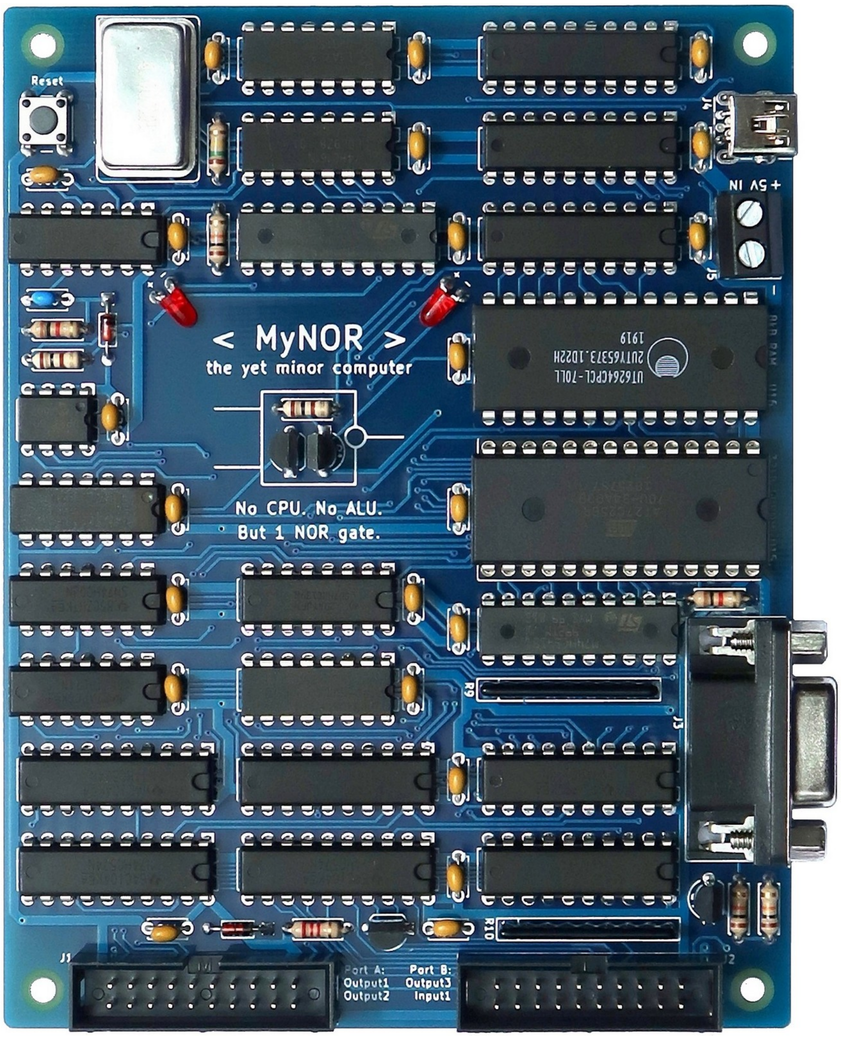
MyNOR Board Construction Manual

Dennis Kuschel
dennis_k@freenet.de
2020-12-21

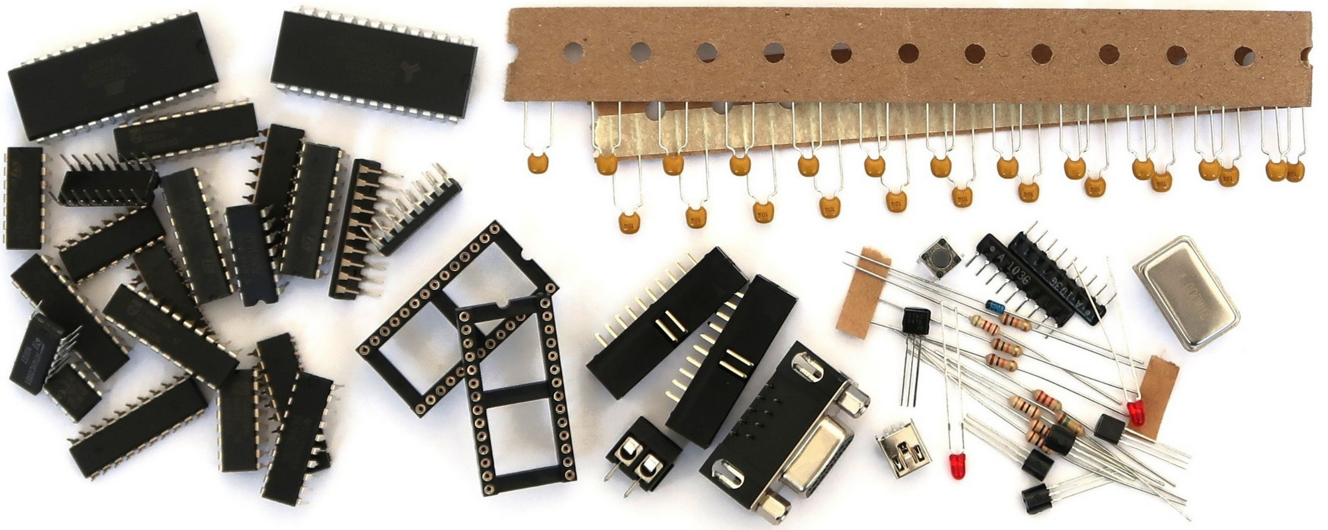
„MyNOR is a single board computer that does not have a CPU. Instead, the CPU consists of discrete logic gates from the 74HC series. This computer also has no ALU. Only a single NOR gate is used to perfom all computations such as addition, subtraction, AND, OR and XOR. This computer is not fast, it is rather slow. MyNOR can only perform 2600 8-bit additions per second, although it is clocked at 4 MHz. This is because everything is done in software. MyNOR has only a 32 kB ROM for program storage, but this is more than enough. The very slim microcode occupies only 9 kB, the remaining 23 kB are used for the application program."

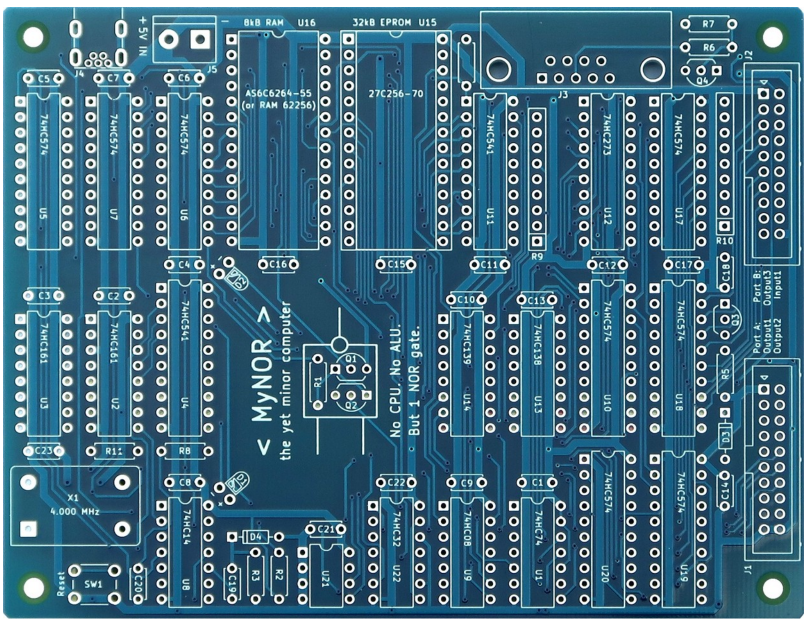
This document will help you assemble your own MyNOR computer. Visit www.mynor.org for more documentation.



Required Components

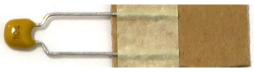
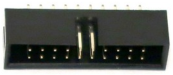
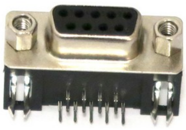
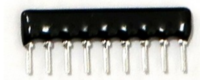
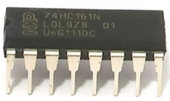
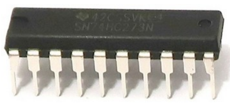
Here is an overview of the required components:

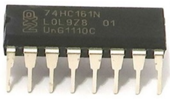
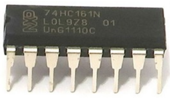
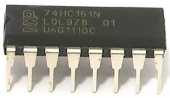
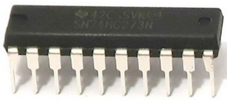
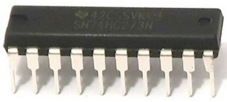
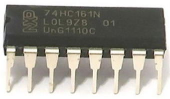
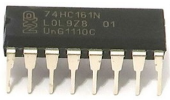
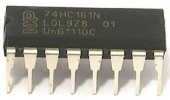
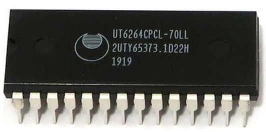
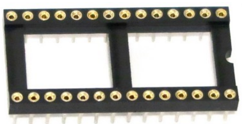
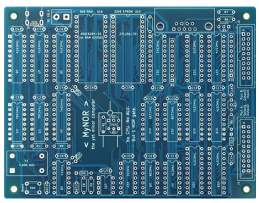




You may also use sockets for all of the ICs. This is optional, but it can simplify troubleshooting a lot:

The complete bill of material is listed in the table on the next page. Note that the table contains also the EPROM 27C256. When you are not able to „burn“ or „program“ it by yourself, please contact me, we will find a solution ;-)

Reference	Qty	Picture	Value	Mouser P/N www.mouser.com	Reichelt P/N www.reichelt.de
C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12 C13 C14 C15 C16 C17 C18 C20 C21 C22 C23	22		100 nF (X7R / 5 mm)	SR215C104K	X7R-5 100N
C19	1		82 pF (5mm)	FG28C0G2A820JNT00	KERKO-500 82P
D1 D2	2		red 3mm LED	LTL-4222N	KBT L-7104LSRD
D3 D4	2		BAT41	BAT41-TAP	BAT 41
J1 J2	2		20 pin header	710-61202021621	WSL 20G
J3	1		D-Sub DB9 (female)	LD09S33E4GX00LF	D-SUB BU 09US
J4	1		USB_B	XM7D-0512	USB BWM
J5	1		Terminal	TB006-508-02BE	AKL 057-02
Q1 Q2 Q3	3		BS170	BS170D27Z	BS 170
Q4	1		BC547B	BC547BTA	BC 547B
R1 R2	2		100	CFR-25JR-52100R	1/4W 100
R3 R4	2		1k	CFR-25JR-52-1K	1/4W 1,0K
R5	2		2k2	CFR-25JR-522K2	1/4W 2,2K
R6 R7 R8	3		10k	CFR-25JR-5210K	1/4W 10K
R11	1		150k	CFR-25JR-52150K	1/4W 150K
R9 R10	2		8x 10k	4609X-1LF-10K	SIL 9-8 10K
SW1	1		SW_Push	TL1105BF160Q	TASTER 3301
U1	1		74HC74	SN74HC74N	74HC 74
U11 U4	2		74HC541	SN74HC541N	74HC 541

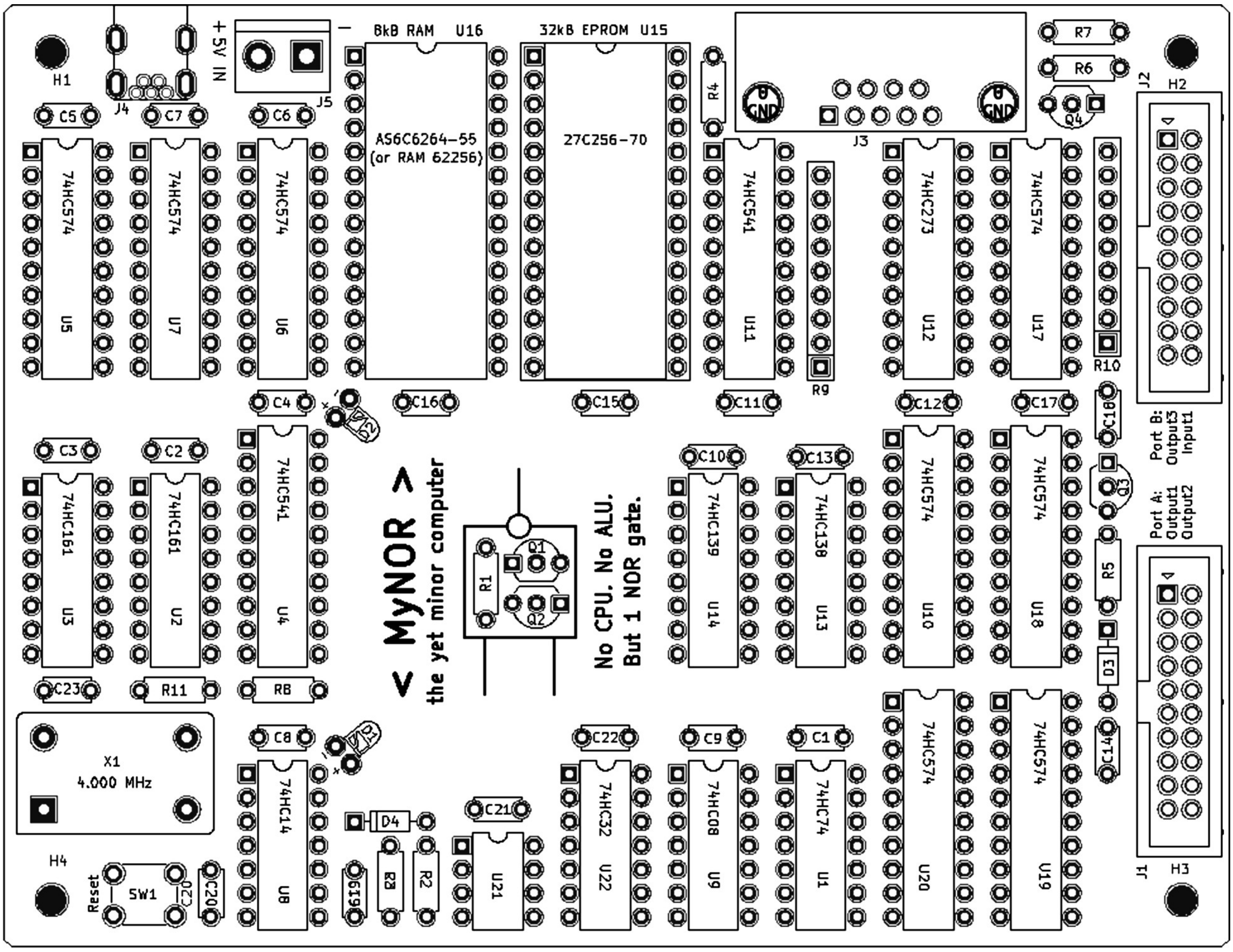
U13	1		74HC138	SN74HC138N	74HC 138
U14	1		74HC139	SN74HC139N	74HC 139
U2 U3	1		74HC161	SN74HC161N	74HC 161
U5 U6 U7 U10 U17 U18 U19 U20	8		74HC574	SN74HC574N	74HC 574
U12	1		74HC273	SN74HC273N	74HC 273
U8	1		74HC14	SN74HC14N	74HC 14
U9	1		74HC08	SN74HC08N	74HC 08
U22	1		74HC32	SN74HC32N	74HC 32
U15	1		27C256 EPROM 32kB (45 - 90 ns)	AT27C256R-70PU -or- AT27C256R-45PU	OT 27C256-90 B
U16	1		6264 SRAM 8kB (45 - 90 ns)	AS6C6264-55PCN (70ns is ok, but 55ns allows up to 10MHz clk) Alternatively: 62256	6264-70 (or 62256-80)
U21	1		24LC512	24LC512-I/P	24LC512-I/P
X1	1		4.000MHz	MXO45-3C-4.0	OSZI 4,000000
Socket	1		DIP28 for EPROM	1104732841001000	GS 28P
PCB Raw Card	1		Use provided gerber files (in zip file) and order the PCB at jlcpcb.com		

The speed of the EPROM and SRAM is not critical. I made good experiences with 70 ns or faster types. If you use chips with 70 ns or faster, you should be able to clock MyNOR also with the higher speed of 8 MHz. MyNOR can run very fast with even slower chips, but there is no guarantee.

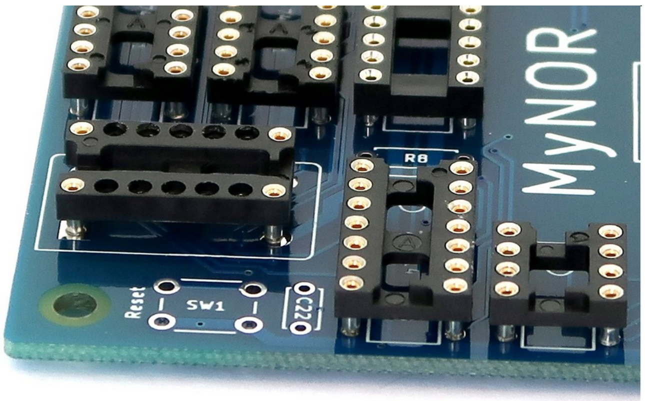
You can buy the components at mouser.com, digikey.com, reichelt.de and many other websites. Regarding the PCB raw card I made very good experiences with JLCPCB in China. Please note that the minimum order quantity is 5 boards, for a price of $8.40 plus shipping costs. That‘s really cheap!

Board Assembly

The picture below shows the position of each part. Start with soldering the low components, in the following order: USB socket, IC sockets, ICs without socket, resistors and the LEDs. After that, continue with the resistor networks R9 and R10, the capacitors and the transistors. At last, mount the switch and the connectors on the PCB.



Hint:
I recommend using a socket for the crystal oscillator X1. You can use a standard 14-pin socket and remove the unused pins by heating the pins with the soldering iron and pushing them out of the housing. The socket makes it easier to change between a 4 MHz and a 8 MHz oscillator.



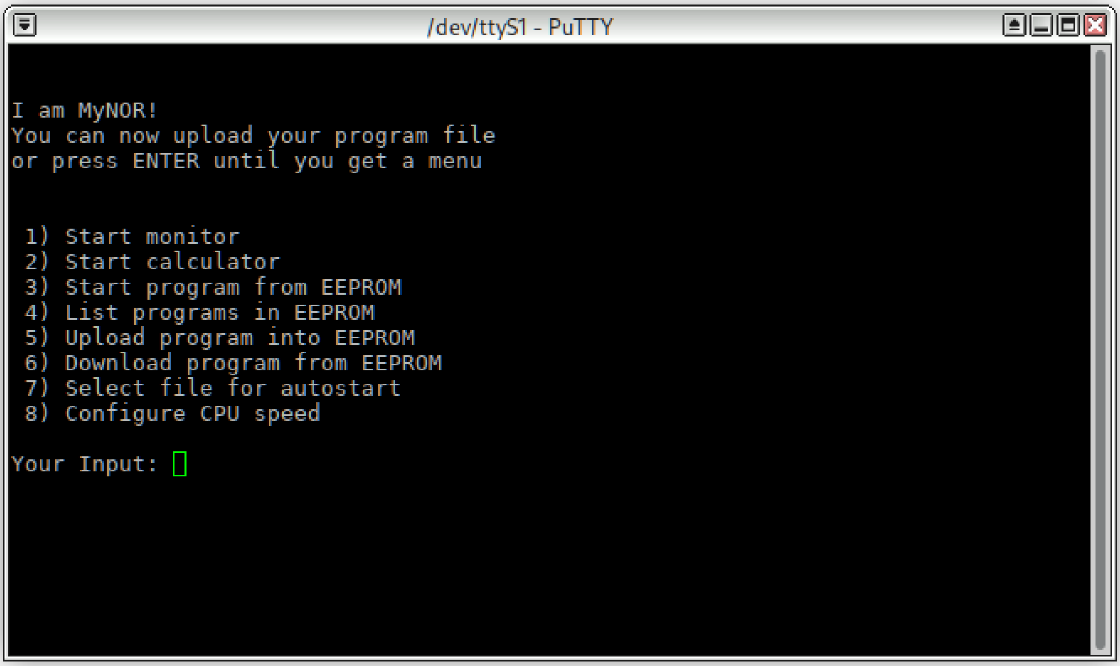
First Test

1. Review your work
Before you apply power to the MyNOR board for the first time, please check all components on the board to make sure they are the correct ones and that the orientation is correct. Also check the solder joints for missing connections or short circuits.

2. RS232 Terminal
Establish a serial connection (RS232) to a suited terminal. I recommend using the terminal emulator „putty“ on MS Windows or Linux. Set putty to 2400 baud, 8 data bits and 1 stop bit. Test the terminal emulation by shorting pins 2 and 3 of the male SUB-D9 connector: When you type something in the terminal window, the echo must be immediately visible. When everything is ok, connect the RS232 cable with the MyNOR board. Warning: Do not use a USB to RS232 converter cable with a Prolific chip. This chip is of poor quality and even at slow baud rates like 2400 baud the cable sometimes loses a byte. The slightly more expensive cables with FTDI chip are better. If possible, please use a real physical RS232 interface (provided by the mainboard of your PC or by a PCIe expansion card).

3. Power Supply
MyNOR can be powered via the 2-pin terminal block or the mini USB socket. I recommend using a „real“ 5V power supply (mains adapter). Tests have shown that the cheap USB chargers for smartphones often do not work properly. They are made to charge batteries, not for supplying devices directly. I have also tested USB powerbanks. Some powerbanks work, some not. You have to find out for yourself. With an improper power supply MyNOR may boot, but it crashes after a while.

4. Apply Power
Now switch on the power supply. Observe the terminal window. If necessary, press the reset button. The two LEDs on the board should flicker a little (they will light up completely while the reset button is pressed, but they get darker when the button is released). The terminal window should display the text shown in the image below. When you press „ENTER“ twice, you should get to the main menu. You can test the calculator integrated in MyNOR by pressing „2“.



Troubleshooting

If your MyNOR does not start at all, please check the following: Do the LEDs flicker as described above? If so, please check your terminal emulation (putty) and the serial connection. Most likely MyNOR is working as expected but you do not see its ouput in the terminal. But your MyNOR has a serious problem if the LEDs do not flicker after reset. Please check all components again, check the solder joints and make sure that your EPROM (27C256) is correctly programmed. And do not swap the SRAM with the EPROM! Any mistake can damage the part in question and also the rest of the parts, so that all ICs on the board need to be replaced. And please also check your power supply! A stable 5V supply with at least 500 mA current output is abosolutely necessary! If you are using a laboratory power supply, please try to keep the connection cables short.

Your MyNOR is working, but crashes immediately after releasing the reset button (you do not see all the text output in the terminal) or it crashes after a while? Then check the following:

1. Power Supply. The power supply must provide stable 5V (4.8V to 5.2V). The charger of your smartphone may not be the best choice. Keep the wires as short as possible.
2. Check your board for missing solder joints. If you are using sockets for the ICs, check that all pins of the ICs are correctly inserted into the socket (and that no pin is bent and has no contact).
3. Check the 4 MHz master clock. The clock must be present at all relevant pins on the board (please look into the schematics for that).
4. Check the clock on pin 8 of U8. If there is no clock, but only a static high or static low level, please replace U8. If you have one, use an IC from another manufacturer.
5. It is always a good idea to replace U8. If you do not have another 74HC14 at hand, try other values for C19 (see below).
6. Try other values for C19. Start with 100pF, then try 82pF, 68pF, 56pF and 47pF. If you have found out that your MyNOR works with 68pF, 56pF and 47pF, then take the value in the middle (56pF).

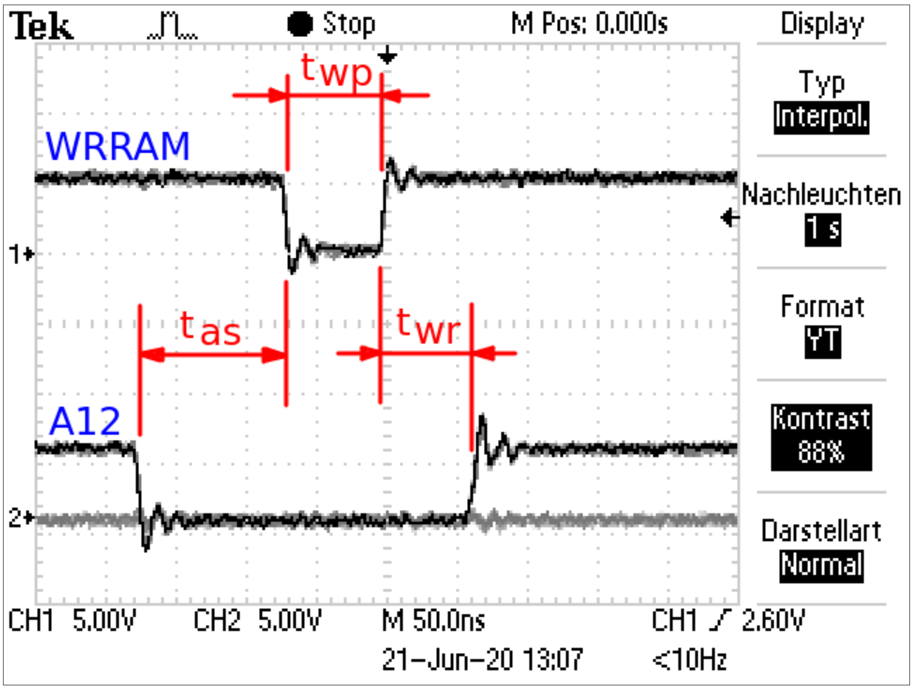
Troubleshooting: The role of R3 and C19

You may have noticed the RC network R3 and C19 in the schematics. Together with D4 this is a „waveform shaping circuit“. It is used to change the duty cycle of the clock signal that is used to generate the strobe signals for the data targets. But why is it required?

In an ideal digital circuit, all signals change their state only on the rising (or falling) edge of the clock signal. This presupposes that all input signals on the flip-flops are stable at the time of the rising clock edge, so that the correct new states can be loaded into the flip-flops. But in real hardware the things are not so ideal – this is due to different signal propagation times through the logic gates, which are caused by the used chip technology, manufacturing tolerances, aging and temperature differences. In order for a digital circuit to function properly, it is important to adhere to the data setup times and the data hold times of the flip-flops. For example, a 8 kB SRAM may need a data- and address bus setup time of 60 ns and a data- and address bus hold time of 30 ns. The setup time means that the data and the address bus must carry valid data for 60 ns before the rising edge of the write strobe occurs. The hold time means that the data and address bus may only change its state 30 ns after the rising edge of the write pulse.

Please have a look at the oscilloscope prints on the next pages. If you have an oscilloscope, please try to verify my mesurements with your MyNOR:

This screenshot shows the signal WRRAM (pin 17 on SRAM U16) and A12 (pin 24 of RAM U16):

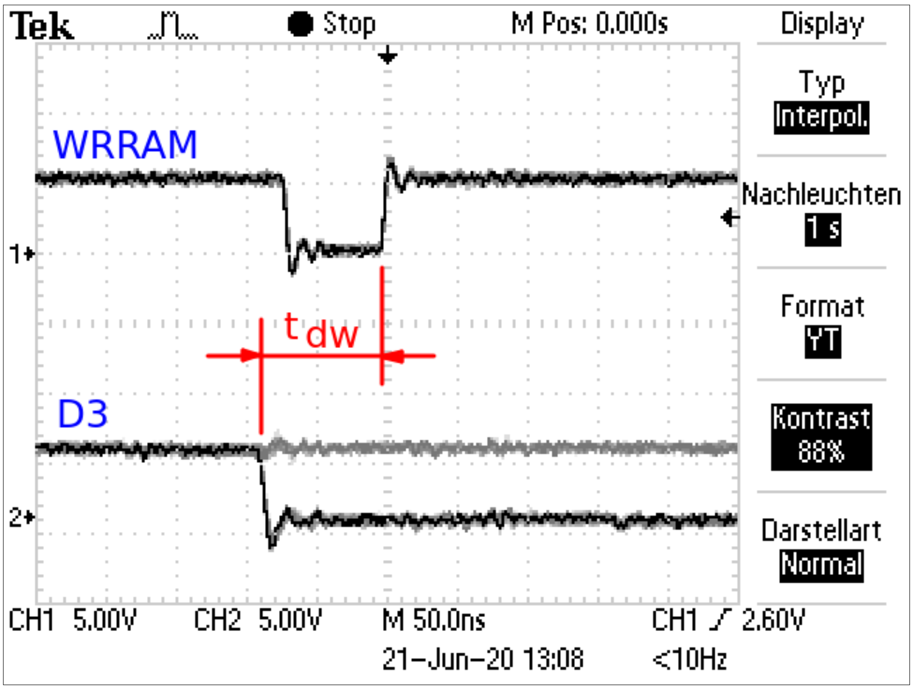


t_{wp} is the width of the write pulse. I am using a slow SRAM with 70 ns access time. According to the datasheet of the RAM, the pulse width must be at least 70 ns. The picture above shows that the requirement is met. If your SRAM has an access time of 55 ns, the write pulse can be shorter. The value of t_{wp} is crucial. It should be in the range 50 ns to 100 ns, but not less than the access time of your RAM. You can adjust the width of t_{wp} by changing the values of R3 and C19. Try using different values for C19 (values in the range of 47 pF to 100 pF are fine). You can also change R3, but please do not make R3 less than 1 kOhm. And please also check the temperature of U8. I had already a “bad” 74HC14 from an unknown brand (purchased from Reichelt) that got very hot. Please try to use only chips from Texas Instruments, STMicroelectronics, Fairchild Semiconductor or NXP (Nexperia). They are known to perform well. Note: The pulse width t_{wp} also depends on the 74HC14 chip used, changing the chip requires a repeated adjustment of R3 and C19.

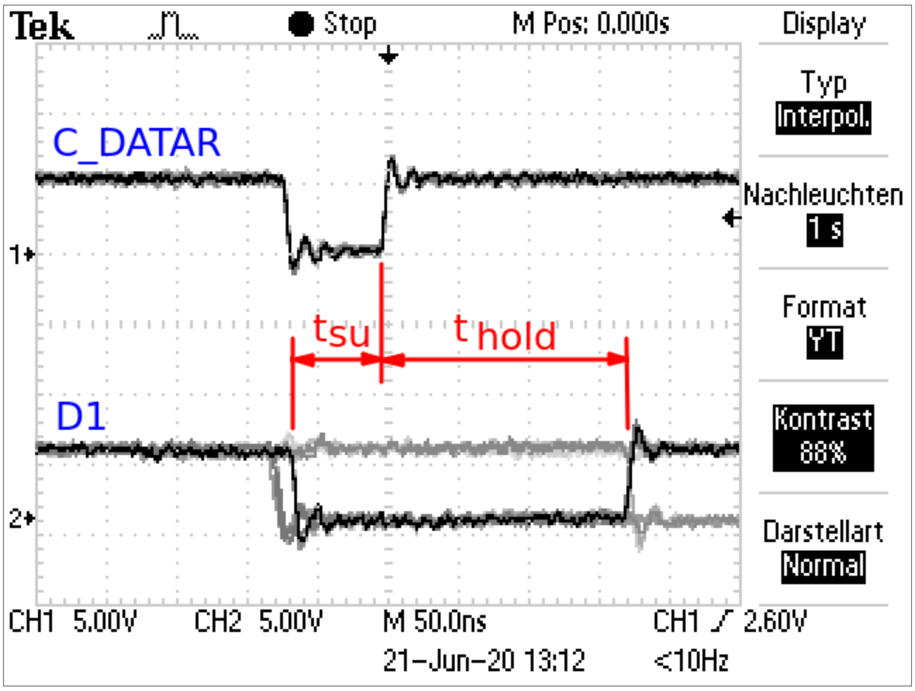
t_{as} is the address setup time. For the SRAM 6264 it can be 0, which means that A12 must be stable at the point where WRRAM goes low. Since A12 is stable long before WRRAM goes low, this requirement is met in the picture above.

t_{wr} is the write recovery time (this is comparable to a signal “hold time”). For the SRAM 6264 it can be 0, that means that A12 can change its state immediately after WRRAM goes high again. This requirement is also met in the picture above, since A12 remains stable for more than 50 ns after the rising edge of WRRAM.

The screenshot on the next page shows WRRAM in relation to the data line D3. As you can see D3 is already stable a few nanoseconds before WRRAM goes low. But it remains stable long time after WRRAM is deactivated again. The screenshot shows t_{dw}, the “Data to Write Overlap Time”. This is comparable to the data setup time. For the SRAM I use, this value must be greater than 30 ns. But because the pulse width of the write pulse is already 70 ns, this requirement is fully met.

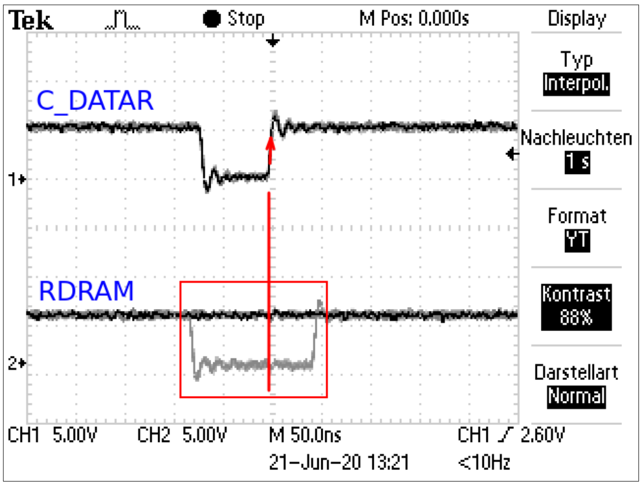


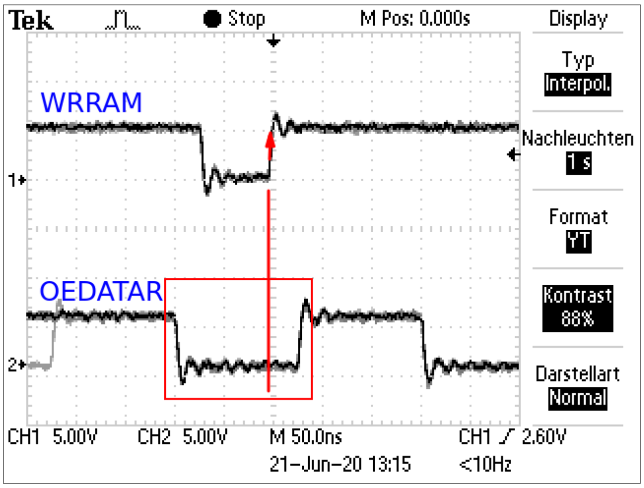
The following screenshot shows the signals C_DATAR and D1 on the DATAR register U10. C_DATAR is measured at pin 11 and D1 at pin 18 of U10. This screenshot shows the meaning of setup and hold time again. Please make sure your MyNOR meets the setup and hold time requirements of U10 (this is a 74HC574).



For the 74HC574 that I am using, t_{su} is 15 ns and t_{hold} is 5 ns. The screenshot shows that these requirements are fully met.

This two screenshots show the data transfer from SRAM to DATAR (left image) and from DATAR to SRAM (right image):



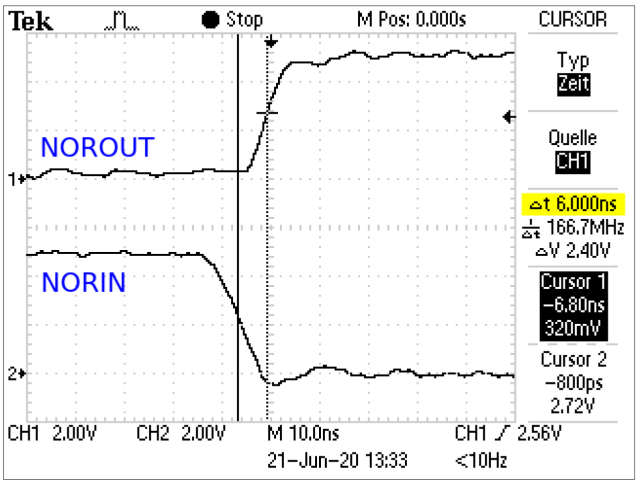


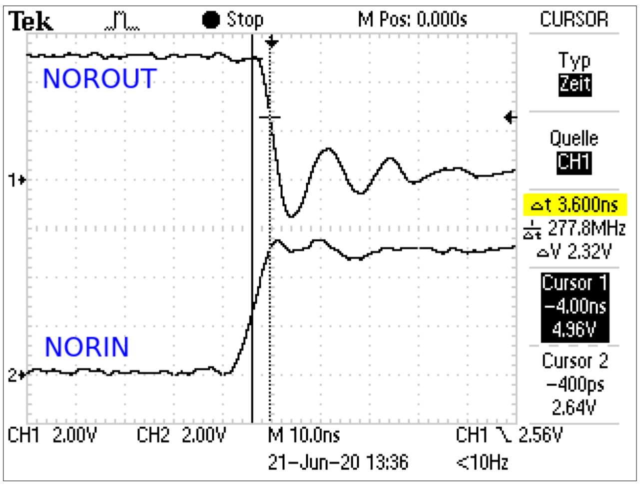
As you can see, the rising edge of C_DATAR and WRRAM is well within the valid window of RDRAM and OEDATAR. Please compare it with the timing of you MyNOR. You may need to change the value of C19 or R3 if the rising edge of C_DATAR and WRRAM is too far to the right side. But do not make the pulse on C_DATAR and WRRAM shorter than the access-time of your SRAM.

Measuring the NOR gate

Below are two screenshots showing the timing of the discrete transistor NOR gate. As you can see, the NOR-gate is *very* fast. P_{delay} is max. 6 ns when the transistors stop conducting. The 100 Ohm pull-up resistor R1 drives enough current into the two transistors so that the charge is completely removed from the inner structure of the transistors in less than 10 ns. With this knowledge you could increase the resistance of R1 to 220 Ohm to save some power. The NOR gate would still be fast enough to run MyNOR at 4 MHz.

The screenshot on the right side shows the falling edge of the NOR output. This happens when one of the transistors begins to conduct. This process is very fast because the transistor's internal RDS_{on} is much lower than R1 (2.5 Ohm versus 100 Ohm). In this case, P_{delay} is approximately 3.6ns.





Software Upload

You can easily run your own program files on MyNOR: Switch on the power supply or reset MyNOR so that the start up message is displayed in the terminal window. Now copy your program and paste it into the terminal window. Some terminal programs also offer the option of sending plain text files, but this function is not really required. The following short program lets the LEDs blink the SOS pattern:

```
@@@@@@:blink-sos@@@@@*@4@0Y4F6Y0W2B4Y0W2B4Y0W2B4Y0B2B4Y0B2B4Y0B2B4Y0W2B4Y0M1B4
Y0M1B4Y0M1B4Y0W2B4Y0B2B4Y0B2B4Y0B2B4X0@0B4[0J0[0K0A0Q0O0Y0L3B4A0Q0E0Y0Z2B4\0K0
\0J0Z0[0J0[0K0A0Q0E0Y0L3B4A0Q0E0Y0Z2B4\0K0\0J0Z0A0Q0O0A0P0@0J0P0U0P0W0]2B4J0Q0
U0Q0W0Z2B4Z0A0P0^0P0N0J0P0U0P0W0O3B4J0Q0U0Q0W0L3B4Z0
```

Below is the source code. Assemble the program with the command „ `myca blink-sos.asm` “ to get the text file for upload. The LEDs light up while MyNOR is executing the „ADD ACCU“ instruction:

```
.name "blink-sos"
#include <mynor/ram-program.hsm>

DIT_LEN   SET  5
DAH_LEN   SET  DIT_LEN * 3

main:     JSR  pause
          JSR  pause
          JSR  pause
          JSR  dot
          JSR  dot
          JSR  dot
          JSR  pause
          JSR  dash
          JSR  dash
          JSR  dash
          JSR  pause
          JSR  dot
          JSR  dot
          JSR  dot
          JMP  main

dash:     PSH  LR_L
          PSH  LR_H
          LD   R1,#DAH_LEN
          JSR  led_on
          LD   R1,#DIT_LEN
          JSR  led_off
          POP  LR_H
          POP  LR_L
          RET
```

```
dot:      PSH  LR_L
          PSH  LR_H
          LD   R1,#DIT_LEN
          JSR  led_on
          LD   R1,#DIT_LEN
          JSR  led_off
          POP  LR_H
          POP  LR_L
          RET

pause:    LD   R1,#DAH_LEN
led_off:  LD   R0,#0
_ml01     DEC  R0
          TST  R0
          JNF  _ml01
          DEC  R1
          TST  R1
          JNF  led_off
          RET

led_on:   LD   R0,#30
_ml02     ADD  ACCU
          DEC  R0
          TST  R0
          JNF  _ml02
          DEC  R1
          TST  R1
          JNF  led_on
          RET
```

Tweaking

When your MyNOR works fine, you can try to increase the clock frequency to 8 MHz. To do this, simply replace the crystal oscillator X1 with an 8.000 MHz type. A side effect of the higher speed is that the baud rate of the RS232 interface doubles to 4800 baud. So please change the setting in your terminal program, otherwise nothing will be displayed when MyNOR is powered up. Now you have to configure MyNOR for 8 MHz: Select the menu item „8“ in the main menu and change the clock frequency to 8 MHz. However, there is no guarantee that MyNOR will operate stably at this high clock frequency under all circumstances. This is because I designed MyNOR to work at 4 MHz in a wide temperature range from 0 to 75 °C. Your MyNOR may become unstable if your ambient temperature deviates significantly from 25 °C. You may also need to adjust R3 and C19 (see the chapter above). With correct values for R3 and C19 I was able to run a MyNOR board at 12 MHz! (but not stable enough...)