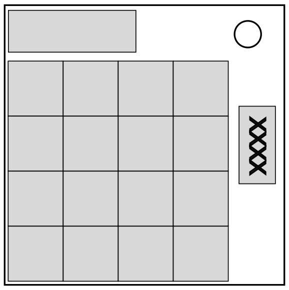
On the Subject of Binary Squares



You'll be sick of binary after this.

The module features a grid of 16 colored squares. They are initially colored randomly and to solve the module, a final configuration of colors must be reached. The possible colors include red, blue, lime, yellow, magenta, cyan, white, and black.

To apply a filter, first create a modified filter by XORing the current grid's pixel colors by the respective pixels in the filter. Then, apply the operator of each color (operators found under Appendix A) in the modified filter to the current grid's respective squares.

Follow all the rules below until you either reach the end of the list or you have followed three rules. For each of the true statements, modify your color grid accordingly.

1. If there are more than four white pixels, apply Filter 1.
2. At this point, if the top-left pixel is either black, white, or blue, apply Filter 2.
3. At this point, apply Filter 3.
4. At this point, if the sum of the digits in the serial number is odd, apply Filter 4.
5. At this point, if there are more green squares on the top two rows than the bottom rows, apply Filter 5.
6. At this point, apply Filter 6.
7. At this point, if the serial number has a letter from A to J, apply Filter 7.
8. At this point, apply Filter 8.

Click each square to change its color. To submit your answer, press the button on top of the grid. If you have 4 or fewer mistakes, you won't get a strike. The module will then display which squares are wrong using the base-7 number system. The digits are as follows:

0	1	2	3	4	5	6

Appendix A: Color Operators

Red: Use XOR.	**Blue: Use OR.**	**Green: Use XOR.**
Magenta: Use XNOR.	**Cyan: Use AND.**	**Yellow: Use XNOR.**
White: Use AND.	**Black: Use OR.**	

Appendix B: Binary Operations with Colors

Since a color's red, green, and blue values are either at 100 percent or 0 percent, a color can be represented by a binary sequence. To do binary operations with this binary sequence, use the binary operator on each respective bit to produce the output bit. For example, red XOR yellow can be represented as 100 XOR 110. This comes out to be 010, or blue.

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

A	B	A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1

Appendix C: Filters

Filter 1

R	C	B	R
Y	R	G	B
G	K	R	G
R	W	M	R

Filter 2

Y	Y	W	C
G	B	Y	C
B	G	W	R
M	M	B	R

Filter 3

K	K	K	W
M	M	M	M
G	G	C	R
B	M	G	R

Filter 4

Y	C	Y	C
R	W	R	W
Y	C	Y	C
R	W	R	W

Filter 5

B	B	B	B
B	C	R	R
K	Y	C	G
W	Y	K	M

Filter 6

B	Y	G	R
W	M	M	C
M	K	R	B
K	G	B	R

Filter 7

M	G	G	Y
K	G	G	Y
K	W	M	Y
R	W	G	C

Filter 8

R	K	M	M
Y	B	Y	G
M	Y	G	B
W	R	G	Y