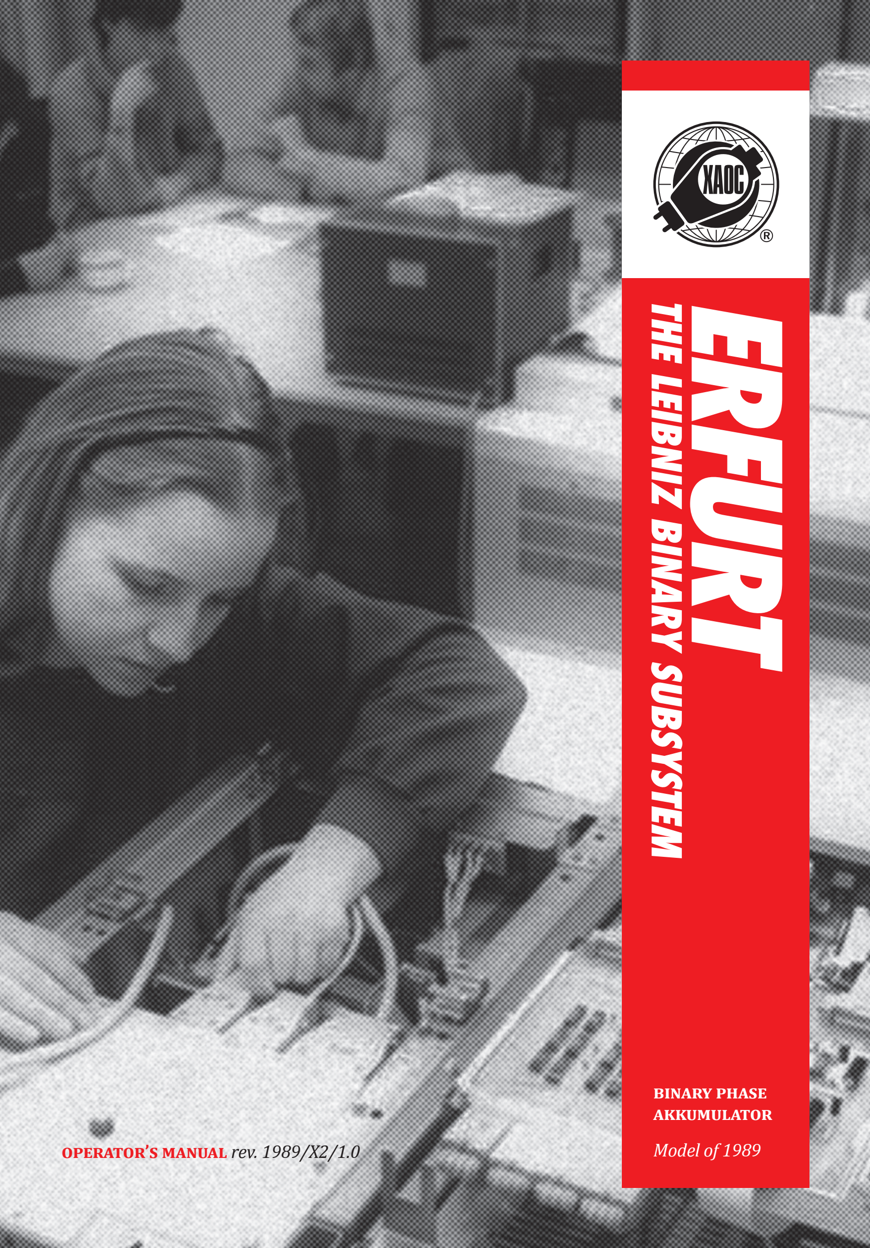


ERFURT

THE LEIBNIZ BINARY SUBSYSTEM

BINARY PHASE AKKUMULATOR

Model of 1989

OPERATOR'S MANUAL *rev. 1989/X2/1.0*

SALUT

Thank you for purchasing this Xaoc Devices product. Erfurt is a bi-directional digital counter, frequency divider, and a component of our 8-bit Leibniz Binary Subsystem. It has multiple applications: the output of the counter may be used as a phase source driving a digital wavetable, connected to other Leibniz modules it may scan waveshapes in Jena, produce stepped voltages useful for making interesting glissandi with Drezno and any VCO, generate gate patterns animating the spectrum of Odessa harmonic banks, spawn pseudo-chaotic sequences when fed back to Lipsk, etc.

It can also operate standalone (without other Leibniz components), and act as an 8-output clock and audio frequency divider.

Since Erfurt is based on CMOS logic (no CPU!), it can be clocked at very high frequencies (exceeding audio rate) without any aliasing or frequency interference. It also features its own internal HF clock source that may be used with clockless data sources.

INSTALLATION

The module requires 6hp worth of free space in the eurorack cabinet. Always turn the power off before plugging the module into the bus board using the supplied ribbon cable. Pay 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. Erfurt is internally secured against reversed power connection, however, rotating 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 particularly close attention to the proper orientation of your ribbon cable on both sides!*

Besides power, you may want to connect Erfurt to other components of your Leibniz Subsystem (unless you intend to use it standalone). Erfurt comes with one 10-pin ribbon data cable to connect with other Leibniz modules alongside any additional cables included with other modules in the subsystem. The 10-pin unshrouded header labeled **OUT** *is where you need to plug the cable that goes to the* **IN** *header of your next Leibniz module (e.g. Drezno, Jena, or Odessa). The incoming data from a module preceding Erfurt (e.g. Ostankino, Drezno, Lipsk, etc.) should be plugged into the 10-pin unshrouded header labeled* **IN.** *Please observe the marks of pin #1 (red stripe) on all connected modules. Depending on the intended use of Erfurt, these connections may not be necessary. We advise you to plan ahead as to how you wish to incorporate Erfurt into your modular setup.*

A jumper labeled **CLK SRC** *selects whether Erfurt runs with its own internal clock generator or uses the clock signal delivered with the Leibniz data. Put the jumper in the* **INT** *position when there is only a static data source (e.g. Lipsk or Ostankino), or when you don't feed Erfurt any data via the Leibniz interface. Set the jumper to* **EXT** *when the preceding module delivers clock alongside its data (usually changing over time), like Drezno or Jena.*

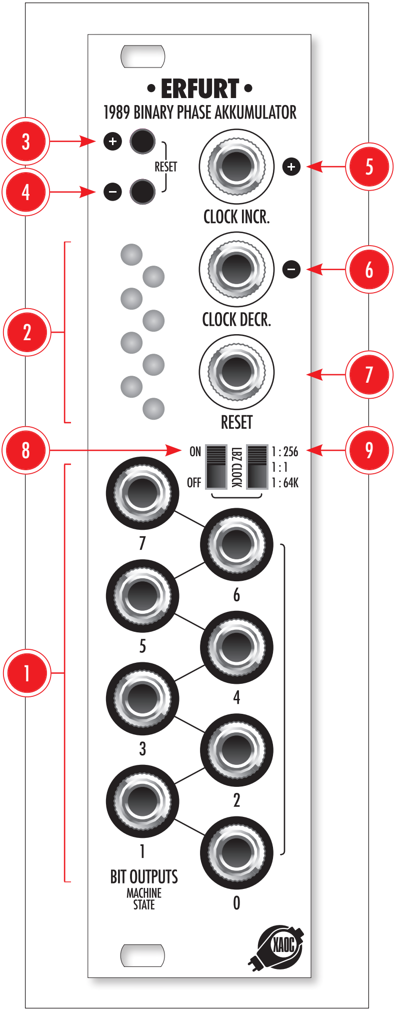


fig. 1

NOTE: DO NOT PLUG A POWER CABLE TO THE LEIBNIZ HEADERS. *This will destroy 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. To better understand the device, we strongly advise the user to read through the entire manual before use.*

MODULE OVERVIEW

The front panel of Erfurt is shown in fig. 1. As with other modules from the Leibniz series, Erfurt features a bank of eight binary outputs ❶ producing 5V gates, where an active gate signal represents logic 1 and an inactive gate represents logic 0. There are eight corresponding yellow LEDs ❷ that display the activity of individual bits of the digital number indicating the internal state of the counter.

The two push buttons allow for manual incrementing ❸ and decrementing ❹ of the state (increasing and decreasing by a predefined value, default to 1). The state can also be reset to zero by simultaneously pressing both buttons. There are three clock inputs for incrementing ❺, decrementing ❻ and resetting the state ❼. All inputs accept arbitrary modular signals (including those of audio rate and above), but are best fed with clocks, gates or triggers.

The two switches allow the Leibniz clock to drive the counter when there is nothing plugged into the **INCR CLOCK** *input ❺. The leftmost switch ❽ turns the Leibniz clock on and off, and the rightmost* switch ❾ offers three frequency pre-scaling options: 1:256, 1:1, and 1:64K (in other words: 256x256 which gives 65,536). With a typical Drezno clock of approximately 2MHz, this yields about 7kHz (high audio rate) in the upper position and about 28Hz (LFO rate) in the lower position. **NOTE:** The switches have no effect when you plug your own clock to the input ❺.

PRINCIPLE OF OPERATION

The heart of Erfurt (fig. 2) is an 8-bit register that holds the internal state of the machine and an adder that combines the internal state with the incoming data value. At every rising edge of the clock signal, the output of the adder is written back to the register. The choice of the clock input (incrementing or decrementing) decides whether it is a sum or difference, though two clocks may be used simultaneously. If two edges arrive at exactly the same time, the incrementing clock has higher priority.

The Leibniz clock that arrives together with the data through the ribbon data cable may be used to drive the counter when you set the leftmost **LBZ CLK** switch to the **ON** position. Since this clock is often very high frequency, it may be conveniently pre-scaled by 1:256 or 1:65,536 via the rightmost switch. This (divided or not) Leibniz clock is always used for counting up, therefore it is normalled to the **INCR CLOCK** input and is replaced by anything patched there.

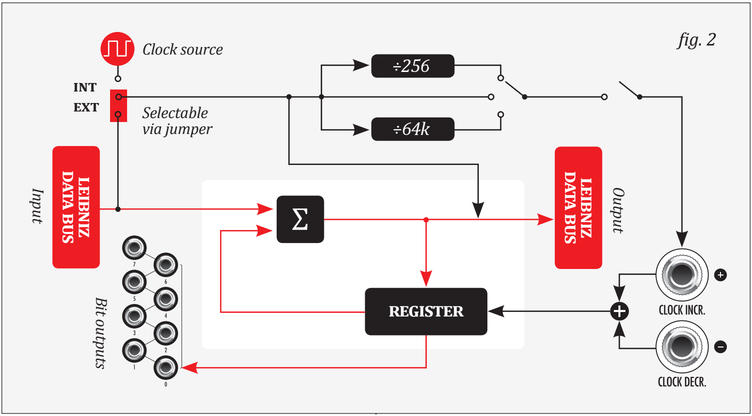
Erfurt also features an internal high frequency (near 2MHz) clock (selected using a jumper at the back) that is useful when the unit is operating standalone or connected to a static Leibniz data source like Lipsk (provided nothing is connected to Lipsk's Leibniz **IN** header).

The Leibniz **OUT** header at the back taps directly from the adder before the data is written back to the register. This allows Erfurt to pass the potentially quickly changing data to subsequent Leibniz modules in the chain (e.g. samples of a signal from Drezno to Jena). The data is being offset (modulo 256) by the content of the register regardless of whether the content stays still, is updated slowly or rapidly (according to the clocks plugged into the front panel jacks), or is being adjusted manually.

COUNTER VS FREQUENCY DIVIDER

When Erfurt is operating without any Leibniz data source, the **IN** header at the back should be left unplugged. There is a normalization on the header, enforcing the value of 1 (binary: 00000001) being fed to the adder. Therefore, Erfurt changes its state by 1 at each impulse from the clock input so that it counts from 0 to 255. When the value exceeds 255 it flips back to 0. In other words, the counter operates modulo 256. Obviously, with a clock plugged into the decrementing input, Erfurt similarly counts down from 255 to 0.

Taking signals out of the individual bit jacks offers the clock frequency divided by 2 (bit 0), 4 (bit 1), 8 (bit 2), 16 (bit 3), 32 (bit 4), 64 (bit 5), 128 (bit 6), and 256 (bit 7). Thus, in this configuration, Erfurt is a classic octave frequency divider.



With two different clocks patched to the inputs, the counter alternately increases and decreases its state, and ultimately counts with a rate proportional to the frequency difference between the clocks.

GENERATING MORE COMPLEX PATTERNS

When there is a Leibniz data source connected at the back (even one as simple as Lipsk), the state of the counter may increase or decrease in larger steps, depending on the data present at the Leibniz bus (e.g. the value programmed by the Lipsk buttons). In that case, the counter overflows quicker and produces patterns that are more complex than regular 4/4 because the remainder from overflow adds to the next cycle. In this case, the simple octave dependence between rates of individual outputs is no longer valid.

When the **DECR CLOCK** *input is used and certain non-trivial input values are programmed with Lipsk, Erfurt becomes a generator of intrinsic rhythms that may be recognized from dance music genres. It is due to the way many rhythmic patterns are defined, which is through various divisions of the common length by different integer factors. These divisions have remainders that contribute to inspiring off-beat incidentals, not unlike the popular Euclidean rhythms. The rule of thumb is that most elaborate rhythms occur when 256 does not divide evenly by the input value N, and the remainder has several non-zero bits or is a prime number. For example, an input value of N=00011011 (which is an 8-bit equivalent of 27) is not an integer divisor of 256. The remainder is 00001101 (which is an 8-bit equivalent of 13) – a prime number. Thus, this combination of bits yields an interesting rhythm.*

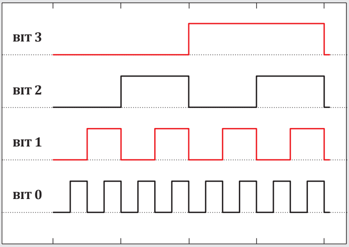


fig. 3a **MATHEMATICAL CLOCK DIVISION**

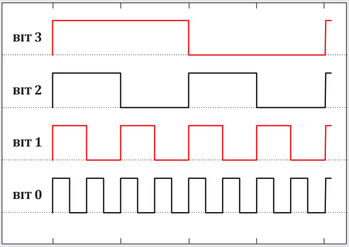


fig. 3b **MUSICAL CLOCK DIVISION**

When a binary counter counts upwards with an increment of 1, the cycle at each output starts with logic 0 in the first half, and proceeds to logic 1 in the second half (fig. 3a). Each completed cycle (going back from 1 to 0) causes the output directly above to advance by half a cycle. Thus, the first state of the counter is binary 00000000, and the last is binary 11111111, which is called mathematical division. After the very last state (11111111 is an 8-bit equivalent to 255) the return to 0 of the lowest bit spawns a quick transition to 0 of all the bits above it.

Counting downward (with a decrementing clock) reverts the situation: at each output, the cycle starts with logic 1 and ends with logic 0 in the second half (fig. 3b). Each completed cycle (going back from 0 to 1) causes the next bit above advancing by half of its cycle. Thus, the first state of the counter is binary 11111111, and the last one is binary 00000000, which is called musical division. After the very last state, the jump to 1 of the lowest bit spawns a quick transition to 1 of all the bits above it (transition from binary 0000000 to binary 11111111).

Erfurt offers both options thanks to its two clock inputs that are active simultaneously. For example, by using the **DECR CLOCK** *input you get a very straightforward 4/4 rhythm on 8 levels (e.g. from 1/32nd up to 4 bars if your source clock corresponds to 1/64ths). Also, if there is any slip in the sequence, you can nudge it forward or backward by using the* **DECR** *and* **INCR** *push buttons respectively.*

Even more complex and less predictable sequences of bits may be obtained by patching feedback from Erfurt's binary outputs to the inputs of the module preceding Erfurt in the Leibniz chain. In such a scenario, the increment value for each consecutive step is being modulated (in a nonlinear way!) by the previous Erfurt counter state.

Such dependence sometimes yields simple cyclic behavior, but it may also produce a pseudo-chaotic sequence. The maximum length of such a sequence *with one Erfurt is 256, but if your goal is total chaotic madness, it may be disrupted by feeding more data to the Leibniz* **IN** *port of Lipsk.*

PATCH IDEAS

• *A Xaoc Moskwa II sequencer connected to its Ostankino II expander can act as a source of Leibniz data. Each Moskwa II sequence step generates an 8-bit value corresponding to the position of the CV potentiometer. Connect Ostankino II's Leibniz* **OUT** *to Erfurt's Leibniz* **IN**, *and then Erfurt's* **OUT** *to Jena's* **IN** *and then – via Jena's* **OUT** *– to Drezno's* **IN**. *Use Erfurt's internal clock, and set the division to 1:256. This will produce a digital stepped ramp in Erfurt (so-called phasor) which can be waveshaped and modulated by Jena and converted to an audio signal by Drezno. Set Jena to* ASYNCHRONOUS MODE *to obtain nicely aliased waveforms resulting from the skipped samples. Use* SYNCHRONOUS MODE *for cleaner results.* **NOTE:** *Moskwa will affect the pitch linearly (not V/oct).*

• *If you set up a chain consisting of Lipsk followed by Erfurt (whose Leibniz output is then fed to the input of Drezno) you may generate interesting CV glissandi that can drive a VCO. Use an LFO to clock the steps and program the width and direction of the steps with Lipsk's buttons. Experiment with using both of Erfurt's clock inputs simultaneously. To limit the length you can use feedback from a selected binary output of Erfurt connected to its own* **RESET** *input. You can also limit the number of bits in Drezno's D/A conversion by patching dummy cables into the* **BIT INPUTS** *to make the quantization steps wider without affecting the speed of the glissando. Alternatively, you can unlink the normalization in Drezno, and patch only a few of the highest* **BIT INPUTS** *to the front panel.*

• *Apart from pitch sequences, the Ostankino II and Erfurt chain may be used to animate Odessa's spectrum. Connect the* **OUT** *header of Erfurt to Odessa's Leibniz* **IN** *header, and activate Odessa's Leibniz connection by holding the* **VOICES** *button until it starts blinking. Remember that the effect very much depends on Odessa's bank length and number of audible overtones.*

• *Lipsk with Erfurt can produce a variety of rhythmic patterns that may trigger drum modules or fire envelopes, but you can also try the special drum bank in Jena. Please consult the pattern guide available on our website (Jena shapes chart). For the best results, use Jena's* SYNCHRONOUS MODE *and special drum* **BANK E**. *Connect Lipsk to Erfurt and then Erfurt to Jena and use the individual bit outs from Jena as triggers. With the Leibniz clock divided by 64K, the value programmed on Lipsk will be your tempo multiplier.*

ACCESSORY

Our Coal Mine black panels are available for all of 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-INFLICTED FAULT.

LEGACY SUPPORT

IF SOMETHING GOES WRONG WITH A XAOC PRODUCT AFTER THE WARRANTY PERIOD IS OVER, 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 AND DISTRIBUTION TERMS, 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

RADA
WZAJEMNEJ
POMOCY GOSPODARCZEJ

MADE IN THE EUROPEAN UNION

MAIN FEATURES

Leibniz Binary Subsystem component

Phase accumulator

Programmable digital oscillator

Bi-directional binary counter

Clock and audio frequency divider

TECHNICAL DETAILS

Eurorack synth compatible

6hp, skiff friendly

Current draw: +25mA/-30mA

Reverse power protection