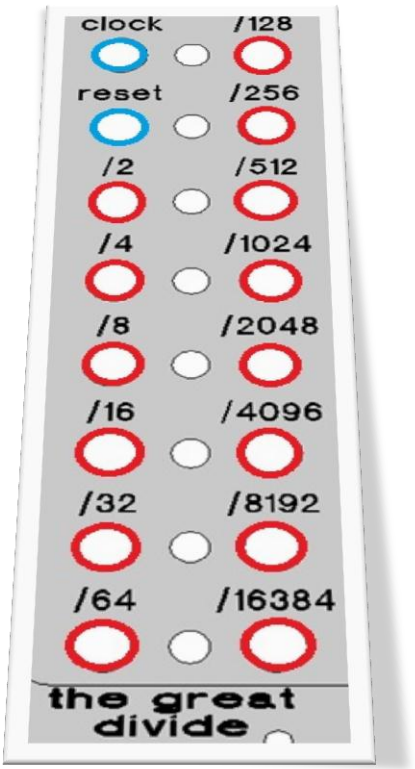
nonlinearcircuits

CMOS Panel User guide vers. 1

Great Divide

This is a regular clock divider capable of dividing down to $\frac{1}{16384}$ of the clock signal. The clock signal can be an output from a VCO (and this is a good source), it means all the CV modules (EGs, sequencers, etc.) will be nicely in sync with your audio. Also, when driven by an audio rate signal, the /2, /4 and /8 outputs can be used to obtain harmonics. The LEDs indicate outputs /128 to /16384, as the /2 to /64 outputs will often be running at rates above 30Hz, an LED for these outputs will appear constantly lit.

A 1Hz signal on the clock means the /16384 output will rise approx. every 4.5 hours.



8 Bit Cipher

This module is based on the Buchla Source of Uncertainty, although it has been trimmed down to be simply a set of clocked shift registers with stage outputs and weighted DtoA networks to obtain CV outputs.

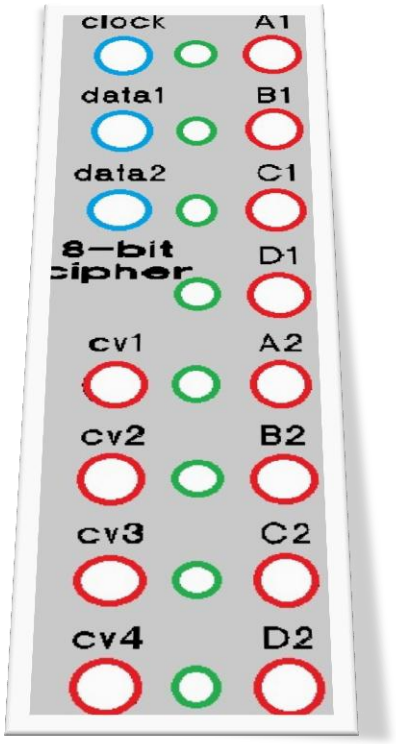
It requires a clock signal AND a signal on at least one of the data inputs to operate. Like virtually everything else in the NLC synth, anything crossing 1V can be used as a clock or data signal.

The best way to see what it does is feed it two signals from different sources, say two LFOs. Leave the clock signal steady and vary the frequency of the signal to the data input. Also try different wave-shapes such as square, sawtooth or triangle. You will see how the data is shifted down thru the stages from A1 to D2. It takes a little bit of tweaking to get the data moving thru in patterns you like, but should be very easy to see what is happening by watching the LEDs.

Once you have it running nicely, connect one of the CV outputs to a VCO and have a listen to the patterns. Tweak the signal on the data input to see how this changes.

The CV outputs are all related to each other but are all different; part of CV1 is fed to CV2, part of CV2 is fed to CV3 etc.

The stage outputs are great for driving drum modules….or anything you like.

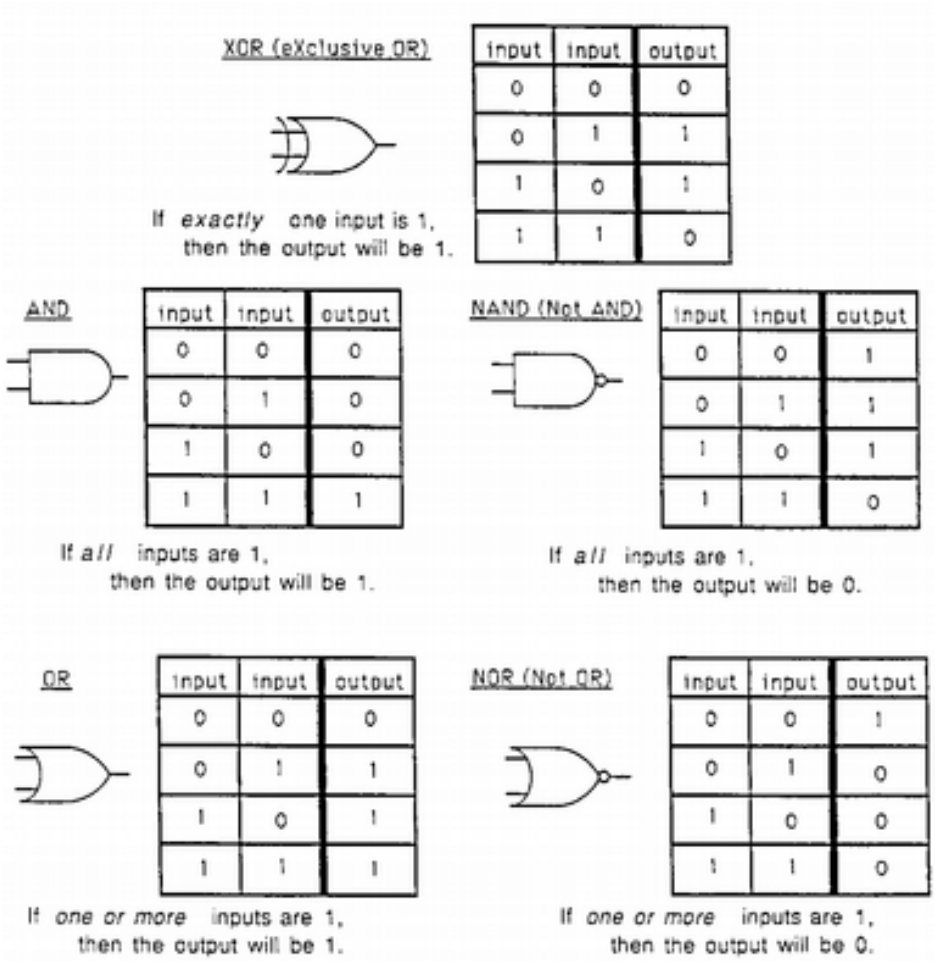


4X4 binary logic

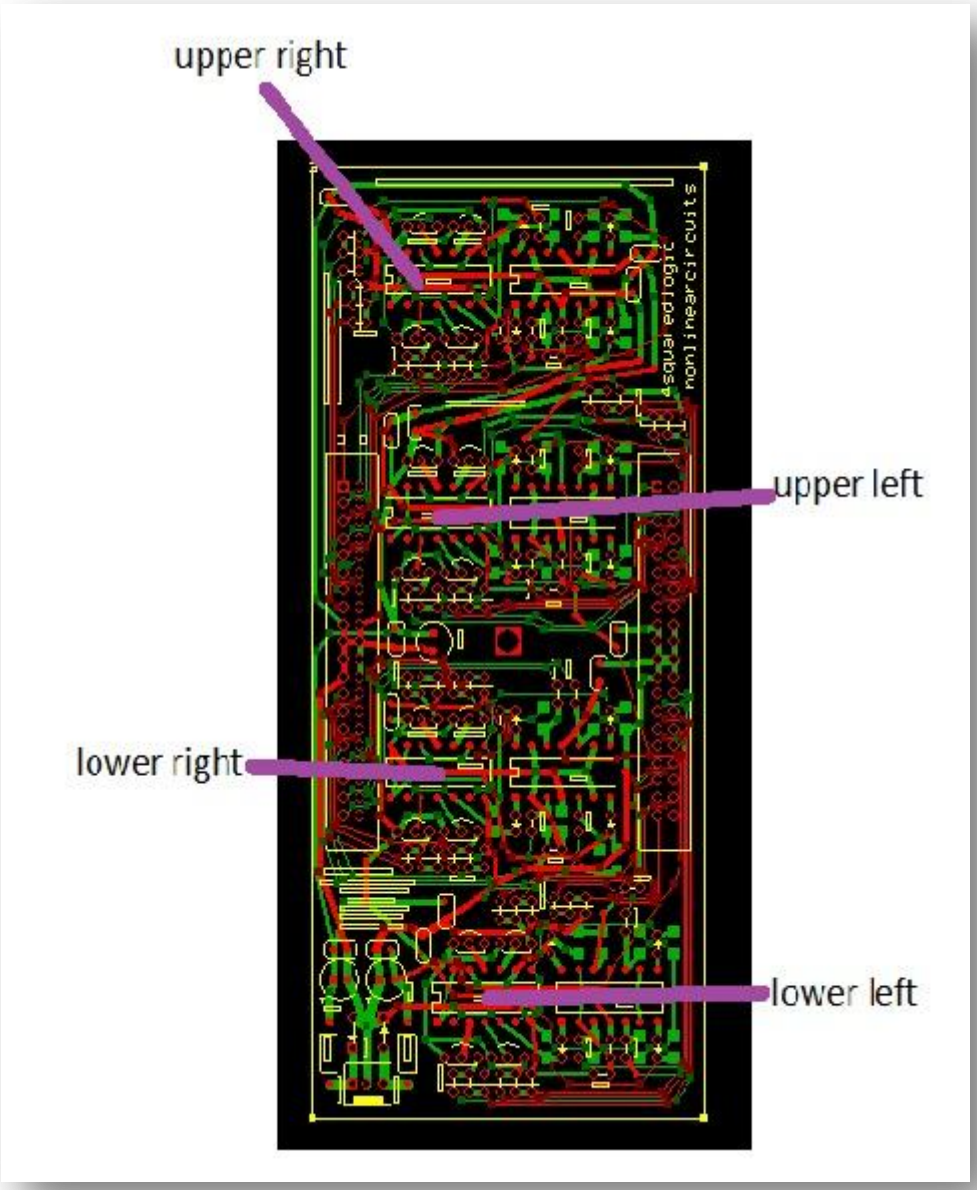
The binary logic modules can use a variety of logic chips.

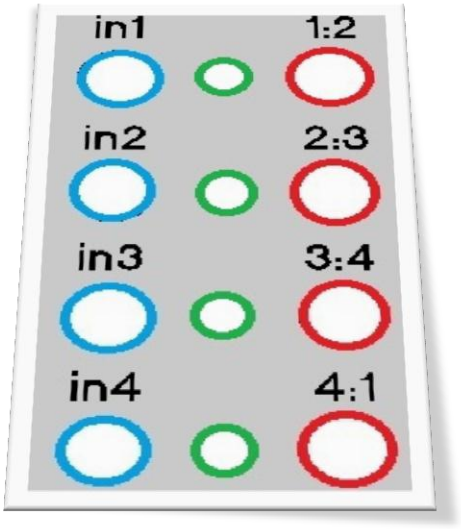
4001	NOR
4011	NAND
4030	XOR
4070	XOR
4071	OR
4077	XNOR
4081	AND

This chart is from psycnet.apa.org



The chips are socketed so can be changed out to obtain the binary functions you wish. When building, I install 4 different functions, usually a XOR, XNOR and an OR, the 4th could be anything. There is nothing on the panel to indicate which is which….that would take the fun out of it. The PCB image below shows which chips to change for each panel section.


upper right
upper left
lower right
lower left



There are 4 logic circuits in each section, so 16 in the entire module. It can be seen the outputs are the binary functions of 2 inputs. As output 1:2 and output 2:3 are both partially controlled by the signal on input 2, these outputs will be different but related to each other. The input signals can, as usual, be anything crossing 1V; audio, gate, trigger, CV, whatever.

To use these modules, get a bunch of signals from the Great Divide, 8 bit Cipher and CMOSC, patch them in to various inputs, feedback outputs to inputs of other modules to get them humping each other each other. Eventually you will get all kinds of poly-rhythmic madness going on to patch out and drive other modules on your synth.

These will run at audio rates and the XNOR module can be used as a set of psuedo ring modulators, very effective to feed in 4 audio signals and get the 4 RM'd outputs…..assuming you can figure out which quadrant contains the XNOR chip!

Sample & Hold / Hold & Thru

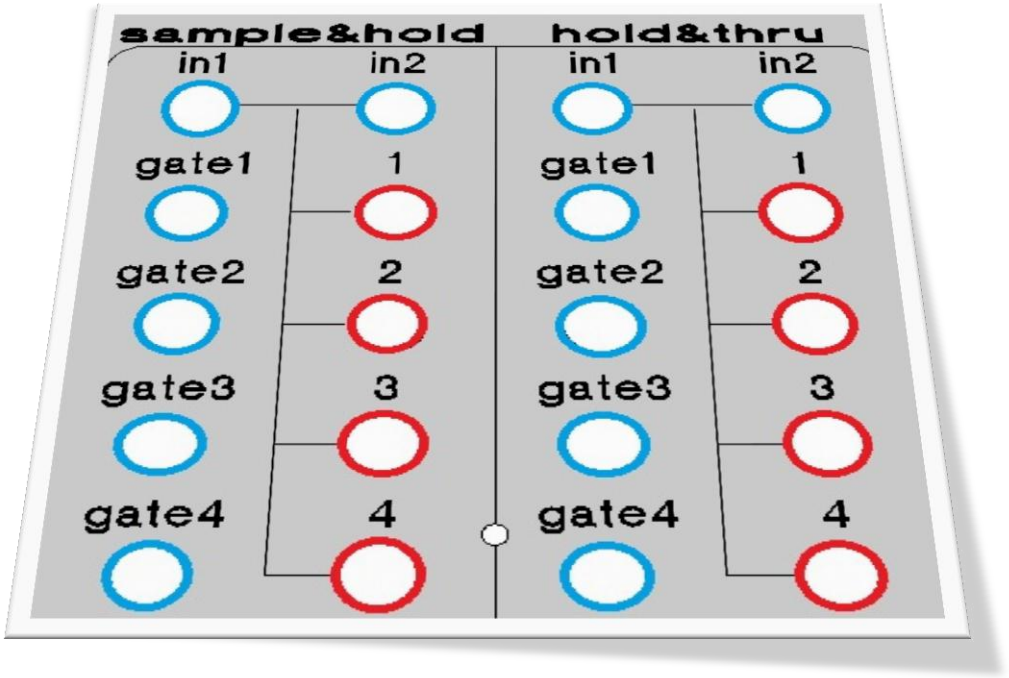
The sample & hold contains four individual S&H circuits all based on Maxim analogue switches. The 4 circuits all use the same input signals to sample but can be controlled individually to choose when to sample. The Hold & Thru is similar in concept except when no gate is present the input signal appears unaltered at the outputs. A gate signal (or anything crossing 1V) will cause the appropriate output to hold the voltage the input signal happened to be when the gate went high. It will hold this voltage until the gate returns to low.

Both modules can sample positive and negative voltages, both can operate at audio frequencies if desired.

To use either module, put a CV signal (eg. Triangle wave from a LFO) into **in 1** or **in 2**. Feed a clock signal to **gate 1**. Connect **output 1** to a VCO and listen to the output of the VCO.

Connect another clock signal to **gate 2**, connect **output 2** to the freq/cutoff input of a VCF. Patch the output of the VCO to the VCF, listen to the results and twiddle knobs a bit

This vid demos this module - http://www.youtube.com/watch?v=ZjlcOY0ZejE&feature=plcp.

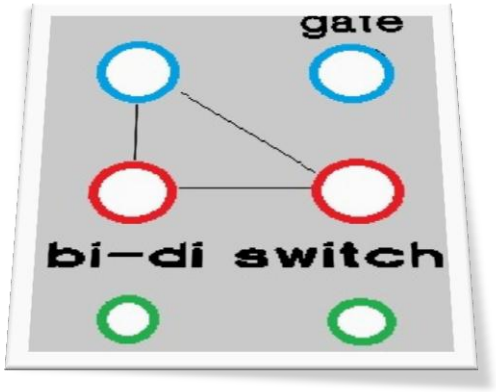


Bi-di switch

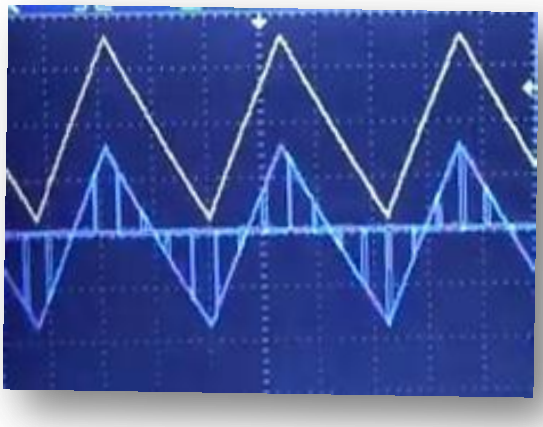
This module is functionally the same as the Bi-directional Router found in vintage Serge synthesizers. It is an entirely different circuit as it uses a Maxim analogue switch, whereas the Serge version used a CMOS 4007.

This module is one exception to the NLC rule of "Blue Jacks = input, Red = output." The switches are bi-directional so a gate can be used to switch one signal between 2 outputs (**A**), or the gate can be used to select which of two signals reaches the output (**B**).

The jacks on the panel have been chosen to give option **A** or **B**, but the colours can be ignored if two **A**s or two **B**s are required



This image shows a triangle wave being switched between the two outputs by a gate signal that is approx 5 times the freq of the triangle wave.



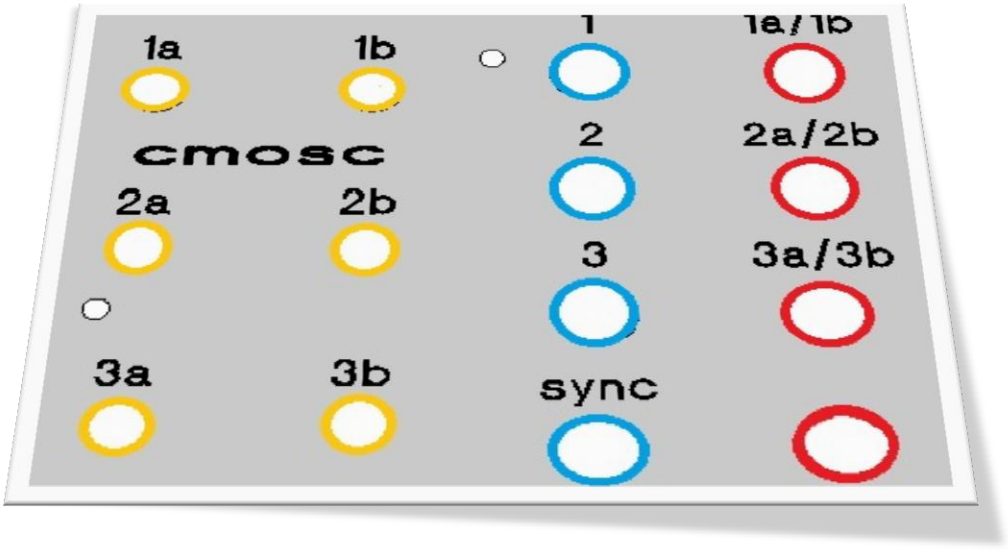
When the two output signals are merged the triangle wave is recreated.

cmosc

There are two CMOSC modules on the panel. Each module has 6 oscillators; the frequency of each oscillator can be set with the pots. Feeding a signal to input will decide which oscillator signal will appear at the output. If the signal is low, oscillator A is heard, if the signal is high oscillator B is heard. This module gets very interesting when the outputs of each oscillator pair are fed into other pairs, which are themselves controlled by others and so on…..complex, evolving drones. Put patchcords into every jack so every pair is controlled by another pair, listen to one of the outputs, twiddle the knobs until you feel satisfied.

The sync input actually disables the oscillators, the unmarked output is a sync output…or if you like; it goes high when the oscillators are disabled. It is most interesting to put an audio signal into the sync input, preferably from the other CMOSC module

Youtube demo: http://www.youtube.com/watch?v=VeeTiJz-PLo&feature=plcp



Feel free to make suggestions on how to improve this guide or ask questions if anything is unclear/poorly explained/glossed over or obviously bullshit

Andrew

Oct 2012