

# Z80 Computer

## Contents

### Overview of the Z80 project

#### Design of the Z80 computer

Z80 computer components, schematic and layout

    Z80 computer components

    Z80 computer component layout

    Z80 computer schematic

    Z80 computer building steps

    Z80 computer signal measurements

    ATF22V10C programmable logic for address selection

    Simulation of the address selection logic

    ATF22V10C programming

#### Z80 test programs

### Thoughts and details about the Z80 computer

    Power decoupling

    Z80 Memory Timing

    Power on Reset

    Pull-up resistors

    Memory and I/O address decoding

        Test PLD programming

        Clock frequency divider

    Serial I/O & baud-rate generator

    Parallel port

    Floppy Disk Controller

    Hard Disk ATA controller

    SD Card interface

    Disk drive over serial interface (or parallel port) ?

### Logic Device Programming

### KiCAD as EDA tool for Z80 computer design

### Z80 Software

    Z80 Monitor and OS

    Handling Z80 MMU / overlays

    Other Z80 Programs

### Memory mapping for Z80 (and other 16 bit address CPUs)

### Z80 References

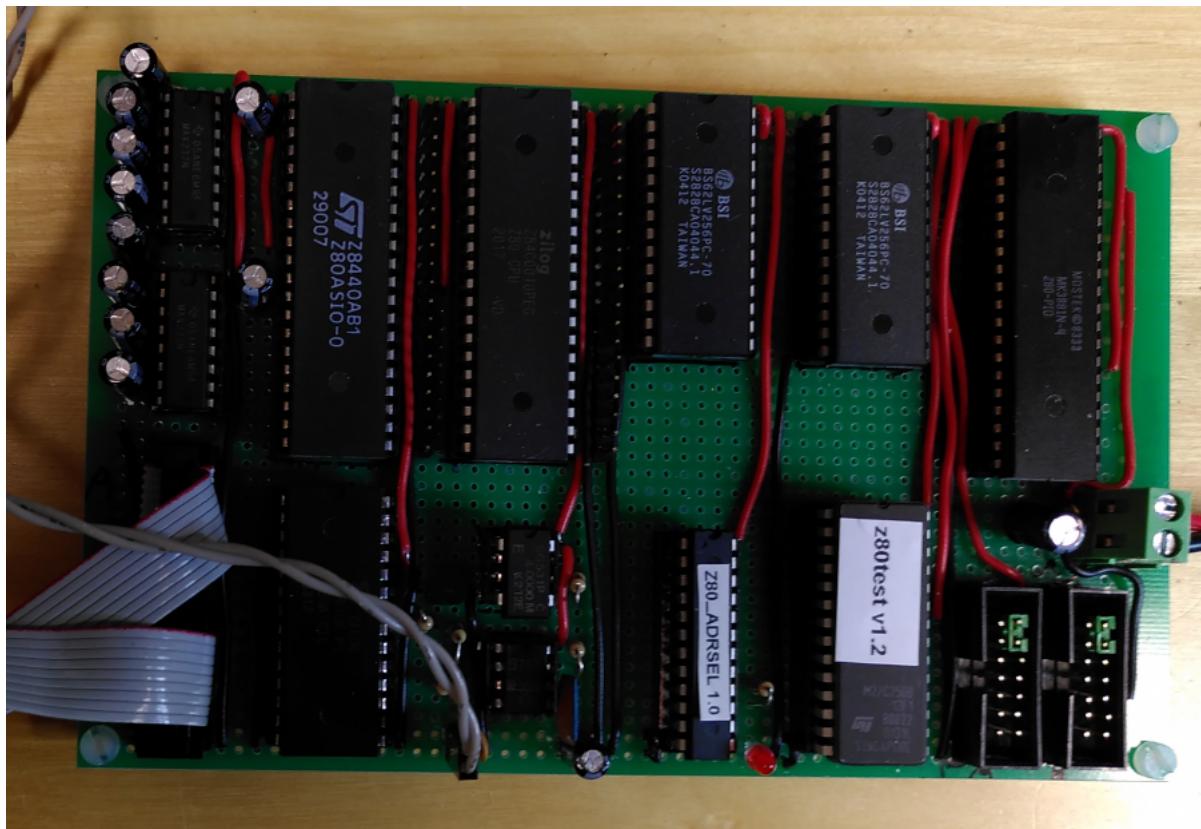
### Links to Z80 Projects

## Z80 related components

The following components are/will be available  
Components ordered from DigiKey

# Overview of the Z80 project

2021-02-12, updated 2021-05-24, 2021-06-30



The design is uploaded to GitHub: [hanske/Z80\\_Computer\\_board: A simple Z80 based computer board](https://github.com/hanske/Z80_Computer_board) ([http://github.com/hanske/Z80\\_Computer\\_board](https://github.com/hanske/Z80_Computer_board))

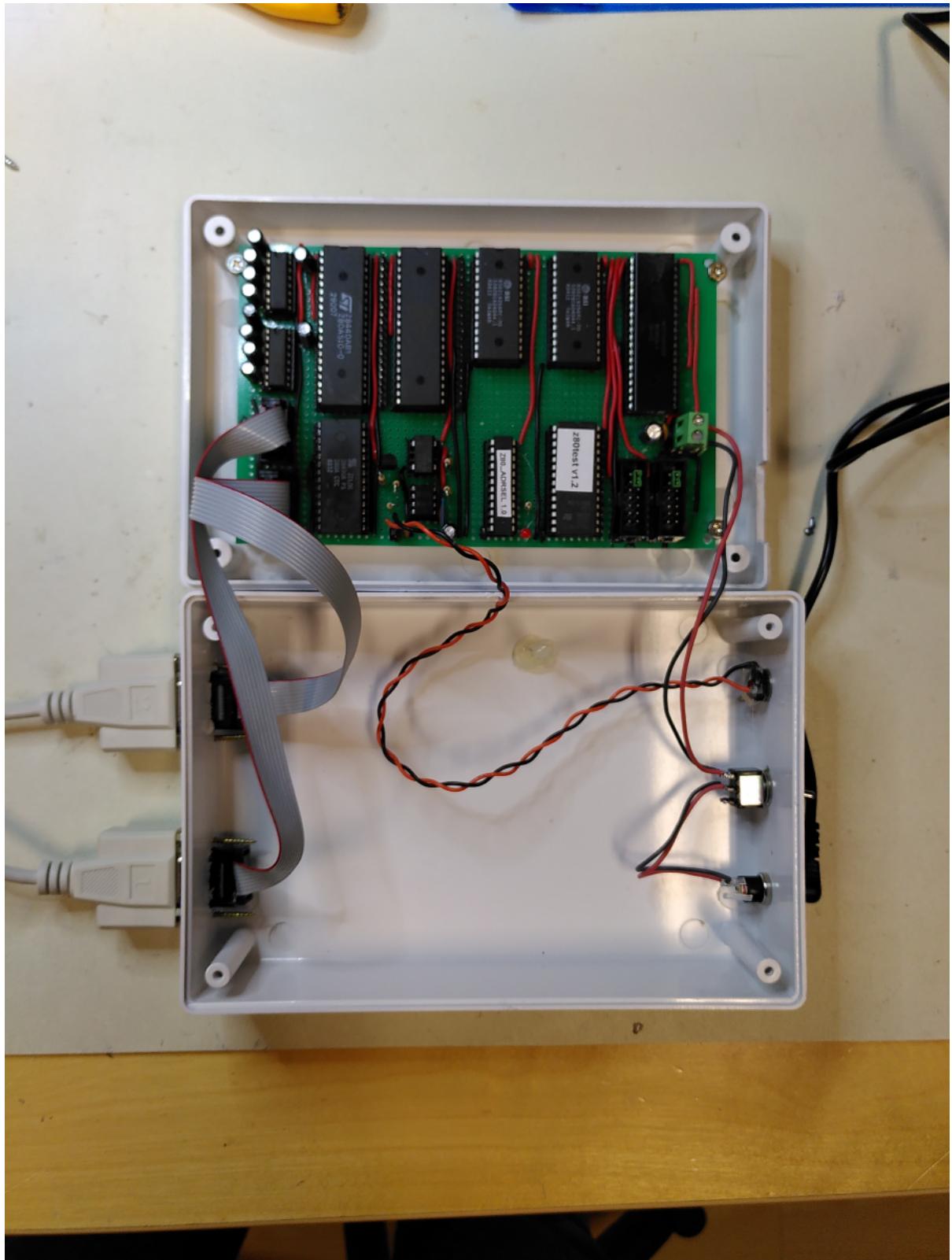
The intention is to do the following:

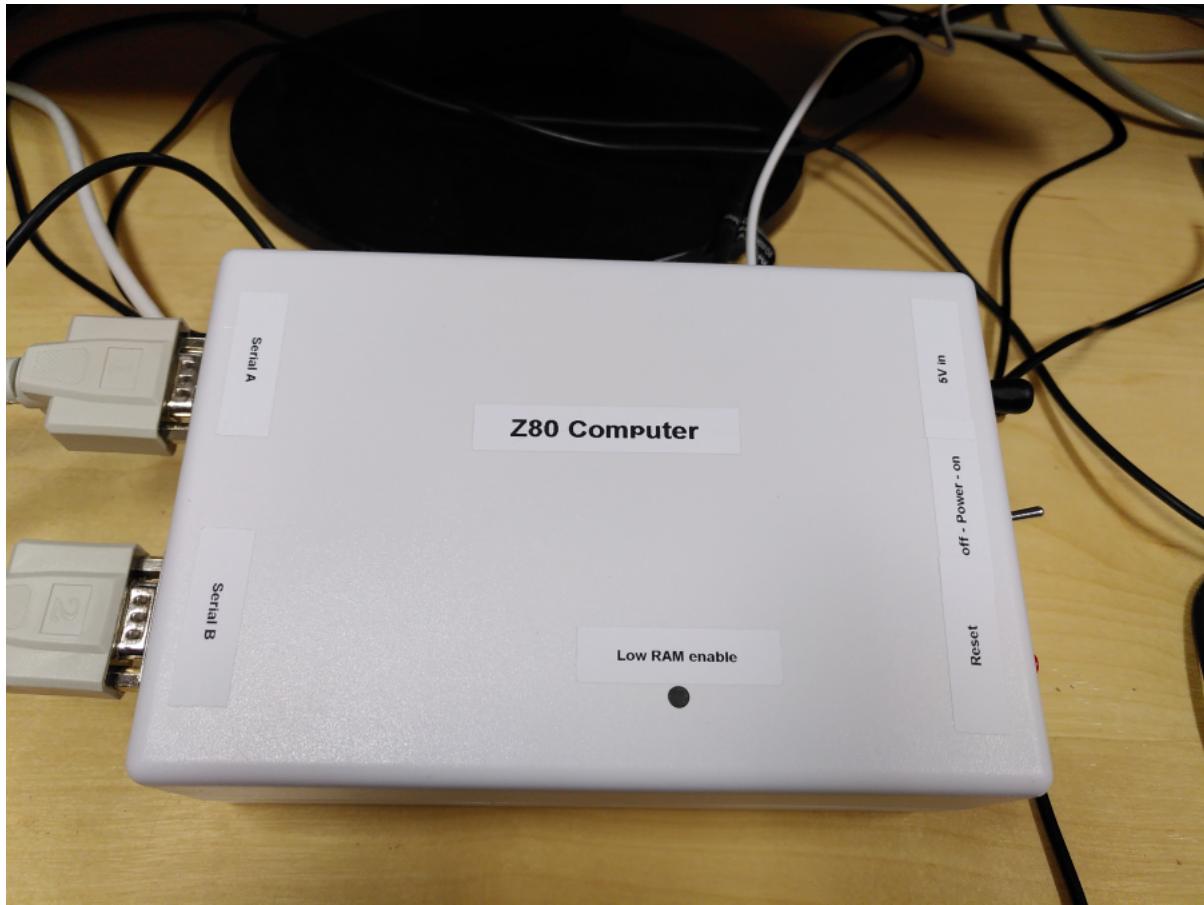
- Design a simple Z80 based computer board with the components that are available
  - [Z80\\_Microprocessor\\_Family\\_Jan90.pdf](http://www.bitsavers.org/components/st_Microelectronics/_dataBooks/Z80_Microprocessor_Family_Jan90.pdf) ([http://www.bitsavers.org/components/st\\_Microelectronics/\\_dataBooks/Z80\\_Microprocessor\\_Family\\_Jan90.pdf](http://www.bitsavers.org/components/st_Microelectronics/_dataBooks/Z80_Microprocessor_Family_Jan90.pdf))
  - [gaby.de/z80/zaks.html](http://gaby.de/z80/zaks.html) (<http://gaby.de/z80/zaks.html>)
  - Thomas Scherrer Z80-Family Official Support Page (<https://www.z80cpu.eu/mirrors/www.z80.info/>)
- The computer board should have:
  - [Z80 CPU Z80 CPU User Manual - um0080.pdf](http://www.zilog.com/docs/z80/um0080.pdf) (<http://www.zilog.com/docs/z80/um0080.pdf>)
  - EPROM memory
  - SRAM memory
    - The higher 32KB of the memory is always RAM

- the lower 32KB of memory is switched between EPROM and RAM, reset selects EPROM
- to use RAM in low memory the start routine will copy code (BIOS etc) from EPROM to high RAM
- Z80 SIO for RS-232 connection [Zilog Z80-SIO Technical Manual.pdf](http://www.hartetechnologies.com/manuals/Zilog/Zilog%20Z80-SIO%20Technical%20Manual.pdf) (<http://www.hartetechnologies.com/manuals/Zilog/Zilog%20Z80-SIO%20Technical%20Manual.pdf>)
- Z80 CTC for timing and serial channel clock [Z80 CTC - Z80 CTC.pdf](https://arcarch.xmission.com/Tech/Datasheets/Z80%20CTC.pdf) (<https://arcarch.xmission.com/Tech/Datasheets/Z80%20CTC.pdf>)
- Z80 PIO for further use [Zilog Z80-PIO Technical Manual.pdf](https://www.hartetechnologies.com/manuals/Zilog/Zilog%20Z80-PIO%20Technical%20Manual.pdf) (<https://www.hartetechnologies.com/manuals/Zilog/Zilog%20Z80-PIO%20Technical%20Manual.pdf>)
  - Could maybe be used for ATA interface?
  - or even an ISA interface to a controller that handles both FD interface and ATA interface
- Maybe Z80 DMA [Z80 DMA ps0179.pdf](http://www.zilog.com/docs/z80/ps0179.pdf) (<http://www.zilog.com/docs/z80/ps0179.pdf>)
  - Proof-of-concept: DMA controller for MOVE/XMOVE by dmcnaugh · Pull Request #74 · udo-munk/z80pack (<https://github.com/udo-munk/z80pack/pull/74/commits/78896e47e8b7d0e2e1e86380d3eaec8c3b0846d5>)
- For other necessary logical functions a GAL IC should be used: [PAL/GAL Programming](#)
- A crystal oscillator
- A power-on and reset circuit, could be based on a 555 IC
  - Maybe. maybe a floppy disk interface?
- The schematic and a simple board layout should be designed with KiCAD
- Wiring should be done with Verowire
- To this project a good C cross compiler and assembler would be nice to have: [Build a Clang-LLVM Z80 Compiler](#).
- A SIMH based simulator would also be very nice. Could be based on: [simh/AltairZ80 at master · simh/simh · GitHub](#) (<https://github.com/simh/simh/tree/master/AltairZ80>)

2021-07-26

Z80 Computer in enclosure





# Design of the Z80 computer

## Z80 computer components, schematic and layout

---

### Z80 computer components

2021-04-03

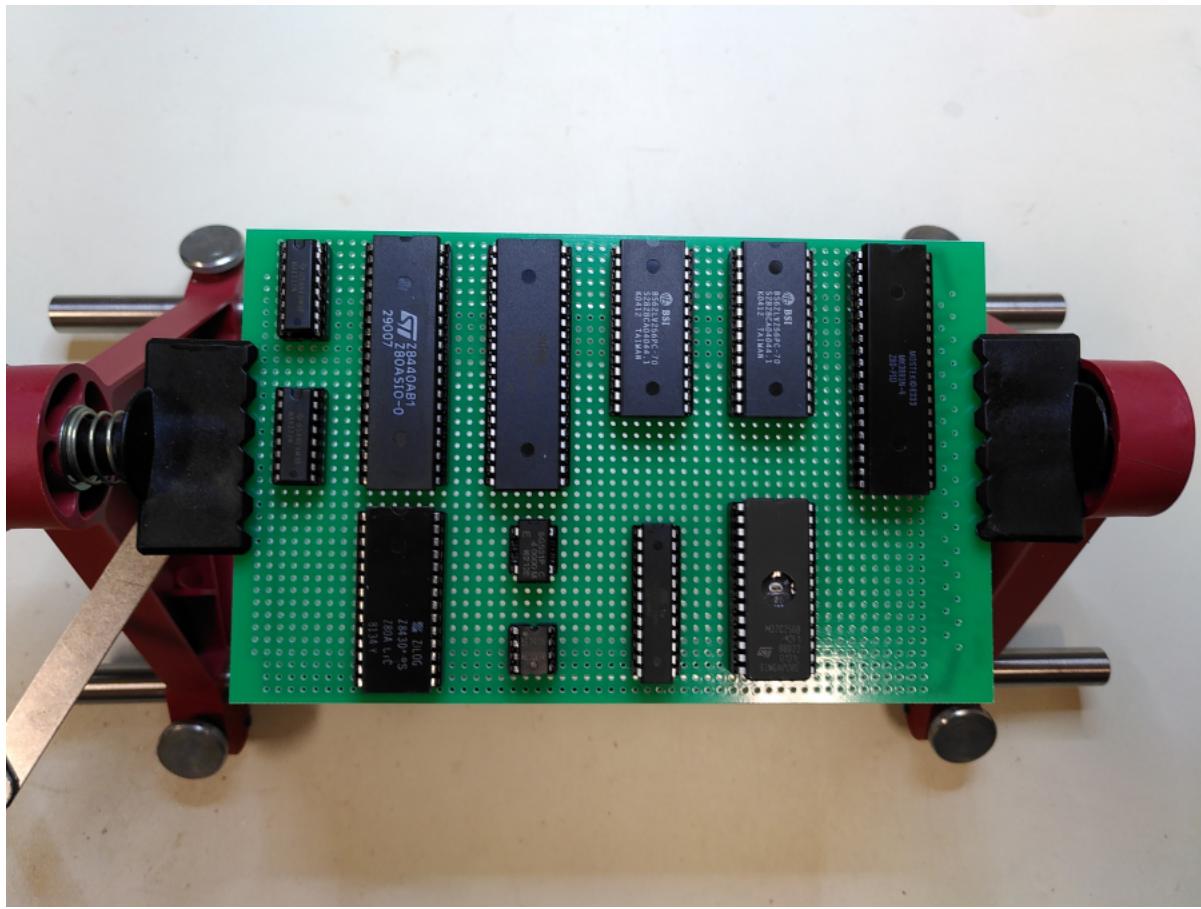
IC components:

- 1x Z80 CPU, Zilog Z84C0010PEG 10MHz
- 1x Z80A SIO/0, Zilog Z-80A SIO/0 / 8440W 0A 4MHz
- 1x Z80A-CTC, Zilog Z80A-CTC 4MHz
- 1x Z80-PIO, Mostek MK3881N-4 4MHz
- 2x SRAM 32Kx8 BSI BS62LV256PC-70 70nS
- 1x EPROM 32x8 ST M27C256 120nS
- 1x 4 MHz Crystal oscillator, ROHS EPSON SG-531P 4.0000MC
- 2x MAX232 RS-232 Transciever, Texas Instruments MAX232N
- 1x 22V10 PLD, Microchip Technology ATF22V10C-15PU
- 1x 555 timer

### Z80 computer component layout

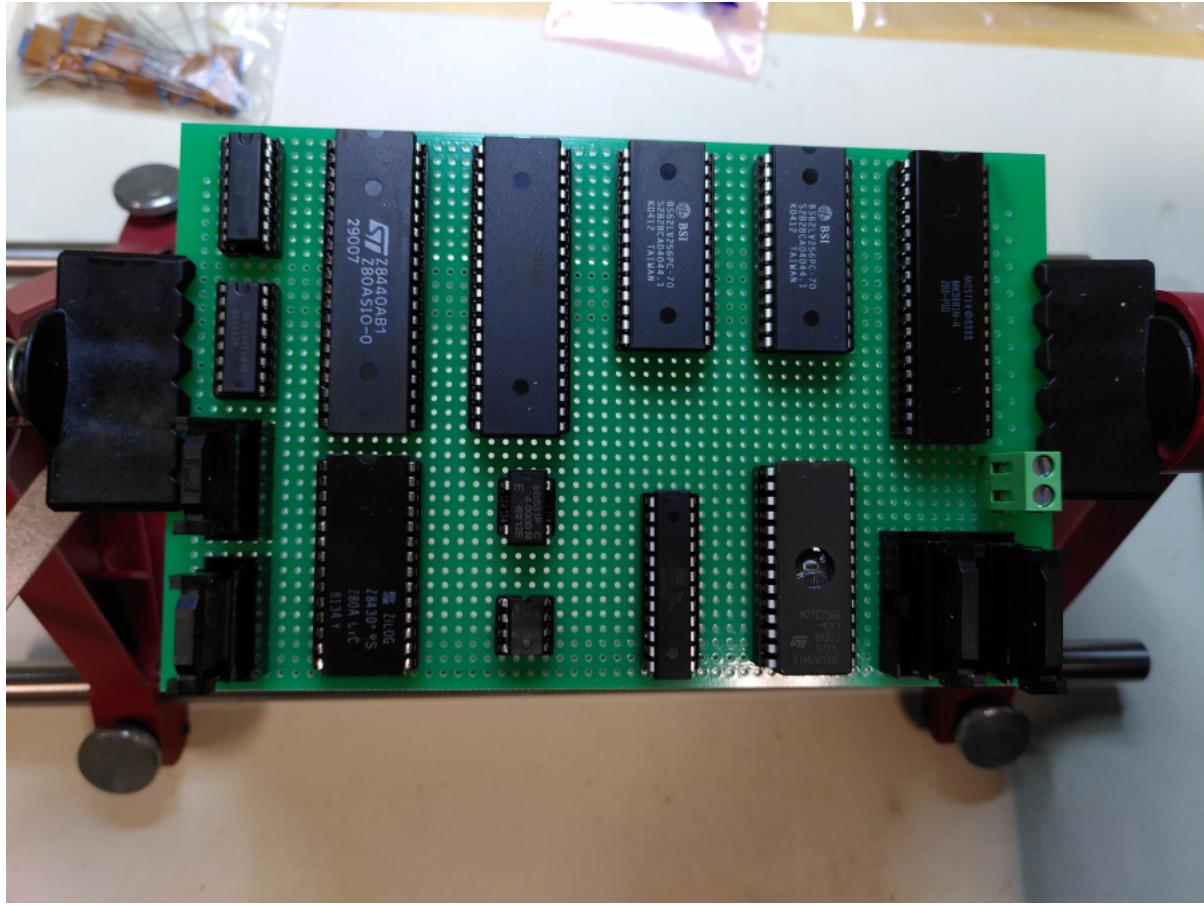
2021-04-03

Z80 computer preliminary component layout:



2021-04-26

Z80 computer final component layout:



## Z80 computer schematic

2021-04-26

Made with KiCAD.

Z80 computer schematic:

[Z80\\_computer\\_schematic\\_2021-04-26.pdf](#) ([http://192.168.42.21/mediawiki/images/f/ff/Z80\\_computer\\_schematic\\_2021-04-26.pdf](http://192.168.42.21/mediawiki/images/f/ff/Z80_computer_schematic_2021-04-26.pdf)) (File:Z80 computer schematic 2021-04-26.pdf)

Z80 computer layout:

[Z80\\_computer\\_layout\\_2021-04-26.pdf](#) ([http://192.168.42.21/mediawiki/images/2/24/Z80\\_computer\\_layout\\_2021-04-26.pdf](http://192.168.42.21/mediawiki/images/2/24/Z80_computer_layout_2021-04-26.pdf)) (File:Z80 computer layout 2021-04-26.pdf)

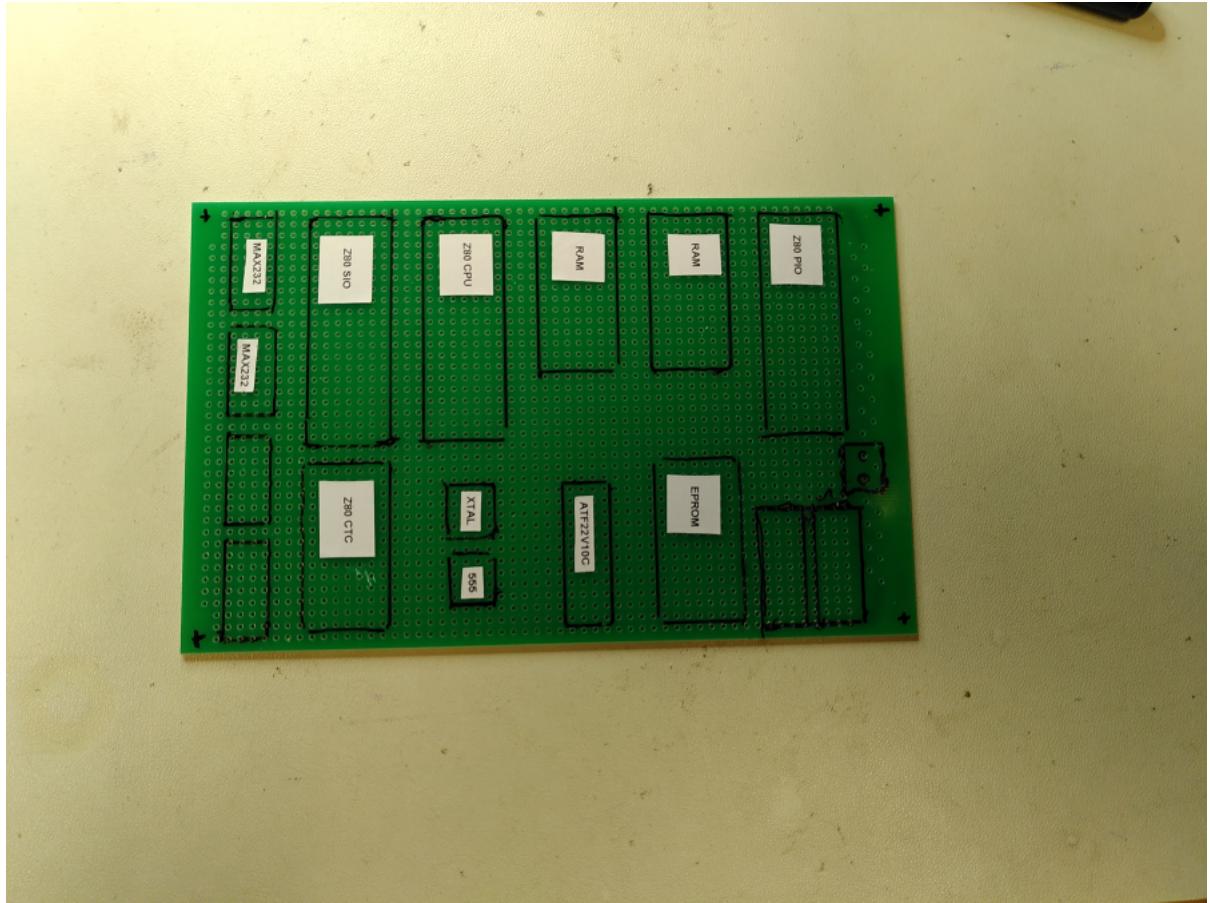
Z80 computer BOM (Bill Of Materials):

[Z80\\_computer\\_bom\\_2021-04-26.pdf](#) ([http://192.168.42.21/mediawiki/images/5/5e/Z80\\_computer\\_bom\\_2021-04-26.pdf](http://192.168.42.21/mediawiki/images/5/5e/Z80_computer_bom_2021-04-26.pdf)) (File:Z80 computer bom 2021-04-26.pdf)

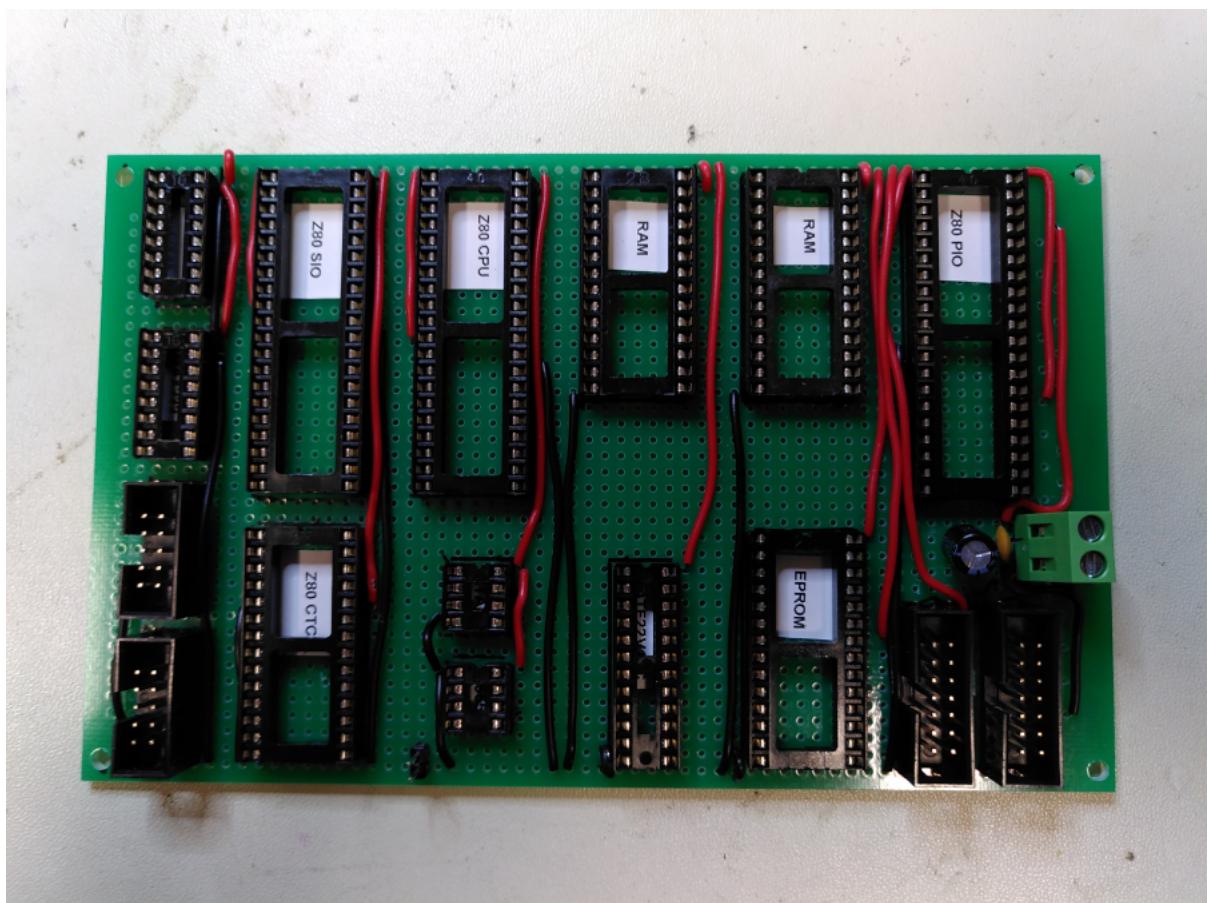
## Z80 computer building steps

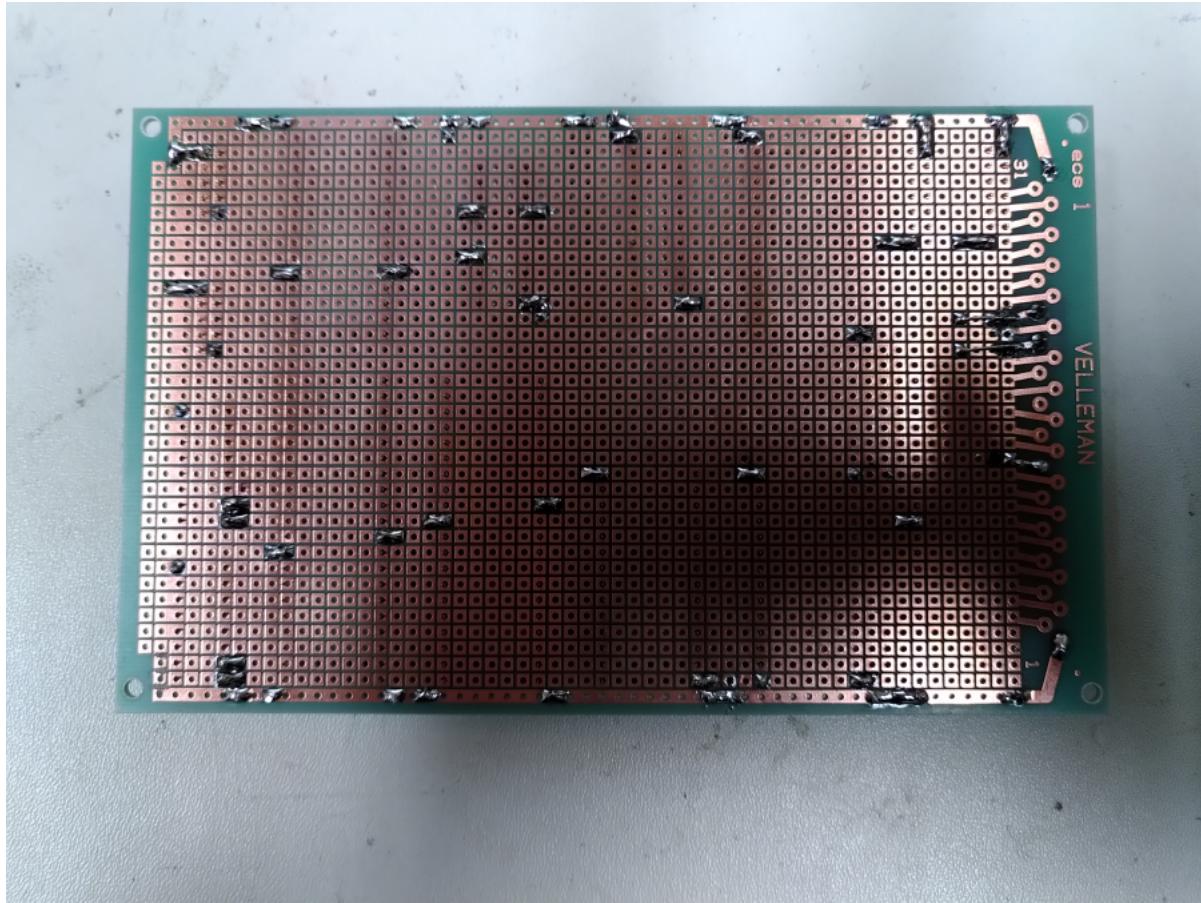
2021-05-03

Z80 computer board



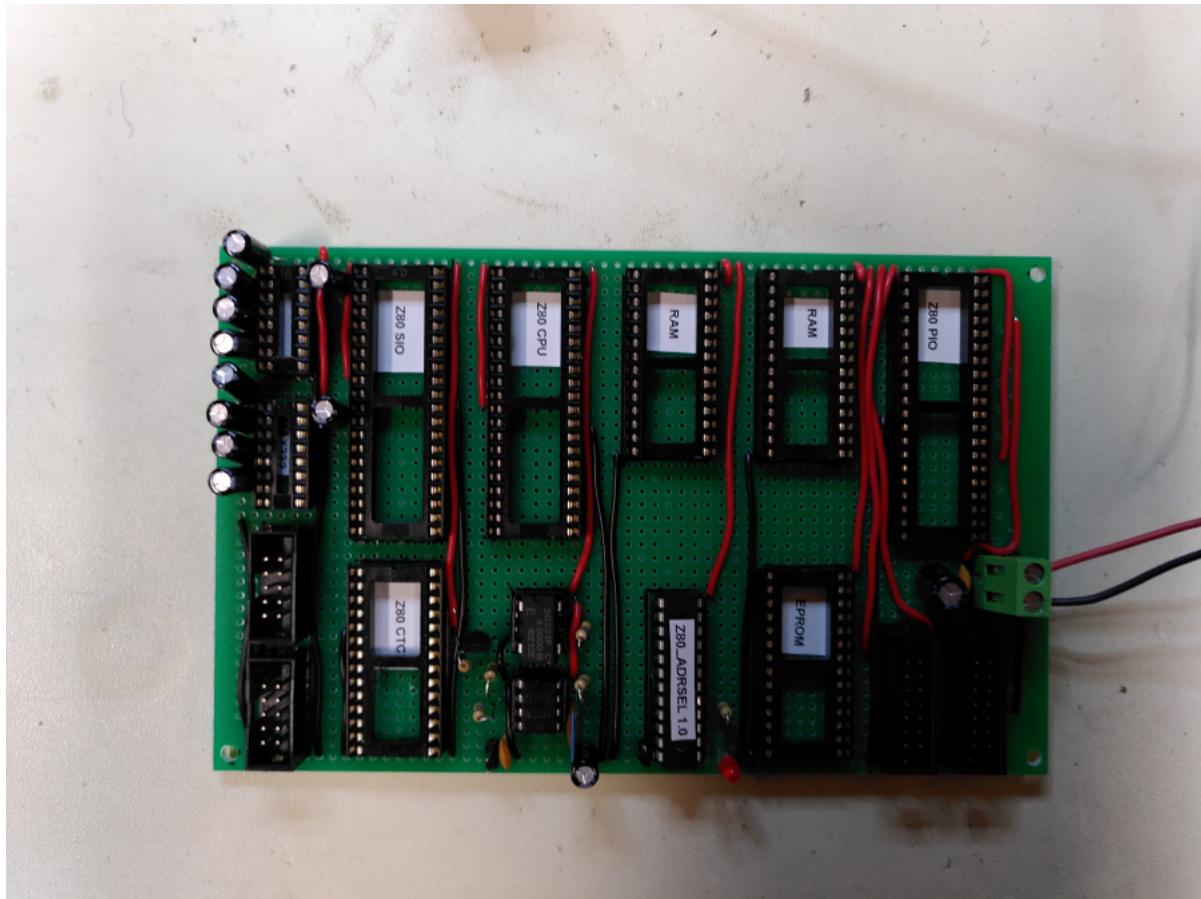
Board with power connections

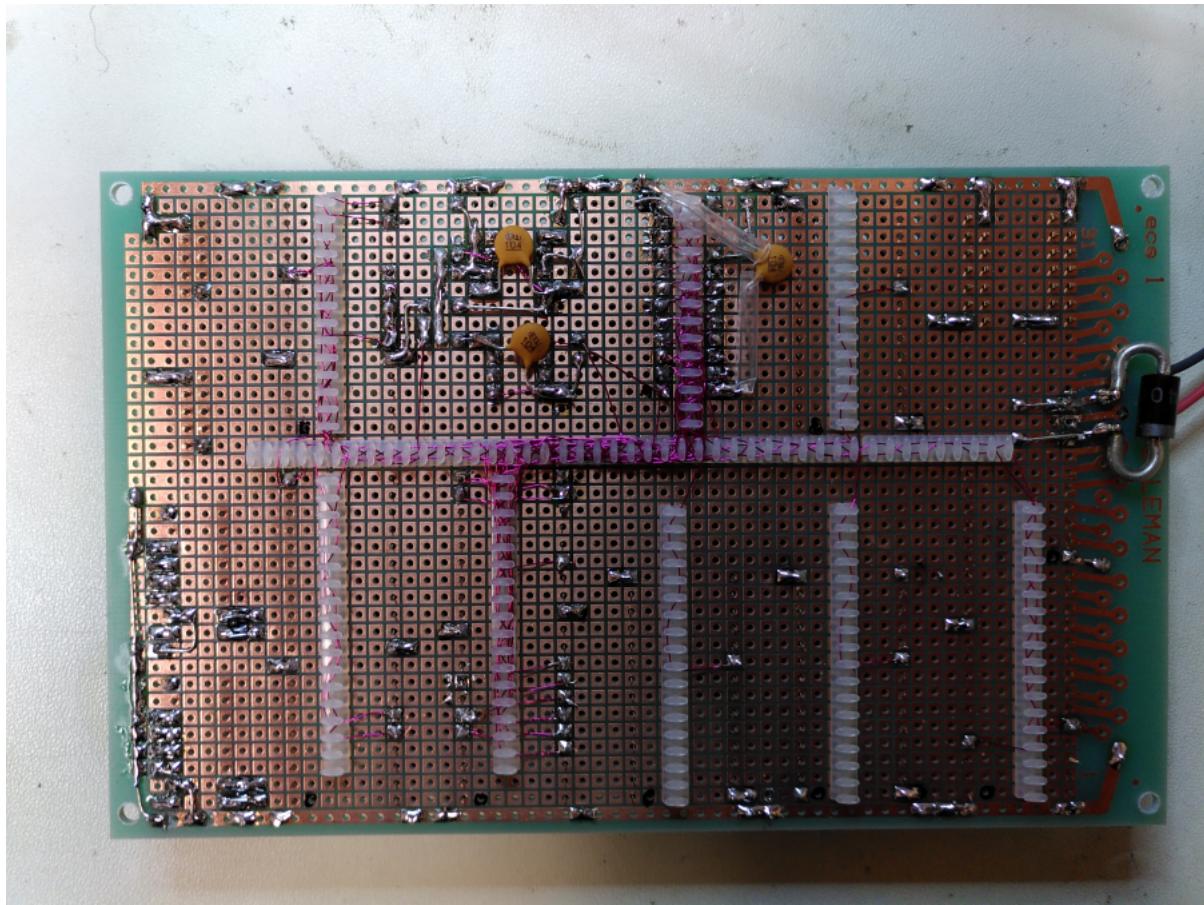




2021-05-10

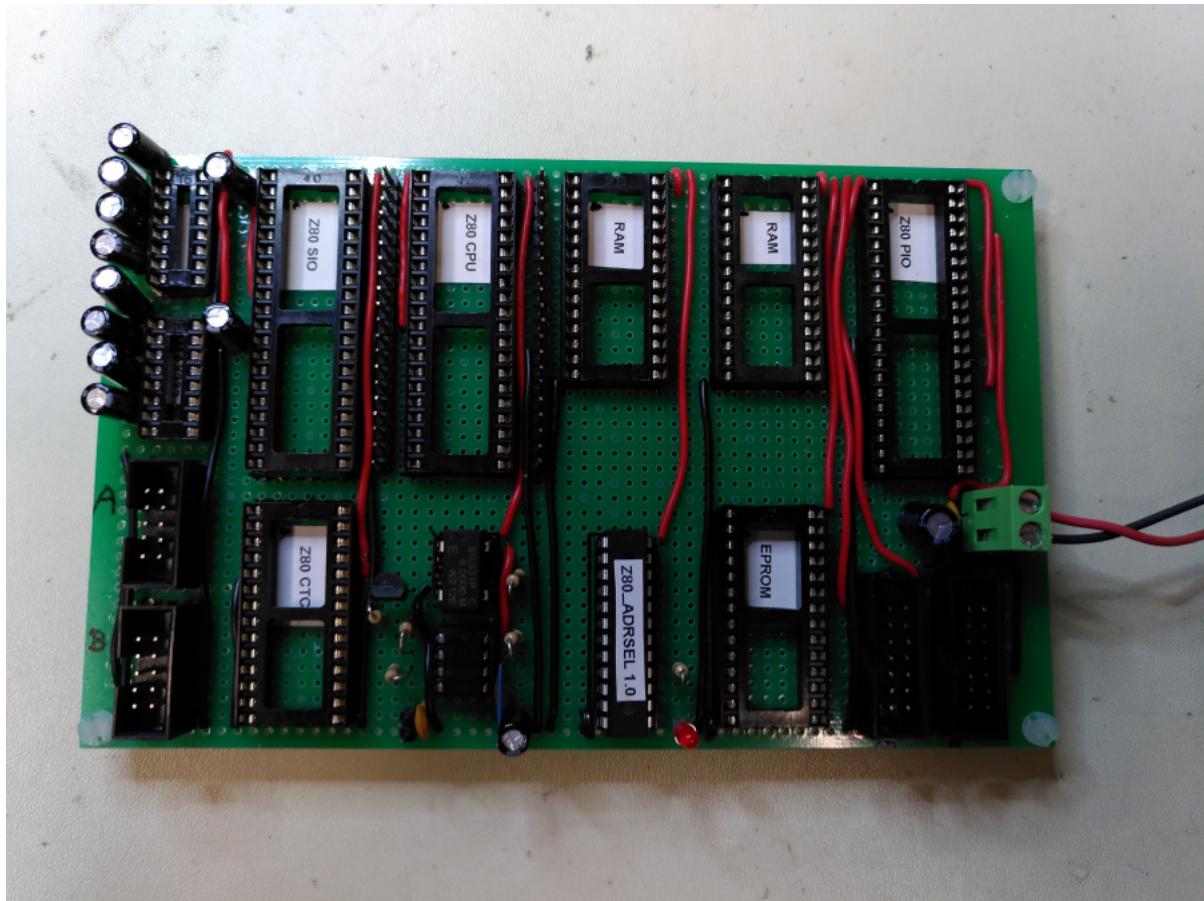
Board with clock and reset

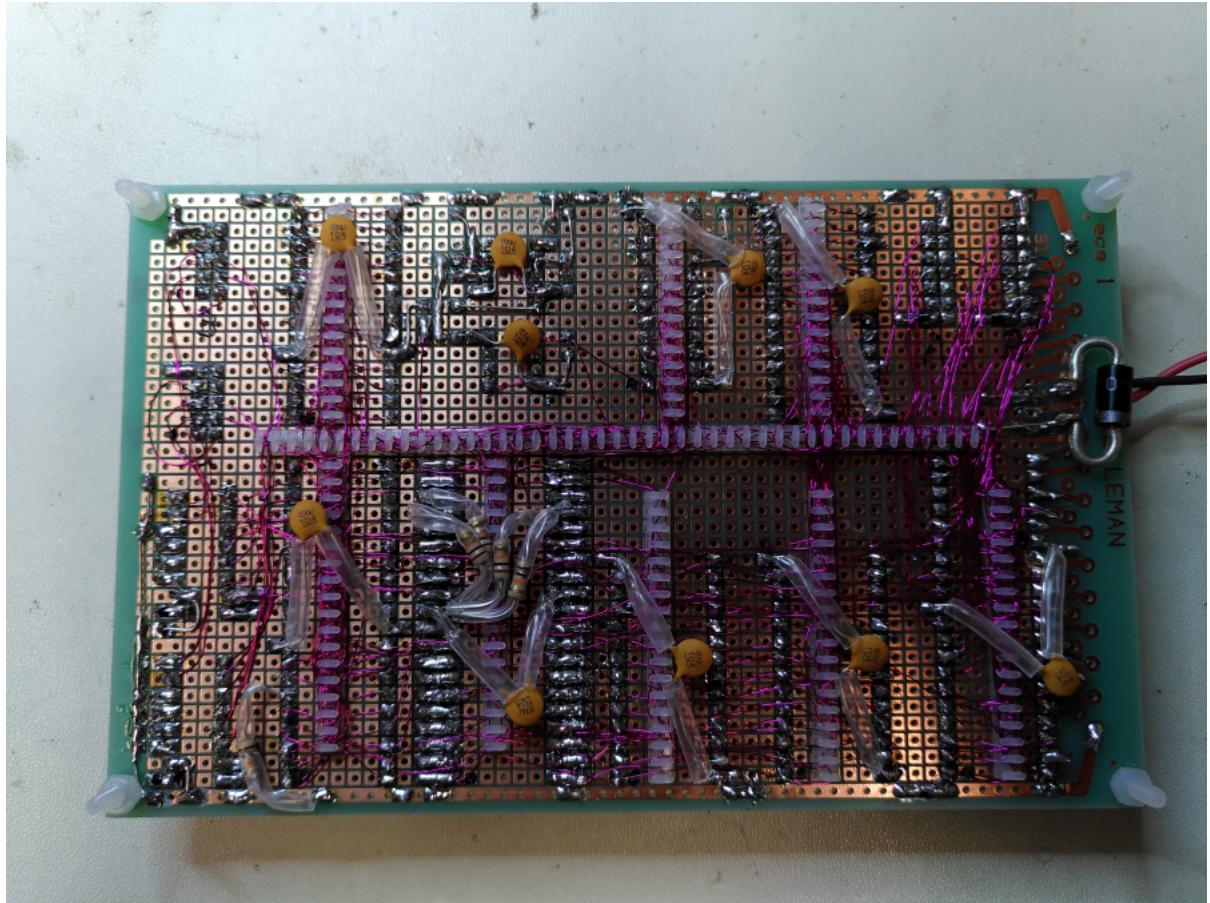




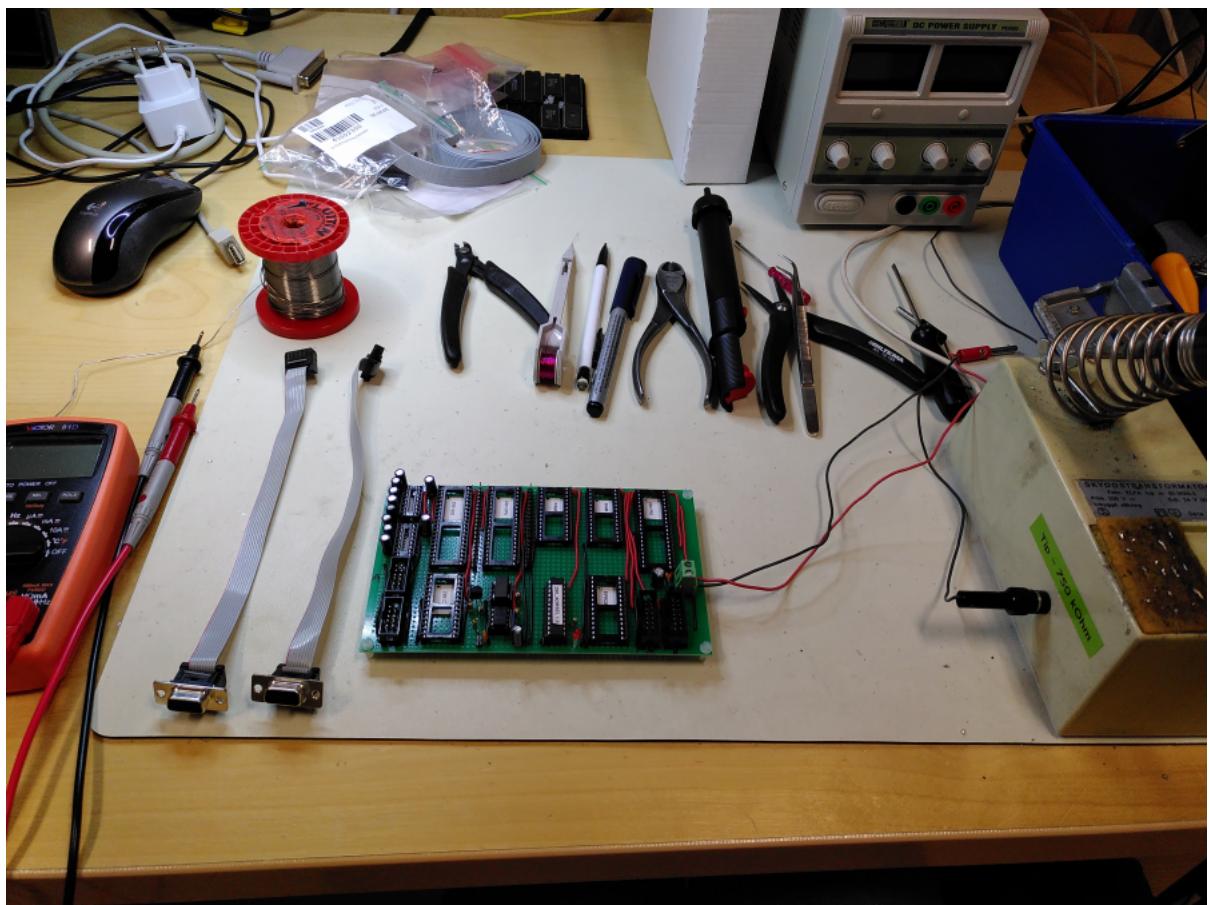
2021-05-16

Completed board





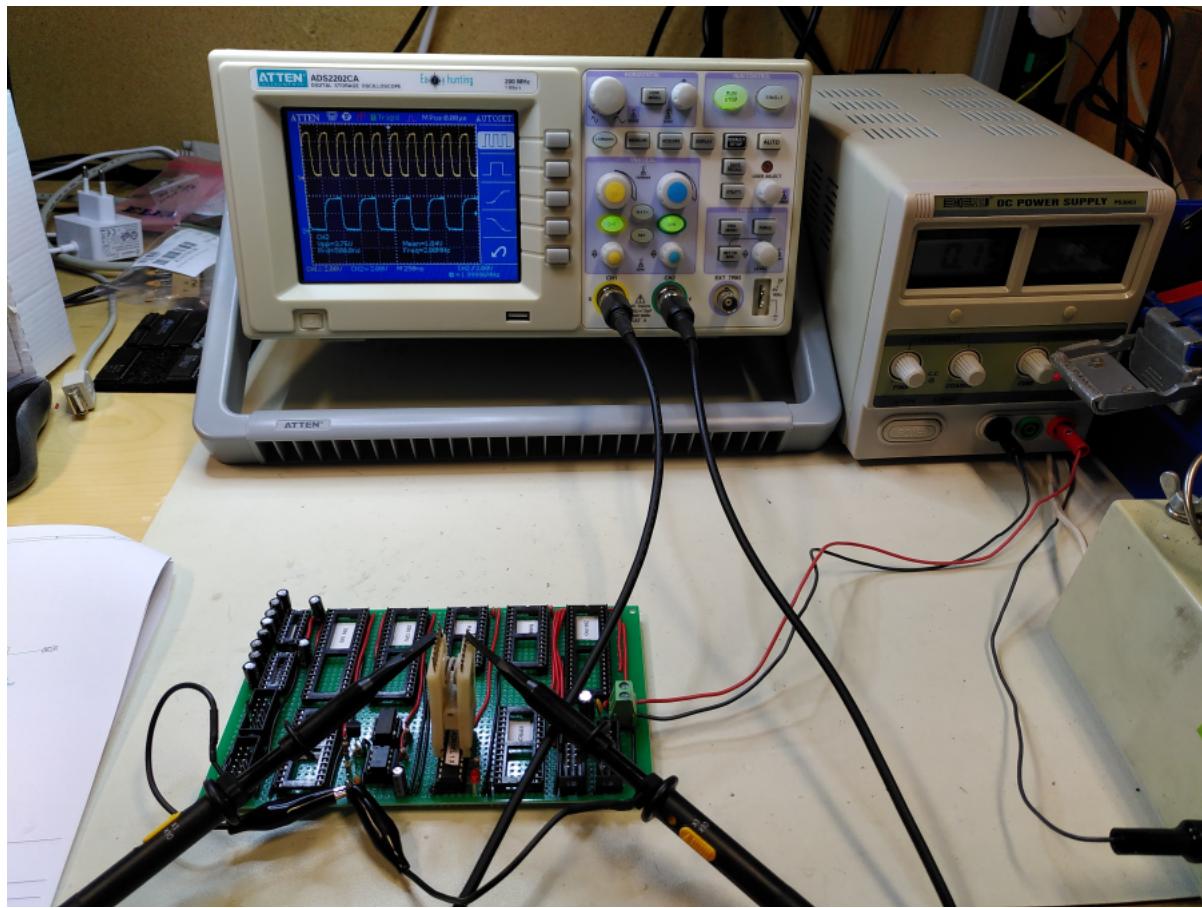
Completed board and tools



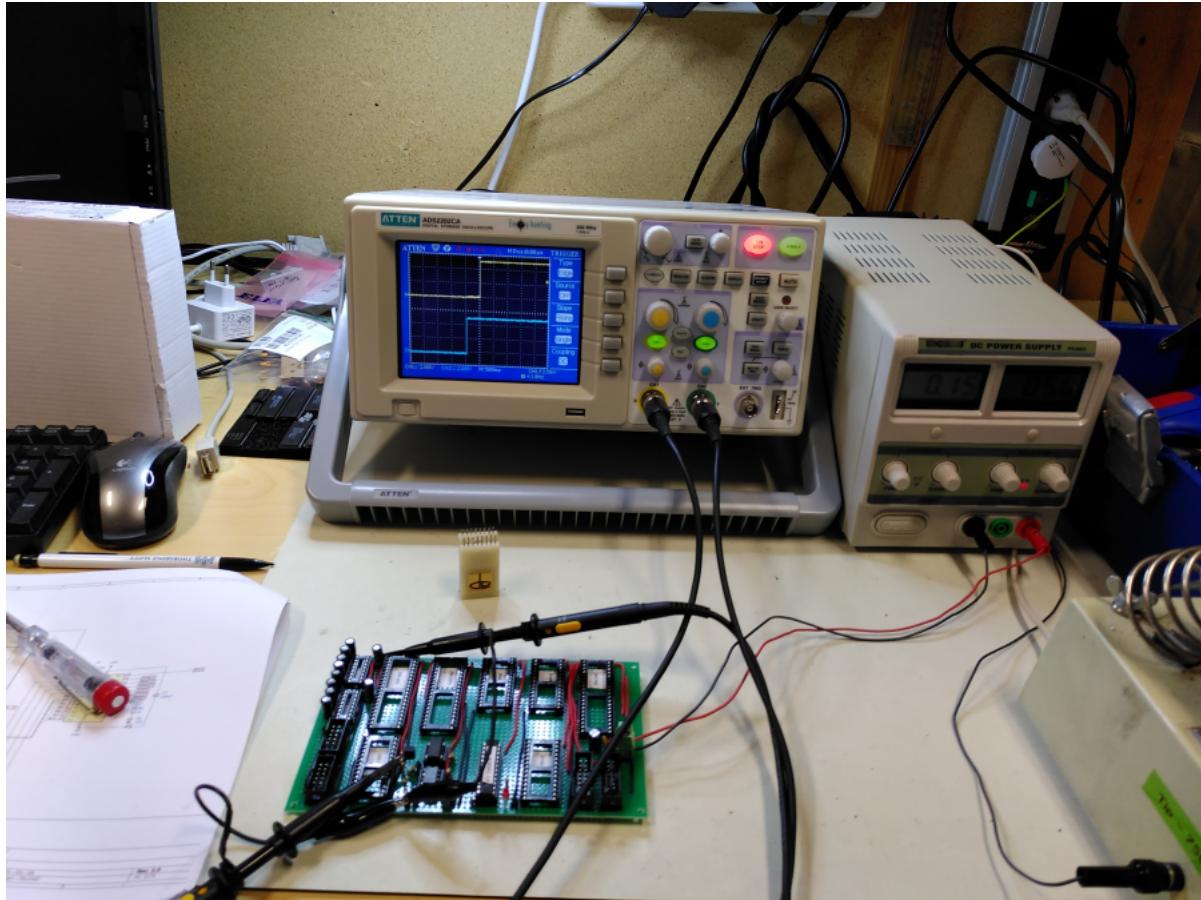
## Z80 computer signal measurements

2021-05-10

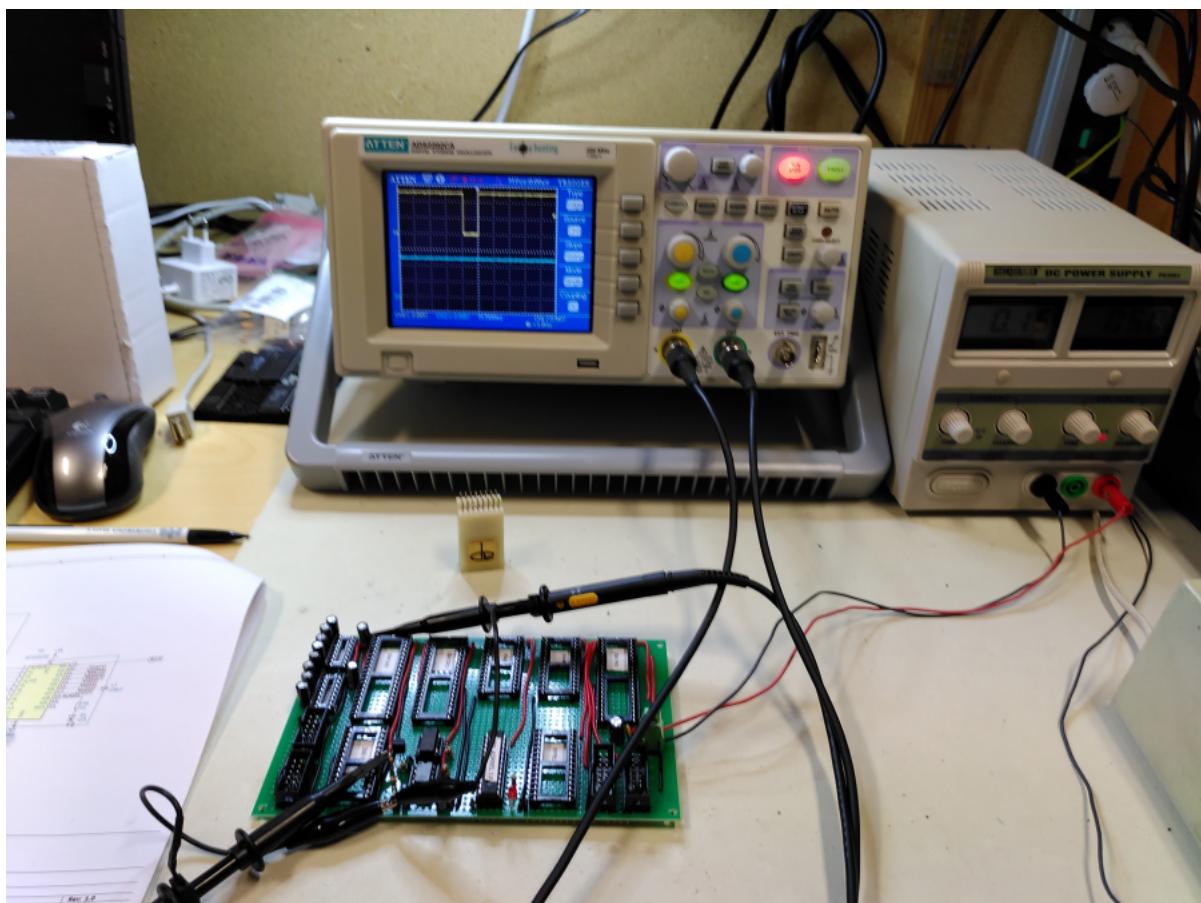
Clock signals, yellow: 4 MHz clock input, blue: 2 MHz clock input to CTC.



Power on reset signal: yellow: reset signal, blue: 5V power.



Manual reset signal: yellow: reset signal.



## ATF22V10C programmable logic for address selection

2021-04-30, 2021-07-27, 2021-09-26

Also handles clock divide.

## Z80\_ADRSEL.PLD

```

Name          Z80_ADRSEL;
Partno       U2;
Revision     1.2;
Date         2021-09-26;
Designer    hal;
Company      atHome;
Location     None;
Assembly    None;
Device       G22V10;

/*
 * Inputs: CPU clock, memory and io access signal and addresses
 */
Pin 1 = CLK;
Pin 2 = MREQ;
Pin 3 = IORQ;
Pin 4 = RD;
Pin 5 = WR;
Pin 6 = A2;
Pin 7 = A3;
Pin 8 = A4;
Pin 9 = A5;
Pin 10 = A6;
Pin 11 = A7;
Pin 13 = A15;
Pin 14 = RESET;

/* Outputs: Serial clock to CTC, io and memory enable
 */
Pin 23 = BCLK;
Pin 22 = ROMCE;
Pin 21 = RAMLCE;
Pin 20 = RAMHCE;
Pin 19 = SIOCE;
Pin 18 = CTCCE;
Pin 17 = PIOCE;
Pin 16 = MEMSEL;
Pin 15 = LED;

/*
 * Logic
 */
/* Divide clock frequency by 2 to get serial clock */

BCLK.D = !BCLK;
BCLK.SP = 'b'0;
BCLK.AR = 'b'0;

/* Decode I/O addresses to select EPROM or RAM in lower 32KB memory */

ROMSEL = !IORQ & !WR & !A2 & !A3 & !A4 & !A5 & !A6 & !A7; /* Write to 0x00 - 0x03 selects EPROM */
RAMSEL = !IORQ & !WR & A2 & !A3 & !A4 & !A5 & !A6 & !A7; /* Write to 0x04 - 0x07 selects RAM */

MEMSEL.D = ((MEMSEL & !ROMSEL) # RAMSEL) & RESET;
MEMSEL.SP = 'b'0;
MEMSEL.AR = 'b'0;

/* Decode I/O chip addresses */

SIOCE = (!(IORQ & !A2 & A3 & !A4 & !A5 & !A6 & !A7); /* Addresses 0x08 - 0x0b selects SIO */
CTCCE = !(IORQ & A2 & A3 & !A4 & !A5 & !A6 & !A7); /* Addresses 0x0c - 0x0f selects CTC */
PIOCE = !(IORQ & !A2 & !A3 & A4 & !A5 & !A6 & !A7); /* Addresses 0x10 - 0x13 selects PIO */

/* Decode I/O addresses to control LED on/off */

LEDOFF = !IORQ & MREQ & !WR & A2 & !A3 & A4 & !A5 & !A6 & !A7; /* Write to 0x14 - 0x17 turns LED off */

```

```

LEDON = !IORQ & MREQ & !WR & !A2 & A3 & A4 & !A5 & !A6 & !A7; /* Write to 0x18 - 0x1f turns LED on */

LED.D = ((LED & !LEDOFF) # LEDON) & RESET;
LED.SP = 'b'0;
LED.AR = 'b'0;

/* Decode memory addresses */

RAMHCE = !(MREQ & A15); /* High 32K RAM */
RAMLCE = !(MREQ & !A15 & MEMSEL); /* Low 32K RAM if selected */
ROMCE = !(MREQ & !A15 & !MEMSEL); /* Low 32K EPROM if selected */

```

Logic generated by WinCupl:

### Z80\_ADRSEL.DOC

```

*****
Z80_ADRSEL
*****

CUPL(WM)      5.0a Serial# 60008009
Device        g22v10 Library DLIB-h-40-1
Created       Sun Sep 26 15:31:11 2021
Name          Z80_ADRSEL
Partno        U2
Revision      1.2
Date          2021-09-26
Designer      hal
Company       atHome
Assembly      None
Location      None

=====
Expanded Product Terms
=====

BCLK.d =>
!BCLK

BCLK.ar =>
0

BCLK.sp =>
0

CTCCE =>
A2 & A3 & !A4 & !A5 & !A6 & !A7 & !IORQ

LED.d =>
!A2 & A3 & A4 & !A5 & !A6 & !A7 & !IORQ & MREQ & RESET & !WR
# IORQ & LED & RESET
# LED & !MREQ & RESET
# LED & RESET & WR
# !A2 & LED & RESET
# A3 & LED & RESET
# !A4 & LED & RESET
# A5 & LED & RESET
# A7 & LED & RESET
# A6 & LED & RESET

LED.ar =>
0

LED.sp =>
0

LEDOFF =>
A2 & !A3 & A4 & !A5 & !A6 & !A7 & !IORQ & MREQ & !WR

LEDON =>
!A2 & A3 & A4 & !A5 & !A6 & !A7 & !IORQ & MREQ & !WR

MEMSEL.d =>
A2 & !A3 & !A4 & !A5 & !A6 & !A7 & !IORQ & RESET & !WR
# IORQ & MEMSEL & RESET

```

```

# MEMSEL & RESET & WR
# A2 & MEMSEL & RESET
# A3 & MEMSEL & RESET
# A4 & MEMSEL & RESET
# A5 & MEMSEL & RESET
# A7 & MEMSEL & RESET
# A6 & MEMSEL & RESET

MEMSEL.ar =>
    0

MEMSEL.sp =>
    0

PIOCE =>
    !A2 & !A3 & A4 & !A5 & !A6 & !A7 & !IORQ

RAMHCE =>
    A15 & !MREQ

RAMLCE =>
    !A15 & MEMSEL & !MREQ

RAMSEL =>
    A2 & !A3 & !A4 & !A5 & !A6 & !A7 & !IORQ & !WR

ROMCE =>
    !A15 & !MEMSEL & !MREQ

ROMSEL =>
    !A2 & !A3 & !A4 & !A5 & !A6 & !A7 & !IORQ & !WR

SIOCE =>
    !A2 & A3 & !A4 & !A5 & !A6 & !A7 & !IORQ

BCLK.oe =>
    1

CTCCE.oe =>
    1

LED.oe =>
    1

MEMSEL.oe =>
    1

PIOCE.oe =>
    1

RAMHCE.oe =>
    1

RAMLCE.oe =>
    1

RESET.oe =>
    0

ROMCE.oe =>
    1

SIOCE.oe =>
    1

```

?

---

Symbol Table

---

Pin Pol	Variable Name	Ext	Pin	Type	Pterms Used	Max Pterms	Min Level
A2		-	6	V	-	-	-
A3		-	7	V	-	-	-
A4		-	8	V	-	-	-
A5		-	9	V	-	-	-
A6		-	10	V	-	-	-
A7		-	11	V	-	-	-
A15		-	13	V	-	-	-

BCLK		23	V	-	-	-
BCLK	d	23	X	1	8	1
BCLK	ar	23	X	1	1	1
BCLK	sp	23	X	1	1	1
CLK		1	V	-	-	-
CTCCE		18	V	1	16	1
IORQ		3	V	-	-	-
LED		15	V	-	-	-
LED	d	15	X	10	10	1
LED	ar	15	X	1	1	1
LED	sp	15	X	1	1	1
LEDOFF		0	I	1	-	-
LEDON		0	I	1	-	-
MEMSEL		16	V	-	-	-
MEMSEL	d	16	X	9	12	1
MEMSEL	ar	16	X	1	1	1
MEMSEL	sp	16	X	1	1	1
MREQ		2	V	-	-	-
PIOCE		17	V	1	14	1
RAMHCE		20	V	1	14	1
RAMLCE		21	V	1	12	1
RAMSEL		0	I	1	-	-
RD		4	V	-	-	-
RESET		14	V	-	-	-
ROMCE		22	V	1	10	1
ROMSEL		0	I	1	-	-
SIOCE		19	V	1	16	1
WR		5	V	-	-	-
BCLK	oe	23	D	1	1	0
CTCCE	oe	18	D	1	1	0
LED	oe	15	D	1	1	0
MEMSEL	oe	16	D	1	1	0
PIOCE	oe	17	D	1	1	0
RAMHCE	oe	20	D	1	1	0
RAMLCE	oe	21	D	1	1	0
RESET	oe	14	D	1	1	0
ROMCE	oe	22	D	1	1	0
SIOCE	oe	19	D	1	1	0

LEGEND    D : default variable    F : field    G : group  
           I : intermediate variable    N : node    M : extended node  
           U : undefined    V : variable    X : extended variable  
           T : function

## Simulation of the address selection logic

2021-04-30, 2021-07-27

Z80\_ADRSEL.SI

```
Name      Z80_ADRSEL;
PartNo   U2;
Date     2021-07-27;
Revision 1.1;
Designer hal;
Company  atHome;
Assembly None;
Location None;
Device   G22V10;
```

ORDER: CLK, BCLK, MREQ, IORQ, RD, WR, A2, A3, A4, A5, A6, A7, A15, ROMCE, RAMLCE, RAMHCE, SIOCE, CTCCE, PIOCE, MEMSEL, RESET;

### VECTORS:

```
0*111100000000*****1
1*011100000000*****1
0*011100000000*****1
1*111100000000*****1
0*111100000000*****1
1*101110000000*****1
0*101010000000*****1
```

```
1*10101000000*****1
0*1111000000*****1
1*1111000000*****1
0*0111000000*****1
1*0111000000*****1
0*1111000000*****1
1*1111000000*****1
0*1111000000*****1
1*1111000000*****0
0*1111000000*****0
1*1111000000*****0
0*1111000000*****1
1*0111000000*****1
1*0111000000*****1
1*1111000000*****1
```

## Z80\_ADRSEL.SO

```
CSIM(WM): CUPL Simulation Program
Version 5.0a Serial#
Copyright (c) 1983, 1998 Logical Devices, Inc.
CREATED Tue Jul 27 12:47:22 2021
```

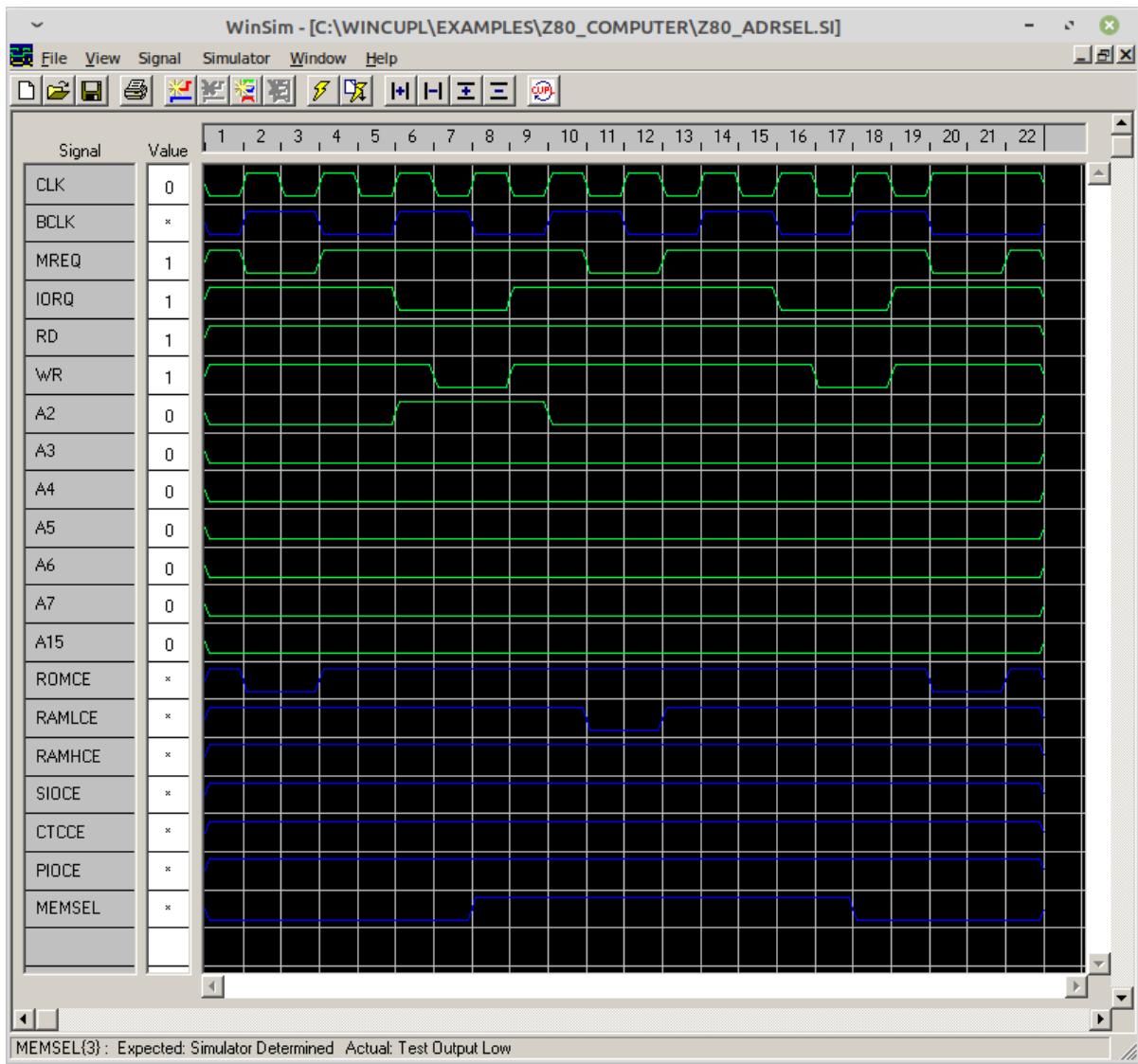
```
LISTING FOR SIMULATION FILE: Z80_ADRSEL.si
```

```
1: Name      Z80_ADRSEL;
2: PartNo    U2;
3: Date      2021-07-27;
4: Revision  1.1;
5: Designer   hal;
6: Company   atHome;
7: Assembly  None;
8: Location   None;
9: Device     G22V10;
10:
11:
12: ORDER: CLK, BCLK, MREQ, IORQ, RD, WR, A2, A3, A4, A5, A6, A7, A15, ROMCE, RAMLCE, RAMHCE, SIOCE,
CTCCE, PIOCE, MEMSEL, RESET;
13:
14:

=====
          RR  M
          RAASCPER
          BMI      OMMITIME
          CCR0     AMLHOCOSS
          LLERRWAAAAAA1CCCCCCEE
          KKQQDR2345675EEEEELT
=====

0001: 0L11110000000HHHHHHL1
0002: 1H01110000000LHHHHHHL1
0003: 0H01110000000LHHHHHHL1
0004: 1L11110000000HHHHHHHL1
0005: 0L11110000000HHHHHHHL1
0006: 1H10111000000HHHHHHHL1
0007: 0H10101000000HHHHHHHL1
0008: 1L10101000000HHHHHHHL1
0009: 0L11111000000HHHHHHHL1
0010: 1H11111000000HHHHHHHL1
0011: 0H01110000000HLHHHHH1
0012: 1L01110000000HLHHHHH1
0013: 0L11111000000HHHHHHHL1
0014: 1H11111000000HHHHHHHL1
0015: 0H11111000000HHHHHHHL1
0016: 1L11111000000HHHHHHHL0
0017: 0L11111000000HHHHHHHL0
0018: 1H11111000000HHHHHHHL0
0019: 0H11111000000HHHHHHHL1
0020: 1L011110000000LHHHHHHL1
0021: 1L011110000000LHHHHHHL1
0022: 1L111110000000HHHHHHHL1
```

Simulation of version 1.0



## ATF22V10C programming

2021-04-30

Check the device:

```
hal@LinuxServer:/mnt/DiskStation_2/files/hal/Electronic_design/Z80_Computer/PLDdesigns/Z80_computer$ afterburner i -t ATF22V10C
PES info: Atmel ATF22V10C VPP=10.0 Timing: prog=10 erase=25
```

Program and verify the device

```
hal@LinuxServer:/mnt/DiskStation_2/files/hal/Electronic_design/Z80_Computer/PLDdesigns/Z80_computer$ afterburner w -f Z80_ADRSEL.jed -t ATF22V10C
hal@LinuxServer:/mnt/DiskStation_2/files/hal/Electronic_design/Z80_Computer/PLDdesigns/Z80_computer$ afterburner v -f Z80_ADRSEL.jed -t ATF22V10C
```

(Needed to program a few times before verification was ok.)

Read the device

```
hal@LinuxServer:/mnt/DiskStation_2/files/hal/Electronic_design/Z80_Computer/PLDdesigns/Z80_computer$ afterburner r -t ATF22V10C
JEDEC file for ATF22V10C
```



[mstead.html\)](#)

## Power on Reset

---

2020-11-19

Using a NE555 (and a transistor to invert the reset signal) seems like a good idea.

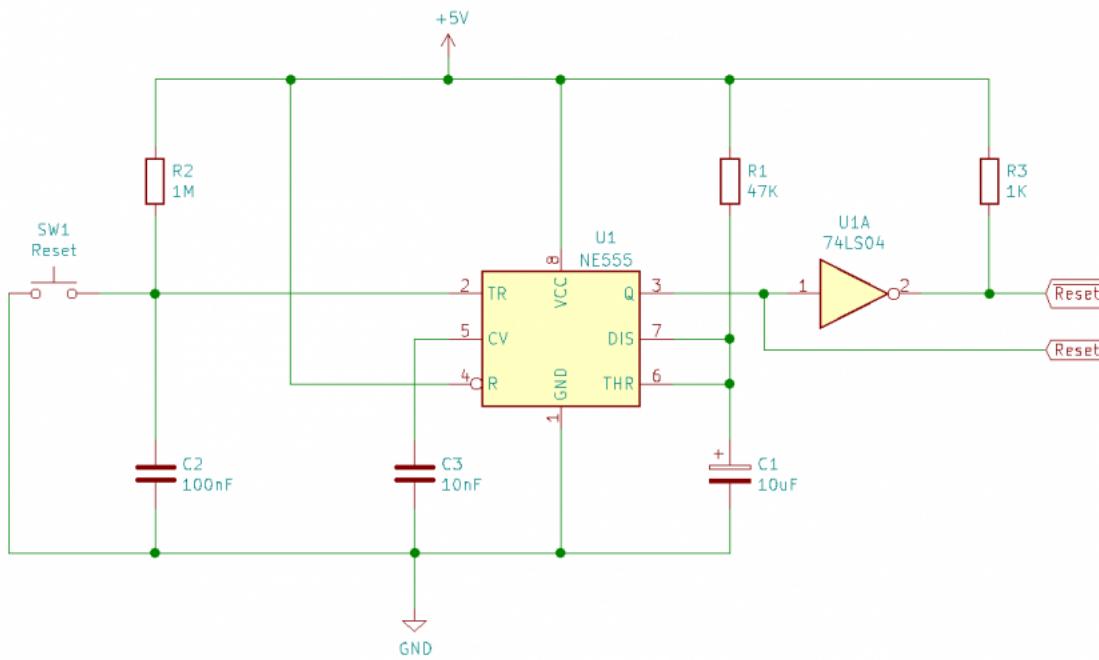
- [Silverlight Developer » Blog Archive NE555 Power on Reset - Silverlight Developer](https://blog.mark-stevens.co.uk/2020/01/ne555-power-on-reset/) (<https://blog.mark-stevens.co.uk/2020/01/ne555-power-on-reset/>)
- [Z80 Power-on reset circuits – Dr. Scott M. Baker](https://www.smbaker.com/z80-power-on-reset-circuits) (<https://www.smbaker.com/z80-power-on-reset-circuits>)
- [Why is the RESET pin set up like this in this Z80 schematic? - Electrical Engineering Stack Exchange](https://electronics.stackexchange.com/questions/420920/why-is-the RESET-pin-set-up-like-this-in-this-Z80-schematic) (<https://electronics.stackexchange.com/questions/420920/why-is-the RESET-pin-set-up-like-this-in-this-Z80-schematic>)

Reset & Watchdog

- [TPS3813 data sheet, product information and support | TI.com](https://www.ti.com/product/TPS3813) (<https://www.ti.com/product/TPS3813>)

2020-04-12

Reset circuit schematic



2021-07-26

The manual reset button does not reset the ATF22V10C, this has to be corrected as if reset is pressed while low RAM is selected, the EPROM start routines does not work. The registers in the Atmel ATF22V10Cs are designed to reset during power-up.

## Pull-up resistors

---

2021-04-19

10k pull-up is proposed: [cpu - Z80, do I need a resistor for pins to Vcc? - Retrocomputing Stack Exchange \(https://retrocomputing.stackexchange.com/questions/18354/z80-do-i-need-a-resistor-for-pins-to-vcc\)](https://retrocomputing.stackexchange.com/questions/18354/z80-do-i-need-a-resistor-for-pins-to-vcc)

2021-05-26

A 10k pull-up to the /INT signal seems necessary and was added.

## Memory and I/O address decoding

---

2021-04-13

- [Programming the memory and I/O address decoder PLD - Z80 Project with AVR system controller \(https://z80avrproject.wordpress.com/2016/10/20/programming-the-memory-and-io-address-decoder-pld/\)](https://z80avrproject.wordpress.com/2016/10/20/programming-the-memory-and-io-address-decoder-pld/)
- [Z80\\_Build\\_Circuits\\_Index\\_0 \(http://zed80.com/Z80-RETRO/Resources/Circuits/Z80\\_Circuits\\_Index\\_0.htm\)](http://zed80.com/Z80-RETRO/Resources/Circuits/Z80_Circuits_Index_0.htm)
- [linker3000/Z80-Board: Z80 computer wirewrapped on perfboard + Some game and utility programs \(https://github.com/linker3000/Z80-Board\)](https://github.com/linker3000/Z80-Board)

2021-04-14

The ATF22V10 is used to:

- Divide CLK/2 to get BCLK for CTC to drive baudrate for SIO
  - [Clock divider /2, Gal22v10B and wincupl - Page 1 \(https://www.eevblog.com/forum/microcontrollers/clock-divider-2-gal22v10b-and-wincupl/\)](https://www.eevblog.com/forum/microcontrollers/clock-divider-2-gal22v10b-and-wincupl/)
  - [embedded - Making a Simple 2-Bit Asynchronous counter in WinCupl - Stack Overflow \(https://stackoverflow.com/questions/2991701/making-a-simple-2-bit-asynchronous-counter-in-wincupl\)](https://stackoverflow.com/questions/2991701/making-a-simple-2-bit-asynchronous-counter-in-wincupl)
- Decode RAM and EPROM address
  - The EPROM and first RAM is switched by writing to one I/O address for using RAM and another I/O address for using EPROM in the lower 32MB of the memory.
  - At power-on EPROM is selected
  - according to: [22v10.pdf \(http://www.da.isy.liu.se/vanheden/pdf/22v10.pdf\)](http://www.da.isy.liu.se/vanheden/pdf/22v10.pdf): "All flip-flops power up to a logic LOW for predictable system initialization."
  - this is valid for PALCE22V10, should be tested for ATF22V10C
- Decode I/O addresses
  - There are two special I/O addresses for switching RAM and EPROM (see above)

- If it is not possible to make a memory selection signal with 22v10 the PIO could be used to select type of memory in the lower 32K. The reset state of the PIO outputs is high impedance, a pull-up resistor could be used to select EPROM at start-up.
- This method was not used, instead the PLD decodes I/O writes to select EPROM or RAM.

## Test PLD programming

2021-04-15

## Clock frequency divider

2021-04-15

Test programmer and IC:

```
hal@LinuxServer:~/Z80_computer$ afterburner i -t ATF22V10C
PES info: Atmel ATF22V10C VPP=10.0 Timing: prog=10 erase=25
```

Compile the logic with WinCupl.

### CLOCK\_DIVIDER.PLD

```
Name          clock_divider;
Partno       U;
Revision     01;
Date         2021-04-14;
Designer     hal;
Company      atHome;
Location     None;
Assembly    None;
Device       g22v10;

/*************************************************/
/*
/*      Clock divider (and address decoder)           */
/*                                                 */ */
/*
/*                                                 */
/*************************************************/

/*
 * Inputs:  define inputs
 */
Pin 1 = clk;

/*
 * Outputs: define outputs
 *
*/
Pin 23 = bclk;

/*
 * Logic: create flip-flop
 */
bclk.d = !bclk;           /* divider */
```

Two warnings were given:

```
[0016cb] Please note: no expression assigned to: bclk.ar
[0016cb] Please note: no expression assigned to: bclk.sp
```

By clicking on the warning message the following information is shown:

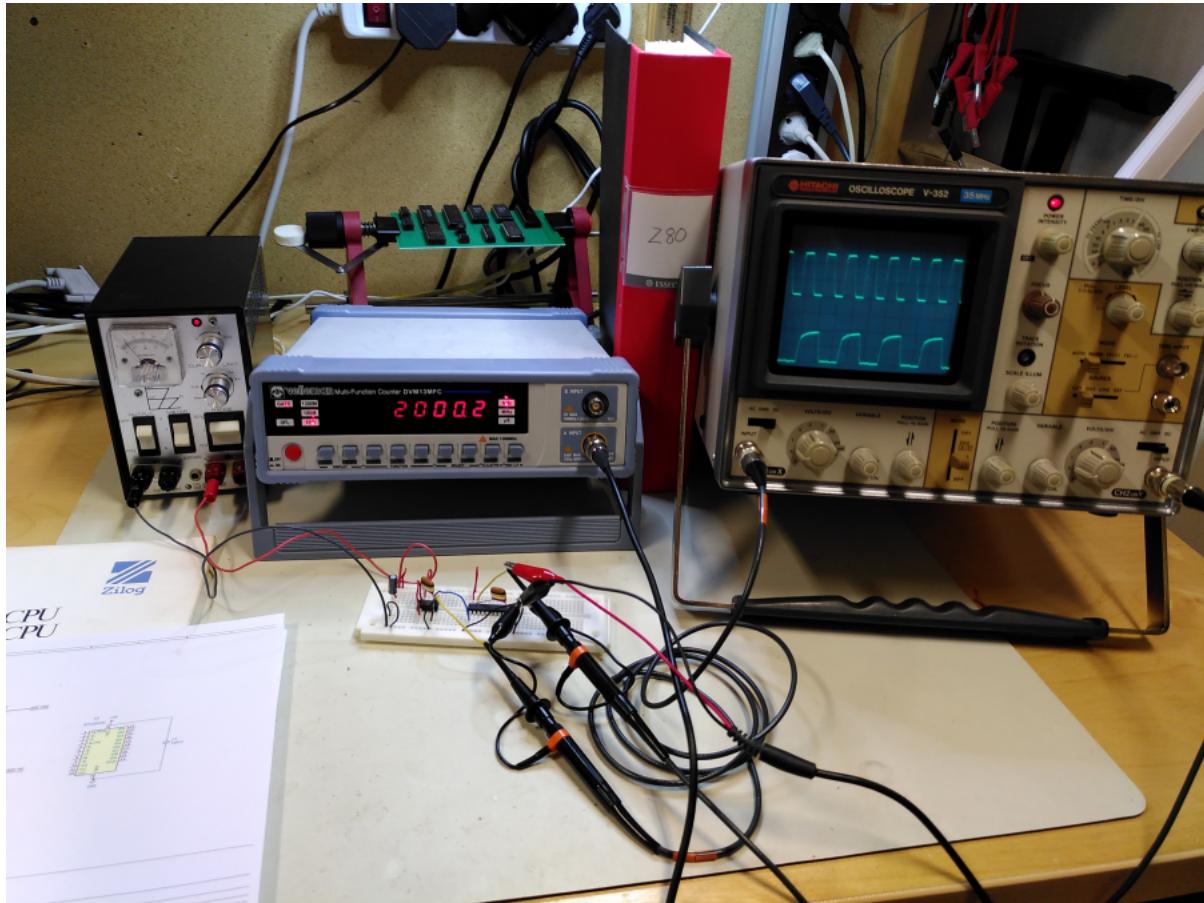
```
Code: 0016cb Level: Warning
Message: no expression assigned to: "variable"
Description: The variable requires an output expression assignment. This warning message is commonly given when all outputs in a bank have the same capability (reset, preset, and so on) and not all the variables have been assigned the same expression. It is given to remind the user that all outputs will be affected.
Information: This warning may be suppressed by assigning the variable to 'b'0 or 'b'1 as appropriate.
```

Program and verify the JEDEC file on the IC.

```
hal@LinuxServer:~/Z80_computer$ afterburner w -f clock_divider.jed -t ATF22V10C
hal@LinuxServer:~/Z80_computer$ afterburner v -f clock_divider.jed -t ATF22V10C
hal@LinuxServer:~/Z80_computer$
```

No output from commands means that all is ok.

Test output frequency and waveform:



Erase IC after test:

```
hal@LinuxServer:~/Z80_computer$ afterburner e -t ATF22V10C
```

2021-04-25

Simulate the clock divider with WinSim

clock\_divider.si

```
Name      clock_divider;
PartNo   U;
Date     2021-04-14;
Revision 01;
Designer hal;
Company  atHome;
Assembly None;
Location None;
Device   g22v10;
```

```
ORDER: clk, bclk;
```

## VECTORS:

```
0*
1*
0*
1*
0*
1*
0*
1*
0*
1*
```

CLOCK\_DIVIDER.sim

```
%SIGNAL
PIN 23 = bclk
PIN 1 = clk
%END

%FIELD
%END

%EQUATION
bclk.d =>
bclk

bclk.oe =>
1
%END
```

CLOCK\_DIVIDER.so

```
CSIM(WM): CUPL Simulation Program
Version 5.0a Serial#
Copyright (c) 1983, 1998 Logical Devices, Inc.
CREATED Sun Apr 25 16:29:53 2021
```

```
LISTING FOR SIMULATION FILE: clock_divider.si
```

```
1: Name      clock_divider;
2: PartNo   U;
3: Date     2021-04-14;
4: Revision 01;
5: Designer hal;
6: Company  atHome;
7: Assembly None;
8: Location None;
9: Device   g22v10;
10:
11:
12: ORDER: clk, bclk;
13:
14:
```

```
=====
b
```

```

cc
ll
kk
=====
0001: 0L
0002: 1H
0003: 0H
0004: 1L
0005: 0L
0006: 1H
0007: 0H
0008: 1L
0009: 0L
0010: 1H

```

## Serial I/O & baud-rate generator

---

2021-04-13

- [LM80C: Z80 SIO and serial communication – LEONARDOMILIANI.com \(https://www.leonardomiliani.com/en/2019/lm80c-z80-sio-e-comunicazione-seriale/\)](https://www.leonardomiliani.com/en/2019/lm80c-z80-sio-e-comunicazione-seriale/)
- [linker3000/Z80-Board: Z80 computer wirewrapped on perfboard + Some game and utility programs \(https://github.com/linker3000/Z80-Board\)](https://github.com/linker3000/Z80-Board)
- [LM80C: Z80 CTC and interrupts – LEONARDOMILIANI.com \(https://www.leonardomiliani.com/en/2019/english-lm80c-z80-ctc-and-interrupts/\)](https://www.leonardomiliani.com/en/2019/english-lm80c-z80-ctc-and-interrupts/)
- [How to program the z80 CTC | Manualzz \(https://manualzz.com/doc/14618543/how-to-program-the-z80-ctc\)](https://manualzz.com/doc/14618543/how-to-program-the-z80-ctc)

Using the CTC with internal prescaler will not work as the prescaler uses only divide by 16 or 256 which will not give the right serial rate. The Tc input frequency to the CTC must not be higher than half the clock frequency.

The Tx/Rx clock input to the SIO must at least be 16 times the baud-rate.

Example with 4MHz xtal oscillator and 9600 baud:  $4000000/(9600*2*16) = 13.0208$  Using a CTC divider of 13 this will give a baud-rate of  $4000000/(2*16*13) = 9615$  baud which hopefully is close enough. If this is not exact enough, another xtal oscillator must be selected. This one would work: [SG-531P 3.6864MC:ROHS EPSON | Kristaller, oscillatorer, resonatorer | DigiKey \(https://www.digikey.se/product-detail/sv/epson/SG-531P-3-6864MC-ROHS/SER1223-ND/1021916\)](https://www.digikey.se/product-detail/sv/epson/SG-531P-3-6864MC-ROHS/SER1223-ND/1021916).

2021-04-16

The following signals are connected for each serial channel:

- Rx - SIO RXD
- Tx - SIO TXD
- RTS - SIO RTS
- CTS - SIO CTS
- DTR and DSR are connected in the external contact

Serial connector IDC to DB9

- [Assembling Serial IDC to DB9 Cable \(https://developer.toradex.com/knowledge-base/assembling-serial-idc-to-db9-cable\)](https://developer.toradex.com/knowledge-base/assembling-serial-idc-to-db9-cable)

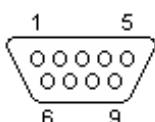
Serial pinout, female D-sub, to work with straight cable from 9-pin D-sub in PC. The PC is DTE and the Z80

computer is DCE.

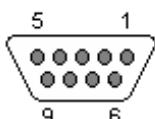
## Serial (PC 9)

Also known as EIA/TIA 574.

### Pinout



9 PIN DE-9 MALE at the Computer.

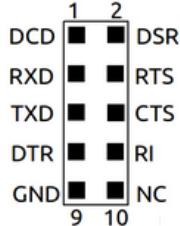


9 PIN DE-9 FEMALE at the Cable.

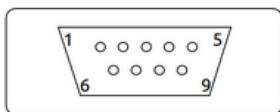
Pin	Name	RS232	V.24	Dir	Description
1	(D)CD	CF	109	←	(Data) Carrier Detect
2	RXD	BB	104	←	Receive Data
3	TXD	BA	103	→	Transmit Data
4	DTR	CD	108.2	→	Data Terminal Ready
5	GND	AB	102	—	System Ground
6	DSR	CC	107	←	Data Set Ready
7	RTS	CA	105	→	Request to Send
8	CTS	CB	106	←	Clear to Send
9	RI	CE	125	←	Ring Indicator

#### Notes:

- Direction is DTE (Computer) relative DCE (Modem).
- RS232 column is RS232 circuit name.
- V.24 column is ITU-TSS V.24 circuit name.



IDC pinout



PIN	SIGNAL
1	DCD
2	RXD
3	TXD
4	DTR
5	GND
6	DSR
7	RTS
8	CTS
9	RI

DB9 pinout

## Parallel port

---

2021-05-22

If the PIO is configured in Output Mode (Mode 0) and connecting the Ready line to the Strobe line, a strobe with a duration of one clock period is generated with no other logic required.

## Floppy Disk Controller

---

2021-02-13

TODO maybe.

How may this be implemented?

- Using old PC ISA board? (I have three FD/HD controller boards, on where the ICs are socketed so it may be used as a component source)
  - I have one ISA board (DT-3767) with IDE and FD interface, the FDC IC is SMC37C65B
  - According to the reference below: Western Digital WD37C65, SMC FDC37C65, or GoldStar GM82C765B floppy disk controllers are equivalent.
- New controller design with FDC IC?
- ...?

References:

- [ISA Floppy Disk and Serial Controller - Malinov Family Web Presence](http://www.malinov.com/Home/sergeys-projects/isa-fdc-and-uart) (<http://www.malinov.com/Home/sergeys-projects/isa-fdc-and-uart>)
- [Zeta SBC V2 - Malinov Family Web Presence](http://www.malinov.com/Home/sergey-s-projects/zeta-sbc-v2) (<http://www.malinov.com/Home/sergey-s-projects/zeta-sbc-v2>)

- [DT-3767 \(https://arvutimuuseum.ee/th99/c/C-D/20330.htm\)](https://arvutimuuseum.ee/th99/c/C-D/20330.htm)
- [Francis' home page - 1.44MB floppy disk controller project \(http://www.g-photo.net/projects/wd\\_fdc/wd\\_fdc.htm\)](http://www.g-photo.net/projects/wd_fdc/wd_fdc.htm)
- [Industry Standard Architecture - Wikipedia \(https://en.wikipedia.org/wiki/Industry\\_Standard\\_Architecture#ISA\\_bus\\_architecture\)](https://en.wikipedia.org/wiki/Industry_Standard_Architecture#ISA_bus_architecture)
- [WD37C65C datasheet \(https://datasheetspdf.com/pdf-file/503579/WesternDigital/WD37C65C/1\)](https://datasheetspdf.com/pdf-file/503579/WesternDigital/WD37C65C/1)
- [Programming the NEC µPD765 and Intel 82072/7 Floppy Disk Controller \(https://www.isdaman.com/alsos/hardware/fdc/floppy.htm\)](http://www.isdaman.com/alsos/hardware/fdc/floppy.htm)
- [floppy\\_drives deramp.com \(https://deramp.com/downloads/floppy\\_drives/\)](https://deramp.com/downloads/floppy_drives/)

## Hard Disk ATA controller

---

2021-02-13

TODO maybe.

How may this be implemented?

- Using old PC ISA board DT-3767?
- New design?
- [Parallel ATA - Wikipedia \(https://en.wikipedia.org/wiki/Parallel\\_ATA#IDE\\_and\\_ATA-1\)](https://en.wikipedia.org/wiki/Parallel_ATA#IDE_and_ATA-1)
- [WD1003-WA2 Hard Disk/Floppy Controller Settings and Configuration \(https://stason.org/TULARC/pc/hard-disk-floppy-controllers/U-Z/WESTERN-DIGITAL-CORPORATION-Two-ST506-412-MFM-devi.html\)](https://stason.org/TULARC/pc/hard-disk-floppy-controllers/U-Z/WESTERN-DIGITAL-CORPORATION-Two-ST506-412-MFM-devi.html)

Since the original ATA interface is essentially just a 16-bit ISA bus in disguise, the bridge was especially simple in case of an ATA connector being located on an ISA interface card. The integrated controller presented the drive to the host computer as an array of 512-byte blocks with a relatively simple command interface. This relieved the mainboard and interface cards in the host computer of the chores of stepping the disk head arm, moving the head arm in and out.

References:

- [DT-3767 \(https://arvutimuuseum.ee/th99/c/C-D/20330.htm\)](https://arvutimuuseum.ee/th99/c/C-D/20330.htm)
- [arvutimuuseum.ee/th99/c/c16ide\\_t.htm \(https://arvutimuuseum.ee/th99/c/c16ide\\_t.htm\)](https://arvutimuuseum.ee/th99/c/c16ide_t.htm)
- [WD1003-WA2 Hard Disk/Floppy Controller Settings and Configuration \(https://stason.org/TULARC/pc/hard-disk-floppy-controllers/U-Z/WESTERN-DIGITAL-CORPORATION-Two-ST506-412-MFM-devi.html\)](https://stason.org/TULARC/pc/hard-disk-floppy-controllers/U-Z/WESTERN-DIGITAL-CORPORATION-Two-ST506-412-MFM-devi.html)
- [mame/hdc92x4.cpp at master · mamedev/mame \(https://github.com/mamedev/mame/blob/master/src/devices/machine/hdc92x4.cpp\)](https://github.com/mamedev/mame/blob/master/src/devices/machine/hdc92x4.cpp)
- [cosam.org - ATA/IDE Interface \(http://www.cosam.org/projects/z80/ata.html\)](http://www.cosam.org/projects/z80/ata.html)
- [An 8-bit IDE interface @ RetroLem \(http://blog.retrolem.co.uk/electronics-articles/an-8-bit-ide-interface/\)](http://blog.retrolem.co.uk/electronics-articles/an-8-bit-ide-interface/)
- [Francis' home page - IDE interface project \(http://www.g-photo.net/projects/ide/ide.htm\)](http://www.g-photo.net/projects/ide/ide.htm)
- <http://blog.retrolem.co.uk/electronics-articles/an-8-bit-ide-interface/>

## SD Card interface

---

2021-09-23

More information in a separate page: [Z80 Computer CP/M with SD card and driver software](#)

2021-09-17

For this computer I think I will make a SPI based microSD disk. It seems that a slow but simple bit-banger SPI interface may be used.

Design of bit-banger SPI interface

- [SPI Bus Z80 RC2014 / Code / \[ab3412\] \(<https://sourceforge.net/p/spi-bus-z80-rc2014/code/ci/default/tree/>\)](#)

Design with TTL shift registers for SPI

- [Galaksija/RC2014\\_SPI\\_SD: SD SPI interface for RC2014 and similar boards \(\[https://github.com/Galaksija/RC2014\\\_SPI\\\_SD\]\(https://github.com/Galaksija/RC2014\_SPI\_SD\)\)](#)

Other reference designs

- [Micro SD card module using shift registers \(<https://groups.google.com/g/rc2014-z80/c/A5lvM31saQo>\)](#)
- [RetroBrew Computers Forum: General Discussion » CPM HDDs on SD Cards - Implementation \(<https://www.retrobrewcomputers.org/forum/index.php?t=msg&th=583&goto=8554&>\)](#)
- [Z80-MBC2: a 4 ICs homebrew Z80 computer | Hackaday.io \(<https://hackaday.io/project/159973-z80-mbc2-a-4-ics-homebrew-z80-computer>\)](#)
- [Overview | A Z80 CP/M emulator for the SAMD51 | Adafruit Learning System \(<https://learn.adafruit.com/z80-cpm-emulator-for-the-samd51-grand-central>\)](#)

2021-09-21

Read, write and format a SD Card

- [GNU ddrescue Manual \(\[https://www.gnu.org/software/ddrescue/manual/ddrescue\\\_manual.html#Top\]\(https://www.gnu.org/software/ddrescue/manual/ddrescue\_manual.html#Top\)\)](#)
- [Is there a "SD Memory Card Formatter" equivalent for Linux? - Unix & Linux Stack Exchange \(<https://unix.stackexchange.com/questions/599977/is-there-a-sd-memory-card-formatter-equivalent-for-linux>\)](#)

A lot of other references

2021-08-31

Z80 CPM with microSD

- [SD / SDHC Interface for Grant Searle's CP/M computer \(<http://xepb.org/dtz/sgsbcsd.html>\)](#)

2021-08-18

TODO maybe.

Why not do a SD card interface? Z180 has an interface usable for SPI:  
[Z180\\_Computer#CSI.2FO\\_based\\_SPI\\_.28or\\_i2c.29\\_interface\\_to\\_SD\\_card](#)

- [SPI SD Card Interface for Z80 Systems](https://www.ecstaticlyrics.com/electronics/SPI/fast_z80_interface.html) ([https://www.ecstaticlyrics.com/electronics/SPI/fast\\_z80\\_interface.html](https://www.ecstaticlyrics.com/electronics/SPI/fast_z80_interface.html))
- [Micro SD card module using shift registers](https://groups.google.com/g/rc2014-z80/c/A5lvM31saQo) (<https://groups.google.com/g/rc2014-z80/c/A5lvM31saQo>)
- [A bit disappointed in the Z80 SPI speed | olduino](https://olduino.wordpress.com/2016/05/17/a-bit-disappointed-in-the-z80-spi-speed/) (<https://olduino.wordpress.com/2016/05/17/a-bit-disappointed-in-the-z80-spi-speed/>)
- [Eric's Projects - Z-80 SPI](https://sites.google.com/view/erics-projects/z-80-projects-page/z-80-spi) (<https://sites.google.com/view/erics-projects/z-80-projects-page/z-80-spi>)
- [Introduction to Bit Banging: SPI communication in Arduino via Bit Banging](https://circuitdigest.com/article/introduction-to-bit-banging-spi-communication-in-arduino-via-a-bit-banging) (<https://circuitdigest.com/article/introduction-to-bit-banging-spi-communication-in-arduino-via-a-bit-banging>)
- [electricimp/SoftwareSPI: Bit Bang SPI library](https://github.com/electricimp/SoftwareSPI) (<https://github.com/electricimp/SoftwareSPI>)
- [RetroBrew Computers Forum: General Discussion » Z80 SPI fast SD](https://www.retrobrewcomputers.org/forum/index.php?t=msg&th=581&goto=8520&) (<https://www.retrobrewcomputers.org/forum/index.php?t=msg&th=581&goto=8520&>)
- [An SPI Master Controller Interface Demo using a Cyclone IV FPGA](http://s100computers.com/My%20System%20Pages/FPGA%20Board%20V2/V2_FPGA_CIRCUITS/SPI_Master_2/SPI_Master_2.htm) ([http://s100computers.com/My%20System%20Pages/FPGA%20Board%20V2/V2\\_FPGA\\_CIRCUITS/SPI\\_Master\\_2/SPI\\_Master\\_2.htm](http://s100computers.com/My%20System%20Pages/FPGA%20Board%20V2/V2_FPGA_CIRCUITS/SPI_Master_2/SPI_Master_2.htm))
- [Overview of Linux kernel SPI support — The Linux Kernel documentation](https://www.kernel.org/doc/html/latest/spi/spi-summary.html) (<https://www.kernel.org/doc/html/latest/spi/spi-summary.html>)
- [Z-One, a Z80 based CP/M machine with SD card storage](http://www.nomad.ee/micros/z-one/index.shtml) (<http://www.nomad.ee/micros/z-one/index.shtml>)
- [Micro SD Card Interfacing Tutorial – HandsOn Tech](https://handsontec.com/index.php/micro-sd-card-interfacing-tutorial/) (<https://handsontec.com/index.php/micro-sd-card-interfacing-tutorial/>)
- [In-Depth Tutorial to Interface Micro SD Card Module with Arduino](https://lastminuteengineers.com/arduino-micro-sd-card-module-tutorial/) (<https://lastminuteengineers.com/arduino-micro-sd-card-module-tutorial/>)
- [AZDelivery 3 x SPI Reader Micro Storage SD TF Card Memory Card Shield Module Kompatibel med Arduino inklusive e-bok!: Amazon.se: Industrial](https://www.amazon.se/dp/B077MCQS9P/ref=asc_df_B077MCQS9P/?tag=shpngadsglede-21&linkCode=df0&hvadid=476555136704&hvpos=&hvnetw=g&hvrand=771615078225805494&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcndl=&hvlocint=&hvlocphy=9062332&hvtargid=pla-666709036717&psc=1) ([https://www.amazon.se/dp/B077MCQS9P/ref=asc\\_df\\_B077MCQS9P/?tag=shpngadsglede-21&linkCode=df0&hvadid=476555136704&hvpos=&hvnetw=g&hvrand=771615078225805494&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcndl=&hvlocint=&hvlocphy=9062332&hvtargid=pla-666709036717&psc=1](https://www.amazon.se/dp/B077MCQS9P/ref=asc_df_B077MCQS9P/?tag=shpngadsglede-21&linkCode=df0&hvadid=476555136704&hvpos=&hvnetw=g&hvrand=771615078225805494&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcndl=&hvlocint=&hvlocphy=9062332&hvtargid=pla-666709036717&psc=1))
- [How to Use MMC/SDC](http://elm-chan.org/docs/mmc/mmc_e.html) ([http://elm-chan.org/docs/mmc/mmc\\_e.html](http://elm-chan.org/docs/mmc/mmc_e.html))
- [Robot\\_Control/SdCard.h at master · arduino-libraries/Robot\\_Control](https://github.com/arduino-libraries/Robot_Control/blob/master/src/SdCard.h) ([https://github.com/arduino-libraries/Robot\\_Control/blob/master/src/SdCard.h](https://github.com/arduino-libraries/Robot_Control/blob/master/src/SdCard.h))

Why not a SD card with WiFi?

- [Guide to the Best WiFi Enabled SD Memory Card 2020](https://nerdtechy.com/best-wifi-sd-card) (<https://nerdtechy.com/best-wifi-sd-card>)
- [32GB ez Share WiFi SDHC Class 10 - Köp på 24.se](https://www.24.se/-minneskort/minneskort/secure-digital-sdsdhcsdxc/32gb-ez-share-wifi-sdhc-class-10) (<https://www.24.se/-minneskort/minneskort/secure-digital-sdsdhcsdxc/32gb-ez-share-wifi-sdhc-class-10>)

- [ez Share Apps User Manual\\_V2.0\\_EN.pdf](https://awesome.nwgat.ninja/ezshare/ez%20Share%20Apps%20User%20Manual_V2.0_EN.pdf) ([https://awesome.nwgat.ninja/ezshare/ez%20Share%20Apps%20User%20Manual\\_V2.0\\_EN.pdf](https://awesome.nwgat.ninja/ezshare/ez%20Share%20Apps%20User%20Manual_V2.0_EN.pdf))
- [Cdrox Micro TF Converter Wifi SD Card Adapter High Quality Camera Photos Wireless Transmit Replacement for Canon: Amazon.se: Electronics](https://www.amazon.se/Cdrox-Converter-Wireless-Transmit-Replacement/dp/B08HD2X3XV/ref=sr_1_1?dchild=1&keywords=sd+card+wifi&qid=1613309299&sr=8-1) ([https://www.amazon.se/Cdrox-Converter-Wireless-Transmit-Replacement/dp/B08HD2X3XV/ref=sr\\_1\\_1?dchild=1&keywords=sd+card+wifi&qid=1613309299&sr=8-1](https://www.amazon.se/Cdrox-Converter-Wireless-Transmit-Replacement/dp/B08HD2X3XV/ref=sr_1_1?dchild=1&keywords=sd+card+wifi&qid=1613309299&sr=8-1))
- [LANMU Micro SD till SD-kort förlängningskabel TF minneskort adapter förlängare konverterare för SD/RS-MMC/SDHC/MMC/bil GPS/mini 3D skrivare: Amazon.se: Electronics](https://www.amazon.se/LANMU-Micro-SD-till-SD-kort-förlängningskabel-TF-minneskort-adapter-förlängare-konverterare-för-SD/RS-MMC/SDHC/MMC/bil-GPS/mini-3D-skrivare:Amazon.se-Electronics) (<https://www.amazon.se/LANMU-Micro-SD-till-SD-kort-förlängningskabel-TF-minneskort-adapter-förlängare-konverterare-för-SD/RS-MMC/SDHC/MMC/bil-GPS/mini-3D-skrivare:Amazon.se-Electronics>)

Maybe a Compact Flash disk?

- [Z80 8-bit Compact Flash Card Interface | Z80 Computer Project](https://z80project.wordpress.com/2015/07/06/z80-8-bit-compact-flash-card-interface-part-1/) (<https://z80project.wordpress.com/2015/07/06/z80-8-bit-compact-flash-card-interface-part-1/>)
- [8-bit compact flash interface < waveguide.se](http://www.waveguide.se/?article=8-bit-compact-flash-interface) (<http://www.waveguide.se/?article=8-bit-compact-flash-interface>)

AVR and USB

- [Weeks 11-12: AVR USB Devices and Programming](http://fab.cba.mit.edu/classes/863.14/people/andrew_mao/week11/) ([http://fab.cba.mit.edu/classes/863.14/people/andrew\\_mao/week11/](http://fab.cba.mit.edu/classes/863.14/people/andrew_mao/week11/))
- [V-USB - A Firmware-Only USB Driver for Atmel AVR Microcontrollers](https://www.obdev.at/products/vusb/index.html) (<https://www.obdev.at/products/vusb/index.html>)

## Disk drive over serial interface (or parallel port) ?

---

2021-08-31

- [cp m - How viable is a universal PC parallel/serial connected CP/M disk controller/drive? - Retrocomputing Stack Exchange](https://retrocomputing.stackexchange.com/questions/8735/how-viable-is-a-universal-pc-parallel-serial-connected-cpm-disk-controller-drive) (<https://retrocomputing.stackexchange.com/questions/8735/how-viable-is-a-universal-pc-parallel-serial-connected-cpm-disk-controller-drive>)
- [Parallel Port Disk Drive for CPM](http://ve7it.cowlug.org/pphd.html) (<http://ve7it.cowlug.org/pphd.html>)

## Logic Device Programming

PLD/GAL could be good for logic around the CPU and peripherals.

PAL/GAL Programming

## KiCAD as EDA tool for Z80 computer design

Electronic schematic and PCB design software

Tutorials on KiCAD: KiCad 5 Tutorial - YouTube (<https://www.youtube.com/playlist?list=PL3by7evD3F51fKkyrUbH-PCdwPCWc9F8a>)

2020-11-21

Installed on Perceval (Acer laptop) and LinuxServer:

- [Acer\\_Laptop#Install\\_KiCad](#) (as the old system crashed this installation does not exist any longer).
- [LinuxServer\\_-\\_Linux\\_Mint\\_Server#KiCad\\_EDA](#)

Some Z80 schematic symbols are available in the distribution, more are available in:

- [flypie/Z80-CPU-for-KiCAD: Z80 & Z180 CPU for KiCAD DIP/DIL/PLCC/QFP \(https://github.com/flypie/Z80-CPU-for-KiCAD\)](#)
- [How to get a downloaded symbol, footprint or full library into KiCad version 5? - FAQ - KiCad.info Forums \(https://forum.kicad.info/t/how-to-get-a-downloaded-symbol-footprint-or-full-library-into-kicad-version-5/19485\)](#)

2021-01-17

Get started with Z80 computer design using KiCAD

Installed KiCAD again on Perceval  
[Electronic\\_schematic\\_and\\_PCB\\_design\\_software#KiCad\\_installation\\_on\\_Perceval\\_Linux\\_Mint](#)

Started KiCAD and created project /home/hal/KiCAD/Z80\_computer/Z80\_computer.pro

Opened Eeschema and placed Z80CPU and Z80PIO.

Cloned other Z80 family symbols:

```
hal@Perceval:~/sourcecode$ git clone https://github.com/flypie/Z80-CPU-for-KiCAD.git
Cloning into 'Z80-CPU-for-KiCAD'...
remote: Enumerating objects: 142, done.
remote: Total 142 (delta 0), reused 0 (delta 0), pack-reused 142
Receiving objects: 100% (142/142), 153.48 KiB | 734.00 KiB/s, done.
Resolving deltas: 100% (94/94), done.
```

In Eeschema open Symbol Editor, File -> Add Library, open ~/sourcecode/Z80-CPU-for-KiCAD/Zilog\_Z80\_Peripherals.lib This will import several Z80 peripherals.

2021-03-28

In Eeschema open Symbol Editor, File -> Add Library, open ~/sourcecode/Z80-CPU-for-KiCAD/Zilog\_Z80.lib

Opened Eeschema and placed Z80SIO0.

Later moved this library to ~/KiCAD/symbols/Z80

2021-04-08

Downloaded from: [Memory\\_RAM \(https://kicad.github.io/symbols/Memory\\_RAM\)](#) Rename to Memory\_SRAM otherwise name conflict. Put in ~/KiCAD/symbols/Memory\_RAM

In Eeschema open Symbol Editor, File -> Add Library, open ~/KiCAD/symbols/Memory\_RAM/Memory\_SRAM.lib

2021-04-09

Made symbol for BS62LV256 based on KM62256CLP in Memory\_SRAM lib.

Made symbol for ATF22V10C and put in ATF lib, based on GAL16V8 in Logic\_Programmable lib.

2021-04-20

When trying to create a BOM an error message was given. It seems that the KiCAD BOM scripts are using "python" instead of "python3".

Installed Python-is-python3 with Software Manager.

Now the BOM scripts work.

## Z80 Software

### Z80 Monitor and OS

---

- Retro Software :: Z80 monitor, OS and utilities

2021-02-14

- Porting CP/M to a new computer (<http://cpuville.com/Code/CPM-on-a-new-computer.html>)

### Handling Z80 MMU / overlays

---

2021-02-13

Should the Z80 computer implement a simple MMU? There is support in the GNU linker for this:

- Using LD, the GNU linker - Overlays ([https://ftp.gnu.org/old-gnu/Manuals/ld-2.9.1/html\\_node/ld\\_22.html](https://ftp.gnu.org/old-gnu/Manuals/ld-2.9.1/html_node/ld_22.html))

### Other Z80 Programs

---

Updated 2021-05-27

- Build a Clang-LLVM Z80 Compiler
- Home · z88dk/z88dk Wiki (<https://github.com/z88dk/z88dk/wiki>)
  - library:interrupts [z88dk] (<https://www.z88dk.org/wiki/doku.php?id=library:interrupts>)
- R.T. Russell's Z80 BBC Basic is now open source (<http://cowlark.com/2019-06-14-bbcbasic-opensource/index.html>)
  - Github - sehugg/zmac: zmac z80 assembler (<https://github.com/sehugg/zmac>)
  - zmac - Z-80 Macro Cross Assembler (<http://48k.ca/zmac.html>)
- Retro Software :: CP/M Programs

- [Math - z80 Heaven \(http://z80-heaven.wikidot.com/math\)](http://z80-heaven.wikidot.com/math)

# Memory mapping for Z80 (and other 16 bit address CPUs)

2021-06-16

Using a 74LS670 to expand the 16 bit address to an 18 bit address with the logical page size of 16 kB.

4-by-4 Register Files With 3-State Outputs datasheet - sn54ls670.pdf ([https://www.ti.com/lit/ds/symlink/sn54ls670.pdf?ts=1623832555462&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/symlink/sn54ls670.pdf?ts=1623832555462&ref_url=https%253A%252F%252Fwww.google.com%252F)).

The IC is available at Svebry: [74LS670 \(https://www.svebry.se/product/74ls670\)](https://www.svebry.se/product/74ls670).

## Z80 References

2021-08-28

- [Full Z80 Opcode List Including Undocumented Opcodes \(http://mdfs.net/Docs/Comp/Z80/OpList\)](http://mdfs.net/Docs/Comp/Z80/OpList)

## Links to Z80 Projects

Updated 2021-09-01

- [Retro Z80 computer with CP/M-80 \(http://www.stuartball.com/retroz80/RetroZ80.htm\)](http://www.stuartball.com/retroz80/RetroZ80.htm)
- [Porting CP/M to a new computer \(http://cpuville.com/Code/CPM-on-a-new-computer.html\)](http://cpuville.com/Code/CPM-on-a-new-computer.html)
- [Yet Another Z180 Project \(YAZ180\) | feilipu \(https://feilipu.me/2016/05/23/another-z80-project/\)](https://feilipu.me/2016/05/23/another-z80-project/)
- [RC2014 - Homebrew Z80 Computer \(https://rc2014.co.uk/\)](https://rc2014.co.uk/)
- [RC2014Z80/RC2014: Official RC2014 repository for hardware schematics / software and more! \(https://github.com/RC2014Z80/RC2014\)](https://github.com/RC2014Z80/RC2014)
- [Lumir Vanek - RC2014 \(http://www.valachnet.cz/ivanek/diy/rc2014/index.html\)](http://www.valachnet.cz/ivanek/diy/rc2014/index.html)  
thorough descriptions of building a Z80 based computer. Also contains test program for DMA
- [How to Build a Z80 Computer, Part 1: The CPU | PIC | Maker Pro \(https://maker.pro/pic/projects/z80-computer-project-part-1-the-cpu\)](https://maker.pro/pic/projects/z80-computer-project-part-1-the-cpu)
- [How to Build a Z80 Computer, Part 2: Memory | PIC | Maker Pro \(https://maker.pro/pic/projects/build-your-own-z80-computer-project-part-2-memory\)](https://maker.pro/pic/projects/build-your-own-z80-computer-project-part-2-memory)
- [How to Build a Z80 Computer, Part 3: EEPROM Programmer | PIC | Maker Pro \(https://maker.pro/pic/projects/z80-project-part-3-eeprom-programmer\)](https://maker.pro/pic/projects/z80-project-part-3-eeprom-programmer)
- [Build a Z80 Computer, Part 4: IO & Coding Your First Program | PIC | Maker Pro \(https://maker.pro/pic/projects/z80-project-part-4-basic-io-and-writing-your-first-program\)](https://maker.pro/pic/projects/z80-project-part-4-basic-io-and-writing-your-first-program)

- [Teensy Z80 – Part 1 – Intro, Memory, Serial I/O and Display | "Domipheus Labs"](http://labs.domipheus.com/blog/teensy-z80-part-1-intro-memory-serial-io-and-display/) (<http://labs.domipheus.com/blog/teensy-z80-part-1-intro-memory-serial-io-and-display/>)
- [Harte Technologies: CP/M-80 Information and Download Page](http://www.cpm80.com/) (<http://www.cpm80.com/>)
- [S100 Computers -Z80 CPU Board](http://www.s100computers.com/My%20System%20Pages/Z80%20Board/Z80%20CPU%20Board.htm) (<http://www.s100computers.com/My%20System%20Pages/Z80%20Board/Z80%20CPU%20Board.htm>)
- [Z180 Memory Management](http://www.ganssle.com/articles/ammu.htm) (<http://www.ganssle.com/articles/ammu.htm>)
- [One Week Wonder: Z180 MMu-tiny](http://oneweekwonder.blogspot.com/2017/12/z180-mm-tiny.html) (<http://oneweekwonder.blogspot.com/2017/12/z180-mm-tiny.html>)
- [Z80 C code development with Eclipse and z88dk | feilipu](https://feilipu.me/2016/09/16/z80-c-code-development-with-eclipse-and-z88dk/) (<https://feilipu.me/2016/09/16/z80-c-code-development-with-eclipse-and-z88dk/>)
- [A homebrew Z80 microcomputer - Wenzel Jakob, May 2003](https://www.mitsuba-rendner.org/~wenzel/z80/index.html) (<https://www.mitsuba-rendner.org/~wenzel/z80/index.html>)
- [Build Your Own Z80 Computer: Design Guidelines and Application Notes - Steve Ciarcia - Google Böcker](https://books.google.se/books/about/Build_Your_Own_Z80_Computer.html?id=mVQnFgWzX0AC&redir_esc=y) ([https://books.google.se/books/about/Build\\_Your\\_Own\\_Z80\\_Computer.html?id=mVQnFgWzX0AC&redir\\_esc=y](https://books.google.se/books/about/Build_Your_Own_Z80_Computer.html?id=mVQnFgWzX0AC&redir_esc=y))
- [5V Logic on a Z80 Retrocomputer | Details | Hackaday.io](https://hackaday.io/project/167522-5v-logic-on-a-z80-retrocomputer/details) (<https://hackaday.io/project/167522-5v-logic-on-a-z80-retrocomputer/details>)
- [reZet80/reZet80-DEV: reZet80 DEV - a development system for the low-level Z80 retro enthusiast](https://github.com/reZet80/reZet80-DEV) (<https://github.com/reZet80/reZet80-DEV>)
- [builderpages:plasmo:riz180:riz180r1 \[RetroBrew Computers Wiki\]](https://www.retrobrewcomputers.org/doku.php?id=builderpages:plasmo:riz180:riz180r1) (<https://www.retrobrewcomputers.org/doku.php?id=builderpages:plasmo:riz180:riz180r1>)
- [Z80 Explorer - Baltazar Studios](https://baltazarstudios.com/z80explorer/) (<https://baltazarstudios.com/z80explorer/>)
- [Zilog Z-80 DIY | Facebook](https://www.facebook.com/groups/546186508918605/permalink/1571987366338509/) (<https://www.facebook.com/groups/546186508918605/permalink/1571987366338509/>)
- [ZAP Z80 Alternative Memories - Page 1](https://www.eevblog.com/forum/projects/zap-z80-alternative-memories/) (<https://www.eevblog.com/forum/projects/zap-z80-alternative-memories/>)
- [A tiny Z80 based computer](http://www.vaxman.de/projects/tiny_z80/) ([http://www.vaxman.de/projects/tiny\\_z80/](http://www.vaxman.de/projects/tiny_z80/))
- [dekuNukem/FAP80: A Z80 retro computer without the retro baggage](https://github.com/dekuNukem/FAP80) (<https://github.com/dekuNukem/FAP80>)
- [reZet80](https://reZet80.blogspot.com/) (<https://reZet80.blogspot.com/>)
- [Zalt/Under Construction/IDE v0.2 at master · obianjacobi/Zalt · GitHub](https://github.com/obianjacobi/Zalt/tree/master/Under%20Construction/IDE%20v0.2) (<https://github.com/obianjacobi/Zalt/tree/master/Under%20Construction/IDE%20v0.2>)
- [Building CERBERUS 2080, my dream childhood computer - YouTube](https://m.youtube.com/playlist?list=PLDf2ukIC_d2DAXmF9XuOq_-uNc2M9ITd) ([https://m.youtube.com/playlist?list=PLDf2ukIC\\_d2DAXmF9XuOq\\_-uNc2M9ITd](https://m.youtube.com/playlist?list=PLDf2ukIC_d2DAXmF9XuOq_-uNc2M9ITd))
- [Homebrew Z80 Computer \(Part 5\) - L Break Into Program](http://www.breakintoprogram.co.uk/projects/homebrew-z80/homebrew-z80-computer-part-5) (<http://www.breakintoprogram.co.uk/projects/homebrew-z80/homebrew-z80-computer-part-5>)
- [Casio MX-10 Future Upgrade – Retro Depot](http://retrodepot.net/?p=12593) (<http://retrodepot.net/?p=12593>)
- [Downloads – Retro Depot](https://retrodepot.net/?page_id=510) ([https://retrodepot.net/?page\\_id=510](https://retrodepot.net/?page_id=510))
- [reZet80: Welcome to reZet80!](https://reZet80.blogspot.com/2020/05/welcome-to-reZet80.html) (<https://reZet80.blogspot.com/2020/05/welcome-to-reZet80.html>)
- [linker3000/Z80-Board: Z80 computer wirewrapped on perfboard + Some game and utility programs](https://github.com/linker3000/Z80-Board) (<https://github.com/linker3000/Z80-Board>)
- [ZED80\\_Index\\_SBC](http://zed80.com/Z80-RETRO/index_Home.html) ([http://zed80.com/Z80-RETRO/index\\_Home.html](http://zed80.com/Z80-RETRO/index_Home.html))
- [CERBERUS 2080™ | The Byte Attic™](https://www.thebyteattic.com/p/cerberus-2080.html) (<https://www.thebyteattic.com/p/cerberus-2080.html>)

- [Build a Z80 Computer from Scratch - YouTube](https://www.youtube.com/playlist?app=desktop&list=PL_ysBkGwx4rkhgj6vFoWrfl70GmK11n) ([https://www.youtube.com/playlist?app=desktop&list=PL\\_ysBkGwx4rkhgj6vFoWrfl70GmK11n](https://www.youtube.com/playlist?app=desktop&list=PL_ysBkGwx4rkhgj6vFoWrfl70GmK11n))
- [FTP between host and guest | FreeDOS images for VirtualBox](https://www.lazybrowndog.net/freedos/virtualbox/?page_id=157) ([https://www.lazybrowndog.net/freedos/virtualbox/?page\\_id=157](https://www.lazybrowndog.net/freedos/virtualbox/?page_id=157))
- [Cyber-Necromancy](https://retro.zen-room.org/morrow-micronix) (<https://retro.zen-room.org/morrow-micronix>)
- [Cyber-Necromancy - Whitesmith's C](https://retro.zen-room.org/morrow-micronix/witesmiths-c) (<https://retro.zen-room.org/morrow-micronix/witesmiths-c>)
- [cm68/micronix: Morrow Designs Micronix and tools](https://github.com/cm68/micronix) (<https://github.com/cm68/micronix>)
- [GitHub - ntate6630/SC126\\_to\\_MPFI1B\\_Project: MPF-1B add-on for the SC126](https://github.com/ntate6630/SC126_to_MPFI1B_Project) ([http://github.com/ntate6630/SC126\\_to\\_MPFI1B\\_Project](https://github.com/ntate6630/SC126_to_MPFI1B_Project))
- [MatthewWCook/Z80Project: Z80 Computer Project](https://github.com/MatthewWCook/Z80Project) (<https://github.com/MatthewWCook/Z80Project>)
- [joelang/z80-sbc: Z80 single board computer with CP/M 2.2](https://github.com/joelang/z80-sbc) (<https://github.com/joelang/z80-sbc>)
- [Designing the RhoCoCo Retro Home COlor COmputer hardware - RevSpace](https://revspace.nl/Designing_the_RhoCoCo_Retro_Home_COlOr_COmputer_hardware) ([https://revspace.nl/Designing\\_the\\_RhoCoCo\\_Retro\\_Home\\_COlOr\\_COmputer\\_hardware](https://revspace.nl/Designing_the_RhoCoCo_Retro_Home_COlOr_COmputer_hardware))
- [Project Gibson](https://www.trashworldnews.com/project-gibson/) (<https://www.trashworldnews.com/project-gibson/>)
- [Porting CP/M to a new computer](http://cpuville.com/Code/CPM-on-a-new-computer.html) (<http://cpuville.com/Code/CPM-on-a-new-computer.html>)
- [Having fun with CP/M on a Z80 single-board computer.](https://blog.steve.fi/having_fun_with_cp_m_on_a_z80_single_board_computer_.html) ([https://blog.steve.fi/having\\_fun\\_with\\_cp\\_m\\_on\\_a\\_z80\\_single\\_board\\_computer\\_.html](https://blog.steve.fi/having_fun_with_cp_m_on_a_z80_single_board_computer_.html))
- [Z-One, a Z80 based CP/M machine with SD card storage](http://www.nomad.ee/micos/z-one/index.shtml) (<http://www.nomad.ee/micos/z-one/index.shtml>)
  - [mastmees/z-one: Z-One is a CP/M machine with SD card storage, 80x30 character VGA output, and PS/2 keyboard](https://github.com/mastmees/z-one) (<https://github.com/mastmees/z-one>)

## Z80 related components

### The following components are/will be available

---

- [1x Z80-CPU \(40 pin DIL\) Zilog Z80-CPU 2.5MHz Zilog Z80 microprocessor family](https://www.cpu-world.com/CPUs/Z80/MANUF-Zilog.html) (<https://www.cpu-world.com/CPUs/Z80/MANUF-Zilog.html>)
- [1x Z80A SIO/0 \(40 pin DIL\) Zilog Z-80A SIO/0 / 8440W 0A 4MHz Support/peripheral/other chips - Z80 family](https://www.cpu-world.com/Support/Z80.html) (<https://www.cpu-world.com/Support/Z80.html>)
- [1x Z80A-CTC \(28 pin DIL\) Zilog Z80A-CTC 4MHz Support/peripheral/other chips - Z80 family](https://www.cpu-world.com/Support/Z80.html) (<https://www.cpu-world.com/Support/Z80.html>)
- [6x Z80A SIO-0 \(40 pin DIL\) SGS Z8440AB1 / Z80ASIO-0 4MHz Support/peripheral/other chips - Z80 family](https://www.cpu-world.com/Support/Z80.html) (<https://www.cpu-world.com/Support/Z80.html>)
- [6x Z80-PIO \(40 pin DIL\) Mostek MK3881N-4 4MHz Support/peripheral/other chips - Z80 family](https://www.cpu-world.com/Support/Z80.html) (<https://www.cpu-world.com/Support/Z80.html>)
- [6x Z80A CTC \(28 pin DIL\) Zilog Z8430A PS / Z80A CTC 4 MHz Support/peripheral/other chips - Z80 family](https://www.cpu-world.com/Support/Z80.html) (<https://www.cpu-world.com/Support/Z80.html>)
- [6x Z80A DMA \(40 pin DIL\) SGS Z8410AB1 / Z80ADMA 4 MHz Support/peripheral/other chips - Z80 family](https://www.cpu-world.com/Support/Z80.html) (<https://www.cpu-world.com/Support/Z80.html>)
- [2x Z180 \(68 pin PLCC\) Zilog Z8018010VSC / Z180 MPU 10MHz Zilog Z8018010VSC](https://www.cpu-world.com/Support/Z80.html)

(<https://www.cpu-world.com/CPUs/Z180/Zilog-Z8018010VSC.html>)

- 1x 64B180 (Z180)(64 pin DIL with adapter board) Hitachi HD64180ROP 8MHz?  
Hitachi HD64180RP8 (<https://www.cpu-world.com/CPUs/Z180/Hitachi-HD64180RP8.html>)

EPROM:

- 2x M27C2001-10F1 (32 pin DIL) ST M27C2001 100nS M27C2001 Datasheet pdf - 2 MBIT (256KB X8) UV EPROM AND OTP ROM - SGS Thomson Microelectronics ([http://www.datasheetcatalog.com/datasheets\\_pdf/M/2/7/C/M27C2001.shtml](http://www.datasheetcatalog.com/datasheets_pdf/M/2/7/C/M27C2001.shtml))
- 6x M27C256B-12F1 (28 pin DIL) ST M27C256 120nS M27C256B.pdf (<http://pdf.datasheetcatalog.com/datasheet/stmicroelectronics/2384.pdf>)
- 12x 27C64-15 (28 pin DIL) SGS 27C64 150nS 27C64 datasheet - 27C64.PDF (<http://pdf.datasheetcatalog.com/datasheet/nationalsemiconductor/DS010331.PDF>)

SRAM:

- 2x BS62LV1024PC-70 (32 pin DIL) BSI BS62LV1024PC-70 70nS BS62LV1024 datasheet (<https://datasheetspdf.com/pdf-file/123572/BrillianceSemiconductor/BS62LV1024/1>)
- 12x BS62LV256PC-70 (28 pin DIL) BSI BS62LV256PC-70 70nS BS62LV256-2.6 - BS62LV256SIP55-Brilliance-Semiconductor-datasheet-11474061.pdf (<https://datasheet.octopart.com/BS62LV256SIP55-Brilliance-Semiconductor-datasheet-11474061.pdf>)

Z80 CPU to buy, sockets etc:

- Inbyggda - mikroprocessorer | Integrerade kretsar (ICs) | DigiKey (<https://www.digkey.se/products/sv/integrated-circuits-ics/embedded-microprocessors/694?k=zilog%20z80>)

Other components:

- Crystal oscillator SG-615\_531\_E187.pdf - SG-531P\_en.pdf ([https://support.epson.biz/td/api/doc\\_check.php?dl=brief\\_SG-531P&lang=en](https://support.epson.biz/td/api/doc_check.php?dl=brief_SG-531P&lang=en))
- Timer for power on reset: LM555 LM555 Timer datasheet (Rev. D) - lm555.pdf (<https://www.ti.com/lit/ds/symlink/lm555.pdf>)

## Components ordered from DigiKey

---

Ordered 2020-11-12, received 2020-11-17.

- Z80: 10x Z84C0010PEG Zilog | Integrerade kretsar (ICs) | DigiKey (<https://www.digkey.se/product-detail/sv/Z84C0010PEG/269-3898-ND/929206/?itemSeq=344728659>)
- 10 MHz Crystal oscillator: 1x SG-531P 10.0000MC:ROHS EPSON | Kristaller, oscillatorer, resonatorer | DigiKey (<https://www.digikey.se/product-detail/sv/SG-531P+10.0000MC%3aROHS/SER1208-ND/1021904/?itemSeq=344732078>)
- 4 MHz Crystal oscillator: 3x SG-531P 4.0000MC: ROHS EPSON | Kristaller, oscillatorer, resonatorer | DigiKey (<https://www.digikey.se/product-detail/sv/SG-531P+4.0000MC%3a+ROHS/SER1203-ND/1021899/?itemSeq=345018179>)

- MAX232 RS-232 Transciever: 6x MAX232N Texas Instruments | Integrerade kretsar (ICs) | DigiKey (<https://www.digikey.se/product-detail/sv/texas-instruments/MAX232N/296-1402-5-ND/277048>)
- 40pin DIP socket: 20x A 40-LC-TR Assmann WSW Components | Kontaktdon | DigiKey (<https://www.digikey.se/product-detail/sv/assmann-wsw-components/A-40-LC-TR/AE10010-ND/821764>)
- 32pin DIP socket: 10x A 32-LC-TR Assmann WSW Components | Kontaktdon | DigiKey (<https://www.digikey.se/product-detail/sv/A+32-LC-TR/123-A32-LC-TR-ND/821761/?itemSeq=345018521>)
- 28pin DIP socket: 20x A 28-LC-TR Assmann WSW Components | Kontaktdon | DigiKey (<https://www.digikey.se/product-detail/sv/A+28-LC-TR/AE10006-ND/821760/?itemSeq=345018474>)
- 68 (4 x 17) Positioner PLCC Uttag Tenn Genomgående hål: 3x PLCC-68-AT Adam Tech | Kontaktdon | DigiKey (<https://www.digikey.se/product-detail/sv/adam-tech/PLCC-68-AT/2057-PLCC-68-AT-ND/9833042>)
- 22V10 PLDer (programmerbar logikenhet) IC 10 Makroceller 24-PDIP: 6x ATF22V10C-15PU Microchip Technology | Integrerade kretsar (ICs) | DigiKey (<https://www.digikey.se/product-detail/sv/microchip-technology/ATF22V10C-15PU/ATF22V10C-15PU-ND/1008580>)
- 14 (2 x 7) Positioner DIP, 0,3" (7,62mm) radavstånd Uttag Tenn Genomgående hål: 10x A 14-LC-TT Assmann WSW Components | Kontaktdon | DigiKey (<https://www.digikey.se/products/sv?keywords=A%2014-LC-TT>)
- 16 (2 x 8) Positioner DIP, 0,3" (7,62mm) radavstånd Uttag Tenn Genomgående hål: 10x A 16-LC-TT Assmann WSW Components | Kontaktdon | DigiKey (<https://www.digikey.se/products/sv?keywords=A%2016-LC-TT>)
- 20 (2 x 10) Positioner DIP, 0,3" (7,62mm) radavstånd Uttag Tenn Genomgående hål: 10x A 20-LC-TT Assmann WSW Components | Kontaktdon | DigiKey (<https://www.digikey.se/product-detail/sv/A+20-LC-TT/AE9998-ND/821752/?itemSeq=345022521>)
- 24 (2 x 12) Positioner DIP, 0,3" (7,62mm) radavstånd Uttag Tenn Genomgående hål: 10x A 24-LC/7-T Assmann WSW Components | Kontaktdon | DigiKey (<https://www.digikey.se/product-detail/sv/A+24-LC%2f7-T/AE10285-ND/1628667/?itemSeq=345023229>)
- 8 (2 x 4) Positioner DIP, 0,3" (7,62mm) radavstånd Uttag Tenn Genomgående hål: 5x A 08-LC-TT Assmann WSW Components | Kontaktdon | DigiKey (<https://www.digikey.se/product-detail/sv/assmann-wsw-components/A-08-LC-TT/AE9986-ND/821740>)
- USB-C (USB TYPE-C) USB 3.2 Gen 2 (USB 3.1 Gen 2, Superspeed + (USB 3.1)) Uttag Kontakt 24 Positioner Ytmontering, högervinkelad; genomgående hål: 4x 12401610E4#2A Amphenol ICC (Commercial Products) | Kontaktdon | DigiKey (<https://www.digikey.se/product-detail/sv/amphenol-icc-commercial-products/12401610E4-2A/12401610E4-2ACT-ND/5775520>)
- USB-B (USB TYPE-B) Uttag Kontakt 4 Positioner Genomgående hål, högervinkelad: 4x USB-B1HSW6 On Shore Technology Inc. | Kontaktdon | DigiKey (<https://www.digikey.se/product-detail/sv/on-shore-technology-inc/USB-B1HSW6/ED2982-ND/2677743>)
- Stiftlistkontaktdon Genomgående hål 23 positioner 0,100" (2,54mm): 10x PRPC023SAAN-RC Sullins Connector Solutions | Kontaktdon | DigiKey (<https://www.digikey.se/product-detail/sv/sullins-connector-solutions/PRPC023SAAN-RC/S1011EC-23-ND/2775231>)

---

Retrieved from '[http://192.168.42.21/mediawiki/index.php?title=Z80\\_Computer&oldid=9002](http://192.168.42.21/mediawiki/index.php?title=Z80_Computer&oldid=9002)'

---

**This page was last modified on 27 September 2021, at 12:06.**