
Cascading Register
Clock CV
Clock Out
Clock Rate
Clock in
Data
ØØ
Ø1
Ø2
Ø3
Ø4
Ø5
Ø6
Ø7
CV 1
CV 2
CV 3
-
+
-
+
Ø
+
Seed
omiiindustriies

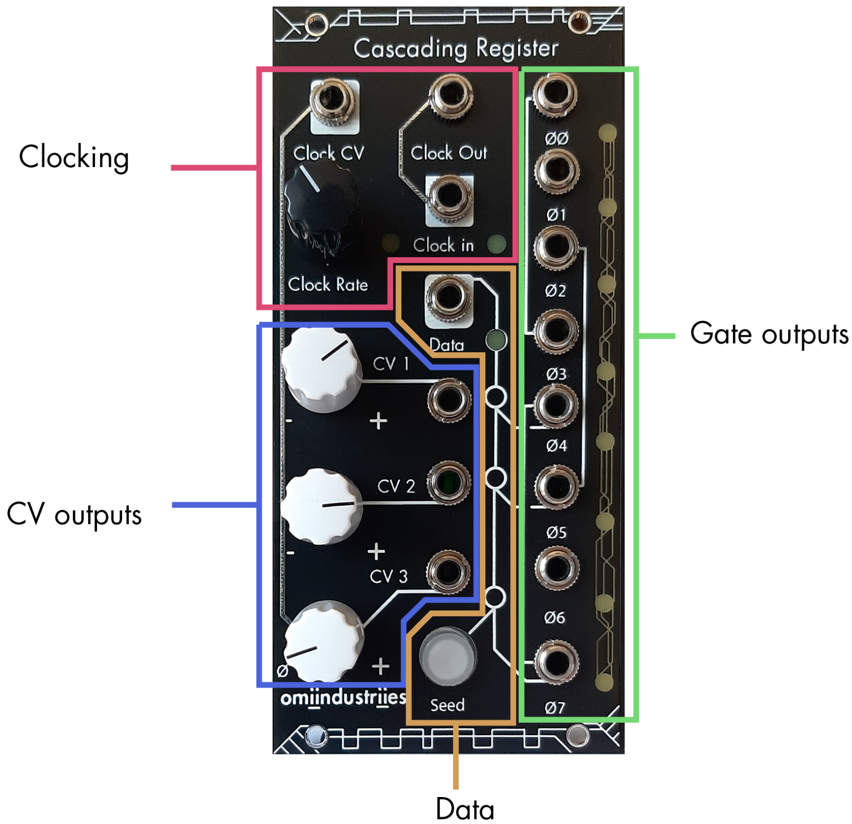
The Cascading Register from omiindustriies is a source of pseudo-random voltages and gates using an eight-bit shift register. It takes inspiration from digital shift registers, runglers, linear feedback shift registers (LFSR), and analog shift registers.

Quick start guide

Plug the Cascading Register into your Eurorack power supply, making sure the power is off and the red stripe indicating -12V is facing down. Turn on the power supply.

When you first power on the module, all of the blue LEDs will be on or off. Pressing the **Seed button** at the bottom of the module or patching a gate signal into the **Data input** will add new data into the register, and you should see the lights begin to move from top to bottom. Don’t be surprised if the rate isn’t steady, as the third CV output modulates the frequency of the internal clock oscillator. Now you’re free to patch the Cascading Register to other parts of your Eurorack system.

Try inputting different gate signals into the **Data input** or **clock** it from an external clock source such as a clock divider. Use the three stepped CV outputs to modulate parameters such as pitch or timbre on an oscillator and use the gate signals to provide timing signals to initialize an envelope or play a sample. Now that you have that going, let’s take a look under the hood and explore the module more in-depth



The Cascading Register is the continuation of ideas put forward with the first omiindustriies module, the Dual Digital Shift Register (DDSR). This module looked to Rob Hordijk's famous (or infamous) circuit at the heart of the Benjolin and Blippoo Box, the Rungler. But before we dive into the specific implementation of historic shift registers, we must look at how they actually work.

Shift registers in detail

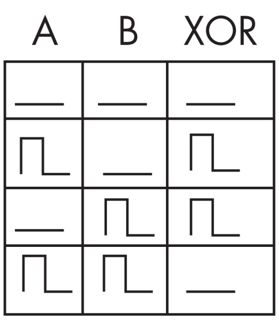
Digital Shift Registers

A shift register has a clock input and a data input. In the case of digital shift registers, the **data input** is a binary input with only two states, on or off. On the rising edge of a timing/clock signal (typically a gate or trigger signal) greater than 0.5V at the **clock input**, the shift register looks at the data input. It takes the state of the signal present at the data input and puts it into the first stage of the register, (ØØ). In this case, it's an eight-bit shift register, meaning it has eight stages, zero-indexed from ØØ to Ø7. If the signal present at the data input is above 0.5V, it moves a high signal into the first stage (ØØ), and if it's low, it moves a low state into the first stage. Additionally, on every clock pulse, whatever data is in the first stage moves to the second (Ø1), whatever is in the second stage moves to the third (Ø2), and so on and so forth. When a bit of information reaches the final stage, Ø7, the data is shifted out of the register.

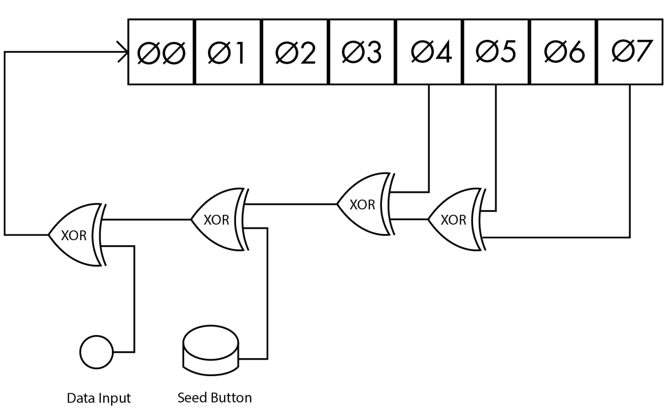
The external data input is not the only source of data within the Cascading Register. The illuminated Seed button on the bottom of the module allows you to manually seed new data into the register. Additionally, three of the stages are fed back into the data stream along with the external data and the button.

Linear Feedback Shift Register (LFSR)

This brings us to our second form of inspiration for the Cascading Register, the linear feedback shift register (LFSR). An LFSR is a common way to generate pseudo-random numbers within software. Due to the fact that you cannot replicate thermal noise present in analog noise generators, programmers must generate randomness within code. In lieu of an external data source, an LFSR uses the data in the register and feeds that back into itself to generate the pseudo-random pattern. The LFSR takes two or more of the stages present in the shift register and combines them through a Boolean logic function, typically XOR. XOR is typically a two-input logic gate, where the output is active if one of the inputs or the other is active, but inactive if both or neither are. Put another way, the output is active if the two inputs don't match, and inactive if the two inputs do match.



The selection of stages within a shift register determines the length of the pseudo-random pattern. In an eight-bit shift register, taking the fourth, fifth, sixth, and eighth stages of the shift register and XOR'ing them together generates the longest stream of pseudo-random numbers, 255 stages long. This is called the maximal length shift register, whose length is 2^n-1, so two to the eighth power, minus one. What happens if all the bits are 0? No data are recycled back into the register, so this is an invalid state. This is the reason that the maximal length of the pattern is 2^n-1: there are 2^n possible combinations of bits, but we remove the one that will not work with the LFSR. The Cascading Register is not a maximal length shift register, as it takes only the 5th(Ø4), 6th(Ø5), and 8th (Ø7) stages and XORs them with both the external data input and the seed button located on the bottom of the module. In order, Ø7 is XOR'd with Ø5, that output is XOR'd with Ø4, that is XOR'd with the seed button, and that is XOR'd with the external data input.



To create the maximal length shift register configuration with the Cascading Register, simply patch the output of the fourth stage Ø3, into the external data input.

In addition to the eight gate outputs located on the right side of the Cascading Register, it includes three CV outputs. The state of the eight stages of the shift register determines the level of the CV outputs. The module uses three simple digital to analog converters (DACs) in order to generate stepped CVs from the gates. CV1 is derived from bits ØØ, Ø1, Ø2, Ø3, CV2 comes from bits Ø2, Ø3, Ø4, Ø5, and CV 3 uses bits Ø4, Ø5, Ø6, Ø7. Notice how the first stage of each of the three sets is shifted by two stages. The four bits of data present in CV1 pass into CV2 after two clock pulses, and then to CV3 after another two clock pulses. This creates an effect reminiscent of an analog shift register.

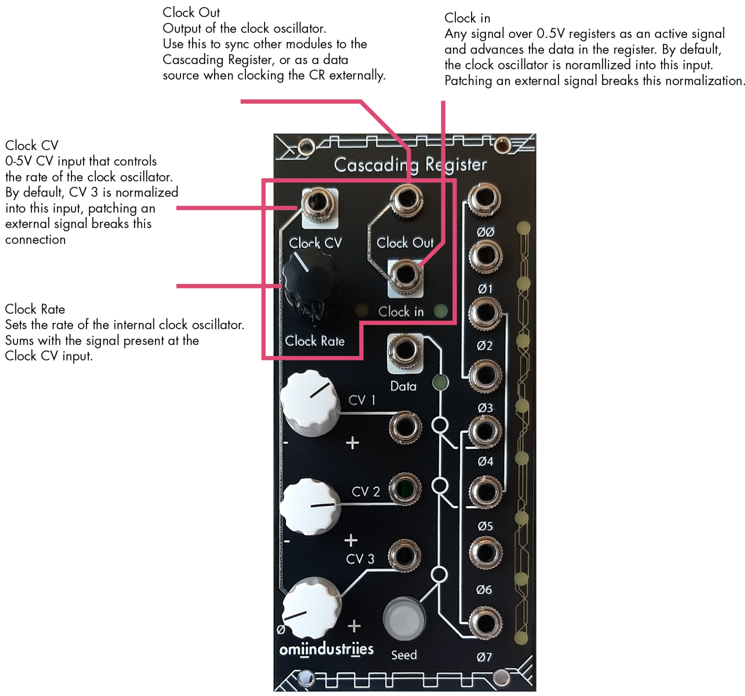
Analog Shift Register (ASR)

An analog shift register is essentially a series of sample and holds. Instead of binary data input, the data or sampling input on a sample and hold accepts continuous voltages. When the sample and hold receives a timing signal at the clock input, it takes a snapshot of the voltage level present at the sampling input and moves it to the output. When it receives another timing signal, the sample and hold looks again at the sampling input and moves that voltage level into the output. With an analog shift register, the voltage level passes from one sample and hold to the next one in the linear chain of voltage information. So every time the analog shift register gets a timing signal, the first sample and hold samples a new value and the old value passes to the second sample and hold in the line, almost like a game of telephone. When used to sequence the pitch of multiple oscillators, it creates a canonic musical structure known as an arabesque. While the Cascading Register uses a digital shift register instead of a series of sample and holds, the stepped pseudo-random sequence passes linearly from the first CV output to the second and then third, harkening back to that Arabesque musical form.

Two of the CV outputs include an attenuverter and one an attenuator. An attenuator allows you to reduce the amplitude of the signal, much like a volume control. All the way down, no signal, all the way up, full-strength signal. With an attenuverter, the off position is in the center. Turning clockwise increases the signal positively while turning counterclockwise increases the signal negatively. The attenuverter allows for both positive and negative modulation, increasing or decreasing a given modulation destination.

Inputs and Outputs

The Cascading Register includes two gate inputs (clock and data) and one CV input (clock CV), all marked with a white border around the corresponding jack. The eight gate outputs, three CV outputs, and clock oscillator outputs do not feature this border. The module includes normalization, preset signal paths from one output to an input which can be overridden by patching another cable into that input jack, marked with a silver line from output to input. This includes the output of the clock oscillator into the clock input and the output of CV3 to the rate CV of the clock oscillator.



Internal and external clocks

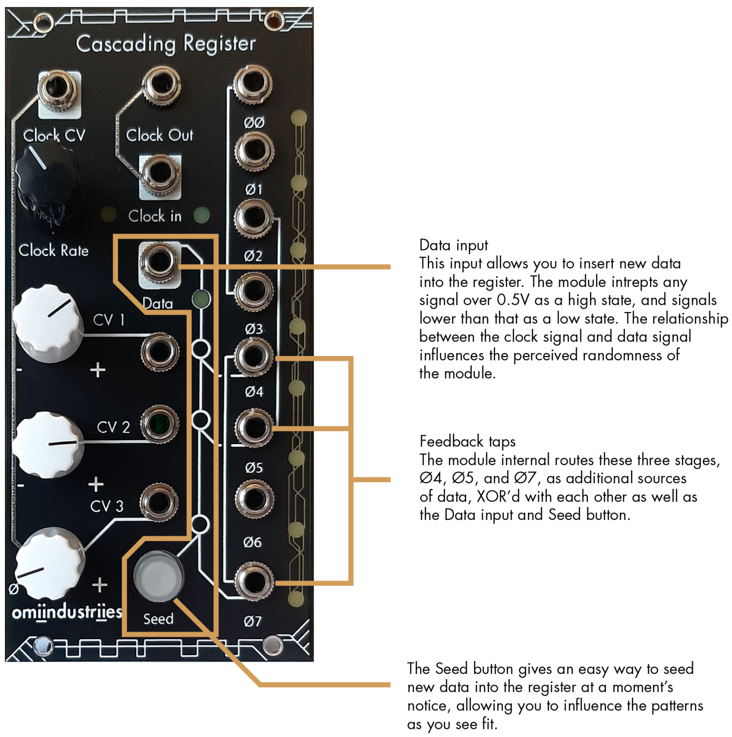
In addition to a shift register, the Cascading Register includes an internal voltage-controlled **clock oscillator**. This clock oscillator runs at sub-audio to low audio rates and features CV control over the rate. With no cable present at the clock input jack, the clock signal is normalized into the input clock. A second normalization routes the output of the **CV3** into the **CV input** of the clock oscillator. This creates a self-modulating feedback loop without the need to patch anything.

Sometimes you want a free-running clocked modulation source, but other times you may feel the need to sync the Cascading Register to an external clock signal. Patching into the **clock input** overrides the internal clock. Good sources of clocks or gate signals in modular synthesis include clock dividers/multipliers, square wave LFOs, sequencers, touch controllers, or another random gate source. Clocking the CR from a shared external clock source is a great way to get your CV and gates in sync with the rest of your patch.

If you do decide to clock the Cascading Register from an external clock source, the internal clock is still active. Because this is modular synthesis, you can patch that clock elsewhere in your system or back into the Cascading Register. Serge Tcherepnin's famous Serge modular synthesizers pioneered a technique called patch programming. Patch programming basically comes down to changing the function of a module by patching one of the outputs directly back into one of the inputs on the same module. Using this concept, we can patch the output of the clock oscillator into the external data input, turning it into a data source. This frees up other gate signals in your system and allows the Cascading Register to function as a fully-fledged clocked pseudo-random modulation and timing source with only one external clock signal needed.

While the internal clock runs at sub-audio to low audio rates, the Cascading Register accepts signals from sub-audio all the way into video rate. At audio rate, the CR becomes a noise generator with both stepped analog and digital noise outputs coming from the CV and gate outputs respectively.

When clocked at video rates, it creates pseudo-random noise and textures. Running one of the CV outputs to modulate the frequency of the video oscillator is a great way to add an additional layer of complexity to the patterns. It works best with video oscillators, with ramps and actual video signals giving mixed results, but experiment with different inputs and see what you get.



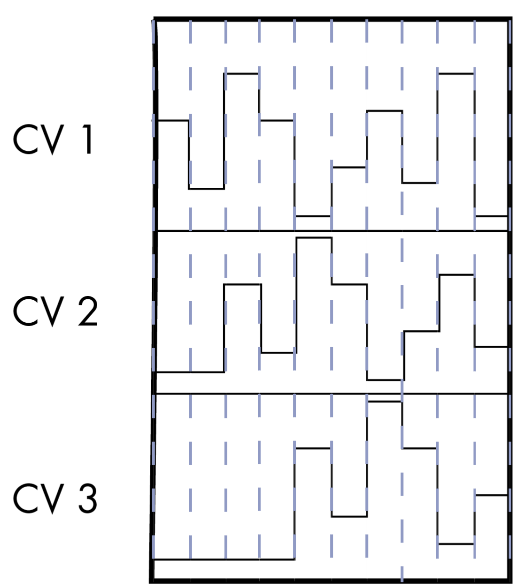
Data input and seed button

A shift register includes a data input to pass information into the register. The **Seed button** gives an easy way to seed new data into the register at a moment's notice, allowing you to influence the patterns as you see fit. This is particularly useful when first turning on the module, as all the stages may be active or inactive. Pressing the button enters new data into the register, and you should see on and off stages moving from the top to the bottom of the module. The CR also includes an external **data input**, allowing you to patch external data into the module. Usually, this takes the form of a gate signal, but any signal over 0.5V will register as a high state and any signal below that (including negative voltages) registers as a low state.

In addition to the external data input and the Seed button, data comes from three of the stages of the shift register,(Ø4, Ø5, and Ø7), as indicated by the white lines on the panel. This addition of the feedback stages means the module gets data from several sources and allows for a wider range of external data sources to be used with the module.

CV outputs

The Cascading Register includes three CV outputs derived from the state of the stages of the shift register. **CV 1** is determined by ØØ, Ø1, Ø2, and Ø3. While Ø2, Ø3, Ø4, and Ø5 correspond to **CV 2** and Ø4, Ø5, Ø6, and Ø7 make up **CV3**. This grouping of stages creates three copies of the same data shifted by two clock pulses.



The CV outputs come from the first four, the middle four, and the last four stages of the shift register which creates three copies of a CV signal, shifted by two clock pulses. The three white knobs control the amplitude of the CV outputs, allowing you to fine-tune the amount of modulation. CV1 and CV 2 include attenuverters (bipolar level control), while CV3 uses an attenuator (unipolar level control).

The three white knobs control the amplitude of the CV outputs, allowing you to fine-tune the amount of modulation. CV 1 and 2 have associated attenuverters, while CV 3 includes an attenuator. Attenuators go from off when the control is all the way counterclockwise, and at full amplitude when the control is all the way clockwise. Attenuverters, on the other hand, are off when centered. Turning the knob clockwise from the center increases the amplitude positively while turning counterclockwise from the center increases the amplitude negatively. Increasing negatively may sound counterintuitive, but if you patch into the pitch of an oscillator, the negative voltage will cause the pitch of the oscillator to decrease rather than increase. (Note, not all modules accept or register negative voltage, so if it seems like the negative voltage isn't doing anything, refer to your module's manual.)

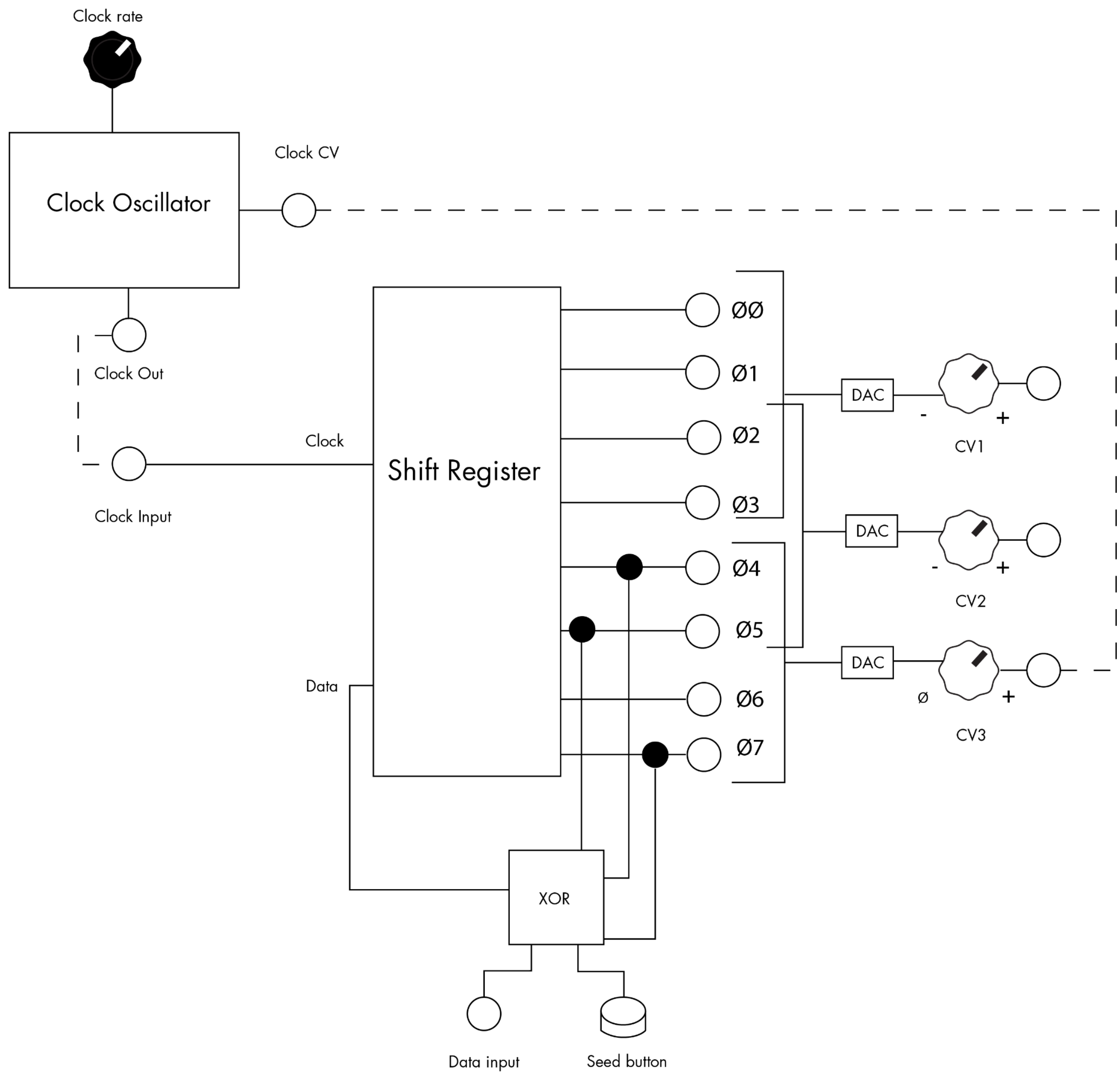
Gate outputs

Gate outputs
The gate outputs are zero indexed from ØØ to Ø7. The state of the gates, or the data in the register, moves from top to bottom, until it reaches the final stage and then is shifted out of the register. Each clock pulse advances the data in the register while inserting new data into the register.

The Cascading Register includes gate outputs for each of the eight stages of the shift register, labeled ØØ to Ø7. The outputs are binary gate signals, meaning they only have two states, on with an amplitude of +5V or off, 0V. Gates fill a variety of uses within a modular synthesizer but are primarily used for timing information. This includes firing off an envelope, triggering a sample, advancing a sequencer, or generating a new random value from another random module.

Making rhythms with the gate outputs is fairly straightforward. The position from ØØ to Ø7 determines where in the bar the beats land. So if you patch to a kick drum with ØØ and a snare drum sound with Ø2, you get a call and response of the two sounds shifted by two clock pulses. Combining several of the stage outputs with a logic module such as the omiindustriies Illyana allows you to increase the complexity of your rhythms.

However, gates are not just for timing, they also come in handy as modulation sources. This works especially well with percussive sound sources and drum modules that benefit from sudden and stepped changes in parameters. Attenuating or attenuverting the gates will help you fine-tune the amount of modulation from a 5V signal to a more incremental change.

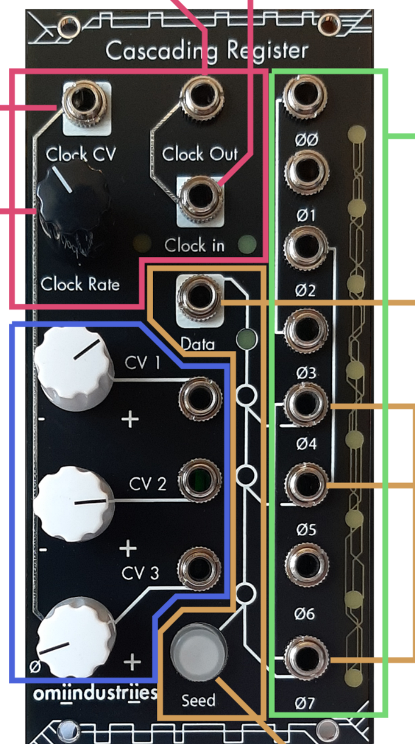

Clock rate
Clock Oscillator
Clock CV
Clock Out
Clock
Clock Input
Shift Register
Data
ØØ
Ø1
Ø2
Ø3
Ø4
Ø5
Ø6
Ø7
DAC
DAC
DAC
-
+
CV1
-
+
CV2
ø
+
CV3
XOR
Data input
Seed button

Clock Out
Output of the clock oscillator.
Use this to sync other modules to the Cascading Register, or as a data source when clocking the CR externally.

Clock in
Any signal over 0.5V registers as an active signal and advances the data in the register. By default, the clock oscillator is normalized into this input. Patching an external signal breaks this normalization.

Clock CV
0-5V CV input that controls the rate of the clock oscillator. By default, CV 3 is normalized into this input, patching an external signal breaks this connection

Clock Rate
Sets the rate of the internal clock oscillator. Sums with the signal present at the Clock CV input.



Gate outputs
The gate outputs are zero indexed from ØØ to Ø7. The state of the gates, or the data in the register, moves from top to bottom, until it reaches the final stage and then is shifted out of the register. Each clock pulse advances the data in the register while inserting new data into the register.

Data input
This input allows you to insert new data into the register. The module intrepts any signal over 0.5V as a high state, and signals lower than that as a low state. The relationship between the clock signal and data signal influences the perceived randomness of the module.

The CV outputs come from the first four, the middle four, and the last four stages of the shift register which creates three copies of a CV signal, shifted by two clock pulses. The three white knobs control the amplitude of the CV outputs, allowing you to fine-tune the amount of modulation. CV1 and CV 2 include attenuverters (bipolar level control), while CV3 uses an attenuator (unipolar level control).

Feedback taps
The module internal routes these three stages, Ø4, Ø5, and Ø7, as additional sources of data, XOR'd with each other as well as the Data input and Seed button.

The Seed button gives an easy way to seed new data into the register at a moment's notice, allowing you to influence the patterns as you see fit.