



Rostock

THE LEIBNIZ BINARY SUBSYSTEM

**BINARY DATA
PIPELINE**

Model of 1989

OPERATOR'S MANUAL rev. 1989/X6/1.0

SALUT

Thank you for purchasing this Xaoc Devices product. Rostock [*ˈrɒstɒk*] is a data pipeline, a FIFO (first in—first out) shift register, or a very short digital delay. Its length is variable from 1 to 64 stages (with optional looping and scrambling). Rostock is a component of the Leibniz Binary Subsystem—a family of digital modules that operate on 8-bit data. It works by processing data in the Leibniz bus and must be connected to other Leibniz modules using data ribbon cables at the back. Rostock may process data sequences representing rhythms, control voltages, audio-rate signals, and even video signals because the bits can change at extreme rates (up to 2MHz). In addition, its memory may be digitally looped. Delaying and looping is a fundamental building block for sequence automation, pattern and chaos generation, and various cybernetic modular patches. Thus, Rostock is a multi-purpose open-ended device that invites creative thinking.

To better understand the device and avoid common pitfalls, we strongly advise the user to read through the entire manual before use.

INSTALLATION

The module requires 8hp worth of free space in the Eurorack cabinet. Always turn the power off before plugging the module into the bus board using the supplied 16-pin to 16-pin ribbon cable, paying close attention to power cable pinout and orientation. The red stripe indicates the negative rail and should match the dot or **-12V** mark on the bus board as well as the unit. Rostock is internally secured against reversed power connection; however, flipping the 16-pin header **MAY CAUSE SERIOUS**

DAMAGE to other components of your system because it will short-circuit the +12V and +5V power lines. Always pay close attention to the proper orientation of your ribbon cable on both sides!

Besides power, you need to connect Rostock to other components of your Leibniz Subsystem. We advise you to plan ahead as to how you wish to incorporate Rostock into your modular setup. The module ships with one 10-pin ribbon data cable, so you will need to use the additional 10-pin cables included with your other Leibniz modules to complete the connection. First, connect the 10-pin unshrouded header labeled **OUT** to the **IN** header of your next Leibniz-compatible module (e.g., Drezno, Poczdam, Lipsk, Jena, Erfurt, Odessa, etc.). Then connect the **OUT** header of the module that will provide input data to Rostock (e.g., Poczdam, Drezno, Erfurt, Ostankino, etc.) to the 10-pin unshrouded header labeled **IN**. Please observe the marks of pin #1 (red stripe) on all connected modules.

WARNING: DO NOT PLUG A POWER CABLE INTO THE LEIBNIZ HEADERS; THIS WILL DAMAGE YOUR UNIT AND MAY ALSO JEOPARDIZE OTHER LEIBNIZ MODULES CONNECTED TO IT!

The module should be fastened by mounting the supplied screws before powering up.

MODULE OVERVIEW

The front panel of Rostock is shown in fig. 1. Contrary to many other modules in the Leibniz series, Rostock does not offer data input or data output jacks. Hence it relies entirely on the data connections at the back of the module. It features a bank of eight clock inputs ①,

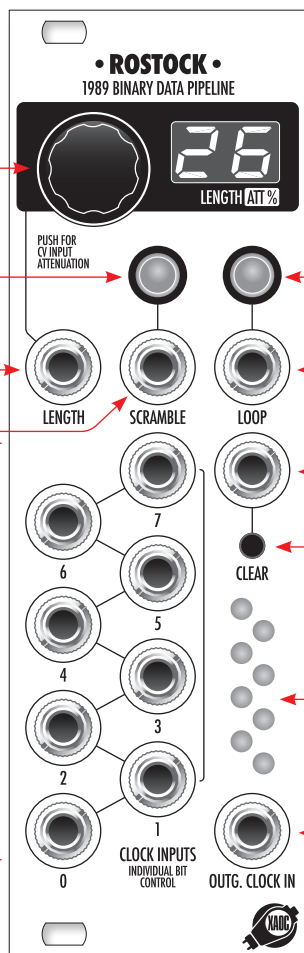


fig. 1

THE INTERFACE

which allows you to replace the original Leibniz clock arriving with the source data (independently for each bit!). Therefore, the timing of each individual bit line may be different.

An additional clock input ② enables you to substitute the output data clock with any other signal. This clock does not affect the pipeline; it only affects any subsequent modules in the Leibniz data chain.

The bank of 8 LEDs ③ displays the status of each bit output of the delay line, where the highest one represents the most significant bit (bit 7), and the lowest one represents the least significant bit (bit 0).

The 2-digit LED display ④ at the top of the front panel and the endless encoder ⑤ offer two functions. By default, the display shows the length of the delay line (the number of stages of the shift register) while the encoder changes this value in the range of 1-64. In addition, this length may be modulated by an external CV plugged into the input below ⑥, which accepts bipolar voltages in the range of -10V to 10V.

Pressing the encoder knob switches the display and encoder function to adjust the modulation depth from 0 to 99%. This secondary function is indicated by an additional dot on the display. Pressing the encoder again returns to the default operation.

There are two illuminated buttons below the display. The right button labeled **LOOP** ⑦ switches between new data being written 1:1 to the pipeline or closing the loop around it. The left button labeled **SCRAMBLE** ⑧ toggles

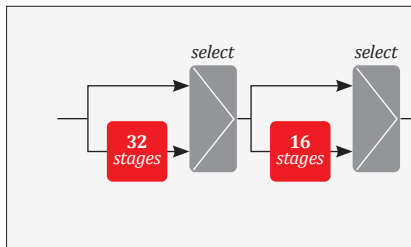
between a 100% loop (input data is ignored and the content losslessly recirculates) and an XOR-style loop (input data is combined with the output data and written back to the input).

Two gate input jacks ⑨ and ⑩ below each of these two buttons allow for remote looping control. Note that the **SCRAMBLE** function has no effect when looping is off. The entire content of the pipeline's memory may be cleared by pressing the small button labeled **CLEAR** ⑪ or by a trigger or gate signal sent into the jack above it ⑫.

THE PRINCIPLE OF OPERATION

Rostock operates similarly to a BBD delay or an ASR (analog shift register), except that it is digital. It consists of a chain of memory cells that pass bits from one to another at each clock cycle.

There are eight such chains inside Rostock operating in parallel: one for each of the bits in the 8-bit Leibniz data. The length of these chains may be set in the range of 1 to 64 stages via the encoder and screen as well as external CV. For example, when the length is set at 48, the input data passes through 48 memory



cells and is delivered to the output after 48 cycles of the clock. In other words, the data is digitally (and losslessly) delayed.

The physical delay is a product of the clock period and the length of the pipeline. For example, with a 2MHz clock, it can vary from 0.5 μ s to 32 μ s (inaudible, video rate range); while with a 5kHz clock it can vary from 0.2ms to 12.8ms (very short audio rate range); and with a 10Hz clock it will be 100ms to 6.4s (rhythm and LFO range).

Varying the length changes the number of memory cells used. Changing the length while a signal is being processed yields some idiosyncratic digital distortion. To understand this distortion, you need to understand the internal structure of the pipeline (fig. 2).

The delay length selects a combination of memory blocks offering 32 stages, 16 stages, 8 stages, 4 stages, 2 stages, and 1 stage, plus there is one fixed 1-stage cell so that the delay is never 0. While the input data always passes through all blocks, some blocks are omitted in the current pipeline by inserting the input data directly into some later blocks. For example, when the length is set

to 33, the delay consists of a 32-stage block, and the remaining blocks are skipped, except for the last fixed single stage. When the length is changed to 32, the 32-stage block is skipped, and the 16, 8, 4, 2, and 1-stage blocks are used. These, combined with the last fixed 1-stage block, offer the required 32. Note that such switching (only by 1) yields significant data shuffling in the memory and, sometimes, an interesting glitch.

LOOPING & SCRAMBLING

When the **LOOP** button is unlit, the device operates as a straightforward delay: new data from the Leibniz **IN** is written to the shift register at each rising edge of the clock, it is delayed by the number of clock pulses set with the encoder and shown on the display, and it is passed to the Leibniz **OUT**.

With the **LOOP** function activated (by pressing the button or plugging a 5V gate into the **LOOP** control input jack), the output data from the pipeline is fed back to the input. This function allows you to catch a sequence of data representing, e.g., one cycle of a waveform or a drum pattern and repeat it as long as needed.

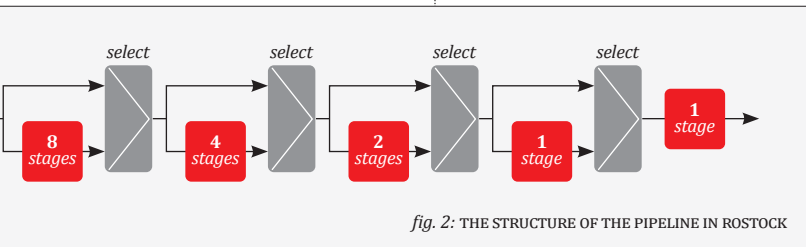


fig. 2: THE STRUCTURE OF THE PIPELINE IN ROSTOCK

NOTE: The length of the loop will equal the length of the pipeline. If you change the length, it will cause glitches, as discussed in the previous section. Shortening the loop while playing back the data will overwrite the unused memory cells with the shortened loop. If the length is increased afterward, the pipeline will still contain the shortened loop only.

The **SCRAMBLE** button (together with the signal sent to the control input jack) changes the looping behavior. When activated, the pipeline is fed with both the input and output bits, combined using logic XOR (fig. 3 shows the truth table for this function). For example, if the input bit is a constant 1, the stream of data bits from the output will be inverted and written back to the input, thus generating a cyclic pattern every two lengths of the pipeline. Feeding the input with more or less varying data and operating the **SCRAMBLE** function allows for creating cyclic binary patterns of various complexity, similar to 8 channels of a binary Turing machine, or a 256-valued discrete state machine, or anything in between, depending on how you patch them.

You can reset the loop's content at any point by pressing the small **CLEAR** button or plugging a gate signal into the control input jack above it.

CLOCKS

In general, the physical delay offered by Rostock may be changed continuously within an extreme range by varying the source clock. Note that this delay is inversely proportional to the clock frequency: you can increase the delay 1000 times by slowing the clock down

INPUT A	INPUT B	OUTPUT
0	0	0
0	1	1
1	0	1
1	1	0

fig. 3: THE TRUTH TABLE OF THE 'XOR' FUNCTION

by 1000 times. This, however, comes at the cost of temporal resolution because data will be sampled 1000 times slower.

By default, Rostock uses the source clock that arrives together with the input data via the Leibniz interface to drive its internal shift registers. Since each data bit is processed by a separate chip, it is possible to change the timing of each bit's pipeline by replacing the default clock using the front panel jacks. For data representing audio, this will naturally destroy the signal integrity as individual bits will be delayed differently. However, when you use individual Leibniz bits as channels of a trigger sequence, this timing behavior is ideal for creating rhythmic changes.

NOTE: There is no limit to how slow the clock can be, and it doesn't even have to be a regular pulse train. You can plug a stream of random gates or bits of the binary data there. The Leibniz hardware makes no distinction between clocks and binary gate signals.

PATCH IDEAS

- Using Drezno with Rostock allows for experimenting with short delays on analog CV

and audio signals: connect the **Leibniz OUT** of Drezno to the **IN** header of Rostock and the **OUT** header of Rostock back to the **Leibniz IN** header of Drezno. Replace the MHz internal ADC clock with a slower oscillator. Alternatively, you can create an analog feedback loop using a simple mixer. To do that, connect your original signal to one mixer input. Next, connect the output of the mixer to the **ADC INPUT** of Drezno. Finally, connect the **DAC OUTPUT** of Drezno to another input of the mixer and mult this **DAC OUTPUT** to use as the final signal output for monitoring.

TIP: Carefully set the **GAIN** and **OFFSET** sliders in Drezno, especially with deep feedback, as it will tend to accumulate the DC offset and eat your headroom. You can prevent this by inverting or AC-coupling the signal before patching it to the feedback input of your mixer.

- Use Rostock as a delay loop for Jena to achieve a multi-attractor chaotic oscillator patch. Connect the **OUT** header of Rostock to the **IN** header in Jena and the **OUT** header of Jena to the **IN** header of Rostock. If you need an analog output, you can use a splitter Leibniz cable (sold separately or DIY) to connect the same **OUT** of Jena to Drezno's **LBZ IN** header. Alternatively, you can connect the front panel outs of Jena to the front panel inputs of Drezno.

TIP: The device may get stuck in a fixed state, especially if you start with a wave that maps 0 to 0, or is generally monotonic. Use Jena's **PHASE** knob or the **PHASE CV** input to kick it out of this. Also, experiment with different clock frequencies plugged into the **OUTG. CLOCK IN**.

- Connect Lipsk's **OUT** header to one of the **IN** headers of Poczdham, the **OUT2** of Poczdham to the **IN** header of Rostock, and the **OUT** of Rostock back to the **IN** of Lipsk. Activate **LINK**. Use your own clock to drive the sequence through the **OUTG. CLOCK IN**. This patch is a programmable trigger/gate sequencer with 8 channels and up to 64 steps. Use Lipsk's buttons to change the state of individual lines. Poczdham's front panel outputs are your trigger outs, and the sequence length is set in Rostock. At any time you can loop Rostock locally to ignore the state of Lipsk or activate **SCRAMBLE** to generate random patterns.

- Connecting Rostock in a loop with Poczdham gives access to all the bit outputs and inputs to mangle. For example, you can patch an $8 \times 64 = 512$ stage one-bit delay by connecting all channels in series: use seven patch cables to connect Rostock's **OUT 7** with **IN 6**, **OUT 6** with **IN 5** and so on finishing with **OUT 1** to **IN 0**. Your single bit input is **IN 7**, and **OUT 0** is the output of the long chain. Using splitter patch cables and different clocks for the individual bits, you may patch a multi-tap trigger delay to drive a sequence of events in your modular.

CONNECTIVITY

Rostock connects to all modules compatible with the Leibniz Binary Subsystem: Drezno, Lipsk, Gera, Jena, Erfurt, Poczdham, Ostankino II, and Odessa.

ACCESSORY

Our Coal Mine black panels are available for all Xaoc Devices modules. Sold separately. Ask your favorite retailer. •

WARRANTY TERMS

XAOC DEVICES WARRANTS THIS PRODUCT TO BE FREE OF DEFECTS IN MATERIALS OR WORKMANSHIP AND TO CONFORM WITH THE SPECIFICATIONS AT THE TIME OF SHIPMENT FOR ONE YEAR FROM THE DATE OF PURCHASE. DURING THAT PERIOD, ANY MALFUNCTIONING OR DAMAGED UNITS WILL BE REPAIRED, SERVICED, AND CALIBRATED ON A RETURN-TO-FACTORY BASIS. THIS WARRANTY DOES NOT COVER ANY PROBLEMS RESULTING FROM DAMAGES DURING SHIPPING, INCORRECT INSTALLATION OR POWER SUPPLY, IMPROPER WORKING ENVIRONMENT, ABUSIVE TREATMENT, OR ANY OTHER OBVIOUS USER-INFLECTED FAULT.

LEGACY SUPPORT

IF SOMETHING GOES WRONG WITH A XAOC PRODUCT AFTER THE WARRANTY PERIOD IS OVER, THERE IS NO NEED TO WORRY, AS WE'RE STILL HAPPY TO HELP! THIS APPLIES TO ANY DEVICE, WHEREVER AND WHENEVER ORIGINALLY ACQUIRED. HOWEVER, IN SPECIFIC CASES, WE RESERVE THE RIGHT TO CHARGE FOR LABOR, PARTS, AND TRANSIT EXPENSES WHERE APPLICABLE.

RETURN POLICY

THE DEVICE INTENDED FOR REPAIR OR REPLACEMENT UNDER WARRANTY NEEDS TO BE SHIPPED IN THE ORIGINAL PACKAGING ONLY AND MUST INCLUDE A COMPLETED RMA FORM. XAOC DEVICES CAN NOT TAKE ANY RESPONSIBILITY FOR DAMAGES CAUSED DURING TRANSPORT. SO BEFORE SENDING US ANYTHING, PLEASE CONTACT US AT SUPPORT@XAOCDEVICES.COM. NOTE THAT ANY UNSOLICITED PARCEL WILL BE REJECTED AND RETURNED!

GENERAL INQUIRIES

FOR USER FEEDBACK SUGGESTIONS, DISTRIBUTION TERMS, AND JOB POSITIONS, FEEL FREE TO CONTACT XAOC DEVICES AT INFO@XAOCDEVICES.COM. PLEASE VISIT XAOCDEVICES.COM FOR INFORMATION ABOUT THE CURRENT PRODUCT LINE, USER MANUALS, FIRMWARE UPDATES, TUTORIALS, AND MERCHANDISE.

WORKING CLASS ELECTRONICS®

EASTERN BLOC TECHNOLOGIES



MADE IN THE EUROPEAN UNION

ALL RIGHTS RESERVED. CONTENT COPYRIGHT ©2023 XAOC DEVICES. COPYING, DISTRIBUTION, OR COMMERCIAL USE IN ANY WAY IS STRICTLY PROHIBITED AND REQUIRES WRITTEN PERMISSION FROM XAOC DEVICES. SPECIFICATIONS ARE SUBJECT TO CHANGE WITHOUT PRIOR NOTICE. EDITING BY BRYAN NOLL.

MAIN FEATURES

*Leibniz Binary
Subsystem component*

*Shift register /
delay memory of
up to 64 stages*

*Voltage controlled
length*

*Looping and
scrambling*

*Optional individual
clock input to
each bit line*

TECHNICAL DETAILS

*Eurorack synth
compatible*

*8hp, skiff
friendly*

*Current draw:
+70mA/-40mA*

*Reverse power
protection*

*NOT protected
against plugging
power to Leibniz
headers!*