

Make Noise Maths Behavioral Specification for a VCV Rack 2 Emulation

Executive summary

Dragon King Leviathan, the “engineering takeaways” that matter most for a faithful digital Maths are these:

- The **core outer channels (1 & 4)** are a **dual-mode, direct-coupled slope/function generator**: in **Signal In** mode they behave as a slew/lag processor (and thus naturally implement ASR and envelope following); in **Trigger In** mode they generate a **transient 0→10 V→0 envelope with no sustain**, and are **re-triggerable only during the Falling segment** (not during Rising). ¹
- The **EOR/EOC “event” outputs** are **not documented as narrow pulses**; they are described as **binary 0/10 V “event signals”** with **stable default states** (EOR defaults LOW when idle; EOC defaults HIGH when idle). In cycle use, this strongly implies **gate-like phase indicators** (“high during a segment”), not 1 ms triggers. ²
- **Cycle** in the 2013 revision adds a **Cycle Input**: “Gate HIGH cycles on; Gate LOW cycles off”, with **HIGH $\geq +2.5$ V**. This is functionally a **run/stop gate** for self-cycling. ³
- The **BOTH CV input** is explicitly **bipolar exponential** and **inverted in sign** relative to Rise/Fall CV: **positive BOTH speeds up (shorter total time)**; **negative BOTH slows down**. Rise/Fall CV are explicitly **linear**, and **positive increases time** while negative decreases. ⁴
- The **Vari-Response** knob is explicitly described as a continuous curve family **LOG → LIN → EXPO → HYPER-EXPO** (2013 revision) and defines its “LOG vs EXPO” direction by **slope increasing/decreasing with voltage**. The manual explicitly states that changing response **also changes timing** (“affects Rise and Fall Times”). ⁵
- **Channels 2 & 3 are attenuverting offset/mix channels**. Their inputs are **normalized to reference voltages** when unpatched to generate DC offsets: **Classic: CH2 and CH3 normalize to +5 V; 2013 revision: CH2 normalizes to +10 V, CH3 to +5 V**. ⁶
- The **Variable Outs (1–4)** are **normalised into the SUM and OR busses**, and **patching a cable into a Variable Out removes that channel from both busses**. This “break-normal” behavior is essential to emulate for correct bus interaction. ⁷
- **OR is analog MAX over positive voltages only**: it “outputs the highest voltage” but **does not respond to negative voltages**, so in software it should behave like `max(0, inputs...)` (post-attenuverter). ⁸
- There is **no publicly available official Maths schematic** in the primary sources retrieved here; therefore, **exact knob laws, CV scalings, comparator hysteresis, and analog saturation points require measurement** if you want a “hardware-indistinguishable” emulation. (Measurement plan provided below.) ⁹

Module I/O map and signal conventions

Versioning used in this report

- **“Classic”** = original *Maths Classic* (often referred to in community as “Maths 1”). ¹⁰
- **“Revision 2013”** = *Maths revision 2013* manual (often referred to in community as “Maths 2”). The manual explicitly calls it the “direct descendant” of the original with upgrades and evolutions. ¹¹
- A community note indicates that the older unit’s physical LFO/cycle switch state may persist across power-off, while the newer does not (relevant only to hardware UX; in Rack you’ll typically serialize state in the patch). ¹²

Global voltage conventions (module-side)

Verified (Revision 2013):

- **Signal inputs (CH1 & CH4)** are **direct-coupled** and specified as **± 10 V** range. ¹³
- **Rise CV / Fall CV / Both CV inputs** on CH1 & CH4 are specified as **± 8 V range**. ¹³
- **Variable outputs (1–4)** are specified as **± 10 V** output range. ¹⁴
- **SUM output** is specified as **± 10 V**. **INV SUM output** is also **± 10 V**. ¹⁵
- **OR output** is specified as **0–10 V** and documented as “does not respond to negative voltages.” ¹⁵
- **EOR and EOC outputs** are **binary levels: 0 V or 10 V**. ¹⁶

Verified (Classic):

- **Trigger-generated transient** is described as a **0 V \rightarrow 10 V \rightarrow 0 V** envelope. ¹⁷
- **CH2 and CH3 Signal inputs** are **normalized to +5 V** for offsets. ¹⁸

Version conflict to track (input range wording):

- Revision 2013 explicitly says CH1/CH4 Signal In range **± 10 V**, but CH2/CH3 Signal In list “Input Range +/-10Vpp.” ¹⁹
- **Inferred (software choice):** treat all **Signal In** jacks as supporting at least **± 10 V** in Rack emulation (with soft/hard clipping configurable), because the module is designed as a “direct coupled” processor across channels and the same manual elsewhere uses ± 10 V language for outer inputs. ²⁰

Normalizations and busses

Verified (Revision 2013):

- The **Variable Outs 1–4** are **normalised to the SUM and OR busses**. If *no cable* is patched into a Variable Out jack, that channel’s attenuverted signal is injected into the bus; if a cable *is* patched, that channel is removed from the busses. ¹⁵
- **Unity Signal Outs (CH1 & CH4)** are *not* normalised into the SUM/OR bus path, and patching them **does not remove** anything from SUM/OR busses. ²¹

Verified (Classic):

- All four **Signal OUT** jacks are normalized to SUM/OR, and patching removes them from the busses. ¹⁸
- Classic provides a **Signal OUT Multiple** (CH1 & CH4) that is *not* normalized to the busses and therefore does not remove the channel from SUM/OR. ¹⁸

I/O-by-channel

Below, “panel intent” is what the manual states the control/jack is for; “electrical behavior” is what must be emulated.

Channel 1

Revision 2013 (Verified):

- **Signal Input:** direct-coupled input to the CH1 circuit; used for lag/portamento/ASR; specified range ± 10 V. ²²
- **Trigger Input:** gate/pulse triggers CH1 regardless of Signal Input activity; result described as **0→10 V transient**, shaped by Rise/Fall/Vari-Response. ²²
- **Cycle Button + Cycle LED:** engages/disengages self-cycle. ²²
- **Rise knob:** CW increases rise time. ²³
- **Fall knob:** CW increases fall time. ²³
- **Rise CV (linear, ± 8 V):** positive increases rise time; negative decreases relative to knob setting. ²³
- **Fall CV (linear, ± 8 V):** positive increases fall time; negative decreases relative to knob setting. ²³
- **Both CV (bipolar exponential, ± 8 V):** positive decreases total time; negative increases total time. ²³
- **Vari-Response knob:** continuously variable LOG→LIN→EXPO→HYPER-EXPO; tick indicates linear. ²⁴
- **Cycle Input:** gate-controlled cycle state; HIGH requires $\geq +2.5$ V; low disables cycling unless Cycle button engaged. ²⁵
- **EOR Output + LED:** “End of Rise Output”; **0/10 V**; defaults low when idle; goes high when the channel reaches its highest voltage; described as useful for clock/pulse; Rise time sets delay-to-high. ²⁶
- **Unity Signal Output:** tapped from core, not affected by CH1 attenuverter; documented as **0–8 V when cycling**; otherwise follows input amplitude. ²⁷
- **CH1 Attenuverter (center column):** scales/inverts signal processed/generated by CH1; drives **Variable Out 1** and bus injection. ²⁸
- **Variable Out 1:** attenuverted CH1 output; **± 10 V range**; normalled to SUM/OR unless patched. ¹⁵

Classic (Verified differences):

- No Cycle Input jack; “Activity/Cycle Switch” is the combined LED + pushbutton. ¹⁸
- No “Unity Signal Out”; instead there is **Signal OUT Multiple** that duplicates the main output without breaking bus normalization. ¹⁰
- Vari-Response is described as LOG→LIN→EXPO (no explicit Hyper-Expo wording). ¹⁸

Channel 4

Revision 2013 (Verified):

- Functional symmetry with CH1 for Signal/Trigger, Rise/Fall controls, Rise/Fall/Both CV (± 8 V), Vari-Response, Cycle button/LED, Cycle Input ($\geq +2.5$ V). ²⁹
- **EOC Output + LED**: “End of Cycle Output”; **0/10 V**; defaults **HIGH when idle**; event is reaching lowest voltage. ³⁰
- **Unity Signal Output**: tapped from core; **0–8 V when cycling**, otherwise follows input amplitude. ³¹
- **CH4 Attenuverter** drives **Variable Out 4** and bus injection. ³²

Classic (Verified differences):

- Same “Activity/Cycle Switch” concept as CH1; no Cycle Input jack. ¹⁷
- Has **Signal OUT Multiple**. ¹⁸

Channels 2 and 3

Revision 2013 (Verified):

- **CH2 Signal Input**: direct-coupled; normalized to **+10 V** reference for offset generation (when unpatched). ³³
- **CH3 Signal Input**: direct-coupled; normalized to **+5 V** reference for offset generation. ³³
- **CH2 & CH3 Attenuverters**: provide scaling/attenuation/inversion; manual explicitly also says “amplification” for these channels. ³³
- **Variable Out 2 / 3**: attenuverted signals; **± 10 V range**; normalised into SUM/OR unless patched. ¹⁵

Classic (Verified):

- CH2 and CH3 inputs normalized to **+5 V** reference. ¹⁸
- Scaling/Inversion control is explicitly described: **NOON = zero output**, full CW = maximum positive, full CCW = maximum inverted; CH2 and CH3 offer “a small amount of gain.” ¹⁸

Bus and global outputs

Revision 2013 (Verified):

- **OR OUT**: analog OR; “always outputs the highest voltage” among CH1–CH4 variable outs; does not respond to negative voltages; **0–10 V range**. ³⁴
- **SUM OUT**: algebraic sum of CH1–CH4 variable outs; **± 10 V range**. ³⁴
- **INV OUT**: inverted SUM; **± 10 V range**. ³⁴

Classic (Verified):

- SUM and OR bus exist, but **no INV SUM output**. ¹⁰

Channel 1 & 4 function generator model

This section is split into **Verified** behavior (explicit in manuals) and **Inferred** DSP-ready modeling choices.

Verified behavioral rules

Dual role: direct-coupled processing + function generation

- Signal inputs are **direct coupled** (audio + CV capable) and used for lag/portamento/ASR processing. ³⁵
- With **Trigger Input**, the channel produces a transient function described as: rises **0→10 V**, then **immediately falls 10→0**, with **NO SUSTAIN**. ³⁶

Attack/decay and retriggers

- **Rise** determines the time to travel upward to the maximum voltage; CW increases time. ⁴
- **Fall** determines the time to travel downward to minimum voltage; CW increases time. ⁴
- Triggered transient is **re-triggerable during the Falling portion but not during the Rising portion**, enabling clock/gate division by making Rise long enough to ignore incoming triggers. ³⁷
- Trigger input is explicitly called useful for “LFO Reset (only during Falling portion).” ³⁸

Rise/Fall CV vs Both CV

- **Rise CV** and **Fall CV** are explicitly “linear” control inputs; **positive increases time** (slower), negative decreases time (faster), relative to knob setting. ⁴
- **Both CV** is explicitly “bipolar exponential”; **positive decreases total time** (faster), negative increases total time (slower). ⁴
- The manual explicitly reiterates: Both CV changes the rate of entire function and is **inverse** in polarity to Rise/Fall CV inputs. ³⁹

Curve family

- Vari-Response shapes Rise/Fall behavior continuously; manual defines:
- LOG: rate of change decreases as voltage increases
- EXPO: rate of change increases as voltage increases
- LINEAR: constant rate and includes “everything in-between.” ⁴⁰
- The manual explicitly states response curve adjustment **affects Rise and Fall Times**. ⁴¹

Cycle behavior

- In revision 2013, Cycle can be engaged by **Cycle button** or **Cycle Input**; both “do the same thing” (self-oscillate). ⁴²
- Cycle Input is explicitly gate-qualified, with **HIGH \geq +2.5 V**. ⁴³
- Patch example explicitly uses **Cycle Input for Run/Stop control** of pulse/clock behavior. ⁴⁴

EOR/EOC semantics and default states

- **EOR (CH1)**: binary 0/10 V; defaults low when idle; “event” is reaching highest voltage. ²⁶
- **EOC (CH4)**: binary 0/10 V; defaults high when idle; “event” is reaching lowest voltage. ³⁰
- The closely-related Make Noise ⁴⁵ module FUNCTION ⁴⁶ (same “core circuit” family) specifies an **EOR/EOC LED state model consistent with complementary gate states**: “When Red, EOC is High and EOR is Low; when Green, EOR is High and EOC is Low.” ⁴⁷

Time range

- Maths is described as capable of times **as slow as ~25 minutes** and **as fast as ~1 kHz** (audio rate), with mention of adding external control signals for “slow-ver-drive.” ⁴⁸

Inferred DSP model that matches verified constraints

Because no official schematic-level transfer functions were available in the accessible primary sources here, the following is an **inferred** model designed to be *implementation-ready* while matching all documented behaviors.

State machine

Model each outer channel as a **stateful slope generator** with three primary states:

- **IDLE** (resting at low endpoint unless tracking Signal In)
- **RISING**
- **FALLING**

and a boolean latch `cycleEnabled`.

Verified constraints the model must satisfy: - Trigger causes a transient 0→10→0 with no sustain. ³⁵

- Retrigger is ignored during RISING but accepted during FALLING. ⁴⁹

- Cycle input/button engages self-oscillation; Cycle Input is gate-controlled with threshold +2.5 V. ²⁵

- EOR/EOC are 0/10 “event” signals with the described defaults. ⁵⁰

Inferred pseudocode (DSP-ready)

(Use this as a baseline, then refine constants by measurement.)

```
constants:
  V_LOW  = 0.0
  V_HIGH = 10.0          // manual's triggered transient peak
  CYCLE_HIGH_THRESH = 2.5 // V, per manual

state per voice:
  float y                // internal slope output (pre-attenuverter)
  enum Phase {IDLE, RISING, FALLING}
  bool  transientMode    // true if Trigger In initiated the segment pair
  float riseTimeSec, fallTimeSec // effective after CV + response coupling
```

```

    float shapeParam          // normalized vari-response (-1 log .. 0 lin .. +1
hyper-exp)
    bool cycleLatchedButton // UI param
    bool cycleGateHigh       // from Cycle input >= thresh
    bool trigEdge             // rising edge detected at Trigger input
    float xSignalIn           // signal input (direct-coupled)

inputs->mode selection (inferred):
    cycleEnabled = cycleLatchedButton OR (cycleGateHigh)

Trigger handling:
    if trigEdge:
        if Phase != RISING:          // verified: ignore during RISING
            transientMode = true
            if Phase == IDLE: y = V_LOW           // start from 0 when idle
            // if Phase == FALLING, keep current y to avoid discontinuity (inferred)
            Phase = RISING

Cycle start:
    if Phase == IDLE and cycleEnabled:
        transientMode = true
        y = V_LOW
        Phase = RISING

Segment targets:
    if transientMode:
        targetRise = V_HIGH
        targetFall = V_LOW
    else:
        // slew/track mode:
        targetRise = xSignalIn
        targetFall = xSignalIn

Per-sample update:
    if Phase == RISING:
        y += stepToward(targetRise, riseTimeSec, shapeParam)
        if y >= targetRise - eps:
            y = targetRise
            Phase = FALLING
    else if Phase == FALLING:
        y += stepToward(targetFall, fallTimeSec, shapeParam)
        if y <= targetFall + eps:
            y = targetFall
            if transientMode and cycleEnabled:
                Phase = RISING          // self-cycle
            else:
                Phase = IDLE
                transientMode = false

```

EOR/EOC outputs (inferred from manuals + FUNCTION behavior):

```
EOR = (Phase == FALLING) ? 10V : 0V      // matches EOR default low, goes
high at end of rise
EOC = (Phase != FALLING) ? 10V : 0V      // matches EOC default high, goes
low during fall
```

Shape function `stepToward()` and the Vari-Response “time coupling”

Verified requirements: - LOG means slope decreases as voltage increases; EXPO means slope increases as voltage increases. ⁵¹

- Changing response curve affects Rise/Fall times. ⁵²

Inferred implementation approach: 1. Compute a **base time** from Rise/Fall knobs + CV.
2. Apply a **shape-dependent time scale** (because the manual says shape affects time).
3. Apply a **shape-dependent curvature** within the segment.

A practical, stable formulation:

- Let `u` be normalized progress through the segment (0 at start, 1 at target).
- Use `curve(u, p)` where:
- `p < 1` produces fast-start/slow-end (manual's LOG direction for rising)
- `p = 1` linear
- `p > 1` slow-start/fast-end (manual's EXPO direction)

Then: - `p = map(shapeParam)` where `shapeParam` maps the knob (LOG..LIN..EXPO..HYPER-EXPO) into a power/exponent range.

Versioning note (Verified + Inferred): - Revision 2013 explicitly adds “greater logarithmic range” and extends response to “Hyper-Exponential.” ⁵³

- **Inferred consequence:** v2 should allow a wider exponent range (especially on the LOG side for “portamento” feel) than Classic.

Rise/Fall vs Both CV interaction

Verified facts: - Rise/Fall CV are linear; Both CV is exponential; Both polarity speeds up on positive. ⁴

Inferred combined timing model (recommended for emulation): - Treat `Rise` and `Fall` knobs as defining a base time constant (logarithmic knob law likely; see measurement plan).

- Treat RiseCV and FallCV as **linear offsets in a “time control” domain** (not necessarily linear in seconds).

- Treat BothCV as **multiplicative scaling of the rate** (exponential in time), i.e. `time *= exp(-k * bothCV)` with `k` to be measured.

Slew limiter mode behavior (Signal Input)

Verified: - MATHS cannot “increase the rate” of an external voltage; it can only slow it down or allow it through. ⁵⁴

- A gate applied to Signal Input gives an ASR envelope: rise to gate level, sustain until gate ends, then fall to 0. ⁵⁵

Inferred (implementation): - In Signal Input mode, treat the channel as an **asymmetric slew limiter**: it tracks the input with a limited dV/dt on rises (Rise time) and on falls (Fall time), and applies the same curvature family as in triggered mode (because the manual says Rise/Fall/Vari-Response shape responses to signals applied to Signal Input). ⁵⁴

- Ensure stability for audio-rate Signal In: for high-frequency inputs with fast Rise/Fall, you may need **oversampling** if you nonlinearly shape or hard-clip.

Channels 2 & 3 attenuverting / mixing

Attenuverter behavior and offsets

Verified (Classic): - Channels 2 and 3 generate offsets when unpatched because their inputs are normalized to +5 V. ¹⁸

- The Scaling/Inversion control is explicitly defined: **NOON = zero**, full CW maximum, full CCW maximum inverted; CH2/CH3 have “a small amount of gain.” ¹⁸

Verified (Revision 2013): - CH2 input is normalized to **+10 V** reference; CH3 to **+5 V** reference (offset generation). ⁵⁶

- CH2/CH3 attenuverters provide scaling/attenuation/inversion and are described as also providing “amplification.” ³³

Inferred (implementation details): - Implement CH2/CH3 attenuverters as a bipolar gain control around 0 at noon, with at least $\text{gain} \in [-1, +1]$.

- Provide an optional “gain > 1” headroom (e.g. up to 1.2–2.0) *only if measurement confirms it materially affects feel*; Classic explicitly hints at “small amount of gain” on CH2/3. ¹⁸

- Offset when unpatched: - v1 Classic: $x2_norm = +5\text{ V}$, $x3_norm = +5\text{ V}$. ¹⁸

- v2 2013: $x2_norm = +10\text{ V}$, $x3_norm = +5\text{ V}$. ³³

SUM and OR mixing behavior

Verified: - **SUM** is the analog sum of the four attenuverted (variable) channel outputs. ³⁴

- **OR** “always outputs the highest voltage” among its inputs, and “does not respond to negative voltages.” ⁵⁷

- Practical patch notes reinforce OR’s rectifying behavior: half-wave rectification uses OR output. ⁵⁸

Inferred (DSP mapping): - Let $v1..v4$ be Variable Out signals *only if* their corresponding output jack is **unpatched** (break-normal). Otherwise, that channel contributes **0** to busses. ⁵⁷

- **OR output:** $or = \max(0, v1, v2, v3, v4)$; clamp to $[0, 10]$. ³⁴

- **SUM output:** `sum = v1 + v2 + v3 + v4`; clamp/saturate around ± 10 V. 15
- **INV output (v2 only):** `inv = -sum`. 34

Diode-drop / non-idealities

Verified: The manuals describe idealized behavior (max selector, sum, no negative response) without specifying diode drops or precision rectification. 57

Inferred:

- Real analog OR circuits may exhibit diode drops unless compensated; whether Maths does is unknown without schematic/measurement. This is likely perceptually minor for control voltages but might affect precise rectification.
- Recommendation: emulate as **ideal max/rectifier** by default, and optionally include a tiny “drop” mode (e.g. 0.1–0.3 V) toggled via a “hardware non-ideality” setting only if measurement demonstrates it. (See measurement plan.)

Comparator/logic specifics

Gate/trigger thresholds

Verified (Cycle Input threshold): - Cycle Input requires **minimum +2.5 V** for HIGH. 43

Verified (Logic outputs are 0/10 V): - EOR/EOC are explicitly 0/10 V. 59

Inferred (Trigger input threshold and hysteresis): - The manuals do **not** specify Trigger Input threshold, hysteresis, or minimum pulse width. 1

- A safe Rack implementation can adopt a Schmitt trigger detector (see Rack guidelines below), but if you aim for hardware parity you must measure actual thresholds.

EOR/EOC “pulse width” interpretation

Verified textual evidence: - EOR/EOC are described as “event signals” that go high at the end of rise / end of fall and have defined idle defaults. 50

- Patch guides treat EOR/EOC as **pulse/clock sources with variable width**, and explicitly say: - Using EOR: Rise more effectively adjusts frequency, Fall adjusts pulse width
- Using EOC: Rise more effectively adjusts width, Fall adjusts frequency 60
- The related FUNCTION module states EOR/EOC LED indicates mutually exclusive high states, which is consistent with phase gates rather than narrow pulses. 47

Inferred conclusion (strong): - Emulate EOR/EOC as **gate outputs indicating phase** (EOR high during FALLING, EOC high during RISING/IDLE), not short trigger pulses—unless measurement contradicts.

“Comparator / gate extractor” edge cases

Verified (Patch behavior indicating internal timing logic): - The manual’s “Comparator/Gate Extractor (A New Take)” patch indicates: - With CH1 Rise/Fall at 0 (fast), **EOR trips when the signal goes slightly**

positive.

- CH1 Fall lengthens the derived gates; CH1 Rise sets the required time above threshold to trip. ⁶¹

Inferred implication: - This behavior is consistent with a slope-based comparator: a threshold crossing causes the slope generator to switch phase and thus toggle EOR gating, effectively turning analog threshold crossings into gates with controllable delay/width.

Calibration, tolerances, and “feel”

What is known from primary sources

Verified: - Revision 2013 lists several circuit/layout evolutions, including: - Signal Output Multiple replaced by Unity Signal Output

- Added INV SUM output
- Added LEDs for SUM and EOR/EOC state
- EOC output buffered for stability
- Added Cycle Input (gate-controlled cycling)
- Added offset ranges (± 10 on CH2 or ± 5 on CH3)
- Greater LOG range in Vari-Response for “East Coast style Portamento” ⁵³

What is not specified but is likely “feel-critical”

Inferred (requires measurement to perfect): - **Rise/Fall knob law:** almost certainly nonlinear (huge time span), likely closer to exponential/log taper than linear seconds-per-degree. ⁵¹

- **Both CV scaling:** specified as exponential but not quantified (e.g., V/oct or V/decade). ²³

- **Response curve mapping:** the exact transfer from knob position to curve exponent and the “time coupling” factor is undocumented. ⁵²

- **Saturation/clipping points:** outputs are specified as ranges (± 10 , 0–10), but actual analog headroom/soft clipping shapes are undocumented. ¹⁵

- Patch “Pseudo-VCA with clipping” explicitly relies on audible clipping behavior when adding offsets and mixing; that suggests non-ideal saturation matters to some patches. ⁶¹

Emulate vs ignore: recommendations

Recommended to emulate (high perceptual impact): - Retrigger-only-on-fall rule. ⁴⁹

- Cycle Input gate threshold (+2.5 V) and run/stop behavior. ⁶²
- OR = max over positives only with 0 baseline. ³⁴
- Vari-Response curve family and the fact that it changes timing. ⁵²

Reasonable to approximate unless doing “gold standard” emulation: - Exact diode drop in OR.

- Precise analog saturation curvature at ± 10 V.
- Exact hysteresis thresholds of trigger comparators (except Cycle Input which is specified). ⁴³

Experimental measurement plan

If you want the emulation to feel indistinguishable from hardware, measure these items on at least one unit per revision (Classic + 2013).

Gear and capture requirements

- **DC-coupled audio interface** (or oscilloscope with DC coupling).
- Sample at ≥ 96 kHz for accurate edge timing and audio-rate cycling behavior; higher (192 kHz) if you want tight 1 kHz+ measurements.
- Record at **24-bit** to keep quantization noise out of very slow slopes.
- For CV sweeps, a calibrated CV source (or a precision DAC module) is ideal.

Test patches and what to record

Function generator transient (Trigger In mode)

- Patch: nothing to Signal In; feed a single trigger into Trigger In; monitor Unity Out and Variable Out.
- Measure: - Peak amplitude at each output (is Unity 8 V or 10 V in this mode?). ²⁵
- Rise time and fall time vs knob positions (multiple points). ³⁹
- Retrigger behavior: fire a second trigger at multiple phases (early rise, late rise, early fall, late fall). Confirm ignore vs restart semantics. ⁶³

Cycle mode amplitude and frequency mapping

- Patch: engage cycle; monitor Unity Out, Variable Out, EOR/EOC.
- Measure: - Unity Out amplitude (should be 0–8 V when cycling per manual) and whether Variable Out differs. ⁴³
- Frequency vs Rise/Fall settings at multiple curve settings. ⁶⁴
- Effect of Vari-Response on total period (quantify “time coupling”). ⁵²

Cycle Input run/stop edge cases (v2)

- Patch: keep Cycle button off; apply gate to Cycle Input; then remove gate mid-cycle.
- Measure: - Whether oscillation stops immediately or at end-of-cycle (phase). ⁶²
- Minimum voltage for reliable HIGH detection (should be $\sim +2.5$ V). ⁶⁵

EOR/EOC width and phase

- Patch: cycle at slow rate; monitor EOR/EOC with scope.
- Measure: - Whether EOR/EOC are “phase gates” (high for the duration of a segment) or narrow pulses. ⁶⁶
- Decide which internal state generates them (RISING vs FALLING) and confirm idle defaults. ⁵⁰

OR and SUM non-idealities

- Patch: feed known DC levels into CH2/CH3 and known LFOs into CH1/CH4; compare OR output to a software max.
- Measure: - Any voltage drop from OR relative to max input. ³⁴
- SUM saturation onset and whether clipping is hard or soft. ⁶⁷

Table template for results

Use a table per revision and per output point.

Test	Knob/CV settings	Output monitored	Metric	Measured value	Notes
Trigger transient	Rise=0.25, Fall=0.25, Shape=LIN	Unity Out	Peak (V)		
Trigger transient	same	Variable Out	Peak (V)		
Cycle	Rise=0.5, Fall=0.5, Shape=LOG	Unity Out	Period (s)		
Cycle	same	EOR	Duty %		
OR	v1=3V, v2=-2V, v3=5V, v4=4V	OR Out	Output (V)		
SUM	same	SUM Out	Output (V)		

For curve fitting: - Fit Rise/Fall knob to time using a **log mapping** first (e.g. $T = T_{min} * (T_{max}/T_{min})^k$) and refine by least squares. - Fit Both CV scaling as an **exponential in time** ($T \propto \exp(-a*V)$) unless measurements show otherwise (e.g., piecewise).

Implementation notes for VCV Rack 2

Rack voltage/trigger conventions (software-side)

VCV's own manual recommends:

- **Gates:** 10 V when active.
- **Triggers:** 10 V, duration about 1 ms; detect using Schmitt trigger with low threshold ~0.1 V and high threshold ~1–2 V (example code uses `dsp:SchmittTrigger`). ⁶⁸
- **Polyphony:** cables can carry up to 16 channels; modules should process each channel independently. ⁶⁹

These standards are about Rack ecosystem compatibility and are not necessarily identical to Maths hardware thresholds—so if you emulate hardware thresholds, still consider providing Rack-friendly behavior for user experience.

DSP architecture recommendation

Implement Maths as **four per-voice channel blocks + shared bus block**:

- **CH1 block (polyphonic):**

- Slope generator state machine + CV parameter evaluation (Rise/Fall/Both/Shape)
- Outputs: UnityOut1, VarOut1, EOR
- **CH4 block (polyphonic):**
- Same + EOC
- **CH2/CH3 blocks (polyphonic):**
- Input normalization (offset) + attenuverter gain
- Outputs: VarOut2/VarOut3
- **Bus block (polyphonic):**
- For each poly channel index `c`, compute `busIn[i]` as VarOut[i] only if that output jack is **unpatched** (normalised). ⁵⁷
- Compute OR/SUM/INV.

Parameter smoothing / time discretization

- Rack's engine already includes smoothing behavior for parameters in some contexts, but you should not rely on it for "analog feel" because envelope timing is extremely sensitive. ⁷⁰
- Recommended:
- Smooth raw knob/CV-derived **time constants** (Rise/Fall/Both scaling) with a short one-pole (e.g., 1–5 ms) to avoid zipper noise at audio-rate cycling.
- Do **not** smooth Trigger edges; detect edges with Schmitt trigger and feed a one-sample event into your state machine.

Oversampling

- If your implementation includes **nonlinear saturation** (soft clipping) or a **nonlinear curve that depends on signal amplitude**, audio-rate cycling can alias.
- Recommendation: offer optional **2× oversampling** for CH1/CH4 signal generation paths only (not needed for slow CV use), controlled by a "HQ" switch.

Polyphony strategy

- Treat polyphonic cables as **independent copies of the entire Maths circuit per channel index** (including each channel's internal state). This matches Rack's polyphony model expectations. ⁶⁹
- For performance, it's acceptable to keep CH2/CH3 purely per-channel scalar math, but CH1/CH4 require per-channel state arrays.

Deterministic unit tests

Where the manuals specify exact behavior, implement unit tests with known expected outcomes.

Verified test vectors (examples):

- **Cycle gate threshold (v2):**
- Input Cycle = +2.4 V → cycle should remain off
- Input Cycle = +2.5 V → cycle should be on ⁴³
- **Retrigger rule:**

- When in RISING, Trigger edges must not restart (no phase reset). ⁶³
- When in FALLING, Trigger edge must retrigger (restart rising phase) (exact “reset to 0 vs restart from current level” requires measurement). ⁶³

- **OR behavior:**

- Inputs: [-2, 3, 5, 4] → OR = 5
- Inputs all negative → OR = 0 ³⁴

- **Bus normalization:**

- If VarOut2 jack is patched, CH2 contribution must be removed from SUM and OR busses. ⁵⁷

Measurement-dependent (“gold standard”) tests: - Exact Rise/Fall knob→time mapping curve. - Exact Both CV scale (volts-per-octave or similar). - Exact EOR/EOC “gate vs pulse” timing and duty cycle.

Known Unknowns

These are the remaining blockers to a hardware-indistinguishable implementation:

- Exact **Rise/Fall knob law** (seconds vs knob position) and whether it is the same in Classic vs 2013 revision. ⁷¹
- Exact **Both CV exponential scaling** (units, slope) and how it composes with Rise/Fall CV. ²³
- Whether Trigger retrigger during FALL **resets to 0** or **restarts from current voltage** (manual only specifies phase allowance, not restart semantics). ⁶³
- Whether Unity outputs are **0–8 V only in cycle** or also in triggered mode, and how that relates to the manual’s “0–10 V function” language. ⁷²
- Whether OR output exhibits any diode drop / compression, and where SUM clips (hard vs soft). ⁷³
- Trigger input threshold, hysteresis, and minimum trigger width (undocumented in manuals). ⁷⁴
- Any hardware calibration/trimmers affecting timing (not mentioned in accessible primary sources). ⁷⁵

Bibliography

Primary sources (Make Noise):

- Maths revision 2013 manual (PDF). ⁷⁶
- Maths Classic manual (PDF). ¹⁷
- FUNCTION manual (PDF), for corroborating EOR/EOC semantics and “same core circuit family” behavior. ⁷⁷
- Maths product page (time range statement is repeated there). ⁷⁸

VCV Rack developer references:

- VCV Rack Manual: Voltage Standards (triggers/gates thresholds and suggested Schmitt trigger usage). ⁶⁸

- VCV Rack Manual: Polyphony. ⁷⁹
- VCV Rack Manual: Plugin API Guide (polyphony implementation guidance). ⁸⁰

Secondary/community reference used sparingly (version UX note):

- Sequencer.de forum thread noting hardware cycle switch persistence difference between older and newer Maths (non-primary, UX-only). ¹²

1 2 3 4 5 7 8 9 11 13 14 15 16 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
36 37 38 39 40 41 42 43 44 45 46 48 49 50 51 52 53 54 56 57 58 59 60 61 62 63 64 65 66 67

71 72 73 74 75 76 <https://www.makenoisemusic.com/wp-content/uploads/2024/03/>

MATHSmanual2013.pdf

<https://www.makenoisemusic.com/wp-content/uploads/2024/03/MATHSmanual2013.pdf>

6 10 17 18 55 <https://www.makenoisemusic.com/wp-content/uploads/2024/03/>

MATHSCLASSICmanual.pdf

<https://www.makenoisemusic.com/wp-content/uploads/2024/03/MATHSCLASSICmanual.pdf>

¹² <https://www.sequencer.de/synthesizer/threads/make-noise-maths.74948/>

<https://www.sequencer.de/synthesizer/threads/make-noise-maths.74948/>

47 77 <https://www.makenoisemusic.com/wp-content/uploads/2024/03/FUNCTIONmanual.pdf>

<https://www.makenoisemusic.com/wp-content/uploads/2024/03/FUNCTIONmanual.pdf>

⁶⁸ <https://vcvrack.com/manual/VoltageStandards>

<https://vcvrack.com/manual/VoltageStandards>

⁶⁹ ⁷⁹ <https://vcvrack.com/manual/Polyphony>

<https://vcvrack.com/manual/Polyphony>

⁷⁰ <https://community.vcvrack.com/t/parameterquality-change-in-2-3-0/19533>

<https://community.vcvrack.com/t/parameterquality-change-in-2-3-0/19533>

⁷⁸ MATHS – Make Noise Music

<https://www.makenoisemusic.com/modules/maths/>

⁸⁰ <https://vcvrack.com/manual/PluginGuide>

<https://vcvrack.com/manual/PluginGuide>