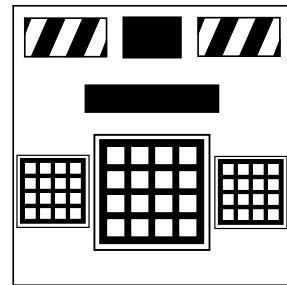


On the Malfunctioning of RGB Logic

*"Don't sit on it, don't sit on it, don't sit on it!" *module cracks**

When the module activates, the two small screens on the sides of the module will each show a 4×4 grid of squares.

The grids can have any amount of squares of each RGB colour channel. These channels additively mix if they overlap.



The screen will display a logical statement with two letters:

- Lowercase letters refer to the left screen.
- Uppercase letters refer to the right screen.
- Each square returns true if its colour contains the primary colour indicated by the letter and returns false otherwise.

Select all the squares in the middle grid whose corresponding squares in the other two grids satisfy the displayed condition.

Avoid selecting any squares that fail the displayed condition, or missing any squares that satisfy the displayed condition.

Squares that are missed will be briefly shown upon deactivation.

Even on the first activation, this module will glitch out and alter its functionality. Multiples of these recorded glitches can occur at once.

- **Needy Timer:** The needy timer will stop displaying the proper time counting down at a certain time. The actual time remaining can still be interpreted by its full duration of 99 seconds. Its 5 second warning will still play normally.
- **Alternating Coloured Square(s):** Two different colours will be displayed on a particular square. XORing the two colours by applying an XOR operation on each channel will result in its true colour.
- **Randomly Flashing Coloured Square(s):** One of every possible colour will all be displayed in a particular square. If a logical statement can be deducted without those particular squares, it must do so. Otherwise, the squares of the center grid in the same position as these squares should never be pressed.
- **Off-Coloured Logical Statement:** Logic statements purely in white expect satisfied squares, and unsatisfied squares otherwise.
- **Shifted Logical Statement:** Its shift can be undone to obtain the original logical statement. Inversions of any boolean states are always in the form " !X. "
- **Faulty Strike Handling:** This needy only truly strikes once per activation. Fake strikes will trigger for each infraction after the first.

Appendix: Logical Symbols

Symbol	Logic Gate	Operation		
		A	B	$A \star B$
\wedge	AND	F	F	F
		F	T	F
		T	F	F
		T	T	T
\vee	OR	F	F	F
		F	T	T
		T	F	T
		T	T	T
$\underline{\vee}$	XOR	F	F	F
		F	T	T
		T	F	T
		T	T	F

Symbol	Logic Gate	Operation		
		A	B	$A \star B$
$\bar{\wedge}$	NAND	F	F	T
		F	T	T
		T	F	T
		T	T	F
$\bar{\vee}$	NOR	F	F	T
		F	T	F
		T	F	F
		T	T	F
\leftrightarrow	XNOR	F	F	T
		F	T	F
		T	F	F
		T	T	T

Appendix: Colour Reference

Colour	Visual	Channel		
		R	G	B
Black		F	F	F
Blue	B	F	F	T
Green	G	F	T	F
Cyan	C	F	T	T
Red	R	T	F	F
Magenta	M	T	F	T
Yellow	Y	T	T	F
White	W	T	T	T

Appendix: Logic Gate Identities

Fixed states are written in its full form, I.E. T is represented as TRUE.

Involution	$!!A = A$
Dominance (OR)	$\text{TRUE} \vee A = \text{TRUE}$
Dominance (AND)	$\text{FALSE} \wedge A = \text{FALSE}$
DeMorgan's (OR)	$A \vee B = !A \bar{\wedge} !B$
DeMorgan's (AND)	$A \wedge B = !A \bar{\vee} !B$