

CELLULAR AUTOMATA SEQUENCER

USERS GUIDE

nonlinearcircuits

#1 rule with nonlinearcircuits panels

blue = input

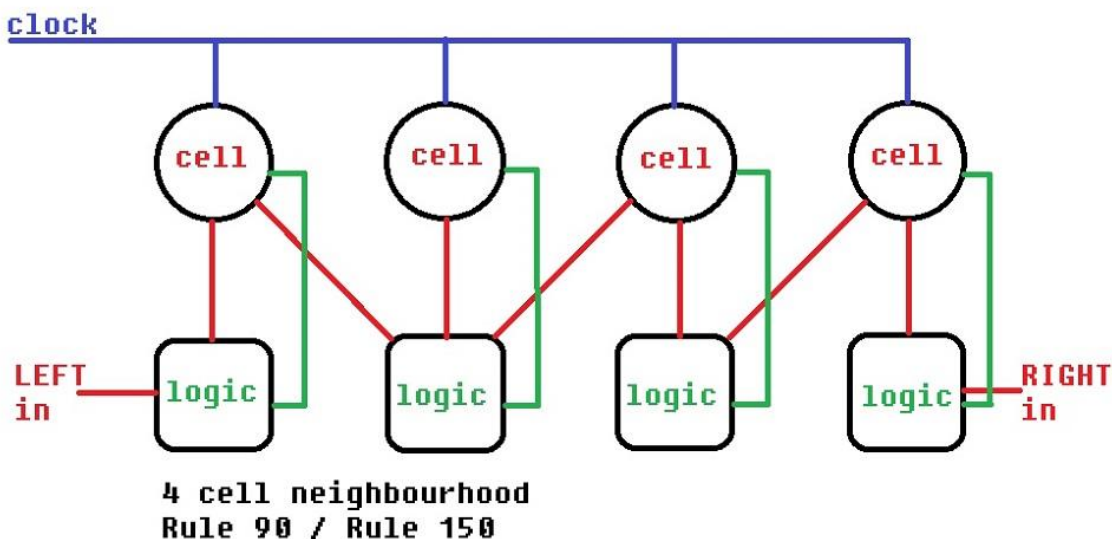
red = output

You can connect one output to several inputs, but do not connect outputs together

Circuit description from the build manual, just FYI:

This circuit was inspired by a paper on using cellular automata as a method of fault detection in analogue circuits (Theory and application of multiple attractor cellular automata for fault diagnosis – P. Kolin, et al). The paper does not supply a circuit but gives block diagrams of sample layouts. Although based on CA rules 90 & 150, the circuit kind of sits between the two.

The block diagram shows the basic function.



blue lines – clock signal for all four cells

red lines – input paths for the logic gates

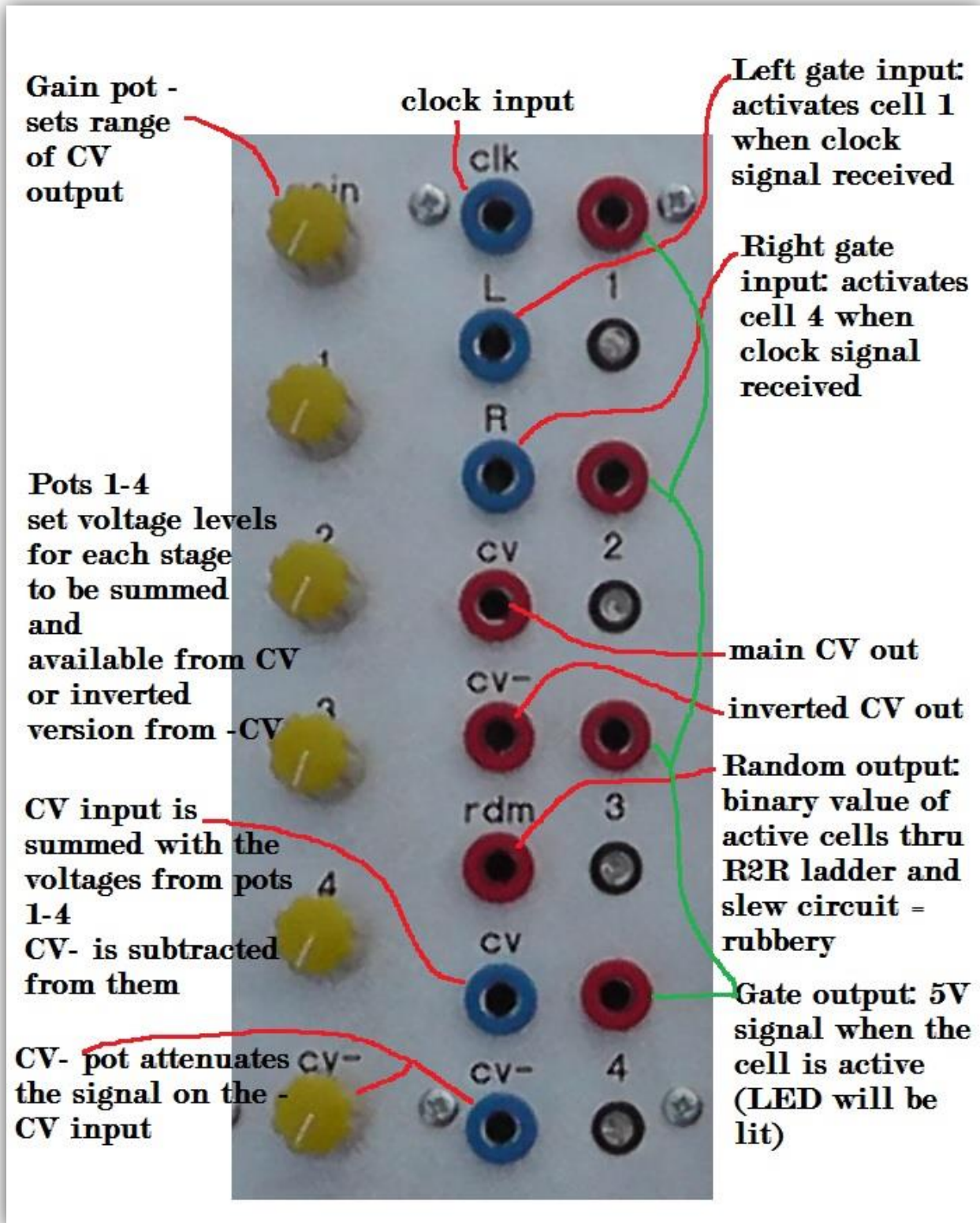
green lines – logic outputs

As each cell lives it puts out a gate for the length of its lifetime and a voltage to a pot that can be attenuated and then summed to get a CV signal (and its inverse...and there is a staircase divider off the 4 cells to give a semi-random CV as well). Of course more than one cell may be alive at the same time so the resulting CV is far more complex than a standard 4 stage CV.

There is also a CV input and inverted CV input with attenuator for each neighbourhood. This means neighbourhoods can be summed together for more complex signals or even the random output can be fed back in to be mixed with the CV.

Each neighbourhood needs a clock and a signal on the left or right input to get started, the cell gate outputs can then be used to activate cells in other neighbourhoods or even act as clocks for other neighbourhoods.

It is best to have at least four neighbourhoods to be able to patch-program a great deal of complex patterns. That said, even one neighbourhood is capable of producing some simply amazing patterns. This video was made using a single neighbourhood - <http://www.youtube.com/watch?v=SINc4EIJL-Y>



To get started, it is best to work with a single neighbourhood.

The module needs a clock input and gates to activate the cells. If a gate signal is high and on, say, the L input and a clock signal is received, then cell 4 will activate. The next clock signal will activate cell 3, if the gate signal on cell 4 is still high, cell 4 will de-activate or die. If the gate signal on cell 4 has gone low by the time the next clock signal arrives, cell 4 will live for another cycle.

Activating cell 1 at the same time and trying to explain the whole process is beyond the scope of this guide.

To sum up:

**The module needs a clock signal and gate signals on the Left (L) and/or Right (R) inputs.
The gate signals can be any signal higher than 1.5V.**

When cells are activated you will see the corresponding LEDs light up, the corresponding gate output will also be high (5V) and can be used to activate or clock other neighbourhoods.

The gain pot can be used to set the range of the CV and inverted CV output, this is very useful allowing you to easily keep the output between 0V and 1V or have a range from 0-11V. If the gain pot is set to zero, you will get no CV signal, so be sure it is always set higher than zero.

The RDM output is the inverted binary value of the active cells fed to an R2R ladder network and then the voltage is slewed. It is not strictly random (nothing is), but it is a very unpredictable and entertaining voltage. It is quite useful to take the RDM signals and input them to the CV- inputs of adjoining neighbourhoods, adjusting the attenuator to get the level as desired.

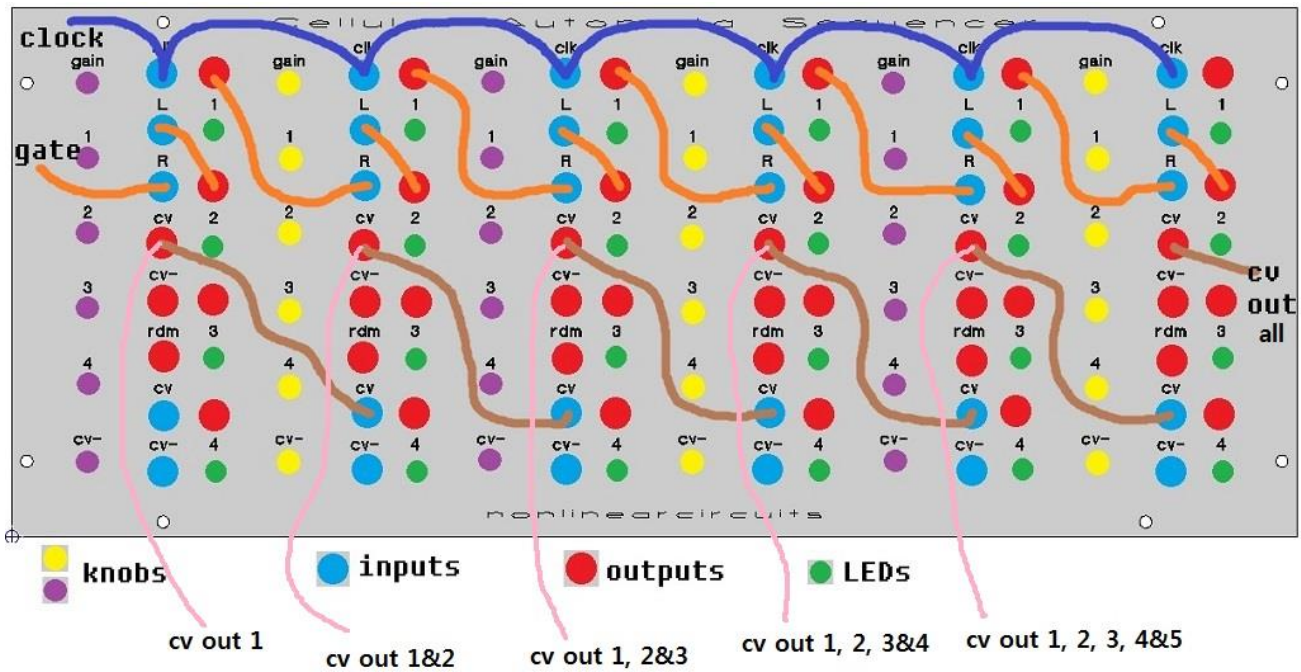
The CV input will add an incoming CV signal with the voltages from Pots 1-4, this means the CV outputs can be chained together

The CV- input with the CV- Pot to attenuate inverts the incoming CV signal and adds it to the mix.

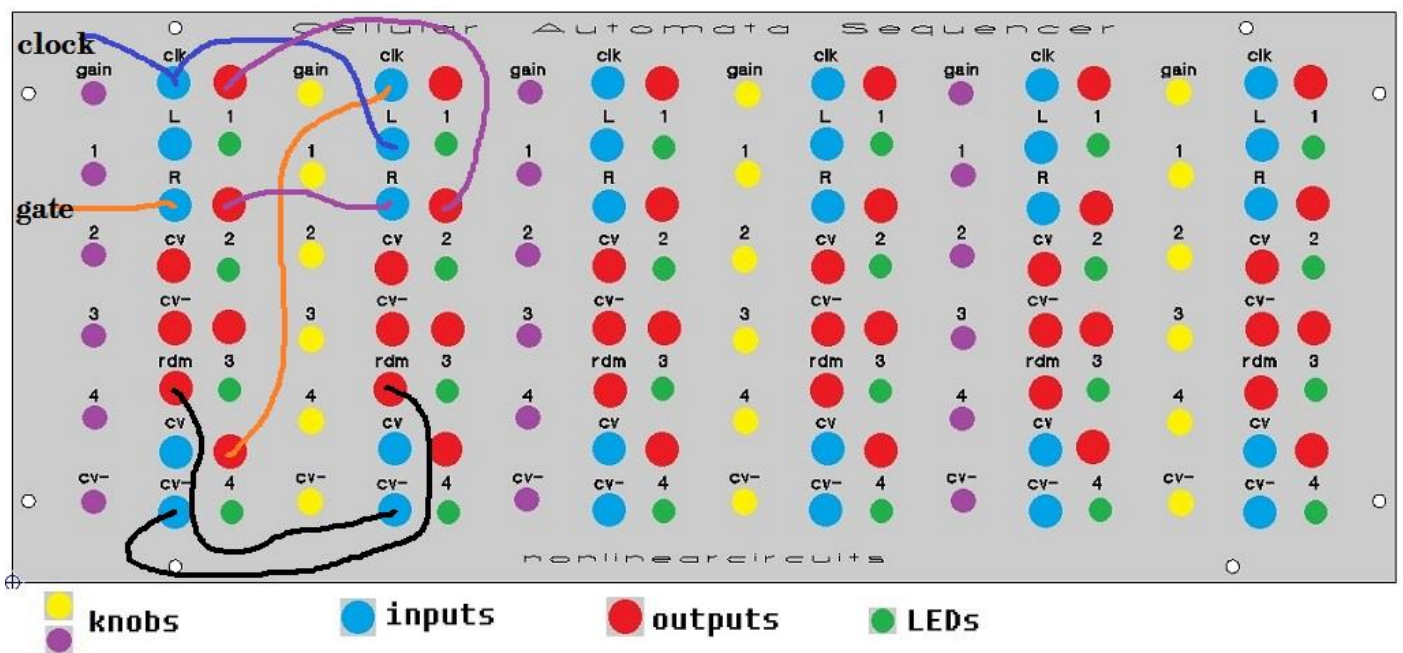
Using multiple neighbourhoods together

Of course the whole point of this panel is to chain multiple neighbourhoods together, chain the CVs together in various combinations and pop in the slewed random voltages here & there for a gamut of CVs to control everything else.

There are many combinations of patches to be tried, I will give a few here as examples.



In the above patch, all of the neighbourhoods run off the same clock signal. Just one gate signal is used and patched through to activate every neighbourhood. The CVs are chained and summed outputs can be used elsewhere in the synth.



A smaller patch, neighbourhood 1 receives a clock signal which also goes to L input of neighbourhood 2.

Cell 4 of neighbourhood 1 supplies a clock signal to neighbourhood 2. The random outputs are fed to the CV- inputs of the other neighbourhood.

I suggest stop reading this, plug the panel in and try it out. Sometimes it can be tricky getting patterns to start as there has to be a 'high' gate on the L or R input when the clock signal arrives. Once you get used to using the panel this is rarely an issue.

The beauty of massive patches using several neighbourhoods, is that adjusting one pot in one neighbourhood, or shifting one patch-cord to get a gate from a new source will have a flow on effect that eventually changes everything going on.

Another interesting effect can occur when no external gate signal are used. The only signal coming from outside of the panel is the clock signal, and all the gates are patched from cell outputs to L & R inputs.

When you turn on the power, usually a number of cells are activated and these start off the pattern. Eventually some neighbourhoods end up in pattern loops and others die out. If you input an external gate signal and then remove it, you can create whole new patterns which will eventually die out or get stuck in loops.