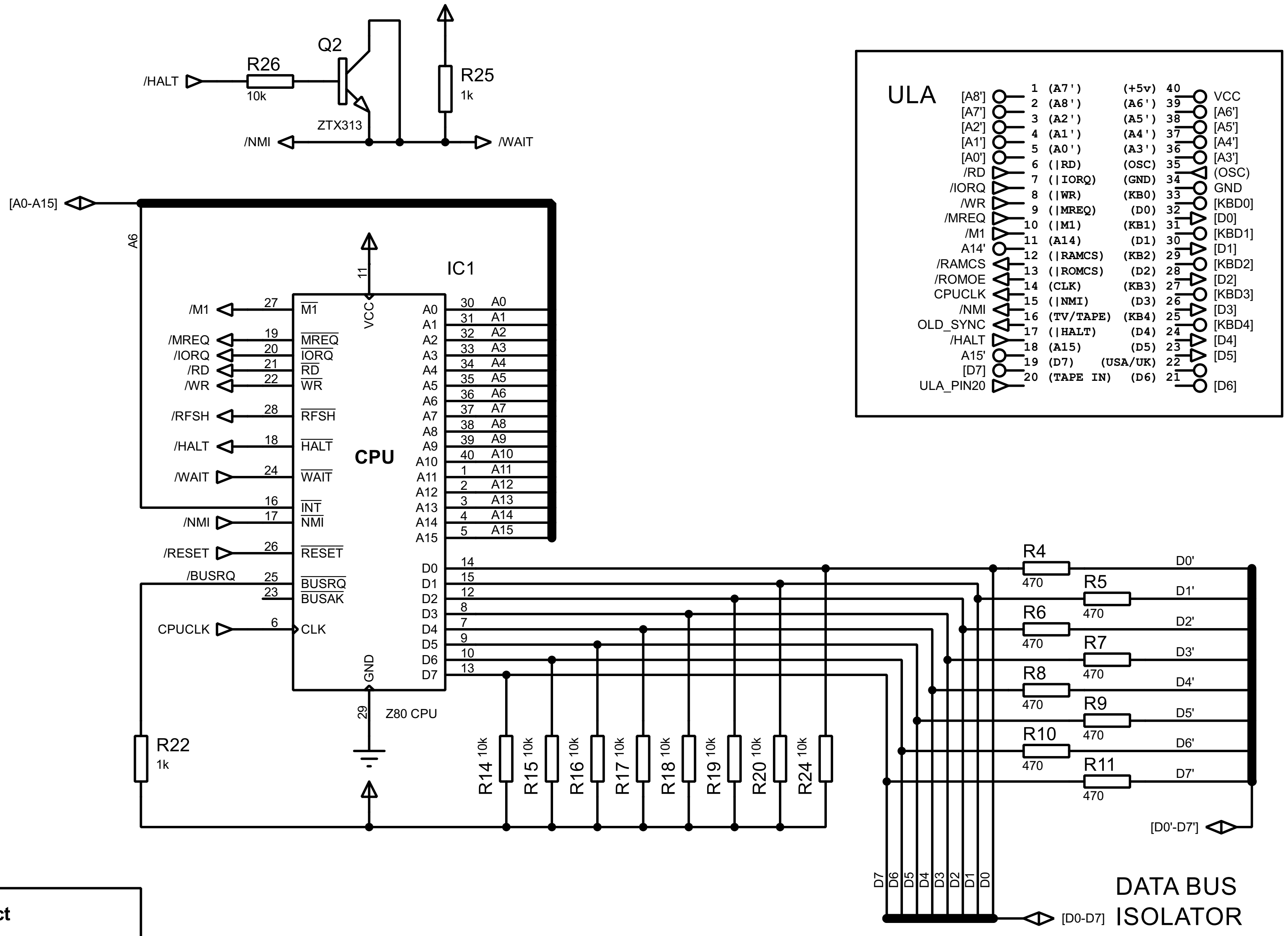
CPU

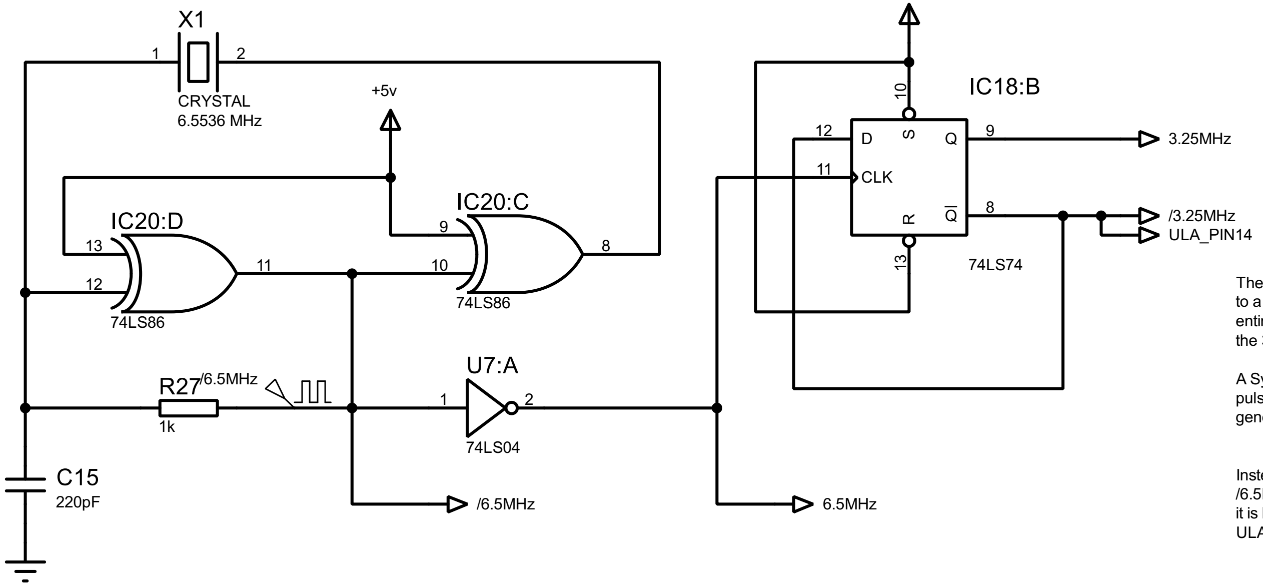


ZX81 Project

Augusto Baffa © Feb/2021

Based on original ZX81 design

CLOCK GENERATOR

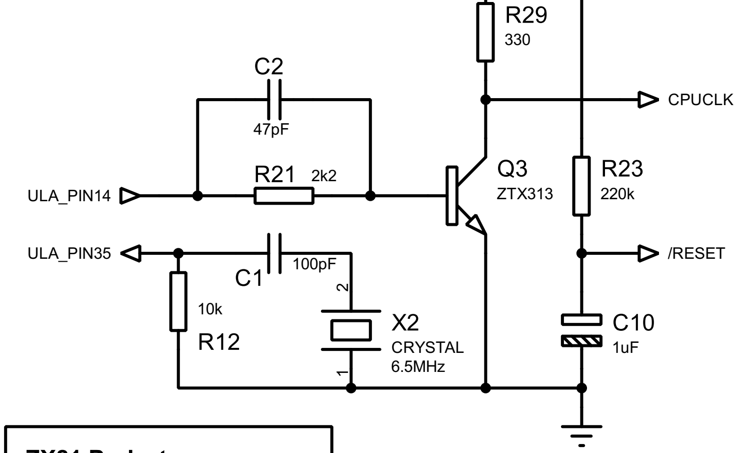


The part of the oscillator circuit that is on the main circuit board looks like it belongs to a Colpitts oscillator, however I decided not to use this part and instead built the entire oscillator on the ULA replacement circuit. The hsyn/nmi generator basically takes the 3.25Mhz clock and counts 207 cycles (resetting at count 207 retuning to 0).

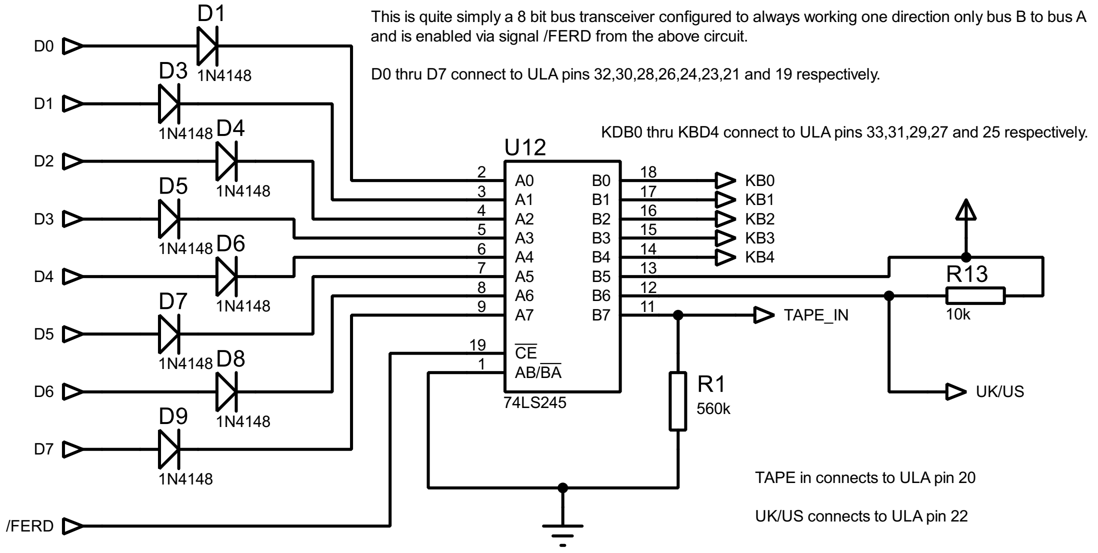
A Sync pulse will reset and stop the counter only allowing it to restart when the VSync pulse is released, the /NMI signal is generated during the last 15 cycles if the NMI generator is turned on.

Instead of using a third XOR gate I use a simple inverter to give a 6.5Mhz signal and a /6.5Mhz. This is further divided by 2 to give the 3.25Mhz signal required by the Z80 CPU it is here that the first mistake can easily be made you need to connect the /3.25Mhz to ULA pin 14 remember it gets inverted by TR2 on the main board.

ORIGINAL CLOCK AND RESET



KEYBOARD/TAPE PORT



This is quite simply a 8 bit bus transceiver configured to always working one direction only bus B to bus A and is enabled via signal /FERD from the above circuit.

D0 thru D7 connect to ULA pins 32,30,28,26,24,23,21 and 19 respectively.

KDB0 thru KBD4 connect to ULA pins 33,31,29,27 and 25 respectively.

TAPE in connects to ULA pin 20

UK/US connects to ULA pin 22

ZX81 Project

Augusto Baffa © Feb/2021

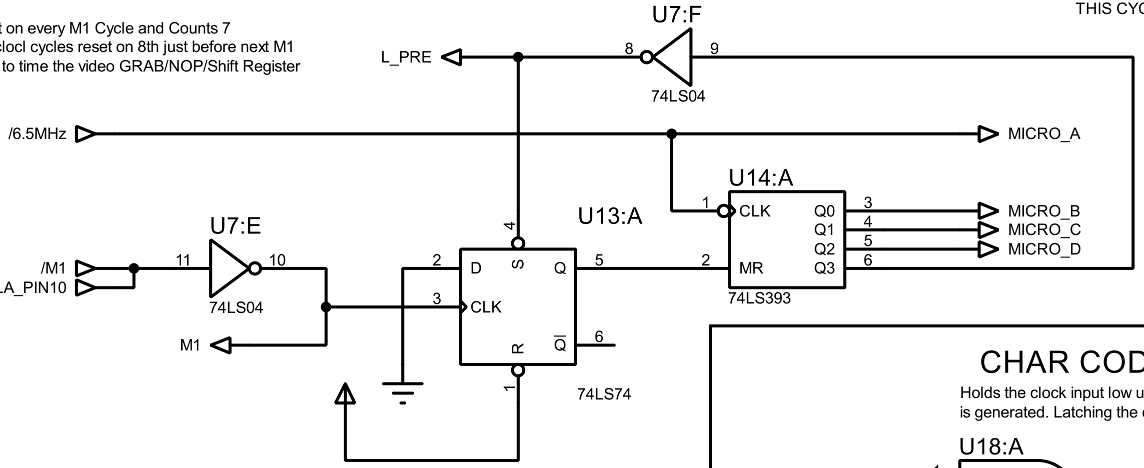
Based on original design by Andy Rea

MICRO COUNTER

The MICRO COUNTER, is a four bit counter that starts to count at the beginning of each /M1 cycles and reset to zero on the rising edge of the 4th t state just in time to start counting again. Using the 6.5Mhz clock signal as the lowest bit its possible to split the 4 t states into 16 time zones. It was found that I only needed to split it into 8 zones corresponding to each half cycle of the CPU clock during the M1 cycle. This counter is used to trigger the various events during the Forced NOP cycle.

When /M1 goes low the inverter clocks the positive edge triggered D flip flop which then outputs a low on it Q output this releases the active high clr input on the counter which is clocked at double the CPU clock by the /6.5Mhz signal when the count reaches 8 it presets the Q output via the other inverter, clearing the counter back to zero ready to be triggered again by the next M1 cycle by resetting at 8 it ensures that our circuit will not interfere with any instruction that has a longer M1 cycle. The signal L_PRE is used to preset the NOP THIS CYCLE Flip flop at the end of each cycle.

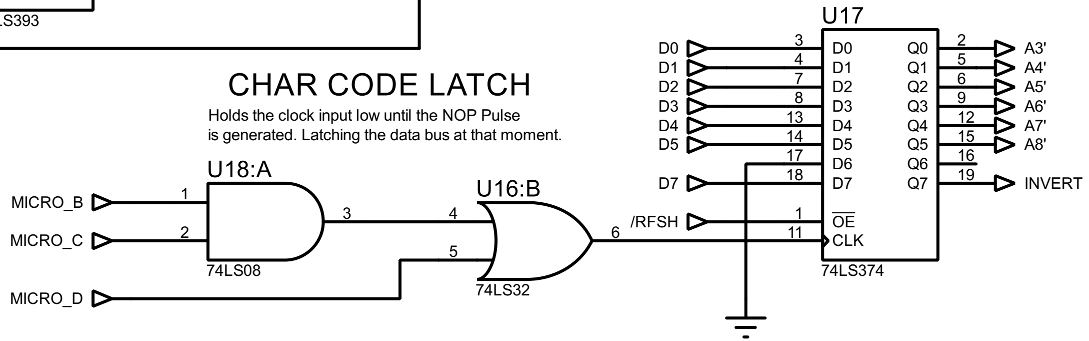
Reset on every M1 Cycle and Counts 7
Half clocl cycles reset on 8th just before next M1
Used to time the video GRAB/NOP/Shift Register



This 8 bit latch is controlled by my counter it latched whatever is present on the D inputs on the rising edge of the CLK input, The and or logic to the left holds the CLK input low until the falling edge of T2 when it is driven high and held high until the end of T4. its output is control by an internal refresh signal. Generated by the following circuit.

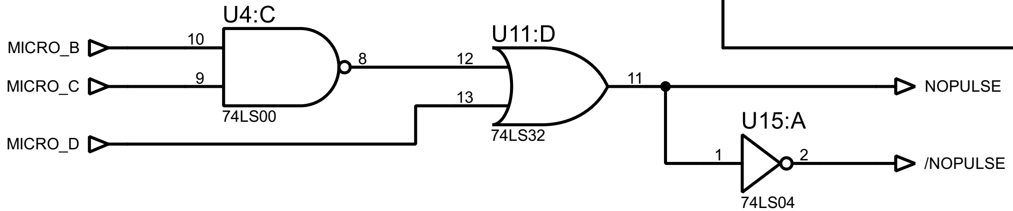
CHAR CODE LATCH

Holds the clock input low until the NOP Pulse is generated. Latching the data bus at that moment.



NOP PULSE

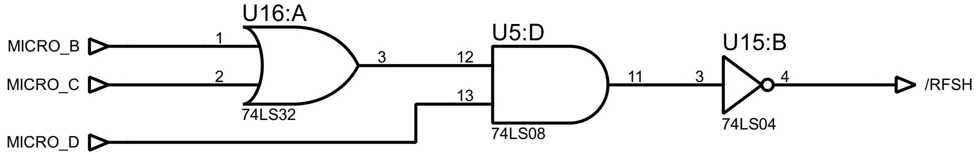
Generates a low pulse from falling edge T2
Lasting 1/2 clock cycle to the rising edge of T3
When the data bus is sampled for the OP-CODE



This little circuit controlled by the count of the micro counter determines in conjunction with the above circuit when a NOP should be forced, when my micro counter reaches 3 at the falling edge of T2 and holds it until the rising edge of t3 when the CPU samples the data bus for the op-code.

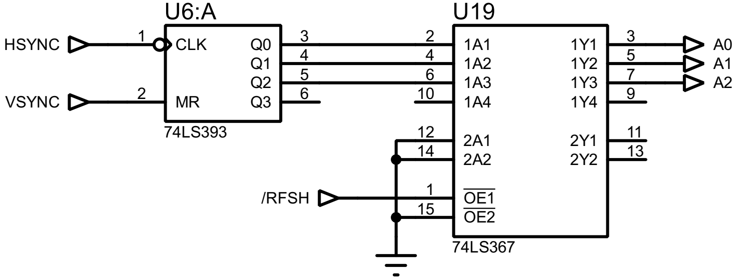
After this my circuit releases the data lines.Also at the falling edge T2 my circuit grab the data that was on the data bus immediately before the NOP is forced.

REFRESH



Which enables the output to the alternative address lines from the falling edge of t3 up until the end of T4.
This signal is also used to enable the line counter output onto alternate address lines A0' , A1' and A2'

The counter is reset on every VSYNC back to 0 and then counts every hsync pulse the lower 3 bits of this count are then gates by the refresh control tristate buffer.



ZX81 Project

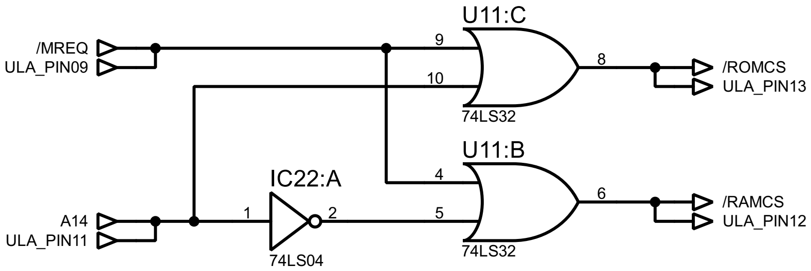
Augusto Baffa © Feb/2021

Based on original design by Andy Rea

/ROMCS & /RAMCS

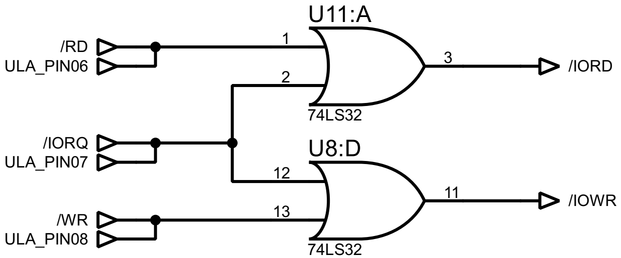
Takes A14 and OR's it with /MREQ to generate the /ROMCS signal (both A14 and /MREQ low).

Takes an inverted A14 and OR's it with /MREQ to generate the /RAMCS signal (A14 high and /MREQ low).



IO READ-WRITE

Gives a low on /IORD or /IOWR depending if its a read or write IO instruction used in the VSYNC & NMI Control circuit.



ZX81 Project

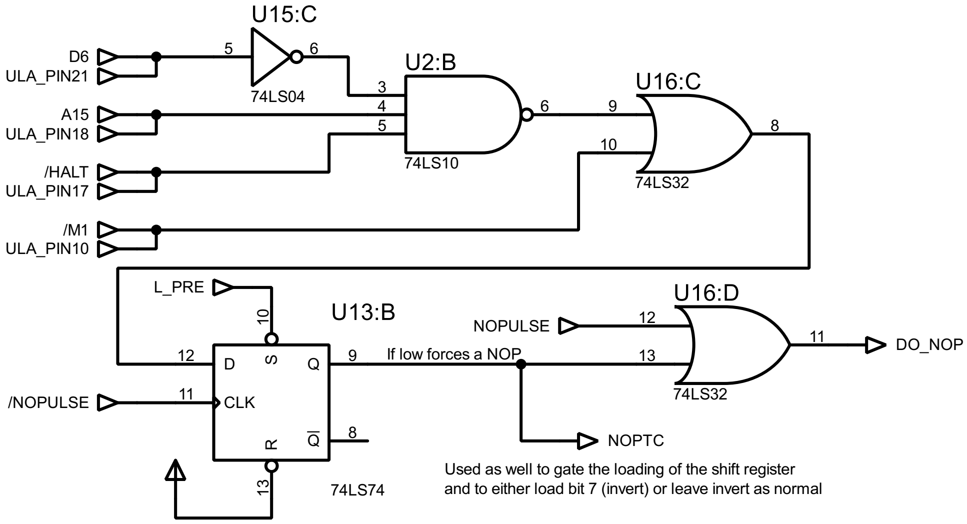
Augusto Baffa © Feb/2021

Based on original design by Andy Rea

NOP THIS CYCLE

Sets the output of the 7474 Low when
a videl cycle is detected

The next Flip flop in the video circuit detects whether this M1 cycle is a video cycle or not, by using D6, A15, /HALT and /M1. The latched output will remain in the low state until the rising edge if the next CPU M1 cycle, it is preset to 1 when my micro counter reaches 8.

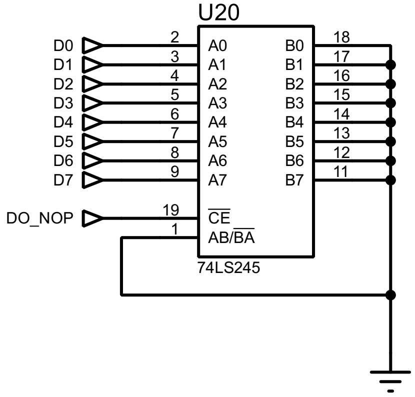


If low forces a NOP

Used as well to gate the loading of the shift register
and to either load bit 7 (invert) or leave invert as normal

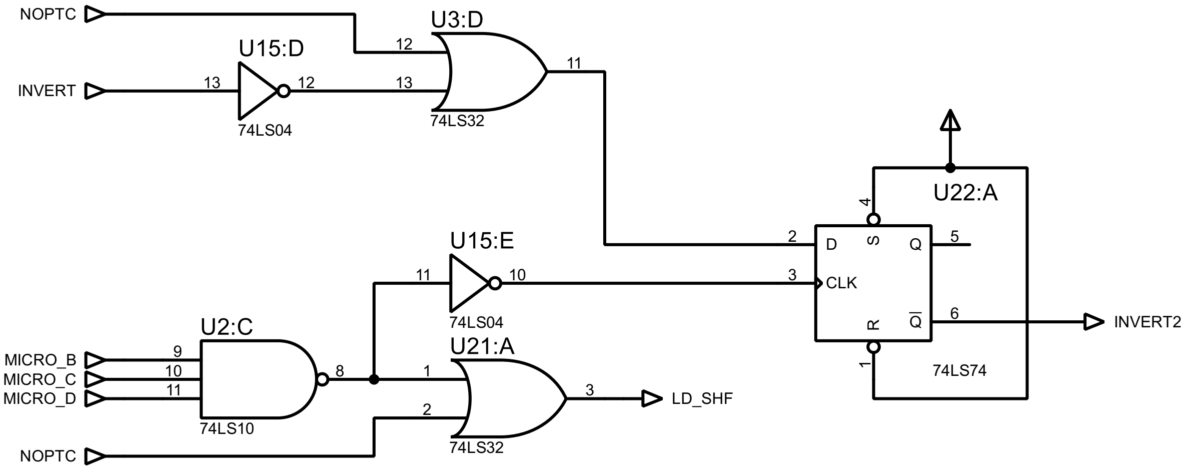
FORCE NOP CYCLE

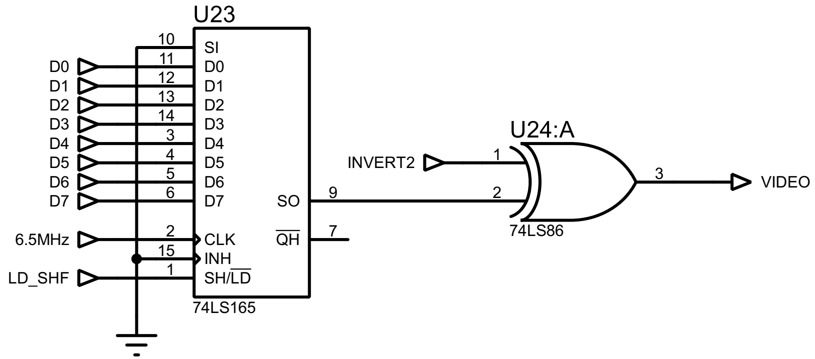
This is the bit that actually forces all logic lows (B inputs tied to ground) when the outputs (the A bus) are enabled the CPU data lines are forced low, forcing the NOP cycle.



VIDEO INVERTER

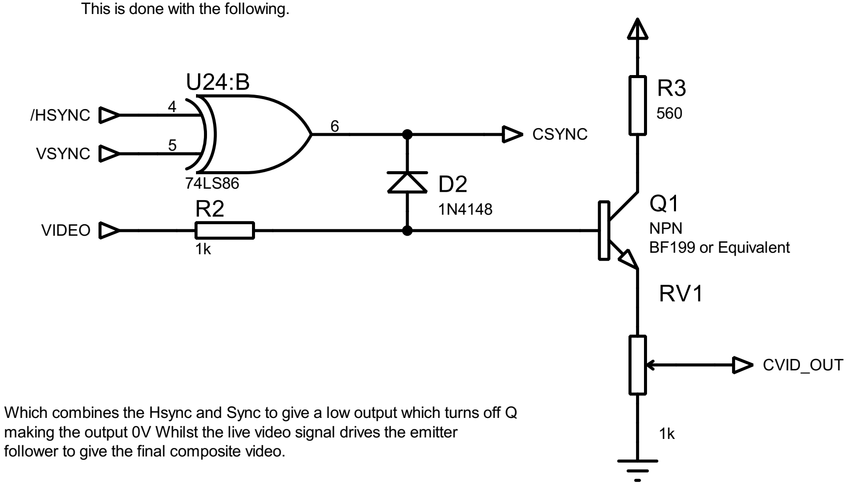
The video inverter circuit consist of a flip flop that latches the state of the invert bit of the character code from the char code latch but only if this cycle was is a forced NOP, if it is not a forced NOP then the top or gate allows the invert bit7 through to the D input ready to be latched every time when the micro counter reaches 7 falling edge of T4, if it is a forced NOP cycle also via the bottom or gate the shift register is allowed to load the new character pattern, otherwise it will continue to load in white video via its serial input below.





All that remains now is to get the VSync, Hsync and Video all together to produce a useable composite video signal

This is done with the following.

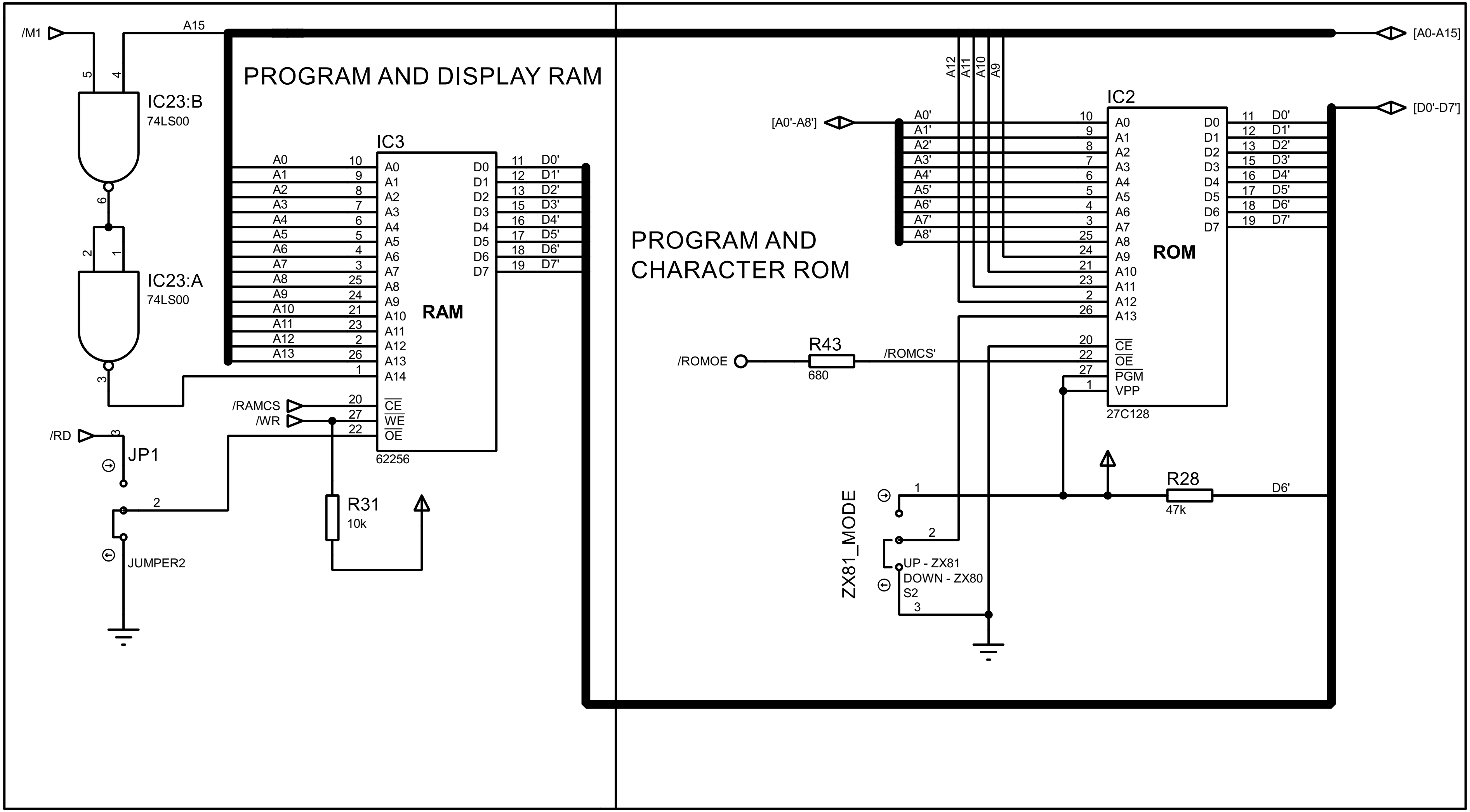


Which combines the Hsync and Sync to give a low output which turns off Q making the output 0V Whilst the live video signal drives the emitter follower to give the final composite video.

ZX81 Project

Augusto Baffa © Feb/2021

Based on original design by Andy Rea

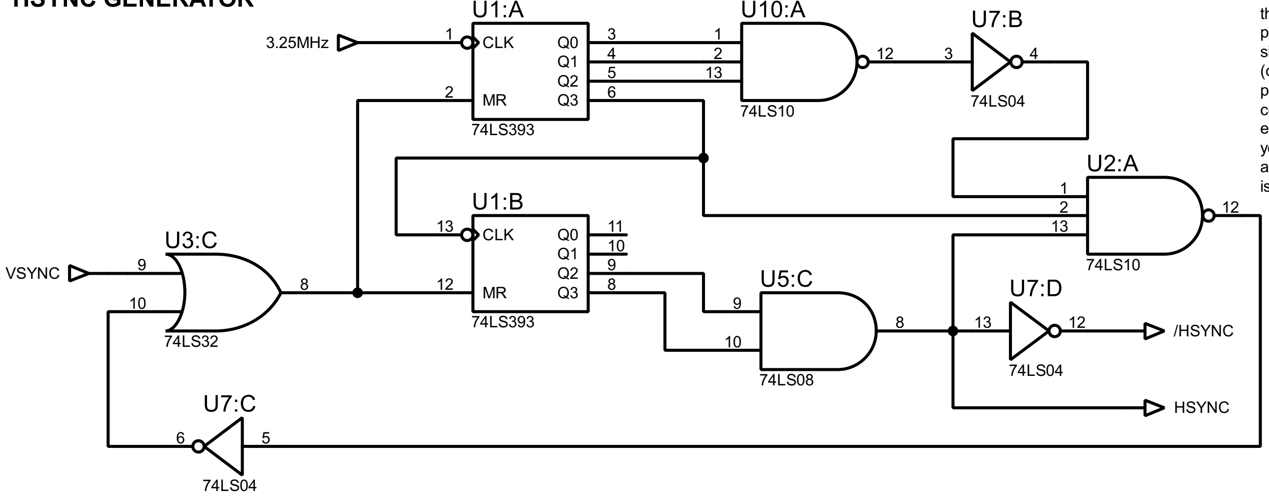


ZX81 Project

Augusto Baffa © Feb/2021

Based on original ZX81 design

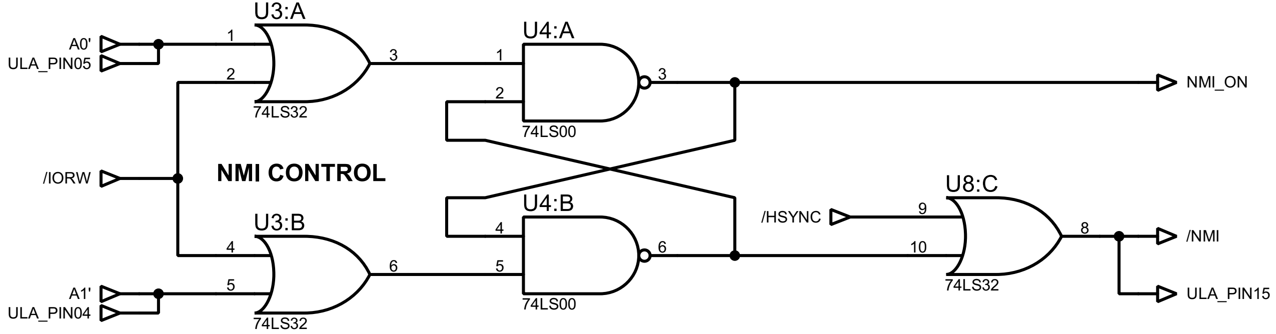
HSYNC GENERATOR



If the VSync pulse is started the OR gate on the left holds the pair of counters in there reset state (all outputs low) otherwise the counters will count each of the clock pulses from the 3.25Mhz signal the lower counter been clocked at 1/16 of the main clock signal when the count reaches 192 the sync becomes active via the 2 input AND gate (connected to Qc an Qd of the second counter) and will remain so for the next 15 clock pulses until the count reaches 207 at which point the pair of 3 input and gates in conjunction with inverters (I didn't have a 7411 which would negate the need for the extra inverters) almost instantly resets both counters back to zero. So left free running you get a 15 cycles long pulse every 207 cycles (4.61uS long, every 63.69uS, we are aiming for 4.7uS and 64uS respectively) due to the way the Zeddy generates the video it is a requirement that the hsync pulses are slightly short so they remain synchronized with the video from the zeddy.

The hsync pulses are also used for the /NMI pulses if the NMI is turned on.

NMI CONTROL



This circuit is quite simple as long as you understand how a nand gate RS latch works.

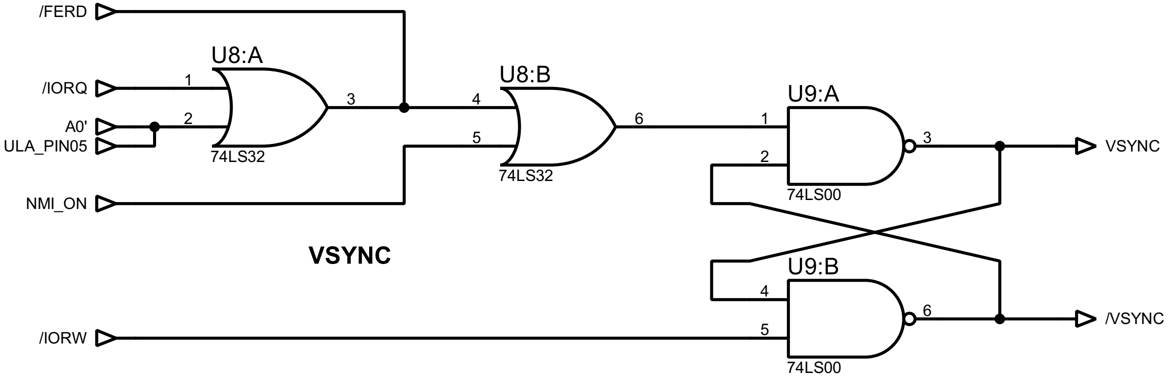
There are 2 latches used in the zeddies one for the VSYNC pulse (the top latch) and one for the NMI on or off control the (bottom latch)

So a read on port $FE [IN A,($FE)] will always read the keyboard and if the NMI generator if off will also toggle the RS latch starting the VSync pulse and stopping the hsync generator.

Any port write will stop the VSync pulse and restart the hsync generator.

The NMI control is done of two separate ports, a write to port $FD will turn on the NMI generator, and a write to port $FD will turn off the NMI generator.

VSYNC



ZX81 Project

Augusto Baffa © Feb/2021

Based on original design by Andy Rea