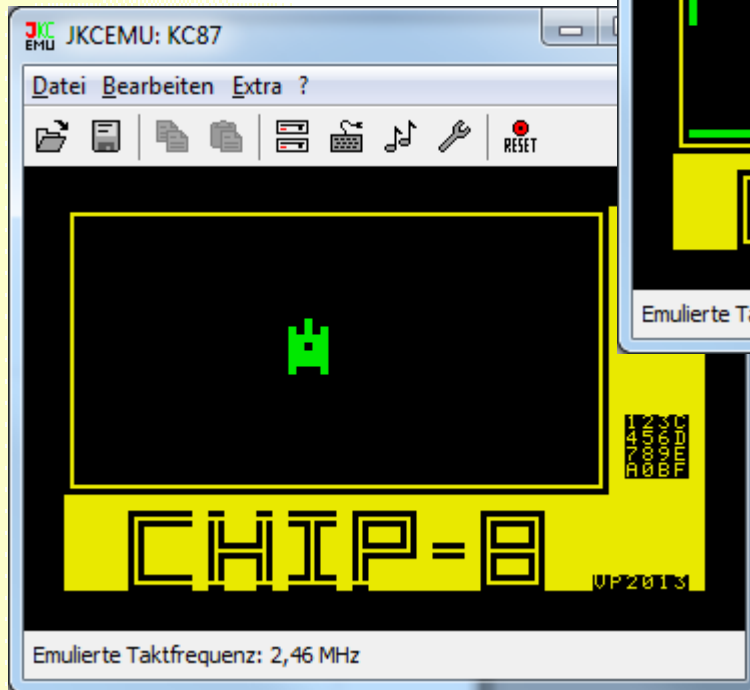


Panzer Code

```
org      200h
init     dw      06120h      ; V1=20      initialize
         dw      06210h      ; V2=10
         dw      0a240h      ; I=240
show     dw      0d127h      ; SHOW 7MI@V1V2      show tank
key      dw      06002h      ; V0=2      key 2?
key1     dw      0e0a1h      ; SKIP;V0 NE KEY      skip next if key ne V0
key2     dw      01216h      ; GO 216      erase, jp if key pressed
         dw      07002h      ; V0+2      next key (2-4-6-8)
         dw      0300ah      ; SKIP;V0 EQ 0A      skip next if last key
         dw      0120ah      ; GO 20A      key2, loop back to check next key
         dw      01208h      ; GO 208      key1, loop back to recheck next key
;erase tank
erase    dw      0d127h      ; SHOW 7MI@V1V2      erase tank
;change x or y
         dw      04002h      ; SKIP;V0 NE 02      key 2?
         dw      072ffh      ; V2+FF      = V2-1; move up
         dw      04004h      ; SKIP;V0 NE 04      key 4?
         dw      071ffh      ; V1+FF      = V1-1; move left
         dw      04006h      ; SKIP;V0 NE 06      key 6?
         dw      07101h      ; V1+01      move right
         dw      04008h      ; SKIP;V0 NE 08      key 8?
         dw      07201h      ; V2+01      move down
; set pointer
         dw      04002h      ; SKIP;V0 NE 02      key 2?
         dw      0a240h      ; I=240      s_up
         dw      04004h      ; SKIP;V0 NE 04      key 4?
         dw      0a253h      ; I=253      s_left
         dw      04006h      ; SKIP;V0 NE 06      key 6?
         dw      0a24dh      ; I=24D      s_right
         dw      04008h      ; SKIP;V0 NE 08      key 8?
         dw      0a246h      ; I=246      s_down
; jump to show
         dw      01206h      ; GO 206      show
```



Z9001





homecomputer-ddr.de.vu

Danke!