

# Making Sound Machines – DivSkip

---

- [Manual PDF](#)
- 

[Skorpion Manual PDF](#)

## Using WMD Skorpion for densely rhythmic, hyper-complex percussion

---

Skorpion is not just a wavfolder. From the manual, it's really a **threshold-driven waveform reanimation system** with lots of internal logic, segment-based behavior, target sequencing, direction detection, comparator counting, and auxiliary outputs. That makes it unusually good for **percussive complexity**, especially if your goal is:

- dense rhythmic layering
- polyrhythms
- irregular meters
- interlocking accents
- self-modulating drum logic
- “one source becomes many correlated rhythms”

The trick is to stop thinking of Skorpion as only a timbre processor and start using it as a **rhythmic event extractor / event generator / modulation network**.

---

# Why Skorpion is good for percussion complexity

---

Several manual features matter a lot here:

## 1. **8 thresholds (THLDs)**

The incoming signal is analyzed by 8 comparators. Every threshold crossing can change the state of the waveform engine. That means one input can create many internal “events.”

## 2. **TRGTs as an 8-step sequencer**

The sliders can define target voltages, and those targets are selected as thresholds are crossed. This is basically an **event-addressed sequencer** rather than a clocked sequencer.

## 3. **COUNT, DAC, G(IN>0), ±G(DIR), ABS(IN), DIFF outputs**

These outputs turn the internal behavior into reusable control signals. For percussion, that means:

4. extract gates

5. derive stepped accents

6. create pseudo-clocks

7. generate fills and ratchets

8. produce correlated but non-identical modulation streams

## 9. **SYNC / HALT / TARGET modes / SHAPE feedback**

These let you force articulation, choking, asymmetry, hard resets, and dynamic segment shapes—exactly the stuff that makes percussion feel alive and complicated.

## 10. **Macro modulation + threshold LFOs**

Skorpion can slowly or quickly animate thresholds and core controls. That lets one rhythmic topology evolve over time into another without repatching.

---

# Core concept for percussion

---

Feed Skorpion with something that already has clear transient or cyclic structure, then use:

- **OUTL/OUTR** as audio percussion voices
- **AUX outputs** as rhythm generators for other modules
- **TRGTs / THLDs / SHAPE / HALT** to sculpt micro-variation

Best source signals into **IN** for percussive work:

- sine or triangle VCO for synthetic tom / kick / click material
- pulse or square for aggressive digital percussion
- a drum loop or noisy transient source
- another sequenced voice
- clocked modulation audio-rate oscillator
- burst generator / noise through VCA / LPG

Because Skorpion responds to crossings and slope behavior, even a simple oscillator can become a highly articulated rhythmic object.

---

## Best strategies for hyper-complex percussion

---

### 1. Use one steady oscillator as a “rhythmic substrate”

---

Patch: - VCO sine/triangle → **IN** - Skorpion **OUTL/OUTR** → mixer - Start with: - **FOLD** low to medium - **SLOPE** medium-high - **TARGET** at **SLIDERS** or between **SLIDERS** and **5V** - **SYNC** = **HARD** or **SOFT** - **EQUALIZE THLDs** off initially

Why this works: - The oscillator provides stable zero crossings. - Threshold crossings create multiple fold events per cycle. - This turns a steady source into a clustered, articulated, almost sequenced percussion tone.

## For denser attacks

- Raise **FOLD**
- Increase **SLOPE**
- Use **SHAPE** with **DIFF** or **OUT**
- Raise some thresholds close together and others far apart

This makes the waveform generate nonuniform fold segments that sound like: - flam-like attacks - synthetic snares - metallic hats - syncopated edge transients

---

## 2. Think of THLD sliders as a rhythm map

---

The threshold sliders are not just timbre controls. They define **where events happen** in relation to the input waveform.

### Good rhythmic threshold layouts

Instead of evenly spacing all thresholds, try:

- **Clustered low + sparse high**
  - dense activity early in the waveform
  - fewer events later
- gives attack-heavy percussive behavior
- **Alternating gaps**
  - short / long / short / long spacing
  - creates built-in rhythmic asymmetry
- **Near-symmetry with one displaced threshold**
  - sounds like a repeating groove with one “wrong” accent
  - excellent for odd-meter feel
- **Two groups of 3 + 5**
  - creates a natural 3-against-5 character

Because thresholds are crossed according to the incoming waveform plus FOLD and SHIFT, these slider positions create **embedded rhythmic subdivisions**.

---

### 3. Use **TARGET = SLIDERS** for event-addressed percussion sequencing

---

This is one of the strongest features for your goal.

Set: - **TARGET** toward **SLIDERS** - Hold spring toggle left to edit **TRGTs** - Create 8 different target voltages

Now every threshold crossing can select a new destination voltage for the vector core.

This means each segment of the waveform can behave like a different drum articulation: - one target = click - another = hollow tom - another = noise-like rasp - another = choked square hit

#### Use **TARGET ORDER** to create rhythmic structure

The manual gives two options:

- **SEQ**: target selected by count of active thresholds
- **TIED**: target selected by most recently crossed threshold

#### **SEQ mode**

Feels more like a staircase / deterministic progression through activity count. Good for: - repeating metric complexity - stable polyrhythmic loops - "sequence-like" percussion phrasing

#### **TIED mode**

Feels more reactive and locally event-based. Good for: - asymmetrical accents - less predictable strikes - pseudo-humanized microstructure

If you want **complicated but playable**, start with **SEQ**.

If you want **glitch-percussion swarm**, use **TIED**.

---

## 4. Use SHIFT as rhythmic asymmetry control

---

From the manual, **SHIFT** pushes the input up and down against the comparators and creates asymmetry. Slow modulation produces a frequency-shift effect.

For percussion, **SHIFT** is extremely useful because it changes **which thresholds get crossed and when**.

That means **SHIFT** effectively alters: - accent placement - density - even/odd balance - left/right rhythmic feel if using stereo widening

### Patch ideas

- slow triangle LFO → **SHIFT CV**
- stepped random → **SHIFT CV**
- sequencer row in odd length → **SHIFT CV**

This is how you get: - 5-step modulation over 4-beat patterns - 7-step accent drift - asymmetrical evolving percussion

A 5-step CV sequence into **SHIFT** while the main rhythm cycles in 4 creates immediate **5:4 polymeric drift**.

---

## 5. Use 1V/OCT to make timbre track pitch-based rhythmic lines

---

The manual notes **1V/OCT controls slope and is necessary for equal timbre across different notes**.

This means you can sequence pitch into **1V/OCT** and keep the percussion's character more consistent across melodic movement. Very useful if you want: - tuned percussion lines - tom patterns in changing pitch - bass-percussion hybrids

## Advanced move

Send one sequencer to: - oscillator pitch - Skorpion **1V/OCT**

Now when your source pitch changes in a non-4/4 sequence—say 5 or 7 steps—Skorpion retains a coherent timbral logic while still generating complex attacks.

This is great for: - tuned polyrhythmic tom systems - melodic percussion ostinati in odd meter

---

# Using the auxiliary outputs as rhythm generators

---

This is where Skorpion becomes a serious rhythmic machine.

## 6. $G(IN>0)$ as a derived clock/gate

---

Output: - 0V when  $IN < 0$  - +5V when  $IN > 0$

This is basically a polarity gate from the input waveform.

Use it to: - trigger envelopes - ping LPGs - clock logic modules - fire one percussion voice

If the input is an oscillator, this is a stable derived gate. If the input is already complex, this becomes a transformed rhythmic clock.

### Polyrhythm trick

Patch: - master audio oscillator → IN -  $G(IN>0)$  → trigger hi-hat envelope -  $\pm G(DIR)$  or **COUNT**-derived comparator elsewhere → trigger snare/clap

Now two percussion streams are derived from different event interpretations of the same source.

---

## 7. $\pm G(\text{DIR})$ as alternating up/down strike logic

---

This output is: - +5V when vector core is going up - -5V when going down

This is amazing for alternating rhythmic layers.

Use a comparator / rectifier / logic after it to split: - upward motion → one drum - downward motion → another drum

That gives you: - left/right hand alternation - kick/snare interlock - open/closed hat alternation - A/B strike logic

### Great patch

- $\pm G(\text{DIR})$  → comparator 1 set > 0V → trigger rimshot
- $\pm G(\text{DIR})$  → comparator 2 set < 0V or inverter/comparator → trigger clap

Because direction changes depend on threshold crossings and target motion, the pattern can get very intricate.

---

## 8. COUNT output as accent staircase

---

COUNT gives 0–4V, each active threshold adds 0.5V.

This is effectively a stepped density meter.

Use it to modulate: - drum VCA amplitude - filter cutoff - decay time - wavefolder amount on another voice - logic threshold comparator to generate triggers only at certain densities

### Rhythmic use

Patch: - **COUNT** → VCA CV for noise hat voice  
Higher threshold activity = louder/noisier hats.

Or: - **COUNT** → comparator → trigger accent when over 2.5V

Then accents only occur when enough thresholds are active. This creates **conditional rhythm**, which is ideal for evolving complex percussion.

---

## 9. DAC output for weighted threshold state

---

DAC is a weighted version of count. Higher-number thresholds contribute more strongly.

This is more nuanced than COUNT. Great for: - subtle accent contour - selective timbral modulation - evolving velocity maps

Because the weighting is unequal, DAC reflects **which thresholds** are active, not just how many. That's musically useful for advanced patterns.

### Example

- **DAC** → FM index of metallic percussion oscillator
- **COUNT** → amplitude
- **±G(DIR)** → trigger

Same source, three related but different rhythm dimensions.

---

## 10. DIFF output for harsh, transient-rich percussion

---

The manual says **DIFF** is the difference between target voltage and the current vector core state and tends to be high in harmonic content.

This is one of the best outputs for: - clicks - zaps - snares - metallic bursts - digital percussion edges

Patch **DIFF** directly as audio, or through: - LPG - VCA with short envelope - filter - wavefolder - resonator

### Excellent percussion use

- **DIFF** → LPG → mixer

- use **G(IN>0)** or external trigger to ping LPG
- modulate SHAPE with **DIFF** itself for unstable, sharp attacks

This gives very aggressive, articulate transient percussion.

---

## 11. ABS(IN) for full-wave-derived rhythm

---

ABS(IN) is the full-wave rectified version of the input. That doubles periodicity relative to a bipolar cycle.

This is useful for: - doubling trigger rates - making denser hat pulses - deriving 2x activity from a slower source

Example: - oscillator → IN - **ABS(IN)** → comparator or envelope follower elsewhere → hat clock - **G(IN>0)** → kick trigger

Now hats occur at double the polarity-related event rate.

---

## 12. TRGTs output as an event sequencer

---

The TRGTs output is direct from the target sequencer.

This is extremely useful if you want Skorpion to be the **brains of a rhythm ecosystem**.

Use **TRGTs output** to modulate: - another oscillator pitch - drum decay - filter cutoff - probability/skipping threshold in another module - sample selection in a sampler - VC clock divider ratio if you have one

Because TRGT selection depends on threshold crossing behavior, this is not a regular clock sequencer—it's a **gesture-driven sequencer**. Perfect for advanced percussion mutation.

---

# Techniques for polyrhythms and odd meters

---

## 13. Create polymeter by using different cycle domains

---

One of the easiest ways to get complex rhythm is to let Skorpion and your sequencer run at different pattern lengths.

### Example

- 4-step pitch sequence into oscillator
- 5-step CV sequence into **SHIFT**
- 7-step CV sequence into **FOLD**
- TRGT slider pattern defines 8 event destinations

Now your overall result resolves over a very long combined cycle.

This creates: - non-repeating percussion phrases - long-form rhythmic drift  
- emergent accents

Skorpion excels here because threshold crossings are already nonlinear. Adding multiple pattern lengths compounds that complexity in a musical way.

---

## 14. Use unequal THLD layouts for embedded triplets

---

If you want the feeling of nested triplets or weird subdivisions, the threshold spacing is key.

Try arranging sliders to mimic: - 3 close crossings, then 2 spaced, then 3 close - 5 gradual steps against 3 abrupt ones - one threshold nearly unreachable except at peaks

This creates internal subdivisions that don't feel like standard 16th-note grids.

### **Musical interpretation**

- closely packed thresholds = ratchets / buzz / rolls
- widely spaced thresholds = anchor beats
- one rare threshold = occasional accent or "ghost downbeat"

This is one of the best ways to imply **11s, 13s, 7-over-4**, etc. without needing a literal sequencer in those lengths.

---

## **15. Use EQUALIZE THLDs strategically**

---

The manual says equalized thresholds make it more like a classic wavefolder.

For complex percussion: - **Equalize OFF** = more irregular, human, asymmetrical - **Equalize ON** = more grid-like, repeatable, machine-like

A very good live-performance move is to modulate or switch between equalized and unequal thresholds.

Result: - groove snaps from chaotic asymmetry into regimented machine rhythm - excellent for transitions between "free" and "locked" sections

The switch/jack behavior matters: - **ON** = always equalized - **XOR** = equalized unless high signal turns it off - **JACK** = external high signal turns equalization on

This is useful for rhythmic structuring with gates from another sequencer.

---

## **16. Use HALT for chokes, mutes, and suspended strikes**

---

The manual says HALT stops the vector core at its current voltage.

This can be used like a rhythmic choke.

## Patch ideas

- trigger pattern → **HALT**
- one drum voice uses halted states as stuttered sustain
- another voice continues normally

This creates: - abrupt gated freezes - pseudo-sidechain interruptions - broken-grid accents - stop/start fills

## HALT IF TARG=0 trick

If a target is set to 0 and **HALT IF TARG=0** is on, that segment halts only for that target.

This is huge for percussion sequencing.

You can program certain steps in the TRGT sequence as: - choke points - hard rests - square-like held plateaus - syncopated “mute steps”

That means the target sequencer can contain not only voltages but also **structural rests**.

For complex rhythms, use a few zero targets among active targets to create: - pauses inside a dense roll - missing strokes in odd places - hocket-like segmentation

---

## 17. SYNC for attack discipline

---

SYNC resets the vector core at zero crossings of IN: - **SOFT** ramps to 0V at current SLOPE rate - **X** no sync - **HARD** fast reset to 0V

For percussion: - **HARD sync** = sharper, more repeatable attacks - **SOFT sync** = rounded, elastic attacks - **No sync** = more smeared, unstable, chaotic timbre

If you want hyper-detailed rhythmic clarity in dense mixes, **HARD** is often best.

If you want rolling, swarming, unstable metallic percussion, use **SOFT** or none.

---

## **18. DRY IF NO THLDs as a continuity trick**

---

This forces dry signal to output when no thresholds are active.

For heavily modulated percussion patches, this is very useful because it prevents total dropout when FOLD or threshold states move into non-active zones.

Use it when: - aggressively modulating FOLD - using rare threshold crossings - making evolving odd-meter percussion textures

This gives continuity—important if you want complexity without losing the rhythmic line entirely.

---

# **SHAPE for micro-rhythm and transient design**

---

## **19. SHAPE source selection is a percussion goldmine**

---

SHAPE modulates SLOPE with different feedback or control sources. This changes segment curvature and therefore transient structure.

The sources listed in the manual are all useful, but for percussion these are especially strong:

### **SHAPE = DIFF**

Produces spiky harsh timbres. Great for: - snares - zaps - glitch hats - tearing transients

## SHAPE = COUNT

Slope responds to active threshold count. Great for: - complexity follows density - busier moments become brighter/harder - automatic dynamic phrasing

## SHAPE = DAC

Like COUNT but subtler. Great for: - nuanced accent shaping - less obvious but more musical complexity

## SHAPE = OUT

Classic feedback-ish log/exp shaping. Great for: - punch - thwack - body-vs-click contour

## SHAPE = TRGTs

Each segment can have different slope shaping. Very powerful for making the waveform feel like a sequence of different drum strokes.

---

## 20. Use SHAPE symmetry creatively

---

The manual says: - no feedback at 12 o'clock - **SYM up** makes modulation symmetrical for positive and negative portions

For percussion: - **symmetrical** = more even, machine-like - **asymmetrical** = funkier, more lopsided, more broken-grid feeling

If you want complicated rhythms that still feel danceable, asymmetry is often better than pure randomness.

---

# Stereo rhythm design with OUTPUT and DELAY

---

## 21. Use WIDE as rhythmic spatial multiplication

---

The OUTPUT control goes: - lower half: **DRY** ↔ **WET** - upper half: **WET** ↔ **WIDE**

WIDE adds an ultra-short delay and optional mid/side behavior.

For percussion this can create: - flam-like stereo doubles - widened hats - metallic side splashes - center-stable kick / wide highs if FILTER mode is used

### Best settings

- **OUTPUT switch = FILTERs**
- lows under 240Hz stay centered
- highs are delayed and widened
- great for:
  - centered kick/tom body
  - wide hats/snare fizz

This is excellent for dense rhythmic music because it prevents low-end smear while exaggerating upper-layer complexity.

---

## 22. DELAY output as a separate rhythmic voice

---

The DELAY jack outputs the delayed waveform from the WIDE section.

That means you can split Skorpion into: - main percussion voice from OUTL/OUTR - a delayed companion from DELAY

Process DELAY separately: - filter it brighter for hats - gate it for ghost hits - distort it for side percussion - send to a different spatial processor

Since delay time changes across OUTPUT settings, this can create tiny temporal offsets that feel like: - flams - doubles - stereo call/response - high-speed polymetric shadowing

---

## Macro modulation for evolving rhythmic systems

---

### 23. Use the Macro Envelope to animate complexity over long time scales

---

The macro envelope controls the amplitude of all LFOs in the system.

This is perfect for structure: - verse = low complexity - build = increasing threshold movement - drop = full complexity - breakdown = release back to simplicity

Since attack and release can be very long, you can create **large-form rhythmic evolution**.

#### Suggested macro setup

- slider 1: long attack
- slider 2: long release
- slider 3: moderate THLD LFO amount
- slider 4: low-mid THLD LFO rate

Then set sliders 5–8 for LFO or ENV modulation of: - FOLD - SLOPE - SHIFT - SHAPE

This lets Skorpion slowly morph from one rhythmic topology to another.

---

## 24. Use different internal modulation roles for different parameters

---

A great setup for complex percussion:

- **FOLD** = envelope mode  
Gives per-hit articulation changes.
- **SLOPE** = slow LFO mode  
Alters density and brightness over long cycles.
- **SHIFT** = LFO mode at different rate  
Creates drifting asymmetry and changing threshold access.
- **SHAPE** = envelope or slow LFO  
Controls attack sharpness and segment contour.

If each internal modulation source runs at a different slow rate, the result is deep polyrhythmic evolution even before external patching.

---

## Concrete patch recipes

---

### Patch 1: Polyrhythmic synthetic hats and snare swarm

---

**Goal:** Dense, machine-like but evolving upper percussion

Patch: - triangle VCO → **IN** - **OUTR** → HPF / VCA / mixer - **DIFF** → second VCA / mixer - **G(IN>0)** → trigger hat envelope - **±G(DIR)** through comparator/split → trigger snare envelope

Settings: - TARGET = **SLIDERS** - TARGET ORDER = **TIED** - SHAPE source = **DIFF** - SHAPE slightly asymmetric - SLOPE medium-high - FOLD medium-high - SYNC = HARD - OUTPUT in upper half toward WIDE - FILTER mode on output switch

Programming: - make 8 TRGTs alternate between low and high voltages - set 2 of them to near 0 if using HALT IF TARG=0

Result: - one oscillator produces hats, snares, and unstable accents - up/down direction creates alternation - DIFF adds tearing transient layer - widening gives fast stereo metallic complexity

---

## Patch 2: 5-against-7 percussion engine

---

**Goal:** Long-cycle polyrhythmic percussion

Patch: - VCO → IN - 5-step sequencer CV → SHIFT - 7-step sequencer CV → FOLD - clocked envelope or slower odd-length mod source → SHAPE CV - OUTL/OUTR → mixer

Settings: - EQUALIZE THLDs = OFF - TARGET = between **SLIDERS** and **CLIP** - SHAPE source = **COUNT** or **DAC** - TARGET ORDER = **SEQ** - DRY IF NO THLDs = ON

Why it works: - 5-step SHIFT changes threshold access - 7-step FOLD changes crossing intensity - SEQ targeting stabilizes the pattern enough to remain musical

Result: - repeating but very long composite cycle - accents drift without becoming random - ideal for odd-meter techno/IDM/percussion studies

---

## Patch 3: Controlled odd-meter kick/tom generator

---

**Goal:** Percussive low-end with asymmetrical phrase structure

Patch: - sine oscillator → IN - pitch sequence → oscillator pitch - same pitch CV → **1V/OCT** - COUNT → VCA CV on lowpass-filtered noise - OUTL → main kick/tom bus - DELAY → secondary percussion bus

Settings: - OUTPUT switch = FILTERs - OUTPUT around noon to 2 o'clock - SLOPE medium - FOLD medium - SHIFT near noon with slight CV - TARGET toward 5V for squarer waveforms - SHAPE source = OUT

Add: - 5-step trigger sequence to HALT for occasional chokes

Result: - centered low-end percussion - count-controlled noise adds variable attack grit - delayed output becomes ghosted side percussion - HALT introduces off-grid mutes and phrase interruptions

---

## Patch 4: Self-modulating glitch percussion laboratory

---

**Goal:** Maximum internal complexity

Patch: - VCO or drum loop → IN - **DAC** → SHAPE CV - **TRGTs output** → SHIFT CV - **ABS(IN)** → external comparator / trigger processor - **DIFF** → audio mixer - **COUNT** → FOLD CV attenuated

Settings: - TARGET = SLIDERs - TARGET ORDER = TIED - SHAPE source = TRGTs or DIFF - EQUALIZE THLDs off - SYNC soft or off - HALT IF TARG=0 on with several zero targets

Result: - Skorpion becomes a self-referential percussion ecosystem - patterns mutate through internal state feedback - excellent for hyper-detailed experimental rhythm

---

# How to make it musical instead of just chaotic

---

## 25. Keep one layer stable

---

In dense rhythmic work, keep one of these fixed: - source oscillator frequency - SYNC = HARD - TARGET ORDER = SEQ - some thresholds evenly placed - one external clock-derived modulation source

That gives your ear something to hold onto while the rest becomes complex.

---

## 26. Separate “body” from “detail”

---

Use: - **OUTL/OUTR** for body - **DIFF / DELAY / COUNT**-modulated **side voice** for detail

This helps dense percussion remain intelligible.

---

## 27. Use FILTER output mode for mix-ready complexity

---

The manual’s FILTER mode keeps lows centered and widens highs. This is especially useful if you want: - kick and low tom solidity - wide hats and snare edges - complex stereo without muddy low-end

---

## 28. Program rarity intentionally

---

Put one or two thresholds or targets in positions that are only occasionally reached. These become: - phrase-ending accents - every-11th-feel anomalies - surprise ghost hits - “rotating downbeats”

That’s often more effective than making everything equally complex.

---

# Best external modules to pair with Skorpion for your goal

---

Skorpion will be strongest for hyper-complex percussion if paired with:

- logic modules
- comparators
- clock dividers/multipliers
- burst generators
- VCAs/LPGs
- envelope followers
- trigger sequencers with odd lengths
- matrix mixers / feedback routing
- resonators / filters / noise sources
- sample players for triggered layers

Especially useful pairings: - **logic** for deriving multiple trigger streams from  $G(IN>0)$ ,  $\pm G(DIR)$ , COUNT comparisons - **comparators** to turn COUNT/DAC/ABS(IN) into gates - **VCAs** for dynamically scaling self-modulation paths - **clocked sequencers with different lengths** into SHIFT/FOLD/SHAPE

---

## Practical performance approach

---

### A performance workflow for complex rhythmic sets

---

1. Start simple:
2. equalized thresholds
3. hard sync
4. target near 5V or clip
5. little modulation

6. Add complexity:
7. target = sliders
8. unequal thresholds
9. shape from DIFF or COUNT
10. internal macro modulation active
11. Introduce polymeter:
12. odd-length CV into SHIFT
13. separate odd-length CV into FOLD
14. Add structural disruption:
15. HALT triggers
16. zero targets with HALT IF TARG=0
17. switch target order
18. Open the stereo field:
19. move OUTPUT into WIDE
20. use FILTER mode

This gives you a clear arc from groove to rhythmic overload.

---

## Best “cheat codes” from the manual for your exact goal

---

If your aim is **densely rhythmic, hyper-complex percussion**, these are the highest-value controls:

- **THLD sliders**: your hidden rhythm topology
- **TARGET = SLIDERS**: event-sequenced articulation
- **TARGET ORDER**: SEQ for structure, TIED for chaos
- **SHIFT modulation**: asymmetrical rhythmic drift

- **SHAPE = DIFF / COUNT / DAC / TRGTs**: transient and segment complexity
  - **COUNT / DAC / ±G(DIR) / G(IN>0)**: extract multiple related rhythms
  - **HALT + HALT IF TARG=0**: chokes and intentional gaps
  - **SYNC HARD**: make dense rhythm articulate
  - **OUTPUT FILTER + WIDE**: wide highs, centered lows
  - **Macro Envelope + internal LFOs**: long-form evolution
- 

## Short answer

---

Use Skorpion as a **threshold-driven percussion sequencer disguised as a wavfolder**:

- feed it a simple oscillator or rhythmic source
- use **uneven THLDs** to define internal subdivisions
- set **TARGET to SLIDERS** so threshold crossings create different per-hit articulations
- derive extra drum triggers and modulation from **G(IN>0), ±G(DIR), COUNT, DAC, DIFF, ABS(IN), and TRGTs**
- drive **SHIFT, FOLD, and SHAPE** with odd-length CV patterns for polymeter
- use **HALT** and **HALT IF TARG=0** for chokes, rests, and broken-grid syncopation
- use **HARD SYNC** for precision and **WIDE/FILTER** for mix-ready stereo percussion

That's how you turn Skorpion into a generator of complex, interlocked, evolving percussion systems rather than just a timbre effect.

[Generated With Eurorack Processor](#)