

Banaan Electric Instrument

Overall Structure

Banaan Electric consists of a chain of effects.



Abbildung 1: Banaan Electric Instrument

The stereo input signal is converted to mono and run through a gate, then through a tube compressor, amp simulator, phaser/flanger, chorus/tremolo, dual delay, reverb, stereo compressor and finally output amplifier.

Stereo to Mono conversion

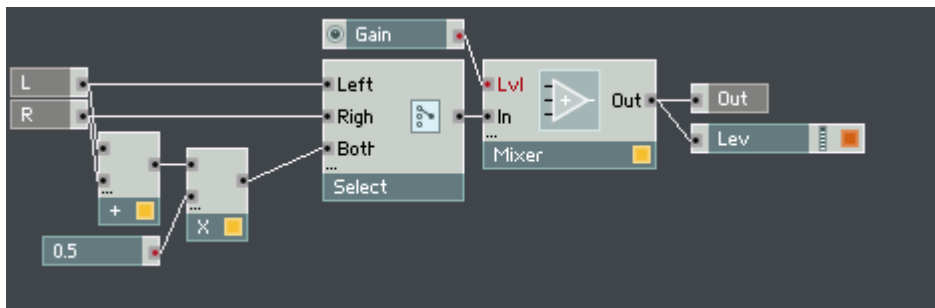


Abbildung 2: Stereo to Mono conversion

The input select switch chooses between L, R and mixed both signal. The signal is amplified by the input gain scaler, and shown using a level meter.

Input Gate

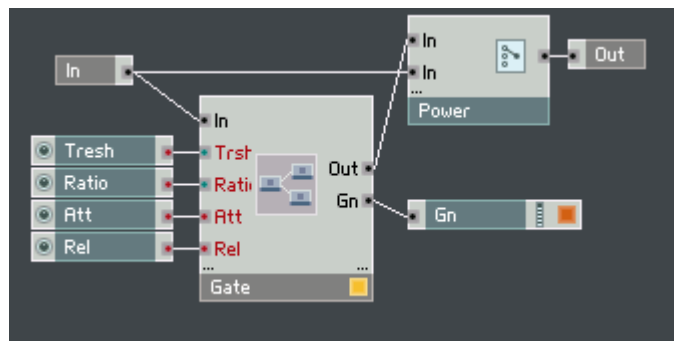


Abbildung 3: Input Gate Macro

The input gate is controlled by Tresh, Ratio, Att, Rel controls. The switch at the output enables the gate.

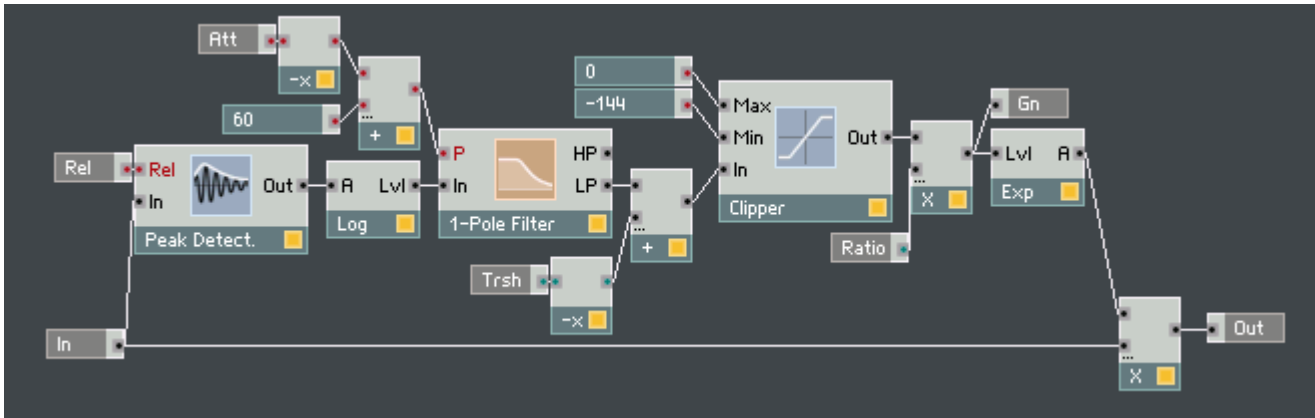


Abbildung 4: Gate Macro

The incoming signal is run through a peak detector controlled using the Rel parameter, the output is converted to db, and run through a LP filter controlled by the Att parameter (the higher attack, the lower the cutoff frequency). The inverse of the threshold is added to the filtered peak signal, clipped between 0 and -144, multiplied by the ratio, and converted back to an amplitude amount to scale the incoming signal.

Tube Compressor

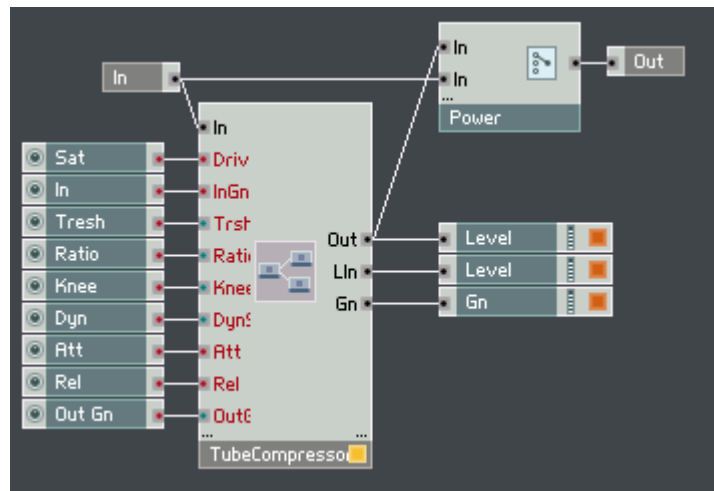


Abbildung 5: Tube Compressor Macro

The structure of the top-level tube compressor is similar to the other macros, it contains only controls, an output switch to enable the effect, and readout level meters.

The incoming signal is amplified using the Drive parameter, saturated, amplified again by the Input Gain. This signal is run through a peak detector chain, distorted using a soft knee distortion, run through a LP filter, multiplied by the ratio, added to the output gain, and used to scale the outgoing audio. The incoming audio is dynamically saturated using the output of the LP-filtered peak detector data.

The Soft Knee Saturator take the incoming audio, adds the knee parameter and clips it between 0 and 144, multiplies the output with $1/(knee/2)$, saturates the signals and adds it back to the original signal before multiplying it with $knee/2$.

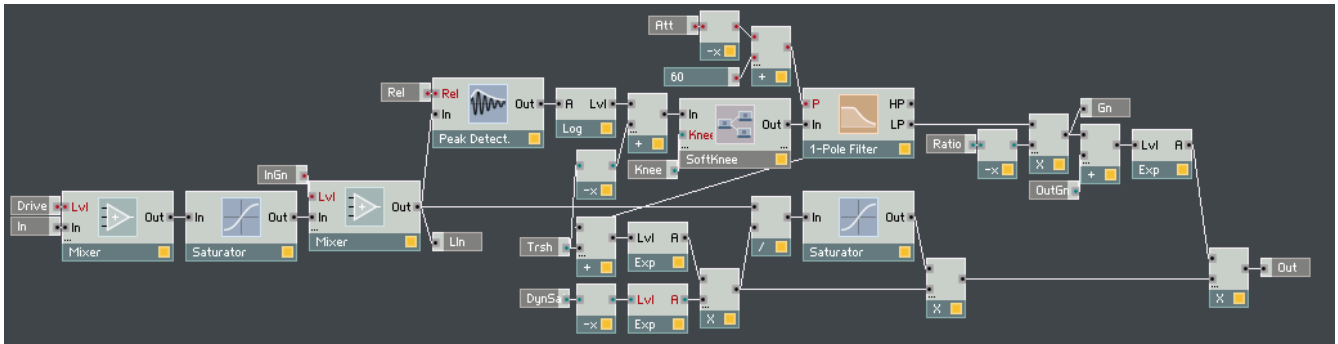


Abbildung 6: Tube Compressor Macro

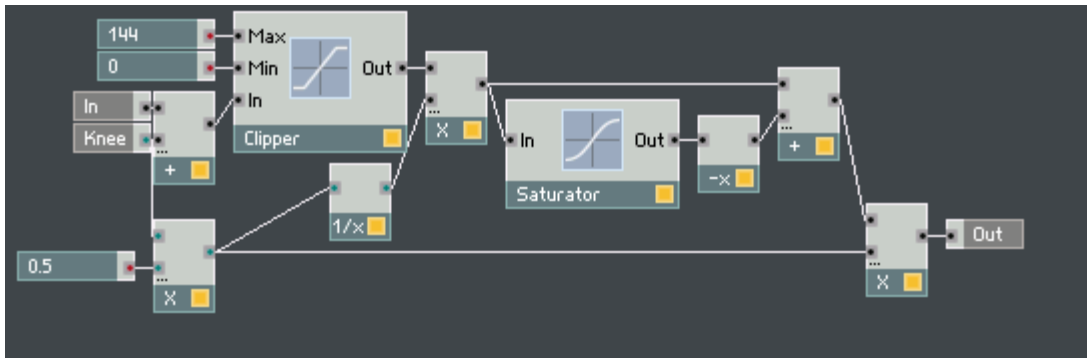


Abbildung 7: SoftKnee Macro

Amp Simulation

For the amp simulation, the audio signal is first run through a pre EQ, then through an amplification stage, through a speaker simulation, and finally a post EQ before being mixed with the incoming audio signal.

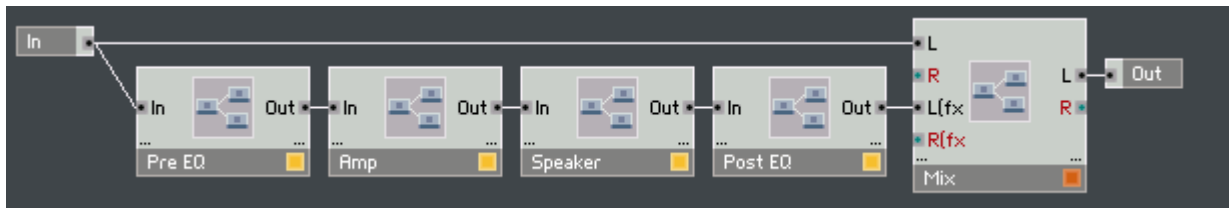


Abbildung 8: AmpSimulation Macro

The pre-EQ is built around a 2-pole filter and a mixer, with a slight boost for the mid signal. The mid and low audio signal is delayed by one sample (don't really understand why).

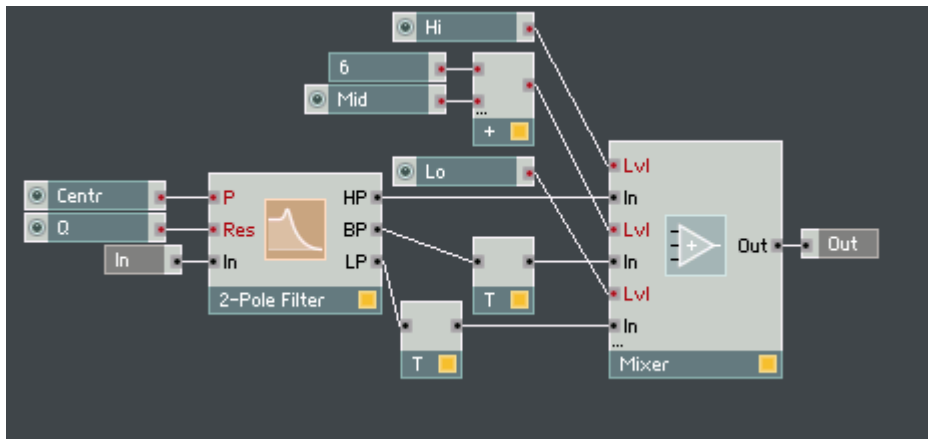


Abbildung 9: Pre EQ Macro

The Amp simulation is built around 2 filters (one 1-pole filter and a ladder 4pole filter), amplified according to the Dist control, and run through an asymmetric parabol saturation macro.

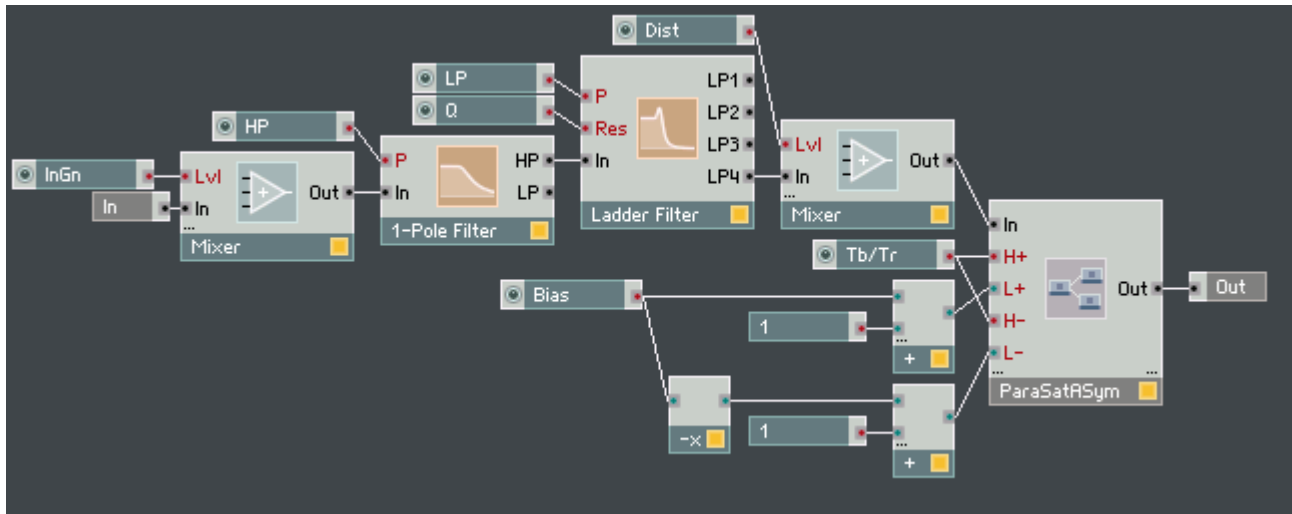


Abbildung 10: Amp Macro

The asymmetric parabol saturation is built around 2 parabol saturation macros. The bias control is fed to the first saturator, and the inverse of the bias is fed to the second saturator.

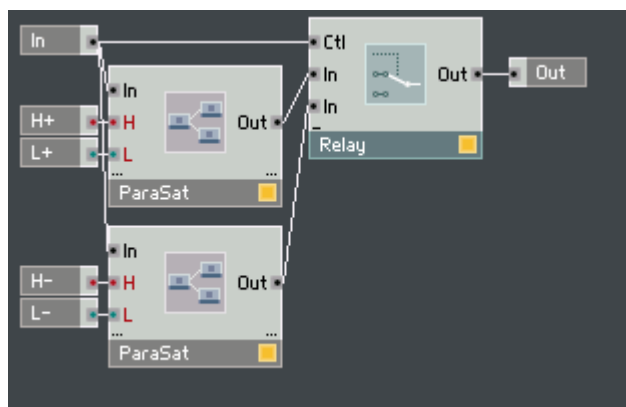


Abbildung 11: ParaSatASym Macro

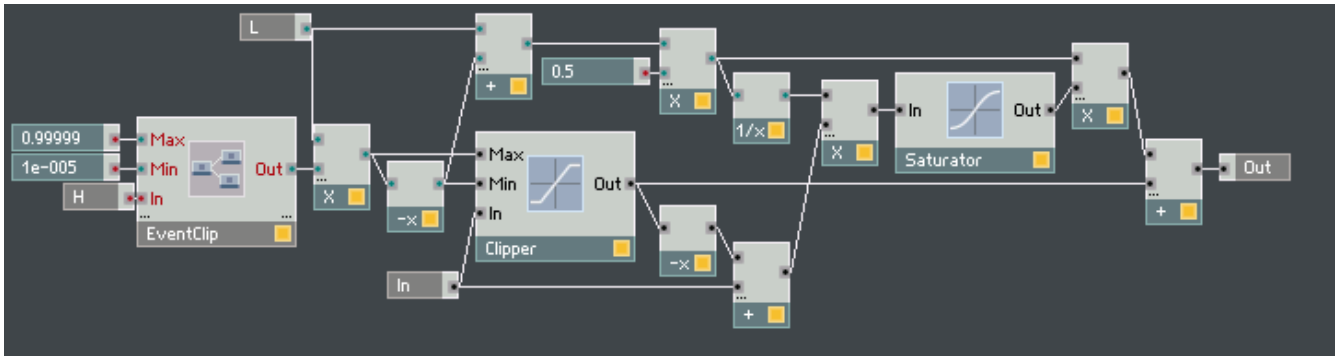


Abbildung 12: ParaSat Macro

The saturation parameter is clipped between 10^{-5} and 0.999 using the EventClip macro.

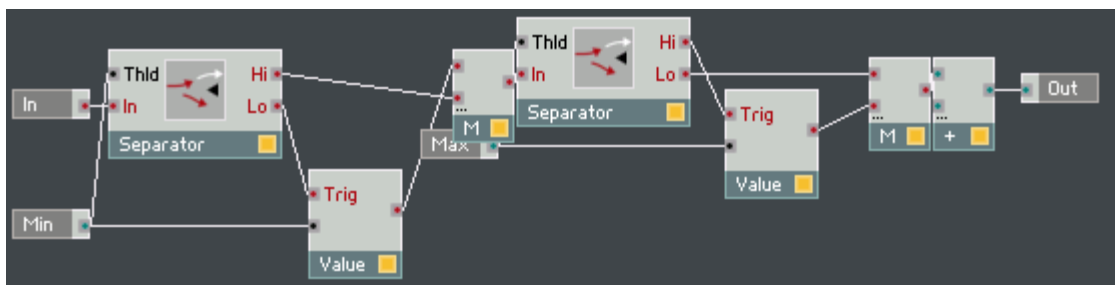


Abbildung 13: EventClip Macro

If In is below Min, Min is latched else In, and if In is above Max, Max is latched, else In. I don't know why the + 0 is used at the end, maybe the remnant of a previous version. The bias is multiplied with the H parameter, and the input signal is clipped between $H * L$ and $-H * L$. The inverse of the clipped signal is added to the input signal, which is then scaled by $1 / 0.5 * (L - H * L)$, saturated, and then multiplied back with $0.5 * (L - H * L)$. The output is added to the clipped H parameter. For negative values of the input signal, the lower saturator is chosen, else the uppersaturator is used.

The ParaSat can advantageously be implemented in Core.

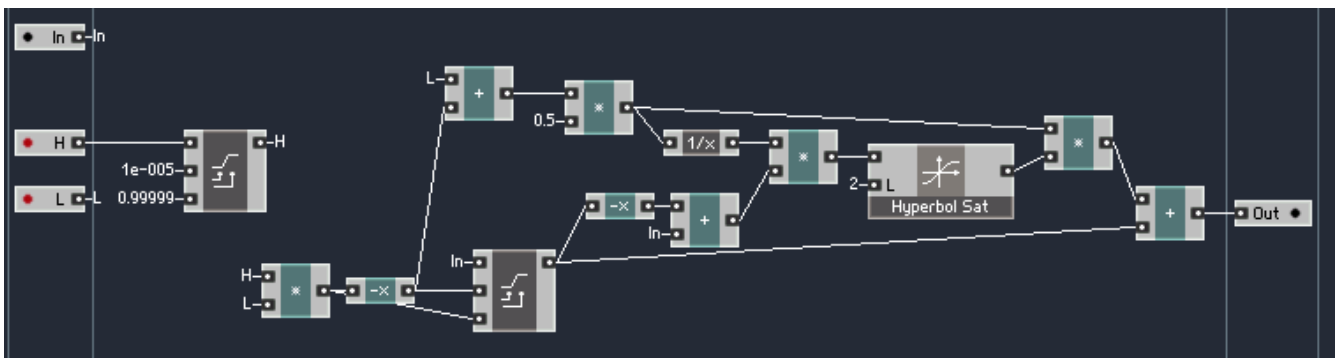


Abbildung 14: Core ParaSat

The Hyperbol Sat is not exactly identical to the reaktor Saturator module, but it does the work.

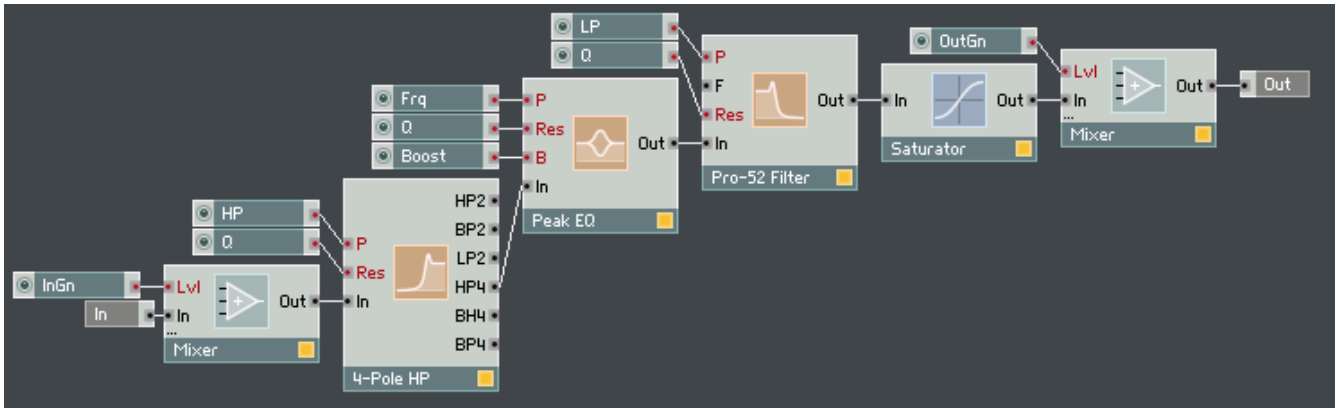


Abbildung 15: Speaker Macro

The Speaker simulator is implemented using a pre-amplifier controlled by InGn, a post-amplifier controlled by OutGn, and 3 filters followed by a saturator. The first filter is used as HP filter, the second filter as a boost/notch EQ, and the last filter as a LP filter.

Finally, a post-EQ is used.

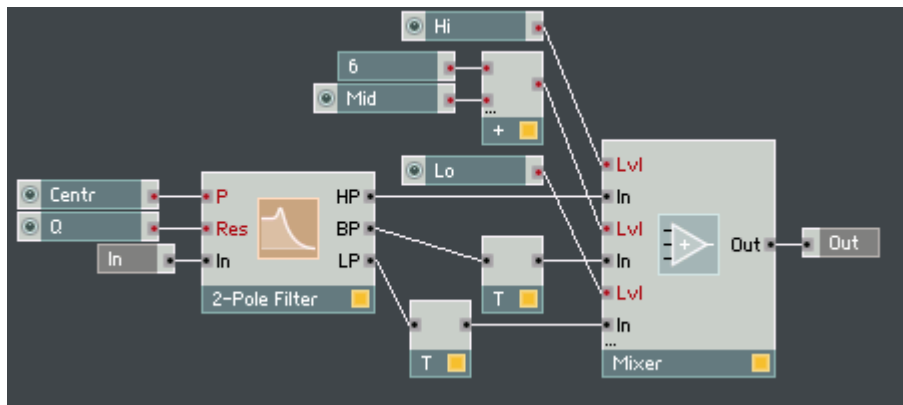


Abbildung 16: Post-EQ Macro

Finally, the amp simulator output is run through a switch in the Mix Macro.

Flanger

The flanger first converts the mono signal to a stereo signal, runs it through the Flanger Macro, and finally through a Mix macro to enable/disable the effect.

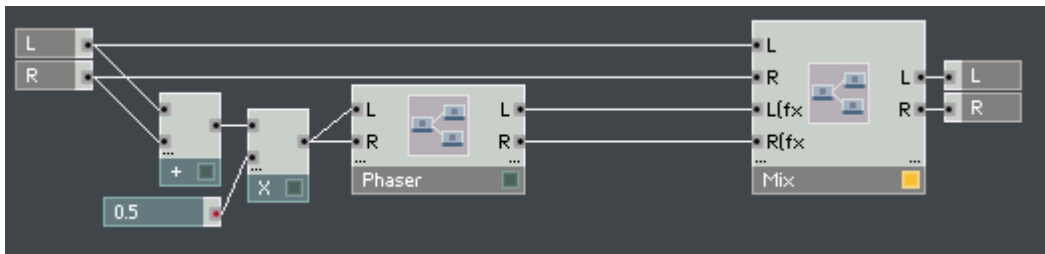


Abbildung 17: Flanger/Phaser Macro

The Phaser is implemented in the Phaser Macro.

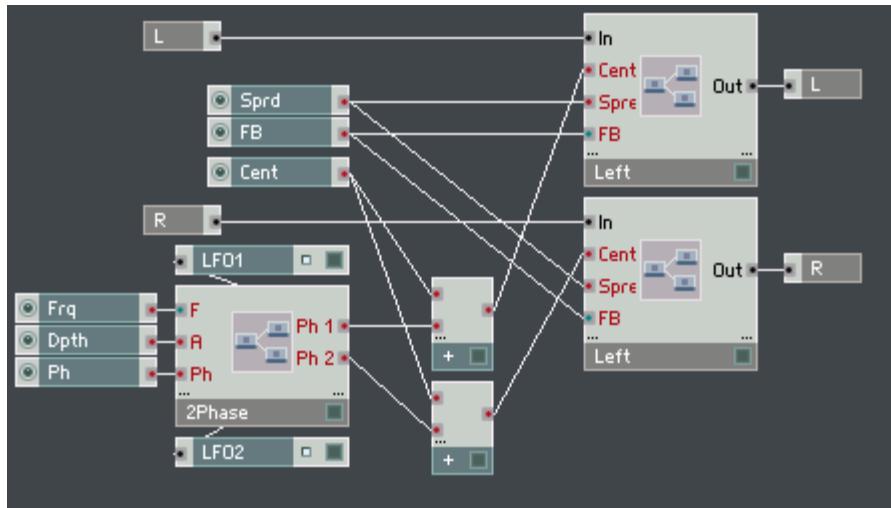


Abbildung 18: Phaser Macro

Two phasing LFOs are controlled by the Frq and Dpth parameters in the 2Phase macro. The outputs of the LFOs are added to the Cent parameter (the pitch at which phasing takes place). The first LFO controls the phasing of the left audio signal, the second LFO is used to phase the right audio signal.

The LFOs are implemented in the 2Phase macro, which uses a ramp LFO run through a sine function and a phase-shifter sin function. They are scaled by the LFO depth.

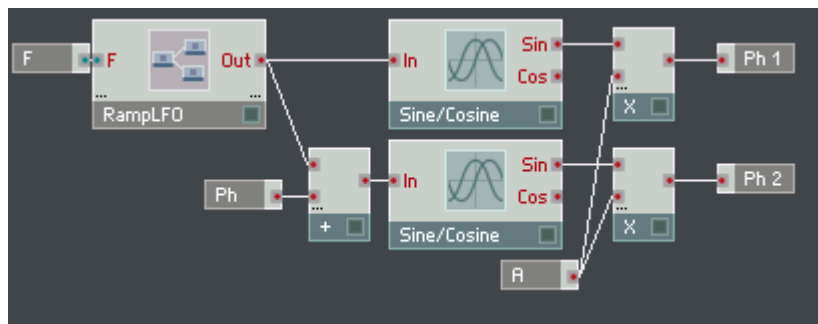
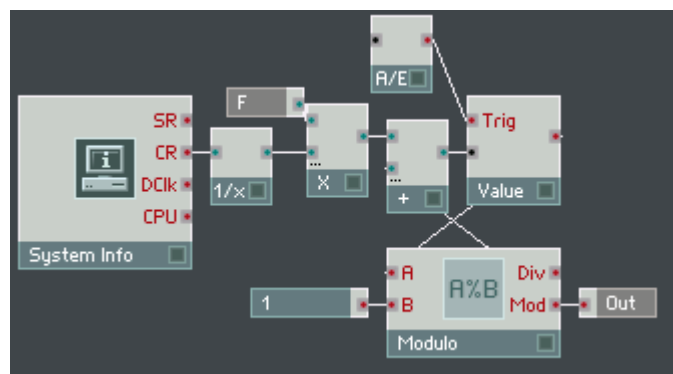


Abbildung 19: 2Phase Macro

The RampLFO is implemented in the RampLFO Macro, which takes F/control rate to increment a counter modulo 1. Thus, 1 is reached F times per second.



The phaser itself is implemented in the Left Macro, which takes the input signal, two control parameters Spread and FB, and of which the center frequency is modulated by the LFOs.

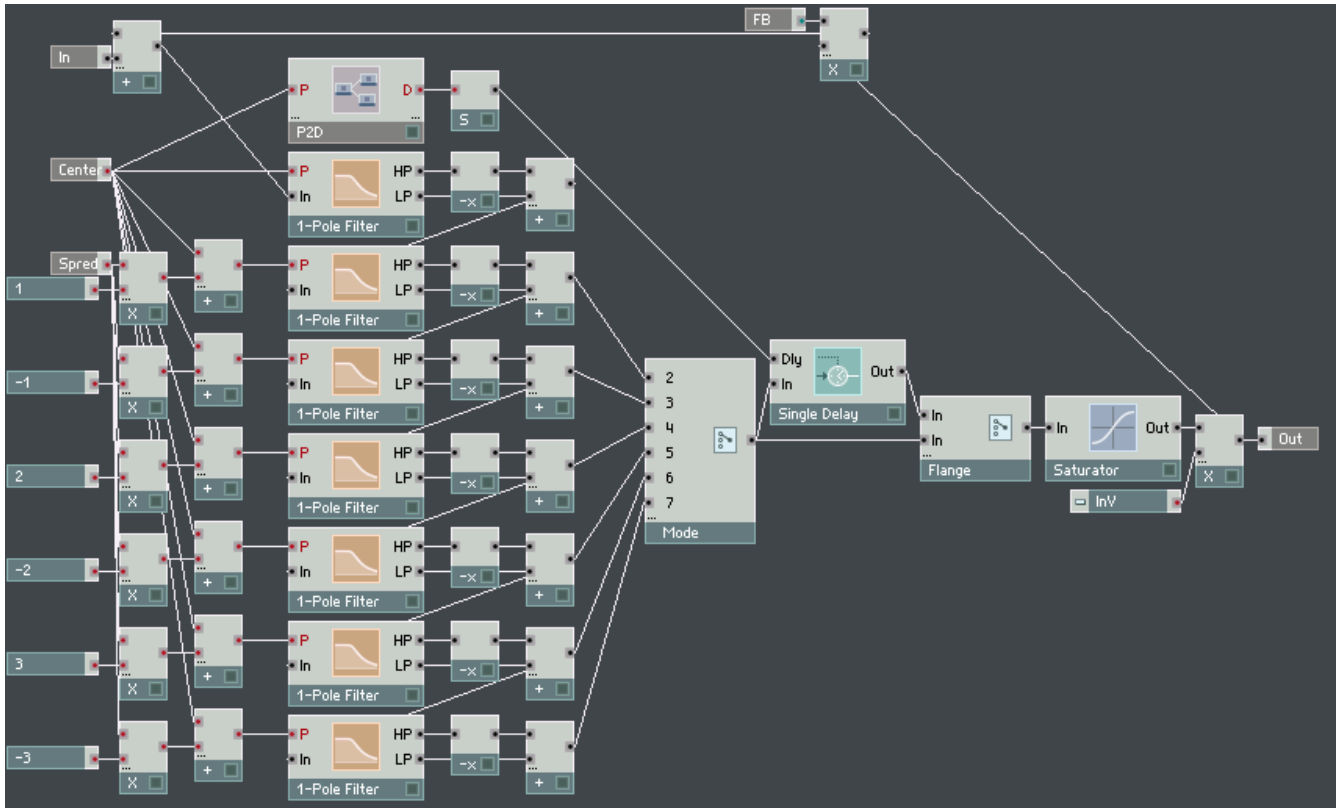


Abbildung 20: Left Macro

The Center frequency is added to the spread parameter multiplied by 1, -1, 2, -2, 3 and -3 and used to control 6 filters. The HP is inverted and added to the LP signal before being fed into the next filter. The output of one of these filters is chosen by the Mode switch, before being delayed shortly (the delay length is modulated by the center frequency too (this is the flanging)). The output is saturated, inverted according to the inv button, and fed to the output and into the feedback loop.

Chorus/Tremolo

The Chorus is implemented in the Rotor cell, and a Mix Macro is used to control the output of the effect.

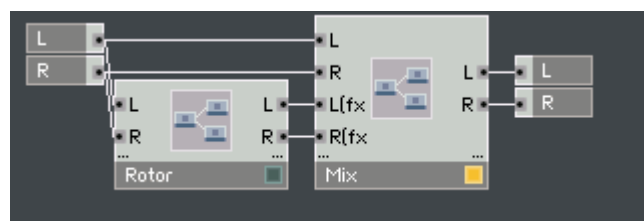


Abbildung 21: Chorus/Tremolo Macro

The Chorus is implemented by 2 modulated delays and 2 modulated amplifiers (2 for the left side and 2 for the right side). The modulation is controlled by 4 phase shifted sine LFOs.

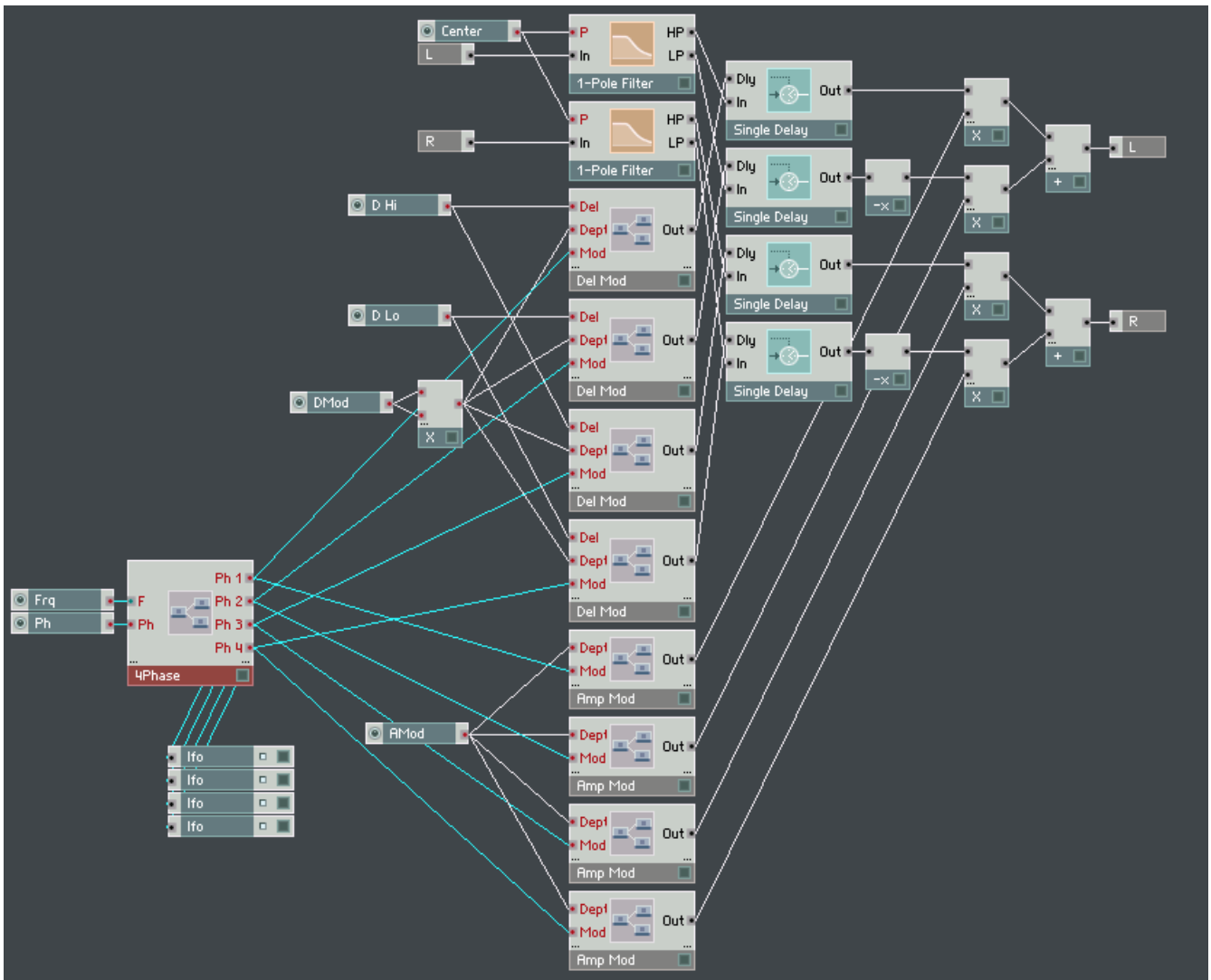


Abbildung 22: Rotor Macro

1 delay and 1 amplifier control the HP-filtered signal, and the other delay and amplifier control the LP-filtered signal. The Del Mod and Amp Mod macros are used to calculate the delay and amplification amount modulated by the LFOs, which are generated in the 4Phase Macro (very similar to the 2Phase Macro).

A RampLFO is transformed to 4 phase-shifted Sine waves.



Abbildung 23: DelMod Macro

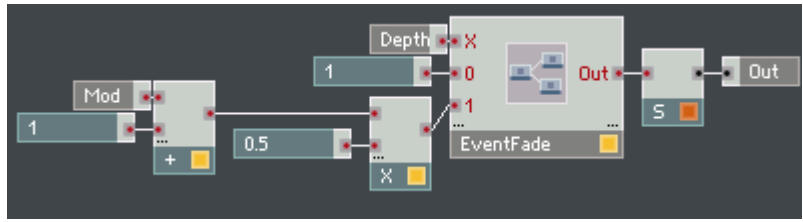


Abbildung 24: AmpMod Macro

The Depth parameter is used to interpolate between 1 (normal amplification, no modification to the input signal) and the modulated amplification value. This interpolation is done in the EventFade macro (a selector with linear interpolation could have been used).

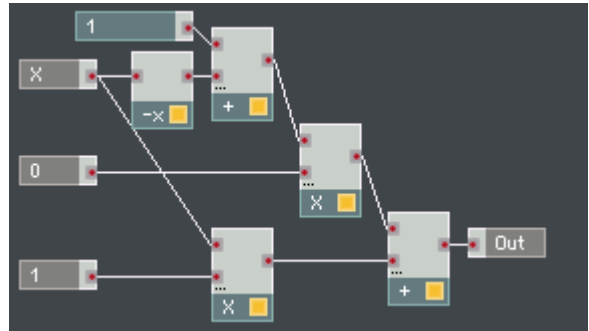


Abbildung 25: EventFade Macro

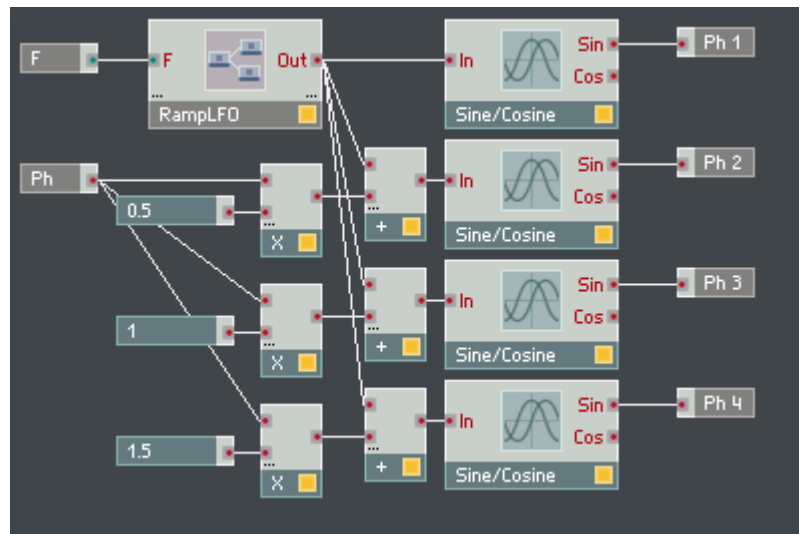


Abbildung 26: 4Phase Macro