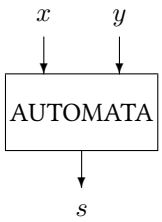
## Block AUTOMATA

The AUTOMATA block calculates an elementary cellular automaton ( https://en.wikipedia.org/wiki/Elementary_cellular_automaton ). The first row (x = 0) is random, every following row is calculated from the previous one with the given rule and periodic boundaries.




| | |
|---|---|
| **Name** | AUTOMATA |
| **Function** | fb0087 |
| **Inputs** | 2 |
| **Outputs** | 1 |
| **Parameters** | 2 |
| **Strings** | 0 |
| **Group** | S |

### Inputs

| | |
|---|---|
| 1 | Row x, from 0 on, incremented by 1 |
| 2 | Cell y in row x, between 0 and 1023 |

### Outputs

| | |
|---|---|
| 1 | State of cell y in row x, 0 or 1 |

### Parameters

| | |
|---|---|
| 1 | Wolfram code of the rule, between 0 and 255 |
| 2 | Seed, 0 for a different random first row at each run, 1 for a single bit in the middle of the first row |

### Strings

None



Each cell of an elementary cellular automaton is either 0 or 1. The cells of the first row ($x = 0$) are random, except with seed 1: the first row then has a single bit in its middle. Since the width of the row is only known once the first row is complete, this bit is set at

the beginning of row $x = 1$, so the output of the first row is all zeros. Every following row is calculated from the previous one: the new state of cell $y$ depends on the states of cells $y - 1$, $y$ and $y + 1$ in the previous row, with periodic boundaries. These three bits form a number $n$ between 0 and 7, and the new state is bit $n$ of the rule.

The width of the rows is defined by the first row, so the calls must loop over all cells $y$ of row $x$ before row $x + 1$, typically with an outer DO block for $x$ and an inner DO block for $y$.

Rule 110 is Turing complete, rule 30 looks chaotic, and rules 90 and 150 draw Sierpinski triangles. See https://en.wikipedia.org/wiki/Elementary_cellular_automaton.