

### Алгоритм простой итерации (2-ая модель сигналов).

Существуют две реализации алгоритма простой итерации: последовательная реализация, параллельная реализация. При последовательной реализации алгоритма простой итерации необходимо задать начальное состояние схемы, затем подать первый входной набор и вычислить значения всех выходов логических элементов схемы на данном наборе. Если результат текущей итерации не равен результату предыдущей итерации, выполняется следующая итерация. Процесс заканчивается при полном совпадении результатов двух соседних итераций либо при достижении заданного максимального числа итераций. Максимальное число итераций в данном случае равно длине максимальной цепочки логических элементов (включая ОС) плюс 1. Также признаком конца процесса моделирования может служить наличие в таблице моделирования записи результата текущей итерации, находящейся на месте, отличающегося от предпоследнего. При параллельной реализации одновременно моделируют несколько входных наборов. Число одновременно моделируемых наборов зависит от языка программирования и особенностей реализации алгоритма. Ускорение происходит за счет меньшего числа обращений к описанию схемы.

Количество тактов (проходов) схемы (максимальное количество итераций) = самый длинный путь + 1.

В данном алгоритме значение на любом элементе в схеме считается как функции, которая зависит от входного набора и значений элементов на предыдущей итерации:

$$Y_{ij}^k = F(x^k, Y_{i-1}), \text{ где}$$

$x = \{x_1, x_2, \dots, x_l\}$  - входной набор (множество входных сигналов);

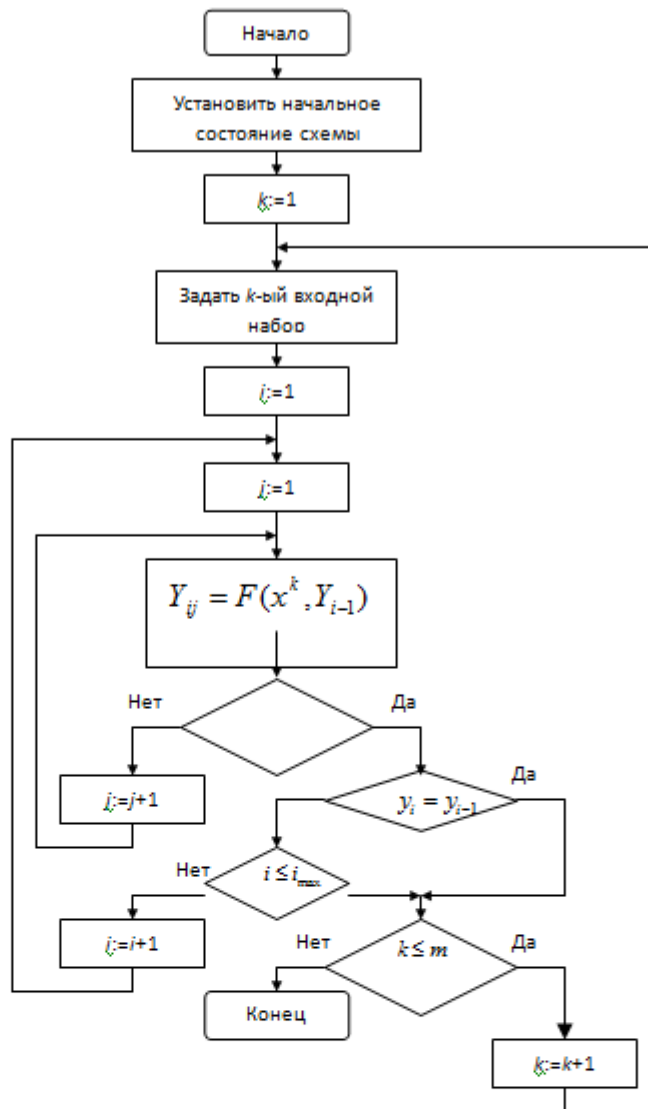
$Y = \{y_1, y_2, \dots, y_n\}$  - выходные сигналы схемы;

$j$  - номер элемента;

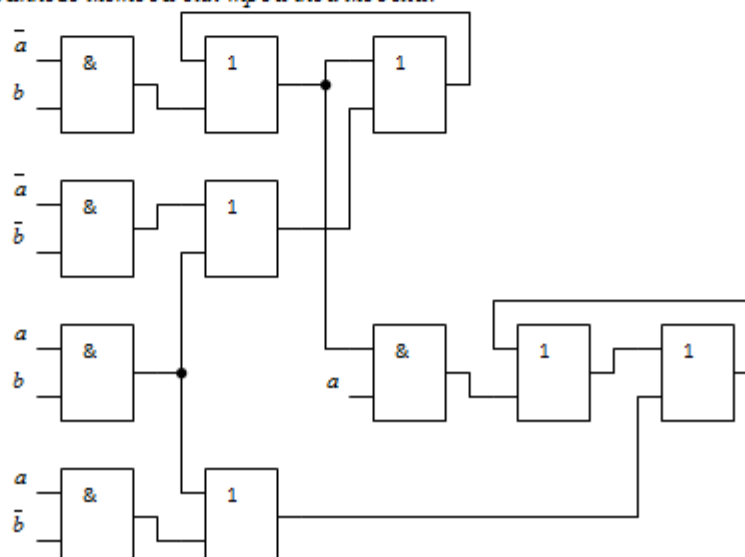
$i$  - номер итерации;

$k$  - номер входного набора ( $k = \overline{1..m}$ ).

Схема алгоритма



Пример использования данного метода для троичной модели.



Максимальное число итераций  $i_{\max} = 10$  (1-2-3-2-6-7-8-7-8 – самый длинный путь).

Для последовательности наборов:

00, 0x, 01, 11.

Для троичной модели начальное значение на элементах не определено.

$a$	$b$	1	2	3	4	5	6	7	8	9	10	11
-----	-----	---	---	---	---	---	---	---	---	---	----	----

		x	X	x	x	x	x	x	x	x	x	x
0	0	0	X	x	1	x	0	x	x	0	x	0
0	0	0	X	x	1	1	0	x	x	0	0	0
0	0	0	X	1	1	1	0	x	x	0	0	0
0	0	0	1	1	1	1	0	x	x	0	0	0
0	0	0	1	1	1	1	0	x	x	0	0	0
0	x	x	1	1	x	1	0	x	x	0	0	0
0	x	x	1	1	x	x	0	x	x	0	0	0
0	x	x	1	1	x	x	0	x	x	0	0	0
0	1	1	1	1	0	x	0	x	x	0	0	0
0	1	1	1	1	0	0	0	x	x	0	0	0
0	1	1	1	1	0	0	0	x	x	0	0	0
1	1	0	1	1	0	0	1	x	x	1	0	0
1	1	0	1	1	0	1	1	1	x	1	1	0
1	1	0	1	1	0	1	1	1	1	1	1	0
1	1	0	1	1	0	1	1	1	1	1	1	0

При использовании троичной модели в схеме можно обнаружить генераторы и состязания сигналов.

Если при походе схемы максимальное число итераций значения на некоторых элементах в схеме не установились, то данные элементы образуют генератор.

Если на одном входном наборе в схеме значение на выходе элемента принимало определенное значение, после чего было не определено, а затем приняло тоже значение, то на данном элементе идет состязание сигналов.