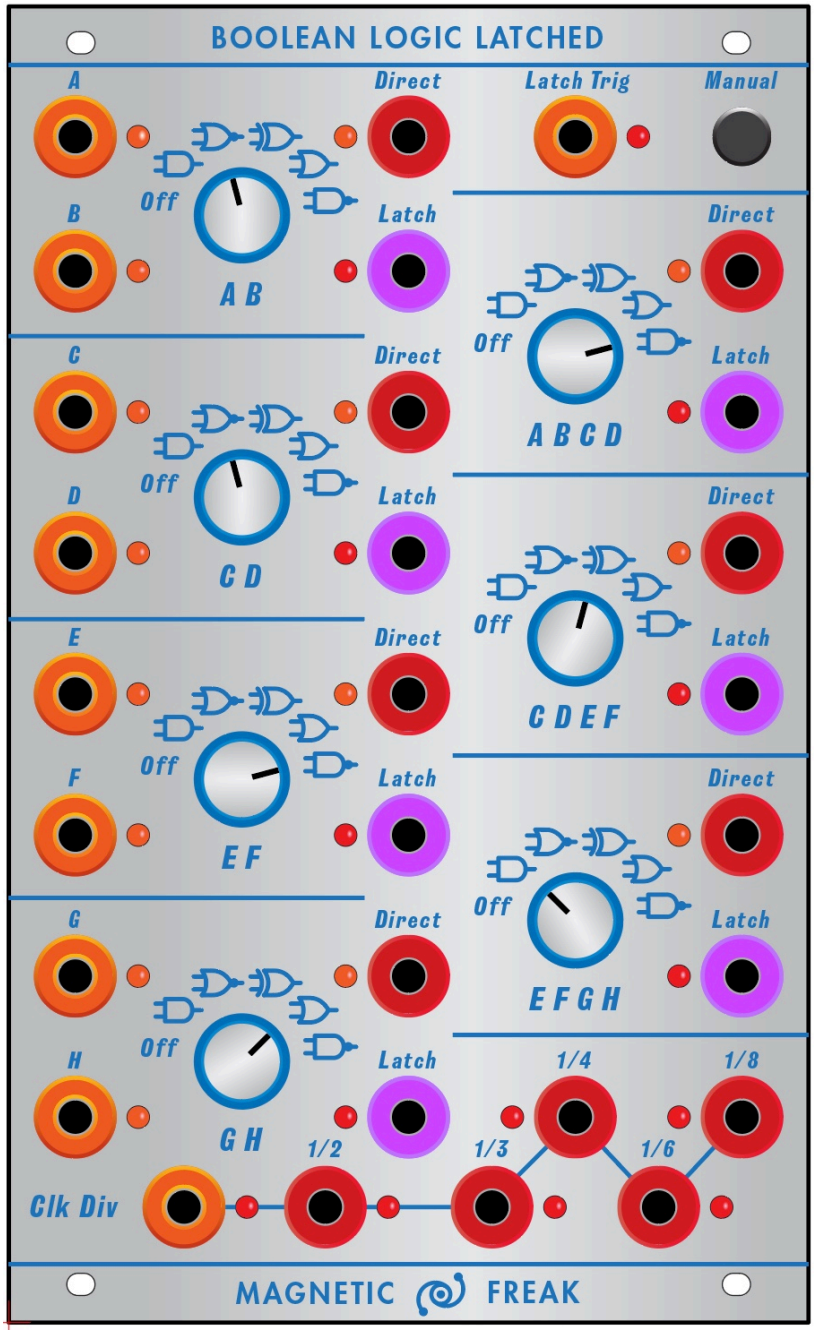
Boolean Logic Latched User Manual

Dr. Robert Kieffer *Last Updated jan 2024*



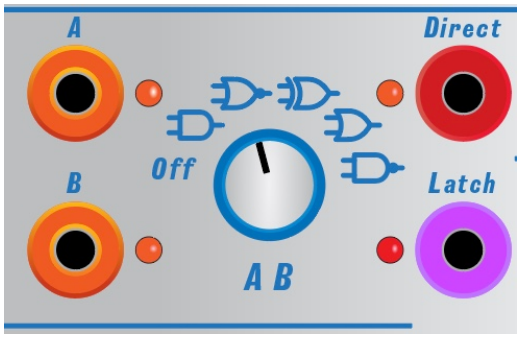
Specs

- Power supply consumption +15V at 50mA
- GATE IN range 0V to 10V (accept negative) detection threshold 3V
- GATE output 13V pulses are emitted
- Each connector have a corresponding LED to visualise timing.
- Panel width : 1U standard
- Sloted fixation holes diam 3.2mm
- Minimum depth needed in the boat 35mm

Introducing the Boolean Logic Latched

Because coincidence is an art and also a mean to measure correlation, the Boolean Logic Latched module offers all possible kinds of existing boolean logic operations. With the seven available gate operators you could perform complex Boolean equation in order to produce all kinds of rhythm sequences from dead simple cycling logic signals. A clock divider lets you derive multiple clocks from a single one and could be patched as a source for the gates operators.

The basic logic operator units



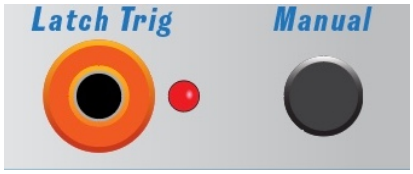
The operator unit uses two logic inputs (here A and B) to perform the operation selected by the rotary switch (Off, AND, NOR, XOR, OR, NAND). The result of this operation is availlable in real time at the red banana output labeled "Direct". The result is also latched on the arrival of an external trigger on the violete banana output labeled "Latch" (see next about it).

Here I just want to give a quick review of boolean logics for those who are not so familiar with it. The following table shows the expected output status with all possible logic operations from the selector given the status of the A and B input. The 0 means LOW (or nothing connected), the 1 means HIGH.

INPUTS		OUTPUT					
A	B	Off	And	Nor	Xor	Or	Nand
0	0	0	0	1	0	0	1
0	1	0	0	0	1	1	1
1	0	0	0	0	1	1	1
1	1	0	1	0	0	1	0

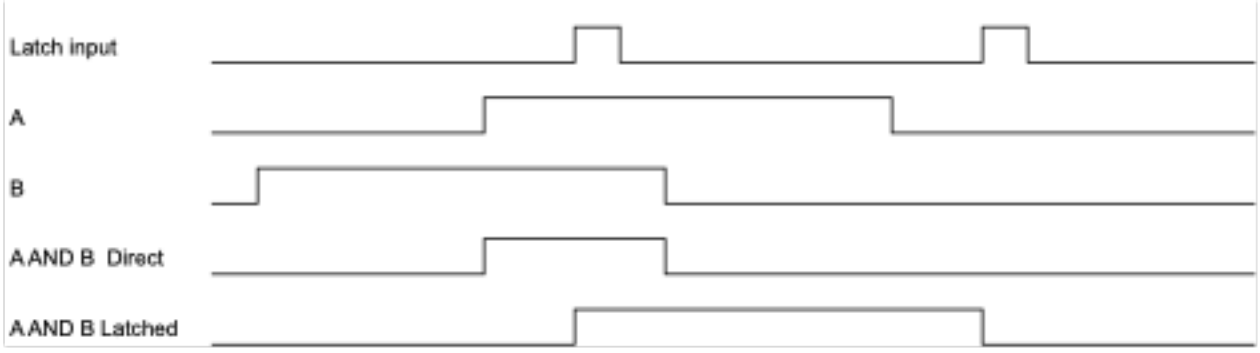
The order of this table is the same as for the rotative selection switch, and it has been choosen so that it should increase the occurence of getting a HIGH output when rotating clockwise from Off to NAND.

The latching inputs and principle

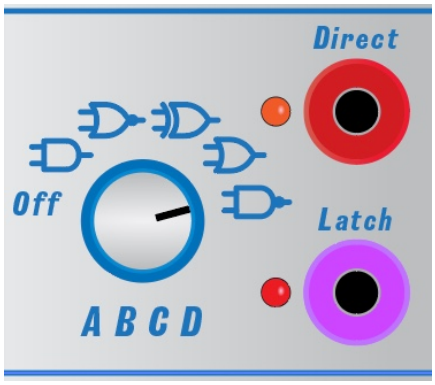


The Latch function acts like a sample and hold of the status of all operator outputs, as soon as a trigger is received in the Latch Input the whole Logic combination status of the module is Latched (sampled) and will hold on the Latched outputs until the next trigger comes. Latching could also be triggered manually by the push-button on the top right of the module.

The following time diagram give an example of the AND operation applied on A and B input signals. The direct output (red) give the result in real time. The rising edge of the Latch input signal copy and hold the A AND B result on the Latched output (pink).



The second level operator units

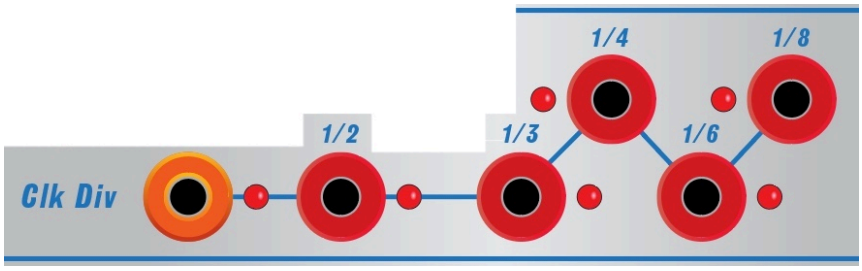


There are three of such operators in the module. These operators basically use the red "Direct" output of the two left side operators (here AB and CD), to make an operation.

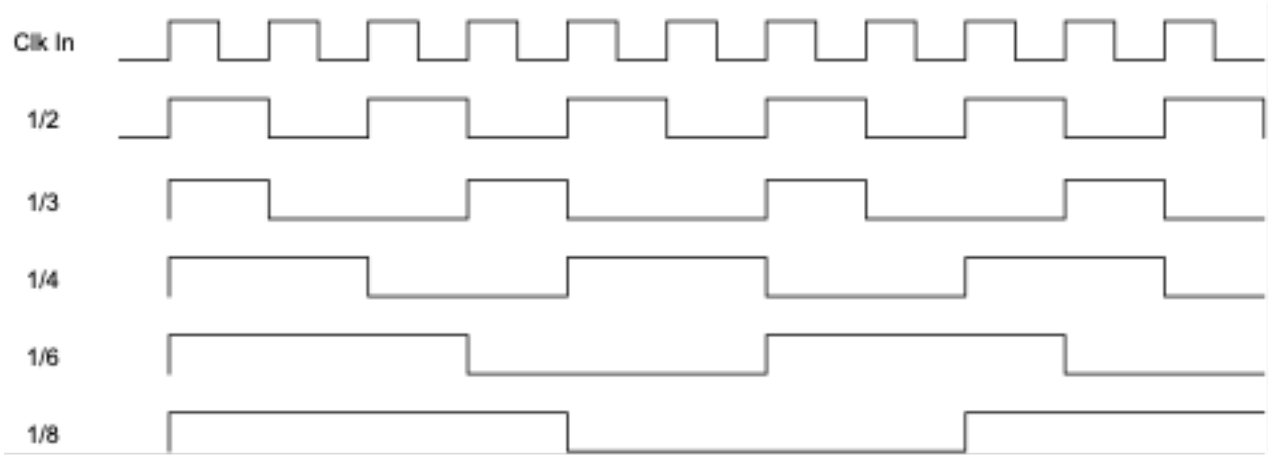
Let me give a short example, if you select the AND on AB, on CD and on the second stage ABCD, you will get a 4 input AND logic overall.

Again as for the first level operators there are "Direct" and "Latched" outputs available.

The clock divider section



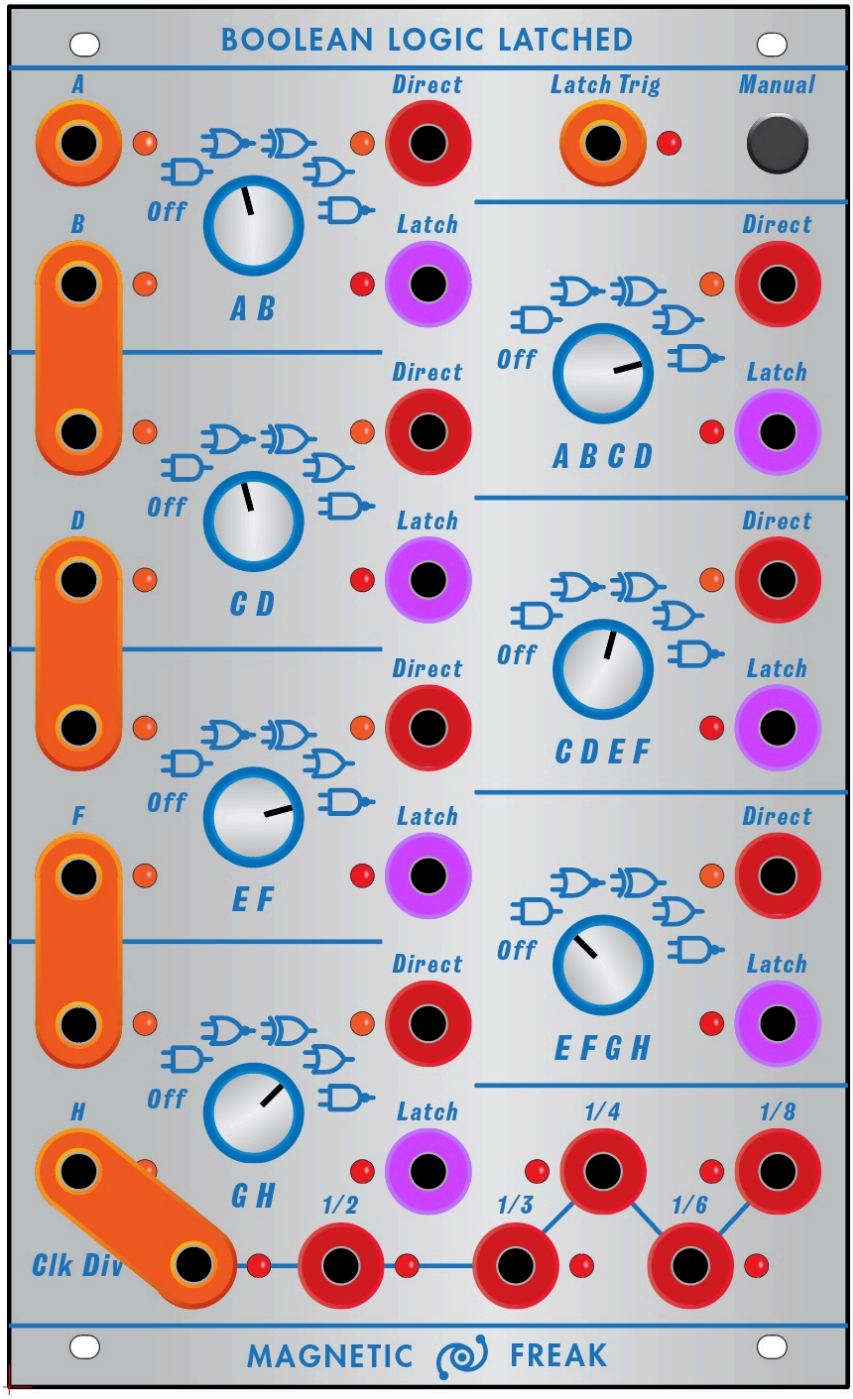
The clock divider provides 1/2 , 1/3, 1/4, 1/6 and 1/8 subdivisions of an incoming clock signal. It is very useful to create source signals for the boolean logic, allowing you to program reproducible rhythmic patterns from a single clock source.



Be aware that if you use a bunch of non-synchronized clock signals as inputs you will start to get interesting evolving patterns with the logic gates and that could also bring wonderfull results with more life in it.

Using the banana bridges

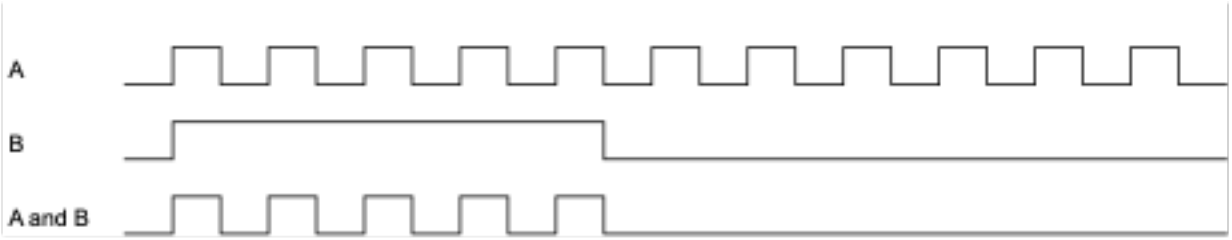
Some zones of this modules are made to allow the fitting of banana bridges for more compact patching, mainly for sharing a signal with two operators. This will help reduce the number of patching cables in your view. Appart from the positions shown there is no further places where it would make sens to use such briges.



Some tips and tricks about boolean logic

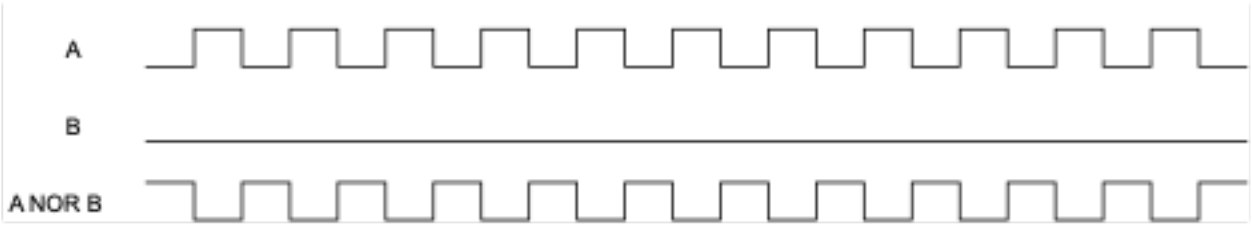
Gating a signal

The AND function could be used to gate a very fast ticking signal with a much slower clock of variable width in order to create a periodic burst.



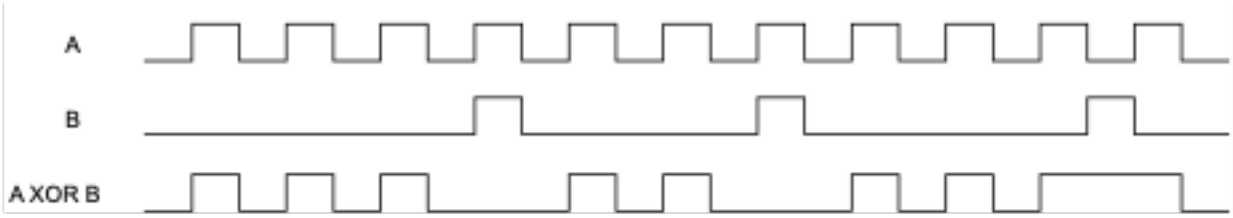
Performing a Not

If you want to perform the NOT operation on a signal just plug it in either inputs and apply the NOR operation. As nothing is plugged in the second input in that case it will stay low state all time an therfore perform a NOT operation on your incoming signal.



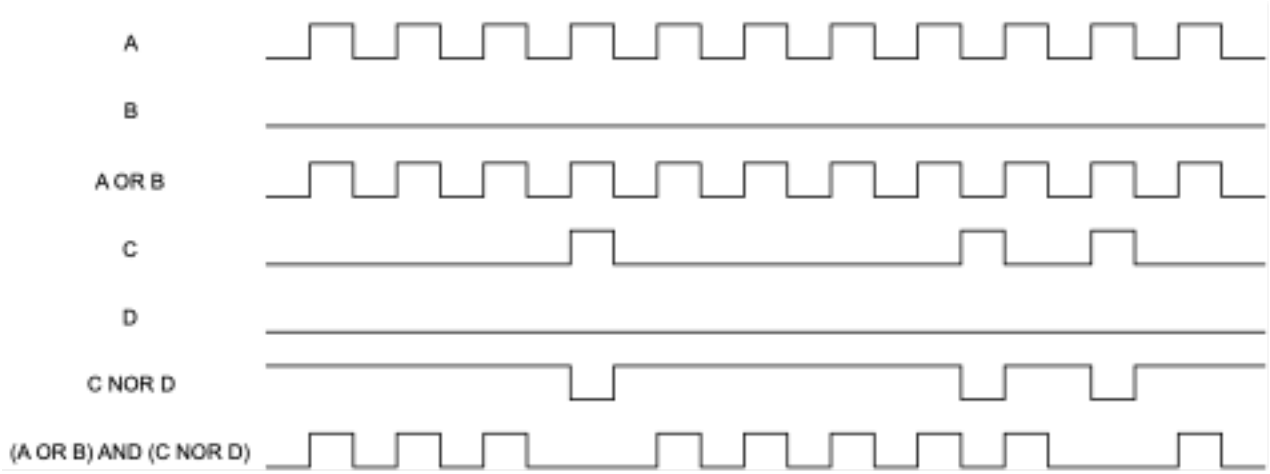
Making a veto

If you want to create a kind of veto on a signal you could use the XOR. In that case when a signal is coming on the input A it will be present on the output only if there is not signal on input B... But if signal B is sent HIGH when A is LOW then you will also get signal out HIGH (check last B pulse on image).



Another possibility is to use two stages, one for inverting a signal using a NOR, the second to do a AND of the signal you want to copy and the veto signal previously inverted. Set a signal on either A or B and select OR. Pluging only one signal in a OR with the second input by default to low, it will effectively copy your signal and forward it to the next stage. Put the second signal on either C or D and select NOR to invert the

signal to be used as a veto, finally select the AND on ABCD second stage, you would phrase it something like (A) AND (NotC) , making C the veto acting on A your pass through signal.



Application examples

Using the Boolean Logic to start the envelopes of all kinds of rythmical events is the most straightforward use of the module, from a single clock, and with the help of a few banana cables you will quickly create some interesting rythmical patterns, something IDM-like (Intelligent Dance Music).

Making use of the Latching system will allow you to resynchronise up to four LFO oscillators using the Latch Trigger Input as the reference resynchronising clock for all signals copied on the Latched outputs. Allowing to perform randomness with groove.

The gates are analog circuitry so it works up to very high frequency. It can reach easily 20khz if you plan to try to feed it with audio-rate signals it would certainly give interesting results.

Further readings

I found this very interesting book in case you want to teach your childrens how to program the Boolean Logic Latched module, or in case you are interested by it yourself.