maths

_travelers guide _for counting to infinity

4 channels	3 bus outs	25 min to 1 khz

[cycle] press · patch › then noon = 12 o'clock cw / ccw = clockwise / counter

an analog computer for music · envelopes, lfos, slew, offsets, logic, mixing

maths 2013 · 20 hp · no menus, no hidden functions

LEDs: green = + · red = –

01 channels 1 + 4

mx_01 · the function generators

[signal in]	slews whatever you patch · lag, portamento, a gate = rise / sustain / fall · ±10 v · also feeds the bus
[trig]	fires a 0 → 10 v rise then fall, no sustain · retriggers only while falling
[cycle]	loops it: an lfo · LED shows on · **[cycle in]** gate high (2.5 v+) does the same
[rise] [fall]	cw = slower · together they set the period
[vari-response]	log ‹ linear (the tick) › expo › hyper-expo (fig.01)
[unity out]	straight from the core, not on the bus · 0–8 v cycling · LED ● + ● –

02 control inputs

mx_02 · ±8 v

[rise cv] **[fall cv]**	linear · + = longer, – = shorter, around the knob
[both cv]	exponential, and backwards: + = faster, – = slower, the whole function
need more range?	add an offset from ch 2 or 3 into the cv jack
need attenuation?	run the cv through ch 2 or 3 first

03 pulse outs

mx_03 · 0 or 10 v

[eor] ch 1	goes high at the end of the rise · low at rest · pulse delay, clock division
[eoc] ch 4	goes high at the end of the fall · **high at rest**, LED on when idle
chain them	eor / eoc into the other channel's trig, signal or cycle input (fig.04)

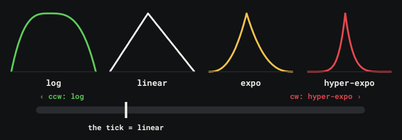
04 channels 2 + 3

mx_04 · scale · invert · offset

[2 in]	nothing patched = a **±10 v** offset from the knob
[3 in]	nothing patched = a **±5 v** offset from the knob
[attenuverter]	noon = zero · cw = more, up to amplifying · ccw = upside down

__fig.01 : vari-response

four curves



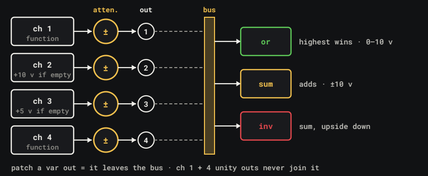
05 the bus

mx_05 · fig.02

[1] [2] [3] [4]	variable outs, after each attenuverter · each feeds sum and or until you patch it
[sum]	adds them, ±10 v · **[inv]** = the same, upside down
[or]	the highest voltage wins, 0–10 v · ignores negatives, so it rectifies
unused 2 / 3	set to noon or plug a dummy cable, or their offset rides the bus

__fig.02 : the bus

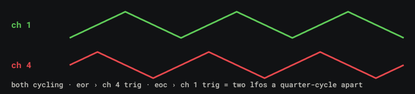
four channels, three answers



patch a var out = it leaves the bus · ch 1 + 4 unity outs never join it

__fig.06 : 281 mode

quadrature lfos



both cycling · eor › ch 4 trig · eoc › ch 1 trig = two lfos a quarter-cycle apart

10 the module

mx_10 · 2013

size	20 hp · 60 mA +12 v · 50 mA –12 v
2013 vs original	unity outs, inv out, bus + eor / eoc LEDs, cycle in, ±10 v offset on ch 2

07 shapes

mx_07 · one channel

triangle lfo
[cycle] on · rise, fall noon · linear · ch 2 into **[both]** sets the rate
unity or variable out

ramp / saw
same, rise full ccw, fall noon or more

attack / decay
gate or trigger into **[trig]**
retriggers while falling

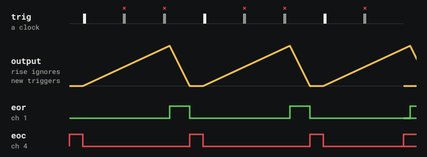
attack / sustain / release
gate into **[signal in]**: rises to the gate level, holds, falls when it ends

clock / pulse
cycle it, take **[eor]** or **[eoc]** · **[cycle in]** = run / stop
ch 1: rise = rate, fall = width · ch 4: the reverse

pulse delay
gate into ch 1 **[trig]**, take **[eor]**
rise = delay · fall = width

__fig.04 : counting clocks

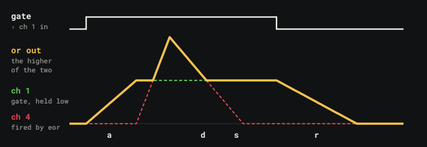
trig · eor · eoc



one eor pulse per accepted trigger · here every third clock gets in

__fig.05 : an adsr from or

ch 1 + ch 4 › or



ch 4 fall = decay · ch 1 knob = sustain level · ch 1 fall = release · set 2 + 3 to noon

06 time

mx_06 · 25 minutes to 1 khz

slowest	rise and fall full cw, log response, plus cv: "slow-ver-drive"
fastest	extreme expo response, knobs ccw: audio rate
response	moving vari-response changes the times too
one rule	maths can slow a signal down but never speed it up

08 two channels

mx_08 · patches from the manual

adsr
gate › ch 1 **[signal in]** · ch 1 **[eor]** › ch 4 **[trig]** · take **[or]**
ch 4 fall = decay · ch 1 knob = sustain (below ch 4) · ch 1 fall = release

281 mode
both cycling · **[eor]** › ch 4 trig · **[eoc]** › ch 1 trig
two lfos 90° apart · nudge ch 1 cycle if they stall

bouncing ball
ch 1 rise ccw, fall 3:00 · ch 4 rise ccw, fall 11:00, linear · **[eor]** › ch 4 cycle in · ch 1 var out › ch 4 fall cv
trig ch 1 to drop the ball · ch 4 out › vca or lpg

arcade trill
ch 4 cycling, expo · **[eoc]** › ch 1 trig and ch 2 in · ch 2 (10:00) › ch 1 both
ch 1 rise noon, fall ccw · tiny ch 1 rise moves = big changes

independent contours
ch 1 var out back into its own **[rise cv]** or **[fall cv]** · its knob shapes that slope alone
vari-response at noon · take the unity out

pseudo-vca
audio into ch 1, rise + fall ccw · ch 2 cw (+ ch 3) until silent · envelope on ch 4, knob ccw › **[sum]**
asymmetrical clipping · try inv

trilling envelope
ch 1 **[eor]** › ch 4 **[cycle in]** · ch 4 rise 1:00, fall 11:00, expo · take **[or]**
trills on the fall · swap channels to trill on the rise

09 utilities

mx_09 · maths as a computer

clock divider
clock into **[trig]** · longer rise ignores more clocks
fall = width · too wide and it stays high

envelope / peak follower
audio into **[signal in]** · rise noon (or 3:00 for peaks) · fall sets the response
or out for a positive follower

rectify
half wave: any input › **[or]** · full wave: same signal into 2 and 3, 2 cw, 3 ccw › **[or]**

voltage mirror
cv into 2, knob full ccw · 3 empty, knob full cw › **[sum]**

gate extractor
signal into 2 · 3 negative = threshold · **[sum]** › ch 1 in, rise + fall 0 › **[eor]**
rise = how long above threshold · fall = gate length

logic inverter
gate into ch 4 **[signal in]** › **[eoc]**

flip-flop
set › ch 1 trig · reset › ch 1 both · rise ccw, fall cw, linear › **[eoc]**
forgets after about 3 minutes

__fig.03 : front panel

ch 4 mirrors ch 1



- **[01] __ch 1 in :** slew · ±10 v
- **[02] __trig :** fire a function
- **[03] __cycle :** loop · LED
- **[04] __rise :** up time
- **[05] __fall :** down time
- **[06] __vari-resp. :** log ‹› expo
- **[07] __rise cv :** + = longer
- **[08] __both cv :** + = faster
- **[09] __fall cv :** + = longer
- **[10] __cycle in :** gate = loop

- **[11] __1–4 :** attenuverters
- **[12] __ch 2 in :** +10 v if empty
- **[13] __ch 3 in :** +5 v if empty
- **[14] __var outs :** feed the bus
- **[15] __or · sum · inv :** the bus
- **[16] __eor :** end of rise
- **[17] __unity :** ch 1 core out
- **[18] __eoc :** end of cycle
- **[19] __ch 4 :** mirror of ch 1

_where things hide

in the normals · mx_11

hidden voltages

- **ch 2 empty** +10 v offset
- **ch 3 empty** +5 v offset
- **var outs empty** all feed sum / or / inv

easy to miss

- **eoc** high at rest, low while busy
- **both cv** runs backwards: + = faster
- **trig** deaf during the rise

LED codes

unity · sum	● positive ● negative brightness = level
cycle	● cycling
eor · eoc	● pulse high