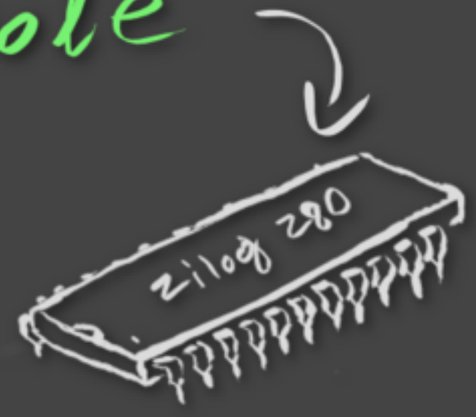
Ο σπιτικός Χειροποίητος αφράτος gluttem-free Z80 υπολογιστής μου

Γιάννης Τσιομπίκας
<nuclear@member.fsf.org>

Γιατί Z80 ?

- Διαθέσιμος σε απλό through-hole package
- Απλή τροφοδοσία 5v
- Απλό bus signalling



- Χωρίς κάτω όριο στη συχνότητα του clock, κάνει απλό το debugging.
- Ενσωματωμένος μηχανισμός για dynamic RAM refresh, κάνει απλή την χρήση DRAM
- <insert joke>

Z80 Instruction Set Architecture

Registers

- a — accumulator
- b, c
- d, e
- f — flags
- h, l
- ix, iy — index
- i — interrupt
- r — refresh
- sp — stack pointer
- pc — program counter

(kaa a' b' c' d' e' f' h' l')

Παραδείγματα εντολών

8 bit load/store

```
ld a, 0
ld a, (hl)
ld (ix+5), 42
```

16 bit load/store

```
ld bc, 666
ld sp, hl
push af
```

block transfers

```
ldi / ldir
ldd / lddr
exx
```

jumps

```
jp foo
jp nz, foo
jr c, e
call foo
ret nc
rst 28h
```

I/O

```
in a, (255)
out (c), a
```

8bit arithmetic / logic

```
add a, b
add a, (iy+3)
sub (hl)
```

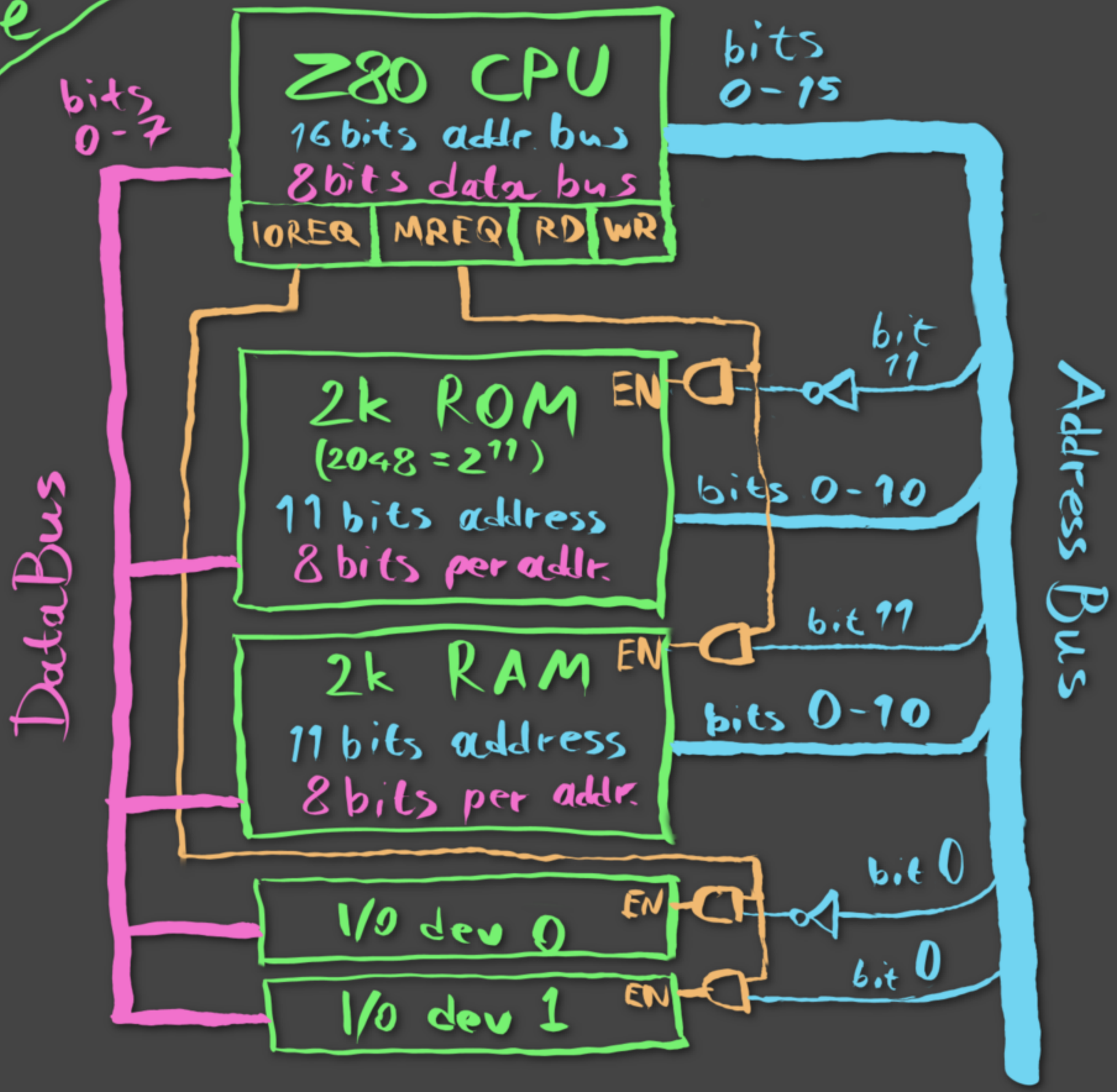
16 bit arithmetic / logic

```
adc hl, de
dec sp
```

rotate / shift

```
rlc a
sla (hl)
```

Computer architecture



My Computer architecture

Z80 CPU
16 bits addr bus
8 bits data bus
IOREQ | MREQ | RD | WR

bits 0-7

bits 0-15

2k ROM
(2048 = 2^{11})
11 bits address
8 bits per addr.
EN

bit 11

bits 0-10

2k RAM
11 bits address
8 bits per addr.
EN

bit 11

bits 0-10

I/O dev 0
EN

bit 0

I/O dev 1
EN

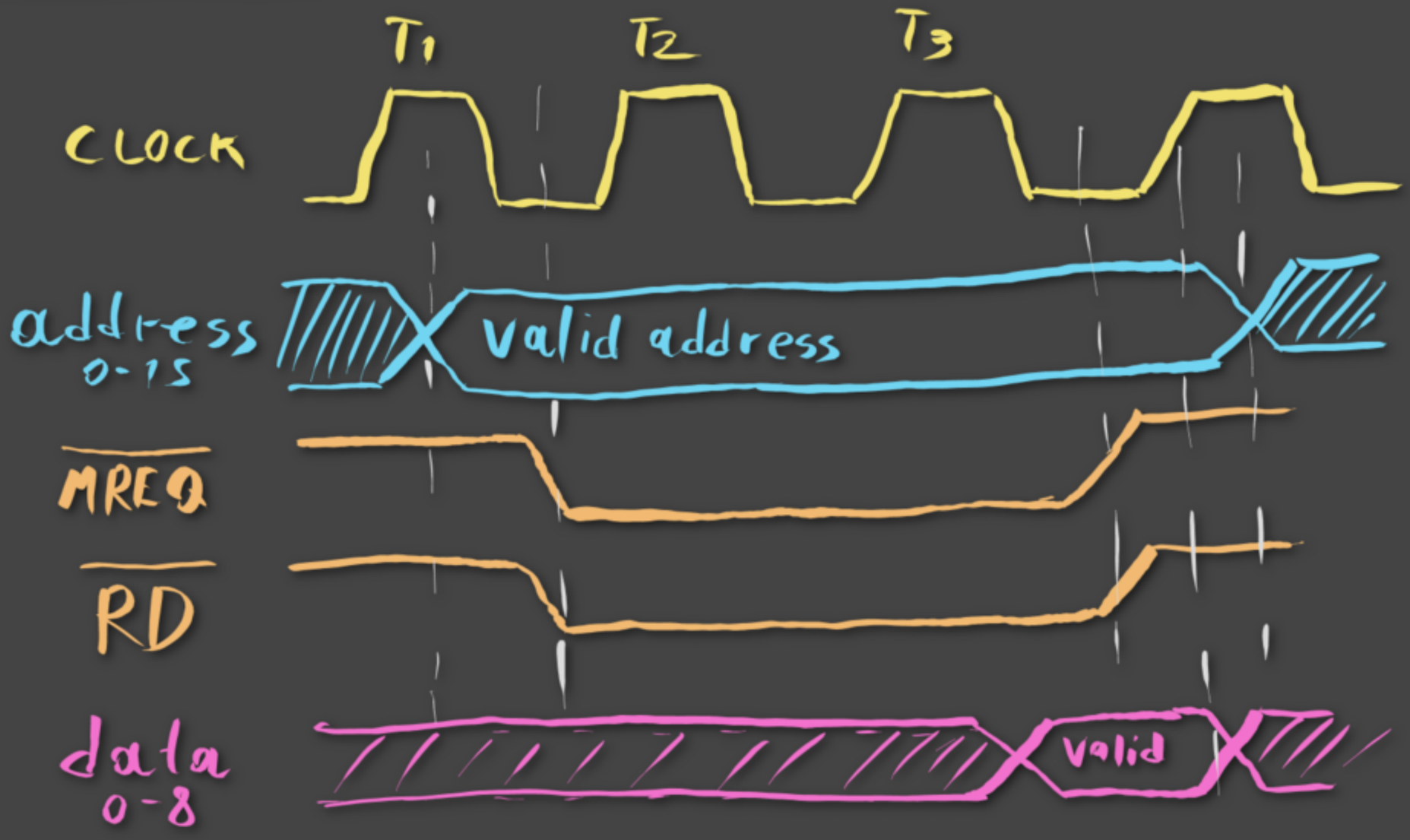
bit 0

DataBus

Address Bus

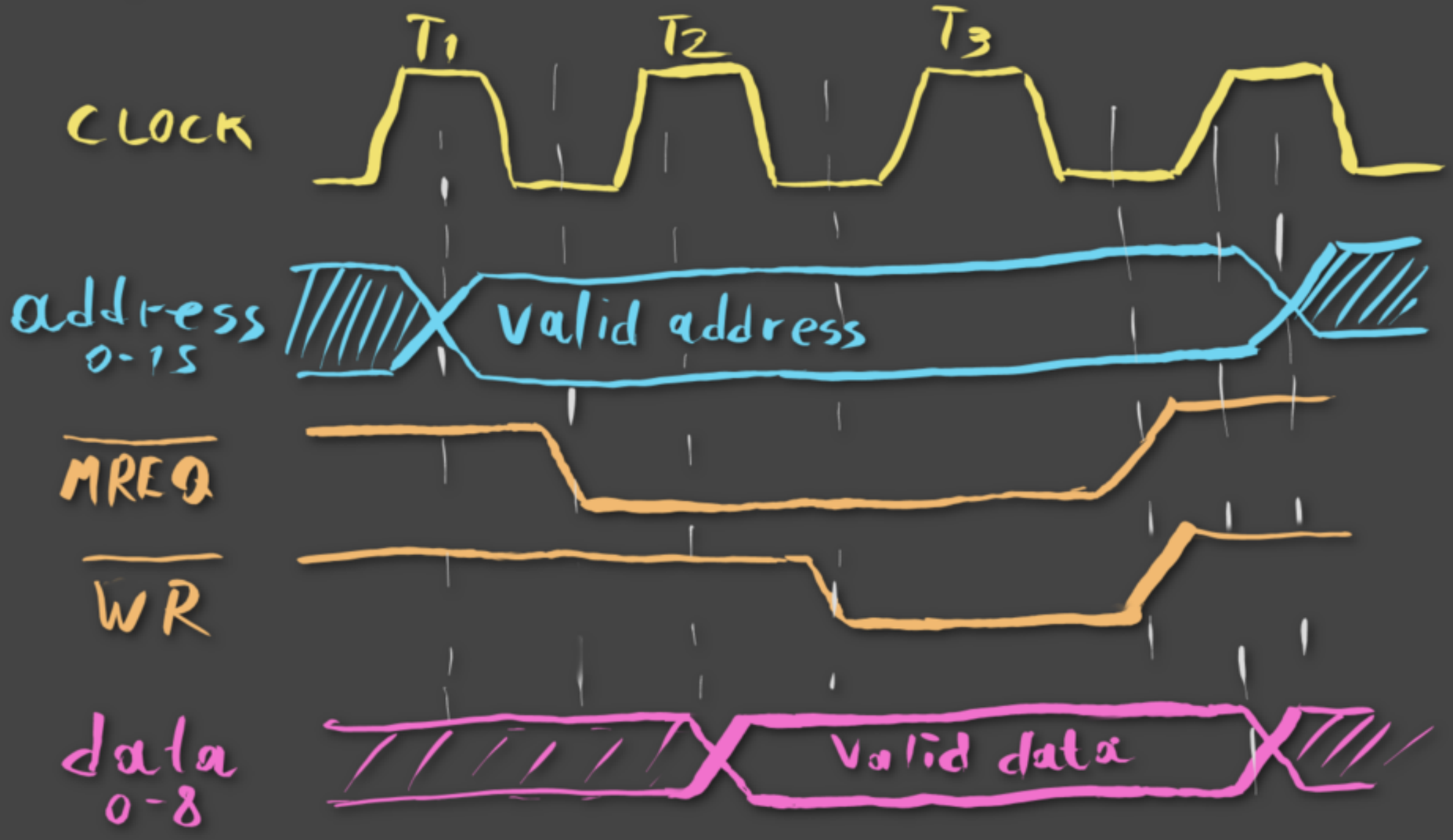
Z80 Memory Read

1. διεύθυνση στο address bus
2. ενεργοποίηση RD και MREQ
3. αναγνωση του data bus



Z80 Memory Write

1. διεύθυνση στο address bus
2. ενεργοποίηση του MREQ
3. data στο data bus
4. ενεργοποίηση του WR



Παραλειπόμενα

- Στο instruction fetch ενεργοποιείται και το M1.

- Στα I/O read/write cycles, ενεργοποιείται το IOREQ αντί για το MREQ.

THIS AREA HAS BEEN INTENTIONALLY LEFT BLANK

Schematic

Voltage Regulator

ROM

address Bus

Data Bus

automatic clock

Manual Clock

Reset

Z80 CPU

Output device

S3 D1 J1 IC6 7805T VI VO GND VCC C9 1uF C8 0.33uF GND

C3 C4 C5 100n 100n 100n GND GND GND

R14 10k R13 1k IC5 TRE VCC+ /RES DIS OUT CON TRI GND NE555 C7 100n C6 100n GND

SU1 CLK

S1 R10 18k R1 18k C1 1uF IC2E 74HC14 GND

S2 R12 18k R11 18k C2 1uF IC2B IC2C 74HC14 74HC14 RST GND

A[0..15] IC3 28C16 A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 CE OE WE D0 D1 D2 D3 D4 D5 D6 D7

LED24 LED23 R9 1k R8 1k R2 1k LED9

IC1 M1 MREQ IORQ RD WR RFSH HALT WAIT INT NMI RESET BUSRQ BUSAK CLK GND VCC Z84C0006

D[0..7]

R3-R7 1k LED18 LED19 LED20 LED21 LED22 GND

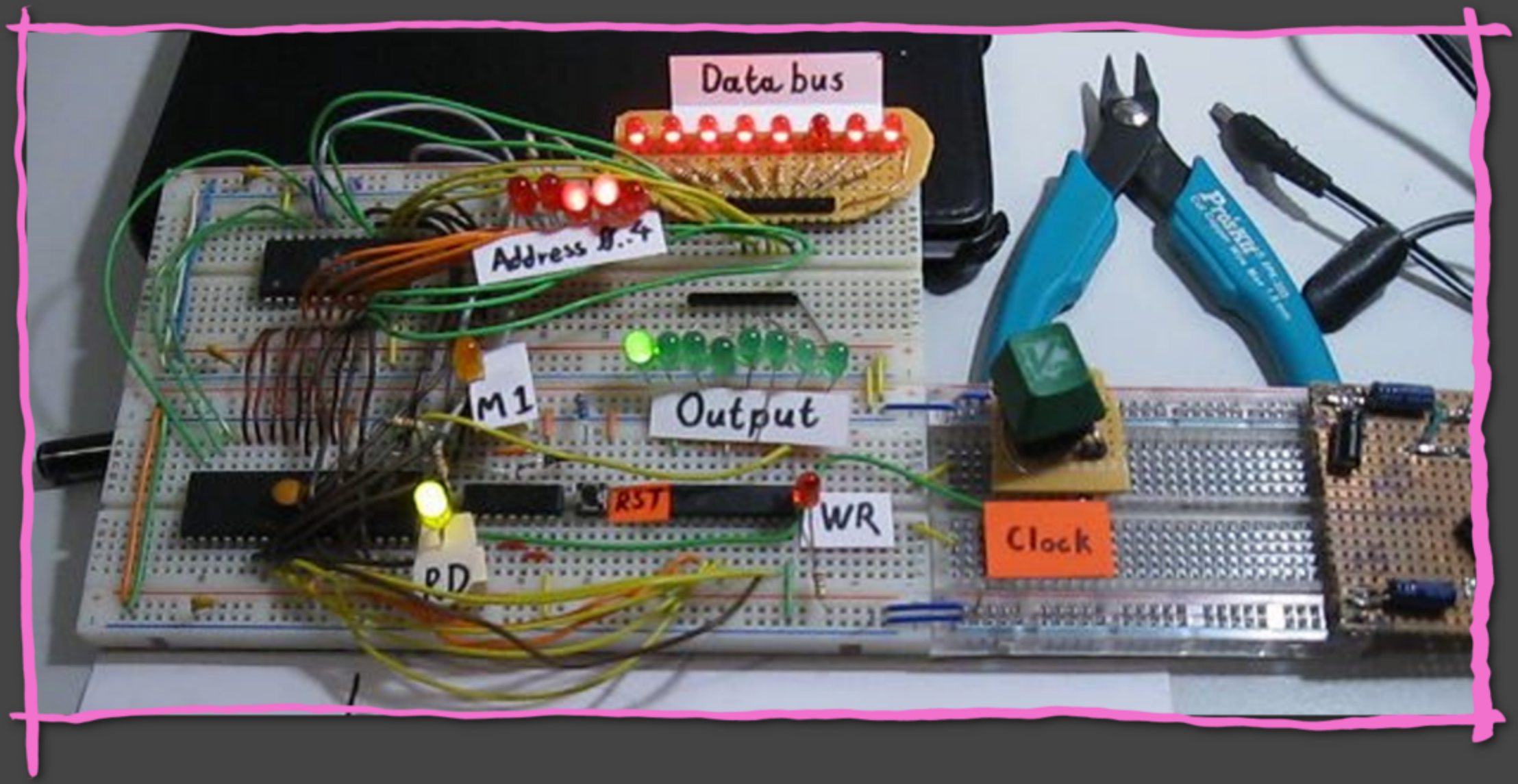
IC2D IC2A IC2F GND

LED10-LED17 RN2 1k GND

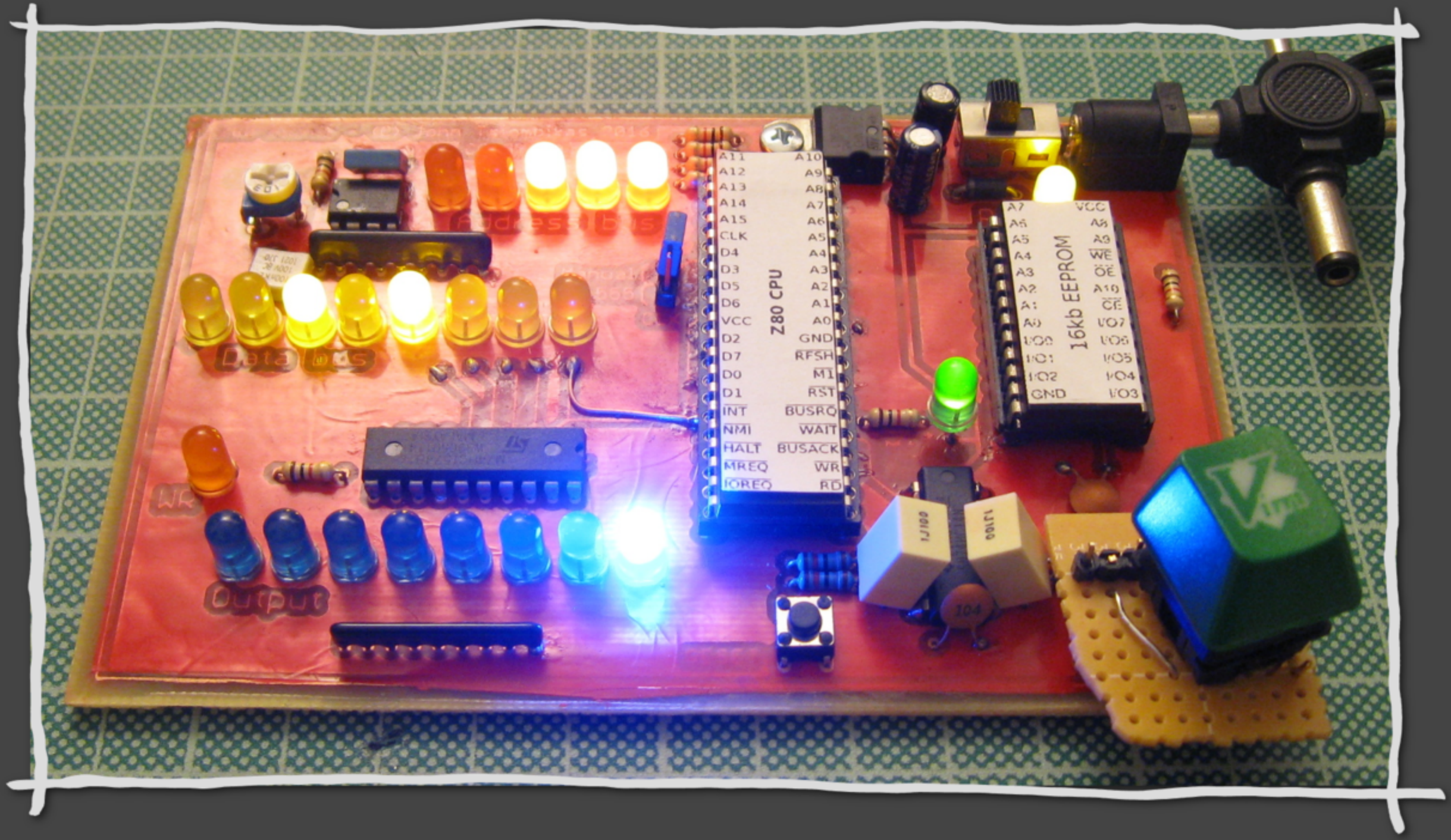
IC4 74HC574N CLK OC GND

LED1-LED8 RN1 330 GND

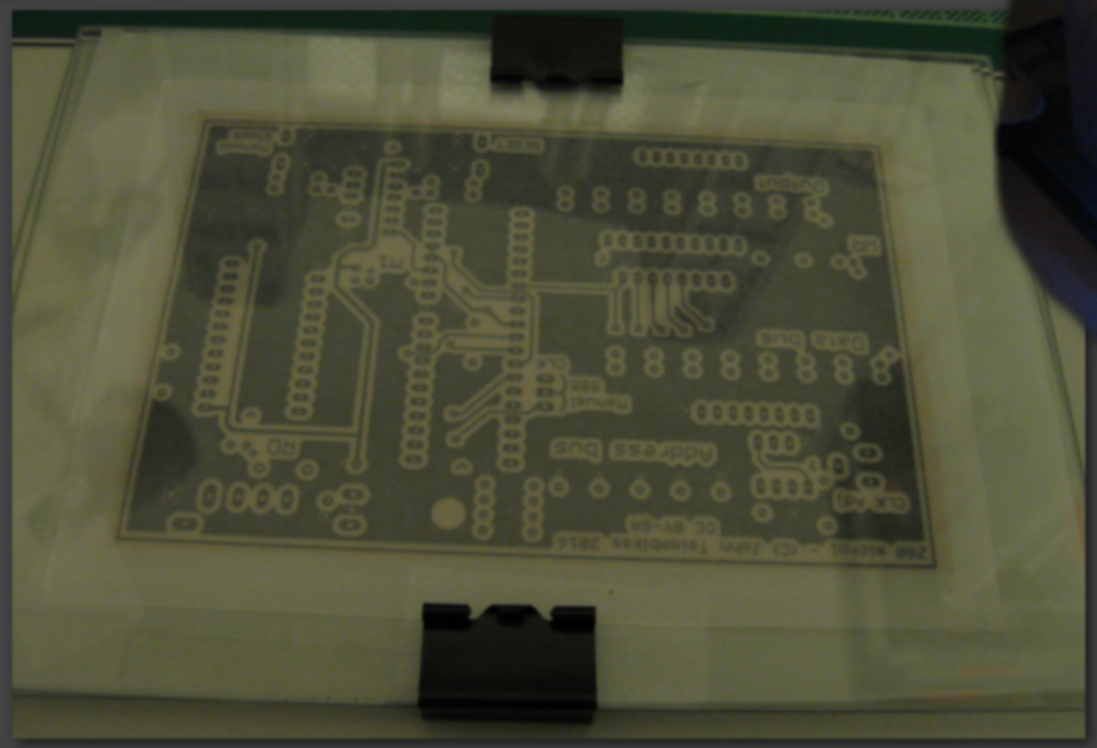
Breadboard Prototype

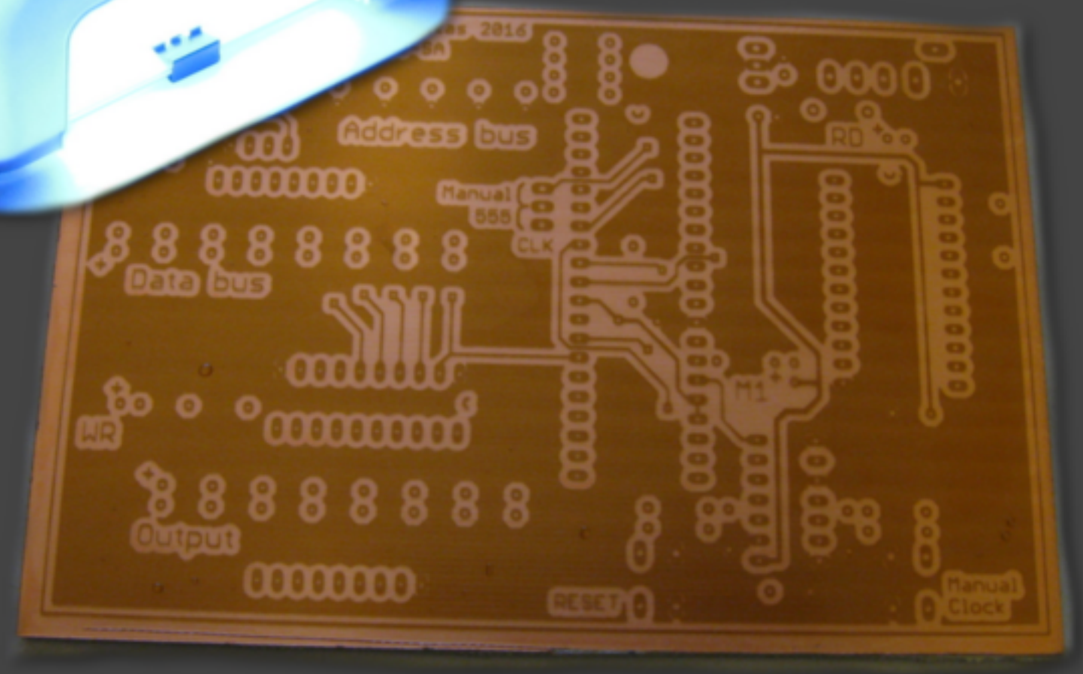


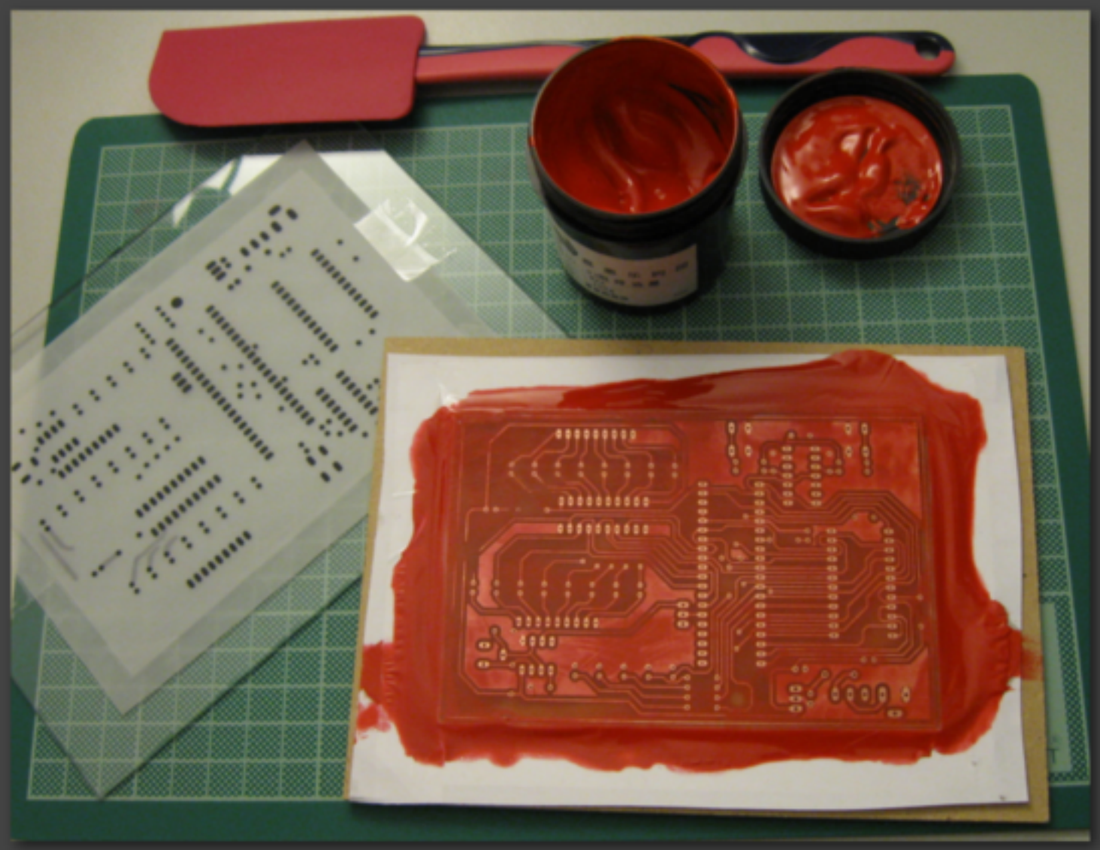
PCB

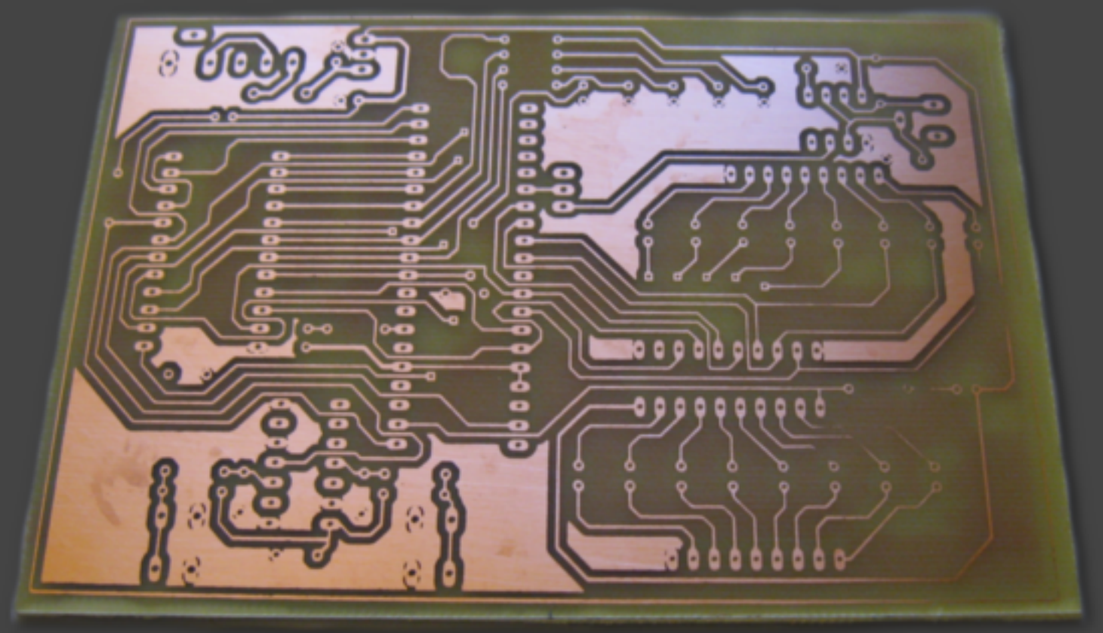

Z80 CPU
A11
A12
A13
A14
A15
CLK
D4
D3
D5
D6
VCC
D2
D7
D0
D1
INT
NMI
HALT
MREQ
IOREQ
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
GND
RFSH
M1
RST
BUSRQ
WAIT
BUSACK
WR
RD
16kb EEPROM
VCC
A8
A9
WE
OE
A10
CE
I/O7
I/O6
I/O5
I/O4
I/O3
GND
I/O2
I/O1
I/O0
A0
Data bus
Output
WR
74LS00
74LS00
104

PCB Construction









Test program

```
        org 0
        ld a, 1
left:   out (255), a
        rlca
        cp  128
        jr  z, right
        jr  left
right:  out (255), a
        rrca
        cp  1
        jr  z, left
        jr  right
```

Demo

address data instruction

live video box

Demo

address	data	instruction
0	3e	ld a, N

Demo

address	data	instruction
0	3e	ld a, 1
1	01	

Demo

address	data	instruction
0	3e	ld a, 1
1	01	
2	d3	out (N), a

Demo

address	data	instruction
0	3e	ld a, 1
1	01	
2	d3	out (255), a
3	ff	

Demo

address	data	instruction
0	3e	ld a, 1
1	01	
2	d3	out (255), a
3	ff	
4	07	rlca

Demo

address	data	instruction
0	3e	ld a, 1
1	01	
2	d3	out (255), a
3	ff	
4	07	rlca
5	fe	cp N

Demo

address	data	instruction
0	3e	ld a, 1
1	01	
2	d3	out (255), a
3	ff	
4	07	rlca
5	fe	cp 128
6	80	

Demo

address	data	instruction
0	3e	ld a, 1
1	01	
2	d3	out (255), a
3	ff	
4	07	rlca
5	fe	cp 128
6	80	
7	28	jr z, DIS

Demo

address	data	instruction
0	3e	ld a, 1
1	01	
2	d3	out (255), a
3	ff	
4	07	rlca
5	fe	cp 128
6	80	
7	28	jr z, 2
8	02	

PC = 9

9 + 2 = b

Demo

address	data	instruction
0	3e	ld a, 1
1	01	
2	d3	out (255), a
3	ff	
4	07	rlca
5	fe	cp 128
6	80	
7	28	jr z, 2
8	02	
9	18	jr DIS

Demo

Two's complement

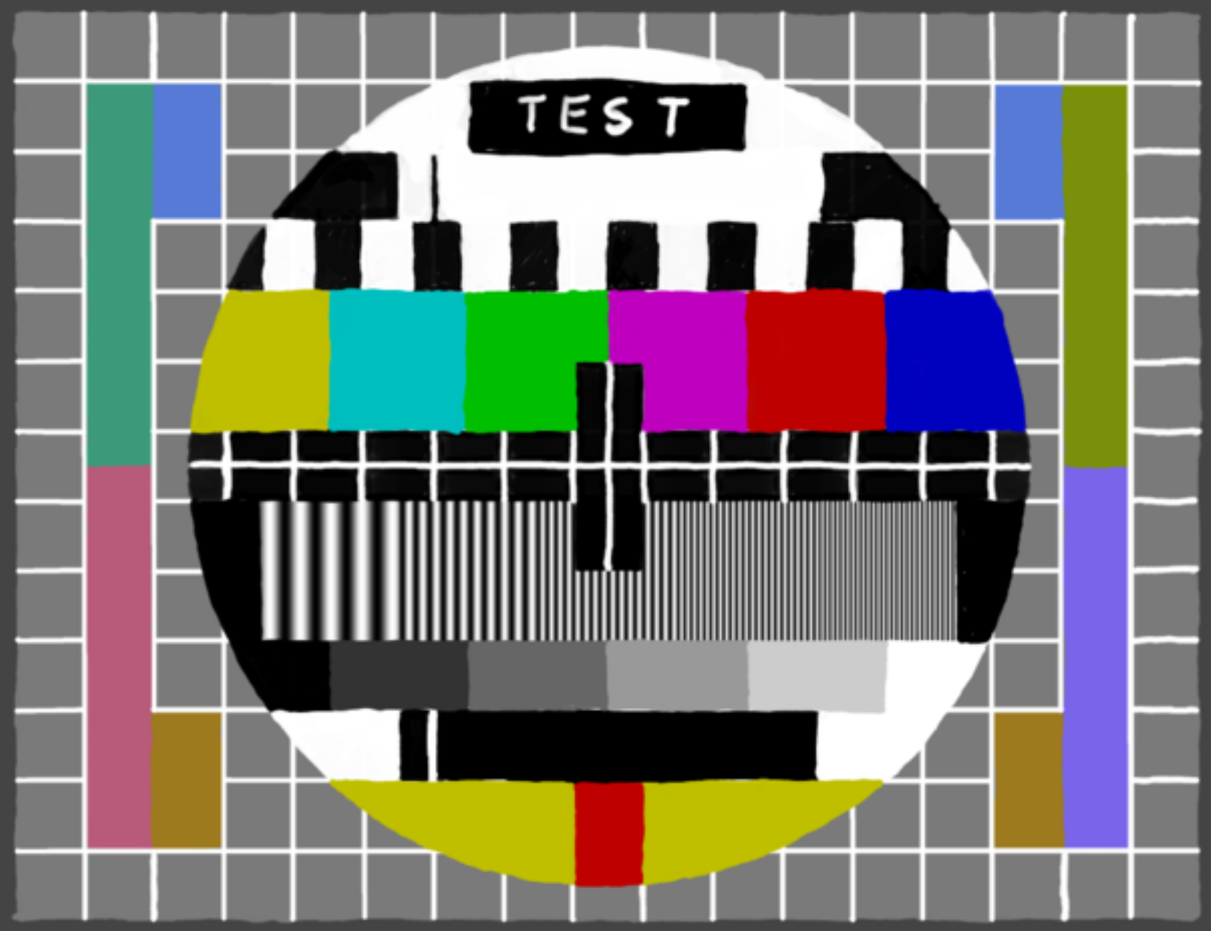
f7 → 11110111

invert → 00001000

add 1 → 00001001 → -9

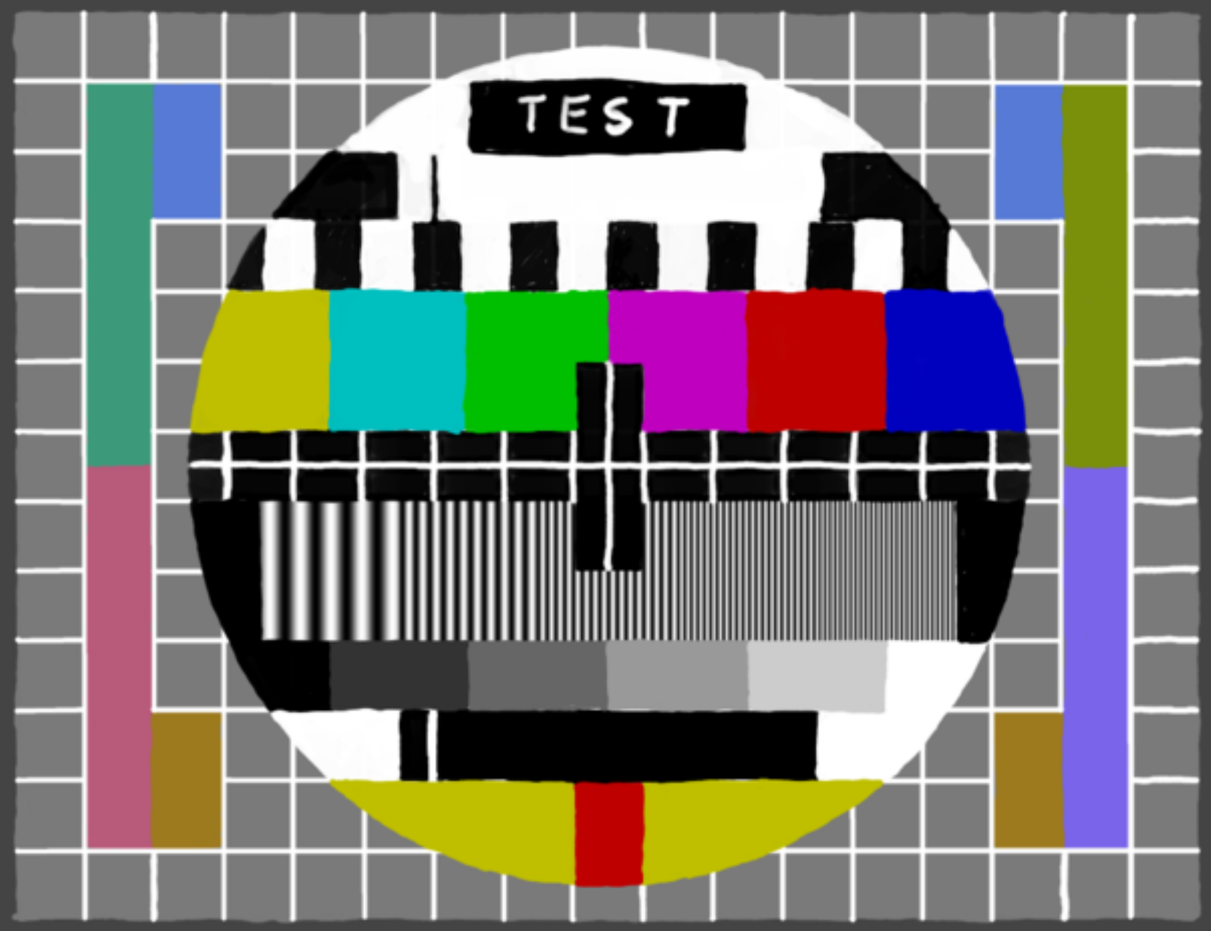
address	data	instruction
0	3e	ld a, 1
1	01	
2	d3	out (255), a
3	ff	
4	07	rlca
5	fe	cp 128
6	80	
7	28	jr z, 2
8	02	
9	18	jr -9
a	f7	

PC = b , b-9 = 2



Demo

address	data	instruction
0	3e	ld a, 1
1	01	
2	d3	out (255), a
3	ff	
4	07	rlca
5	fe	cp 128
6	80	
7	28	jr z, 2
8	02	
9	18	jr -9
a	f7	
.	.	.
.	.	.
.	.	.



Test program reminder

```
        org 0
        ld a, 1
left:   out (255), a
        rlca
        cp  128
        jr z, right
        jr left
right:  out (255), a
        rrca
        cp  1
        jr  z, left
        jr right
```

DI
HALT

http://nuclear.mutantstargoat.com/hw/z80micro

nuclear@member.fsf.org