WMD — Skorpion

- Manual PDF

Skorpion Manual PDF (WMD)

Using WMD Skorpion for densely rhythmic, hyper-complex percussion

Skorpion is not just a wavefolder. Based on the manual, it is a **comparator-driven waveform reanimation core** with: - 8 threshold crossings - target sequencing - internal macro modulation - many derived outputs (`COUNT` , `DAC` , `DIFF` , `TRGTs` , `G(IN>0)` , `±G(DIR)` , `DELAY` , `ABS(IN)`)

That makes it unusually good for **creating percussion logic, event extraction, pseudo-sequencing, and self-modulating rhythmic timbre changes**. If your goal is dense rhythmic music with polyrhythms and complicated patterns, Skorpion can act as:

1. **A percussion voice**
2. **A rhythm extractor from incoming audio/CV**
3. **A modulation generator for other percussion modules**
4. **A self-patched rhythm machine**
5. **A stereo transient shaper for complex drum motion**

Core idea

The most important thing to understand is this:

- The input at `IN` is analyzed by **8 comparators** set by the slider thresholds.

- Each threshold crossing changes the behavior of the **vector core**.
- The number/order of active thresholds can generate stepped and logic-like outputs:
- `COUNT` : staircase, 0–4V, +0.5V per active threshold
- `DAC` : weighted threshold state output
- `TRGTs` : target sequencer output
- `±G(DIR)` : up/down gate
- `G(IN>0)` : polarity gate
- `DIFF` : difference between target and current state, rich in transients/harmonics

For percussion, this means Skorpion can transform a single clocked source, oscillator, envelope, drum loop, or modulation waveform into **many correlated but non-identical rhythmic streams**.

Best ways to use Skorpion for complex percussion

1. Feed it rhythms, not just tones

Most people will patch an oscillator into a wavefolder. For your goals, also try feeding Skorpion:

- trigger trains
- gates
- stepped CV
- envelope shapes
- drum loops
- bursts
- ratchets
- clock-divided square waves
- mixed clocks from several channels
- tom/kick/snare audio
- noise bursts shaped by VCAs

Because thresholds create events whenever the input crosses them, **complex incoming modulation becomes rhythmically decoded** into different outputs.

Excellent input sources

- a swung clock square wave
- a polygonal LFO
- an envelope cycling at a non-integer relation to the main tempo
- mixed trigger/gate patterns run through a slew
- a percussion bus
- a sequence of accent envelopes
- an audio oscillator FM'd by clock divisions

This gives you **rhythmic density from threshold crossings** rather than just from trigger programming.

---

2. Use the sliders as a rhythmic event map

The 8 sliders define where folds/events occur. For percussion use, think of them less as "tone settings" and more as:

- **timing density zones**
- **accent decision points**
- **probability-like contour markers**
- **subdivision extractors**

Practical approach

Set thresholds unevenly: - cluster 3 sliders near center - spread 2 far apart - keep a couple near extremes

This creates uneven crossing behavior, so one input waveform yields **nonuniform event spacing**. That's ideal for: - additive-feeling rhythms - quasi-Euclidean accents - asymmetrical percussion phrases - "7 over 5 over 4" style movement

Equalized thresholds vs uneven thresholds

The `EQUALIZE THLDs` switch is crucial:

- **ON**: equal threshold spacing, more classic wavefolder, more regular rhythmic subdivisions
- **OFF/JACK-controlled**: manually uneven thresholds, more asymmetrical and human/alien rhythmic structures

For your use case: - use **equalized thresholds** when you want machine-like subdivisions - use **uneven slider spacing** when you want weird metric phrasing and polymetric irregularity

---

3. Use `TARGET` and `TRGTs` as an 8-step micro-rhythm sequencer

The manual states the `TRGTs` form an **8-step voltage controlled sequencer**. This is huge for percussion.

Use case

Set `TARGET` to `SLIDERs` and define 8 target voltages while holding the spring toggle left. Now each threshold crossing selects a target.

With `TARGET ORDER` : - **SEQ**: target chosen by count of active thresholds - **TIED**: target chosen by most recently crossed threshold

These two modes can radically change rhythmic behavior.

Why this matters rhythmically

Each threshold crossing doesn't just fold the waveform — it can also shift the vector core toward a different destination voltage. This creates: - segmented transient profiles - repeating-but-not-repeating accent contours - ordered or last-event-dependent phrases

Percussion strategy

Use `TRGTs` for: - per-hit brightness variation - pseudo-accent sequencing - changing the shape of each fold segment - control output to modulate another module's decay, pitch, or filter cutoff

You can patch the `TRGTs` output to: - LPG CV - VCA CV - filter FM - decay CV on a drum voice - sample select on a sampler - clocked switch address

That gives you **threshold-derived sequencing**, which is excellent for dense percussion ecosystems.

---

4. Use `COUNT` and `DAC` outputs as rhythm translators

These are especially useful for complicated patterns.

`COUNT`

- staircase from 0–4V
- each active threshold adds 0.5V

This gives you a crude but very musical **activity meter**.

Great uses

Patch `COUNT` to: - a comparator to create new trigger streams - a quantizer then oscillator for tuned percussion - a VCA controlling noise amount per hit - decay CV on hats/snare - probability/logic thresholding module

If your input is moving irregularly across thresholds, `COUNT` becomes a **meta-rhythm CV** representing pattern density.

`DAC`

Weighted threshold-state output, more subtle than `COUNT`.

This is better when you want: - finer accent nuances - pseudo-melody tied to rhythm - less staircase bluntness - more unstable but controlled modulation

Rhythmic use

Patch `DAC` to: - kick pitch for non-repeating body tones - snare noise color - FM index on metallic percussion - clocked switch crossfade amount - a comparator to derive irregular derived gates

`DAC` often works better than `COUNT` when you want **microvariation inside fast percussion runs**.

---

5. Use `DIFF` as a transient percussion source

The manual says `DIFF` is the difference between target voltage and the vector core, and it slopes toward 0V. It is usually very high in harmonic content.

That description screams: - clicks - spikes - snares - zaps - glitch hats - transient layers

Patch ideas

- `DIFF` → VCA/LPG → mixer as a click layer
- `DIFF` → filter → short envelope/VCA for hats
- `DIFF` → wavefolder/distortion → metallic percussion
- `DIFF` → resonator for tuned impulse percussion
- `DIFF` → comparator → trigger extractor

For dense rhythmic music, `DIFF` can be the “micro-hit generator” while `OUTL/OUTR` supply body and motion.

Tip

Set `SHAPE` to `DIFF` source for even harsher, spikier timbres.

---

6. Exploit `SYNC` to lock complexity to musical timing

`SYNC` resets the vector core at zero crossings of the input: - `SOFT` : ramps toward 0V at current slope - `X` : no sync - `HARD` : fast reset to 0V

For percussion, `SYNC` is how you choose between: - tightly repeatable attacks - semi-chaotic evolving hits - glitched retrigger behavior

Recommended uses

- **HARD sync** for precise, repeatable transient-heavy drum patterns
- **SOFT sync** for more elastic, loping, less robotic rhythms
- **No sync** for drifting polymetric behavior and unstable phrase lengths

If using clock-related inputs, `HARD` often makes Skorpion act more like a controllable percussion synthesizer. If using long LFOs or mixed audio-rate modulation, disabling sync often creates **long-form polymetric drift**.

---

7. Use `SHIFT` for asymmetry and phrase rotation

`SHIFT` pushes the input up/down against the comparators and creates asymmetry. The manual even notes slow modulation can produce a frequency-shift effect.

For rhythmic work, `SHIFT` is one of your best controls because it changes **which thresholds are crossed and when**.

That means modulating `SHIFT` can: - reorder accent structures - create left/right phrase imbalance - alter rhythmic density without changing clock speed - make repeated loops mutate over time

Patch concept

Use a slow odd-cycle modulation source to `SHIFT`: - 5-step sequencer against a 4-beat loop - 7-beat envelope cycle - clock-divided triangle not synced to bar length

This creates **phrase rotation**, where the same input material yields different threshold activations over time.

For hyper-complex percussion, this is gold.

---

8. Use `SHAPE` as a timbral rhythm multiplier

`SHAPE` modulates slope using feedback sources: - `IN` - `OUT` - `DELAY` - `COUNT` - `DIFF` - `TRGTs` - `DAC` - `DIR`

This is one of the most powerful sections for complex percussion.

Best rhythmic `SHAPE` sources

`COUNT`

Makes timbre depend on how many thresholds are active. - More activity = more harmonic or contour change - Great for density-sensitive percussion

`TRGTs`

Each segment gets its own shape behavior. - Excellent for sequenced hit articulation - Makes each threshold region behave like a different drum articulation

`DIFF`

Aggressive and spiky. - Best for glitch percussion, digital-sounding hats, tearing snares

`DIR`

Uses direction signal. - Creates skewed up/down behavior - Good for alternating hit character, almost like built-in call/response

`OUT`

Feedback from output. - More organic/log/exp behavior - Good for toms, wooden percussion, body-rich hits

`DELAY`

Only active when OUTPUT is between noon and fully clockwise. - Gives moving stereo/phase/tap-like articulation - Great for swarming percussion fields

---

9. Use `HALT` for ratcheting, stutters, and broken-grid timing

`HALT` stops the vector core at its current voltage. Audio-rate modulatable.

This is one of the most interesting performance/rhythm inputs.

What it can do

- freeze a transient mid-motion
- create gated terraces in the waveform
- produce abrupt chokes
- impose externally clocked interruptions onto the fold motion

Rhythmic uses

Patch a trigger/gate pattern into `HALT`: - short bursts produce chopped transients - irregular gate lengths create stop/start drum contours - Euclidean trigger streams create punctuated rhythmic lockups - a high-rate square clock gives granular tearing textures

`HALT IF TARG=0`

When enabled, any target set to zero can halt movement for just that segment.

This is perfect for: - inserting rests into timbral motion - creating pseudo-square segments - turning parts of the cycle into “dead air” or held plateaus - building syncopation into the fold structure itself

For percussion, set some `TRGTs` fully down and enable `HALT IF TARG=0`. Now some threshold events become **frozen nodes**, like rests or held accents inside the waveform.

---

10. Use `DRY IF NO THLDs` to preserve groove under heavy modulation

If no thresholds are active, this forces dry signal output and the vector core tries to follow `IN`.

This matters when heavily modulating `FOLD` or `SHIFT`. Without it, your patch may occasionally drop into low-activity or silent zones. With it: - the groove keeps speaking - sparse moments remain useful - modulation can be extreme without losing the pulse

For live rhythmic work, this is often worth enabling.

Specific patch strategies for complex rhythmic music

Patch 1: Polyrhythmic percussion synthesizer voice

Goal

One Skorpion voice producing evolving drum-like patterns with internal asymmetry.

Patch

- Clocked envelope or triangle VCO → `IN`
- Set thresholds unevenly
- `TARGET` → `SLIDERs`
- Program `TRGTs` with alternating high/low voltages
- `TARGET ORDER` → `TIED`
- `SHAPE SOURCE` → `COUNT` or `TRGTs`
- `SYNC` → `HARD`
- `DIFF` → external VCA/filter as click layer
- `OUTL/OUTR` → mixer

Result

Each threshold crossing creates a slightly different transient/body relationship. Because thresholds are uneven and target order is tied to most recent crossing, the pattern feels interdependent instead of step-sequenced.

Patch 2: Extract multiple percussion control streams from one source

Goal

Turn one modulation/audio source into many related rhythms.

Patch

- Mixed clocks + slew or a drum bus → `IN`
- `COUNT` → comparator → trigger for hats
- `DAC` → pitch CV for tom voice
- `G(IN>0)` → trigger/gate for clap envelope
- `±G(DIR)` → switch/select input on sequential switch
- `TRGTs` → decay CV on snare
- `DIFF` → audio click layer
- `DELAY` → modulate stereo VCA or pan

Result

Skorpion becomes a **rhythm decomposition engine**. One source yields several correlated but distinct drum control streams.

This is ideal for polyrhythms because everything shares a common origin but diverges in structure.

Patch 3: Complex time-signature phrase engine

Goal

Create phrase cycles that feel like 5, 7, 9, or mixed meters.

Patch

- Use an LFO/envelope cycling at a musical but not bar-aligned period into `IN`
- Set 8 thresholds in non-even spacing
- Modulate `SHIFT` with a slower sequence of a different cycle length
- `TARGET ORDER`:
- start with `SEQ` for structured phrase counting
- switch to `TIED` for more contingent behavior
- `SHAPE SOURCE` → `DIR` or `DAC`
- `SYNC` off or `SOFT`

Result

You get evolving phrases where the threshold crossing order itself defines the meter feel. Because the `SHIFT` modulator has a different loop length, accents “walk” against the master phrase.

This is a strong route to **apparent odd meters without explicit programming**.

Patch 4: Metallic hat swarm / glitch percussion patch

Goal

Very dense upper percussion with internal variation.

Patch

- Fast oscillator or filtered noise burst train → `IN`
- High `FOLD`
- Medium/high `SLOPE`
- `SHAPE SOURCE` → `DIFF`
- `TARGET` toward `CLIP`

- Patch something other than `IN` into `CLIP`:
- another drum line
- percussion loop
- burst generator
- `OUTPUT` above noon toward `WIDE`
- `OUTPUT SWITCH` → `FILTERS`
- `DELAY` output to separate mixer channel

Result

Tight, metallic, shattered hats and stereo percussion spray. The alternate `CLIP` signal makes the fold target contingent on another rhythm source, creating interlocking complexity.

Patch 5: Self-patched pseudo-generative percussion network

Goal

Make Skorpion recursively generate complex rhythms.

Patch

- Oscillator/envelope into `IN`
- `COUNT` → `SHIFT` CV
- `DAC` → `SHAPE` CV
- `TRGTs` → `TARGET` CV or another module controlling input level
- `±G(DIR)` → `HALT`
- `ABS(IN)` → modulate another oscillator/FM source feeding `CLIP`
- `DELAY` → external VCA or filter on return path

Result

The module begins reacting to its own state. This can produce very intricate but coherent rhythmic mutation.

Start with attenuators low. This patch can get wild quickly.

How to make real polyrhythms with Skorpion

Skorpion doesn't directly output clock divisions, but it excels at generating **event structures whose apparent rhythm depends on threshold crossing geometry**.

Method 1: Different cycle lengths in modulation

Use: - one signal into `IN` - a different-period modulation into `SHIFT` - another different-period modulation into `FOLD` or `SHAPE`

Example: - `IN` : envelope looping every 5 beats - `SHIFT` : modulation every 7 beats - `MACRO ENV` : toggled every 4 bars - `TRGTs` : asymmetrical target pattern

Now the crossing patterns phase against each other. This creates practical polyrhythms.

Method 2: Comparator-derived streams

Use: - `G(IN>0)` as one binary rhythm - `±G(DIR)` as another directional rhythm - `COUNT` through external comparator for another rhythm - `DAC` through external comparator/window comparator for another

One input becomes 3–4 derived rhythmic layers.

Method 3: Threshold geometry as meter

Uneven thresholds create uneven event spacing over a periodic input waveform. If the waveform repeats every quarter note, threshold crossings may imply: - 3+2+3 - 2+2+3 - 5+4 - 7 against 4 feel

This is a fantastic way to get **odd-meter sensation from simple periodic input**.

How to make complicated patterns feel intentional

Hyper-complex percussion can become mush unless there are layers of hierarchy. Skorpion can help structure that.

Use `COUNT` for macro accents

Patch `COUNT` to control: - output VCA level - drum bus filter cutoff - send amount to reverb/delay

Then denser threshold states become louder/brighter.

Use `TRGTs` for phrase identity

Program 8 targets with a contour like: - high - low - mid - zero - high - low - zero - mid

This creates recurring phrase landmarks.

Use `HALT IF TARG=0` for rests

Targets at zero can become built-in pauses or squares. This inserts punctuation into dense streams.

Use `SYNC HARD` for downbeat focus

If everything is getting too smeared, hard sync can restore repeatability at each input crossing cycle.

Using the Macro section for rhythmic evolution

Skorpion's internal macro system is very useful for long-form complex music because it provides: - attack/sustain/release macro envelope - LFOs for thresholds - LFO or envelope normals for `FOLD`, `SLOPE`, `SHIFT`, `SHAPE`

Why this matters

You can make a stable complex percussion patch that: - slowly increases threshold motion - gradually changes fold amount - introduces asymmetry over a phrase - brings in motion only during selected gates

Good macro setup for percussion

- Slider 1: medium attack
- Slider 2: long release
- Slider 3: moderate THLD LFO amount
- Slider 4: slow THLD LFO rate
- Sliders 5–8:
- `FOLD` in LFO mode
- `SLOPE` in ENV mode
- `SHIFT` in LFO mode
- `SHAPE` in ENV or LFO mode

Trigger Macro Env manually or from a phrase gate every few bars.

Result

The patch “breathes” over time while still preserving the intricate local rhythm.

Because LFOs reset on each gate-on, the module can also produce **repeatable complex modulation phrases**, which is excellent when you want intricate music that is still performable and recallable.

Best parameter tendencies for percussion

These are not strict rules, but useful starting points.

For kicks / heavy thuds

- lower `SLOPE`
- moderate `FOLD`
- `TARGET` near `5V` or `SLIDERs`
- `SHAPE` from `OUT` or `DIR`
- `SYNC HARD`

For snares / cracks

- moderate/high `SLOPE`
- `SHAPE` from `DIFF`
- `TARGET` between `SLIDERs` and `CLIP`
- use `DIFF` as parallel layer

For metallic hats

- high `SLOPE`
- high `FOLD`
- `SHAPE` from `DIFF`, `COUNT`, or `DELAY`

- `OUTPUT` in `WIDE`
- `FILTERS` switch on

For weird toms / rubber percussion

- modulate `SHIFT`
- `SHAPE` from `OUT`
- `TARGET ORDER` `TIED`
- `SYNC SOFT`

Performance tips for live complex rhythm making

1. Switch `TARGET ORDER` during performance

- `SEQ` = more countable and structured
- `TIED` = more reactive and tangled

This can feel like moving from composed odd meter to unstable improvisational metric interplay.

2. Modulate or manually change `SHIFT`

This is one of the fastest ways to rotate the rhythm without changing tempo.

3. Use `EQUALIZE THLDs` as a regularity toggle

- equalized = grid-like
- uneven = broken grid

Excellent for transitions.

4. Move between dry/wet/wide with `OUTPUT`

- below noon: keep impact and body
- above noon: spread and animate high-frequency content
- use `FILTERS` to keep low end centered

Very useful for dense percussion mixes where the center can get overloaded.

5. Use `CLIP` as an interlock input

Patch another rhythm source into `CLIP` so one line influences the target behavior of another. This creates musical coupling between drum layers.

A complete advanced patch recipe

“Hypermetric Percussion Engine”

Inputs

- Main sequenced envelope or oscillator → `IN`
- Different rhythmic voice or burst source → `CLIP`
- Slow 5-step CV → `SHIFT`
- Faster 7-step CV → `FOLD`
- Phrase gate every 2 or 4 bars → `MACRO ENV`

Skorpion settings

- Uneven thresholds
- `TARGET` toward `SLIDERs`

- `TRGTs` programmed with highs, mids, and zeros
- `HALT IF TARG=0` on
- `TARGET ORDER` = `TIED`
- `SHAPE SOURCE` = `TRGTs` or `COUNT`
- `SYNC` = `SOFT`
- `OUTPUT` around 1–3 o'clock
- `OUTPUT SWITCH` = `FILTERS`

Outputs

- `OUTL/OUTR` = main percussion stereo voice
- `DIFF` = transient layer
- `COUNT` = modulate external hat decay or comparator-derived triggers
- `DAC` = modulate pitch/brightness of another drum voice
- `TRGTs` = control filter/resonator tuning
- `DELAY` = separate stereo splash layer

Result

You get: - internal asymmetrical subdivisions - phrase-level evolution - self-related but non-repeating percussion layers - stereo complexity without losing low-end center - emergent polyrhythm from modulation cycle mismatch

---

Summary: what Skorpion is best at for your goal

If your aim is **densely rhythmic, hyper-complex percussion with polyrhythms and unusual meters**, Skorpion is especially strong when used as:

- a **threshold-based event shaper**
- a **derived rhythm CV generator**
- a **timbral sequencer**

- a **self-modulating transient machine**
- a **stereo percussion animator**

The most powerful features for you are likely:

1. **Uneven THLD sliders** for non-grid crossing patterns
2. **TRGTs sequencer** for segment-by-segment articulation
3. **COUNT / DAC outputs** for derived control streams
4. **DIFF output** for aggressive transient percussion
5. **SHIFT modulation** for phrase rotation and asymmetry
6. **HALT / HALT IF TARG=0** for stutters, rests, and hard punctuation
7. **Macro modulation** for long evolving rhythmic forms
8. **WIDE/DELAY output behavior** for layered stereo percussion motion

If you want, I can also give you: - **10 concrete Skorpion patch recipes** for IDM/glitch/tribal/polymeter - a **system-level patch plan** showing which other Eurorack modules pair best with it - or a **“Skorpion as drum brain” workflow** for building whole tracks around it

Generated With Eurorack Processor