

Обзор алгоритмических функциональных тестов ОЗУ

Эффективность ФК решающим образом определяется построением теста. Существуют различные способы генерации тестовых последовательностей для контроля ЗУ.

Наиболее широко используются при контроле функционирования ЗУ алгоритмические функциональные тесты (АФТ), содержащие последовательность элементарных тестов, изменяемых по известному закону (алгоритму). Это связано с простотой генерации, малым объемом занимаемой памяти управляющей ЭВМ и высокой воспроизводимостью результатов ФК.

Эталонный сигнал выхода ЗУ вырабатывается, как правило, также алгоритмически генератором тестов, но можно использовать и эталонную схему ЗУ. АФТ должны обладать двумя противоречивыми свойствами: с одной стороны, обеспечивать достаточную полноту контроля БИС ЗУ, а с другой — быть достаточно короткими по времени, чтобы обеспечить производительность проверки БИС ЗУ при их большой информационной емкости и большом количестве БИС.

Внимание! Непосредственный перебор всех $2^{(N+K)}$ возможных состояний ОЗУ (N — число запоминающих элементов, бит; K — число функциональных входов) становится нереальным при $N > 64$. Поэтому алгоритмы ФК ЗУ имеют ограниченный набор входных тестовых комбинаций (циклов обращения), обеспечивающих обнаружение типовых отказов в дешифраторе и матрице памяти ОЗУ.

При разработке алгоритмов ФК ищется минимальная тестовая последовательность входных сигналов, для которой имеет место изменение выходной последовательности сигналов тестируемой схемы при отказе любого из ее элементов. Решение этой задачи осложняется наличием у БИС ОЗУ ряда неисправностей, не описываемых булевыми функциями (например, множественная выборка), а также связанных с динамическими состояниями элементов.

Ниже приводится ряд типовых алгоритмов ФК ЗУ, имеющих практическое применение. Коротко указано о применимости различных алгоритмов.

По количеству циклов обращения тестируемой схеме, выраженному через ее информационную емкость, алгоритмы ФК условно делятся на три типа: N , N^2 , $N^{3/2}$, где N — емкость ЗУ, бит. Линейные алгоритмы типа N используются, как правило, для предварительной оценки ОЗУ на отсутствие катастрофических неисправностей. Для производственного контроля ОЗУ из линейных тестов практически пригоден лишь «Марш», так как достоверность контроля другими линейными алгоритмами недостаточна. Квадратичные алгоритмы (типа N^2) зарекомендовали себя наиболее эффективными для контроля функционирования всех типов ЗУ. Попарные передачи информации между любыми парами элементов памяти позволяют эффективно обнаруживать как статические, так и динамические отказы ЗУ. Применение квадратичных алгоритмов ограничивается резким ростом длительности контроля с увеличением емкости ЗУ. Алгоритмы типа $N^{3/2}$ появились в результате поиска компромисса между длительностью и достоверностью контроля БИС памяти; они достаточно широко используются при контроле ОЗУ большой емкости.

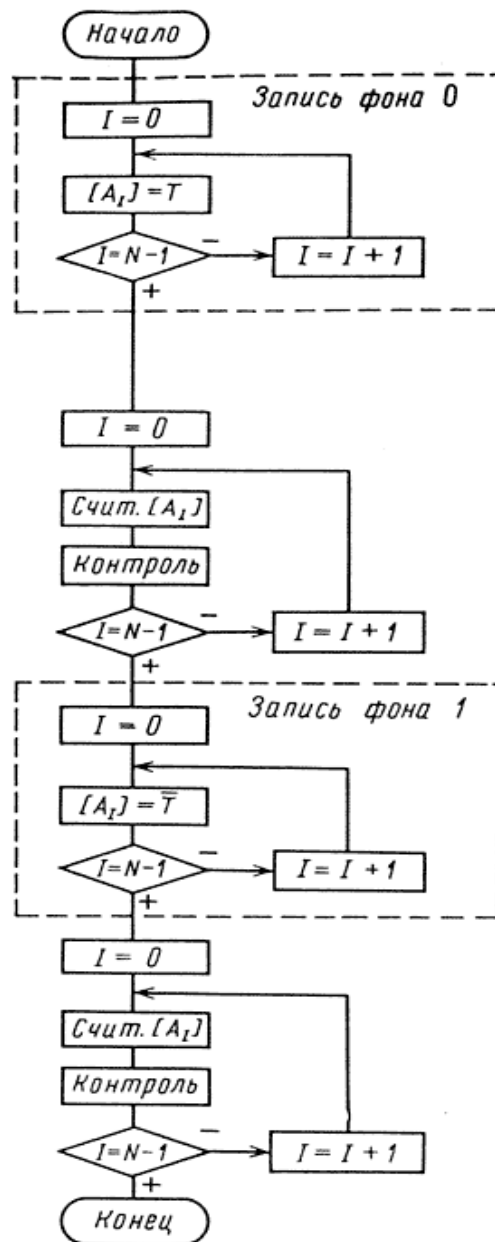
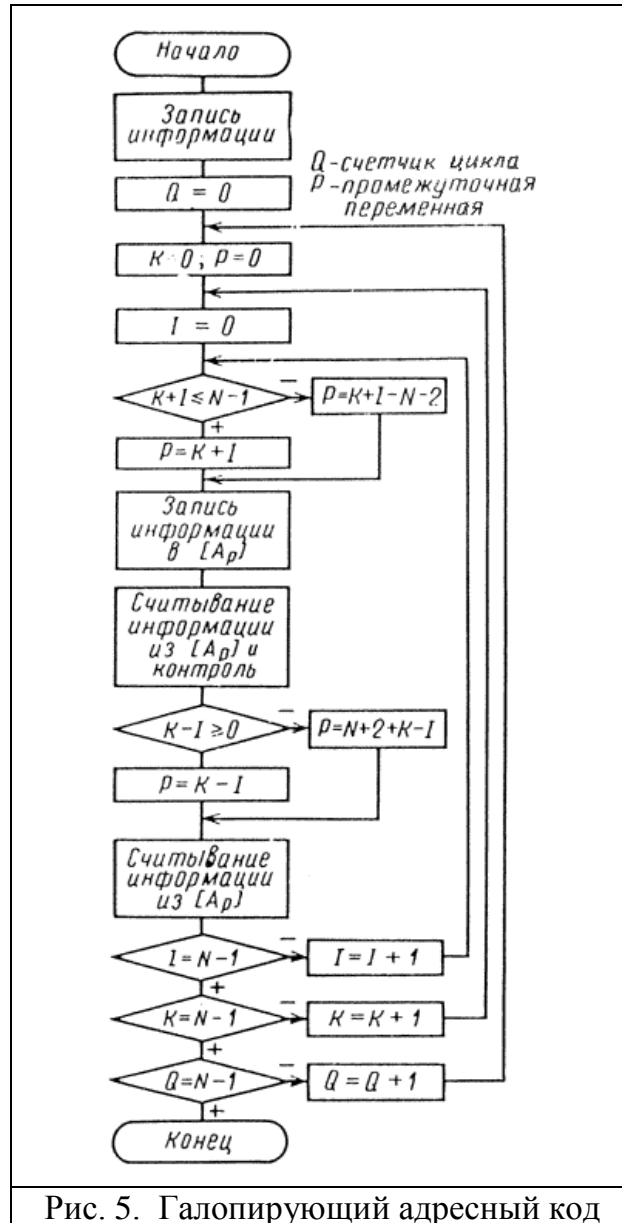


Рис.1. Алгоритм теста «Последовательная запись и считывание».

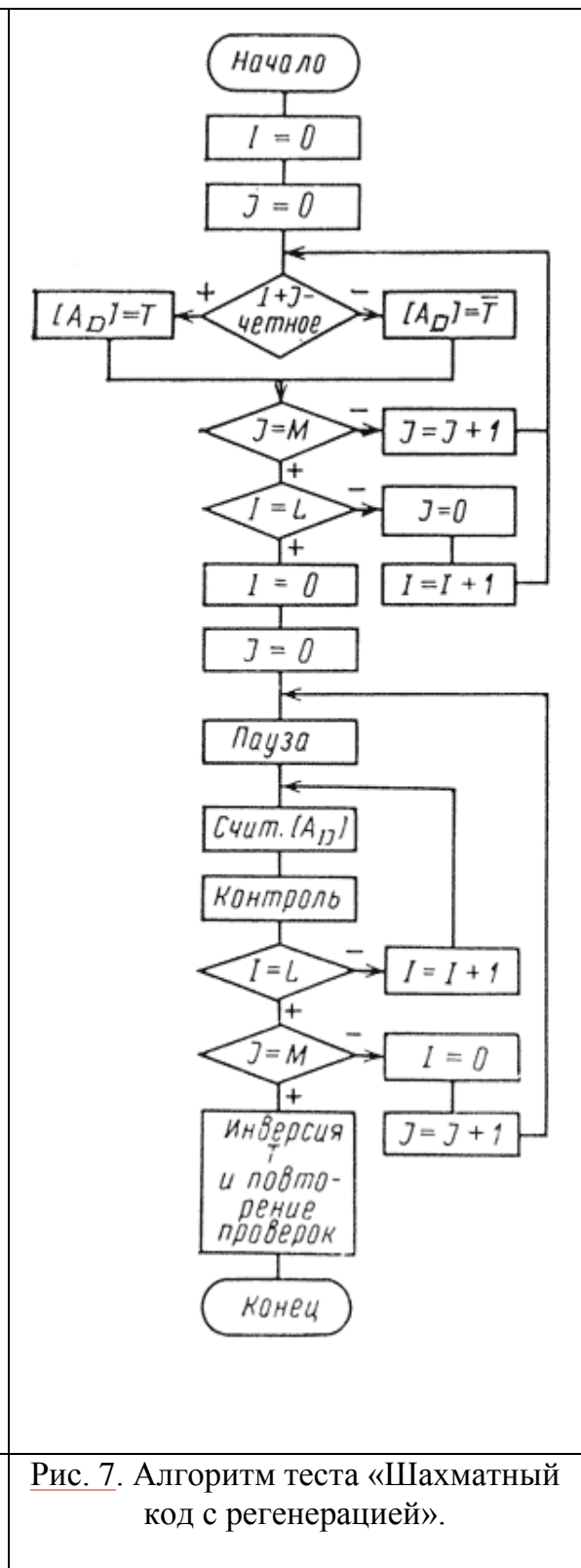
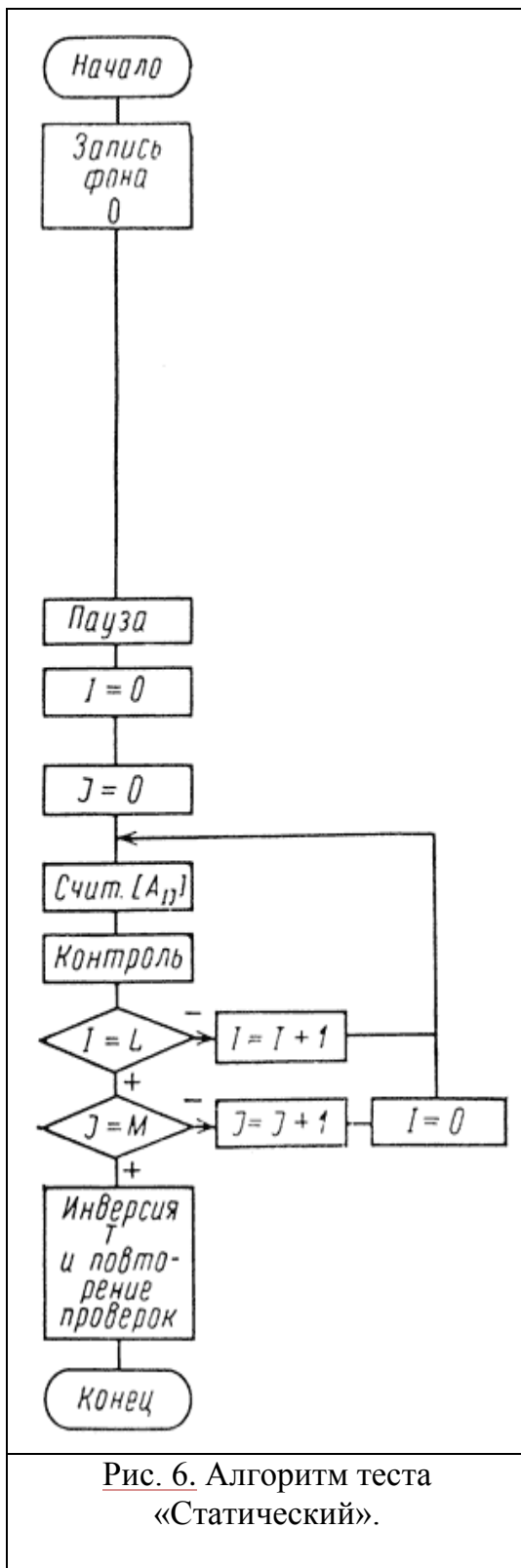
Здесь и далее принимается:

«Контроль» - сравнение считанной информации с эталонной; A_I - текущий адрес ячейки; A_D - дополняющий адрес ($A_D = A_{N-1} - A_I$); $[A_I]$ - содержимое ячейки с адресом A_I ; T - информация логического 0; \bar{T} - информация логической 1. В структурных схемах алгоритмов элементы матрицы памяти могут иметь либо один индекс I , изменяющийся от 0 до $N-1$, где A_K - контролируемый адрес, либо двойной индекс I, J , где I изменяется от 0 до \sqrt{N} (считаем, что матрица ЯП накопителя квадратная) по строкам матрицы, а J - от 0 до \sqrt{N} по столбцам матрицы. В этом случае A_{SR} - контролируемые адреса матрицы памяти, где S - по строкам, а R - по столбцам; A_{LM} - конечный адрес строки L и столбца M .

Тест «Марш» (см. рис. 1.2.6.5). Последовательно по всем адресам производится запись фона 0. Затем для каждого адреса считывается информация T и записывается \bar{T} при изменении от A_0 до A_{N-1} . Далее, начиная с адреса $A_1=A_0$ до $A_1=A_{N-1}$, для каждого адреса считывается информация \bar{T} и записывается T . Затем для каждого адреса считывается информация T и записывается \bar{T} при измерении адресов от A_{N-1} до A_0 (обратный перебор адресов). Далее для каждого адреса считывается информация \bar{T} и записывается T при изменении адресов от A_{N-1} до A_0 . Затем производится инверсия фоновой информации (запись фона 1) и цикл проверки повторяется.



Алгоритм теста «Галопирующий адресный код». Информация T изменяется для каждого текущего адреса и для каждого разряда и определяется как сумма в двоичном коде номеров цикла и адреса. Значение суммы записывается в воображаемый последовательный циклический разрядный регистр и считывается с разряда регистра, номер которого соответствует номеру цикла или кратен ему.



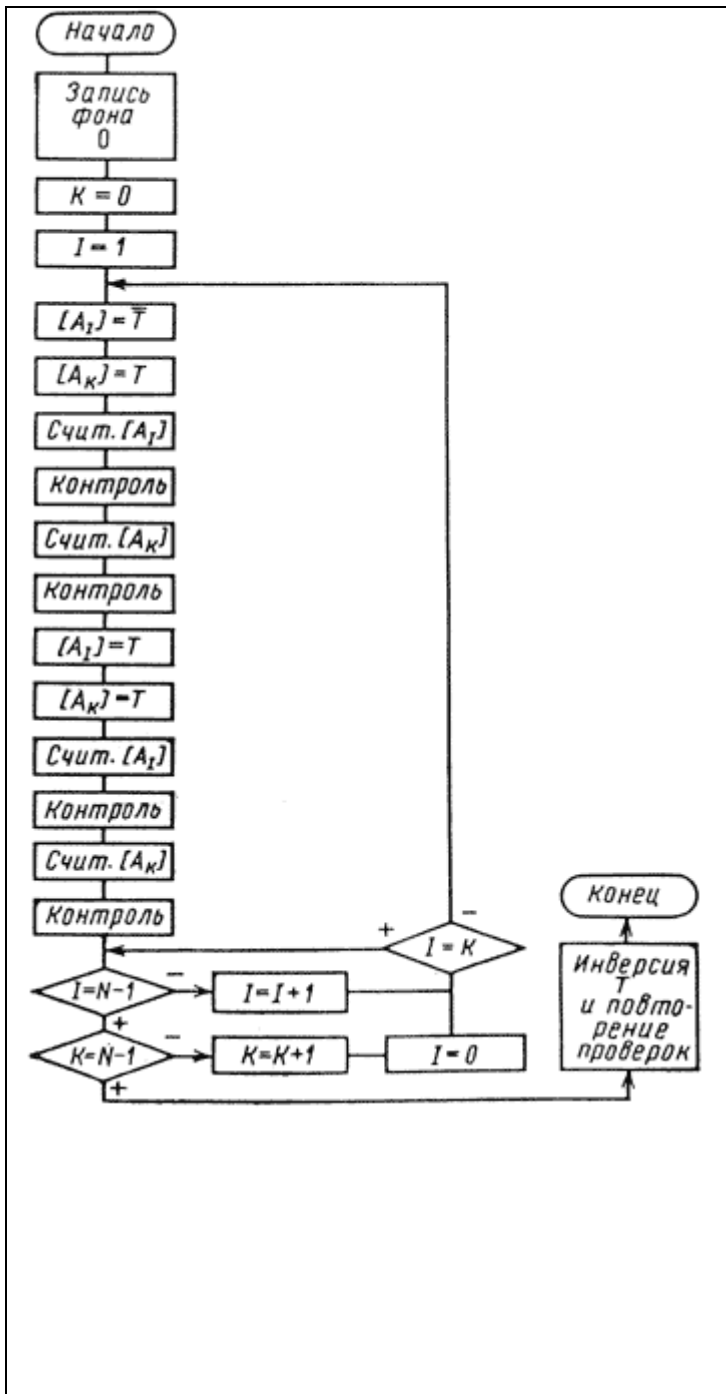


Рис. 8. Алгоритм теста «Попарная запись-считывание с полным перебором».

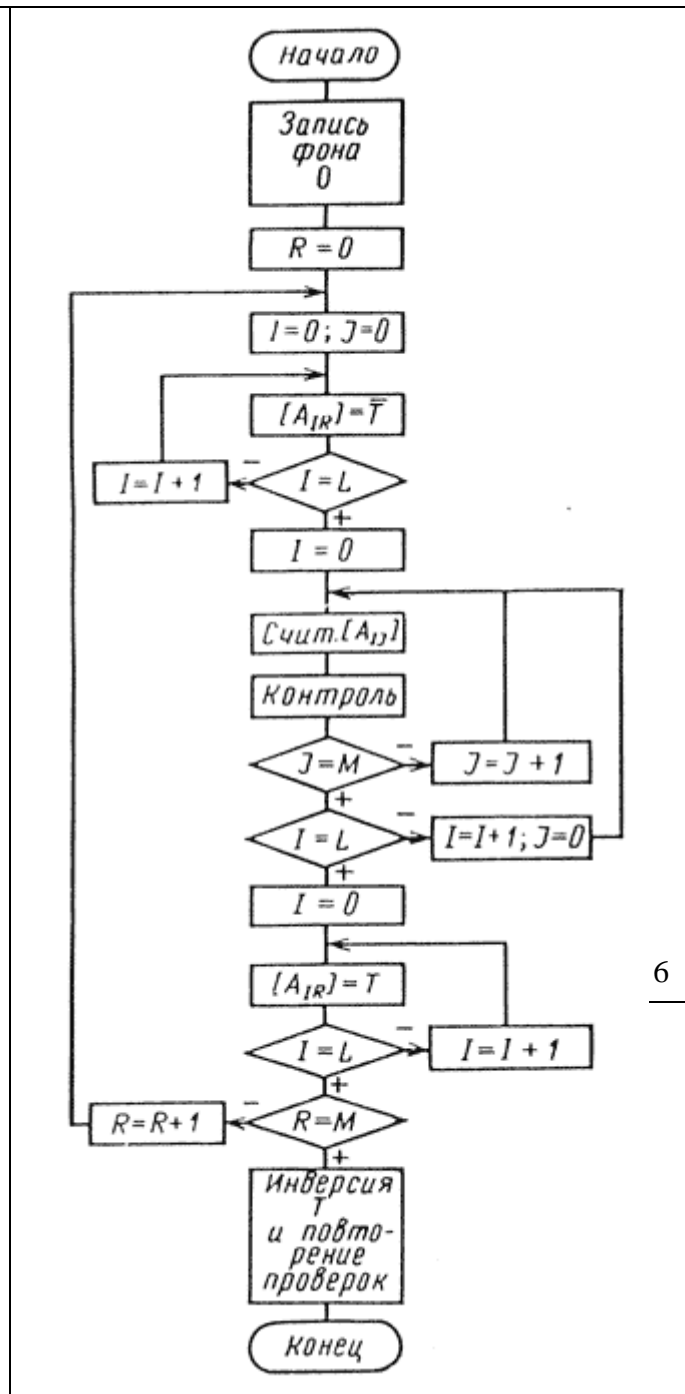


Рис. 9. Тест «Бегущий столбец».

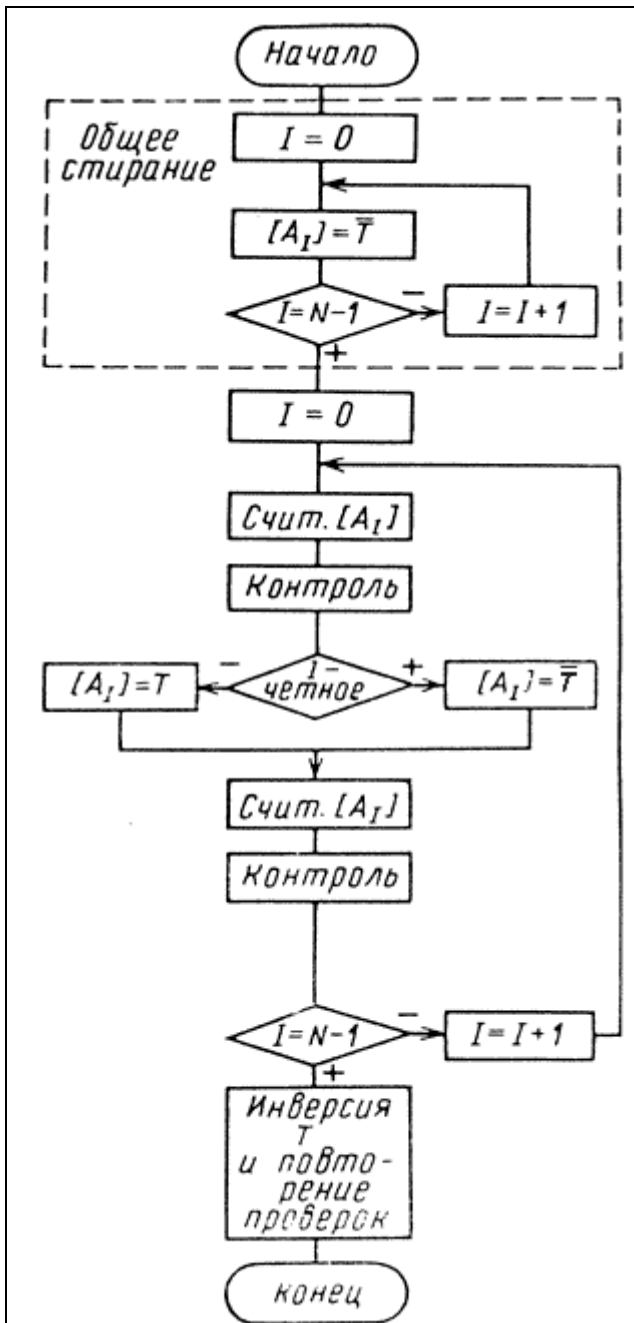


Рис. 10. Алгоритм теста «Марш – шахматная доска».

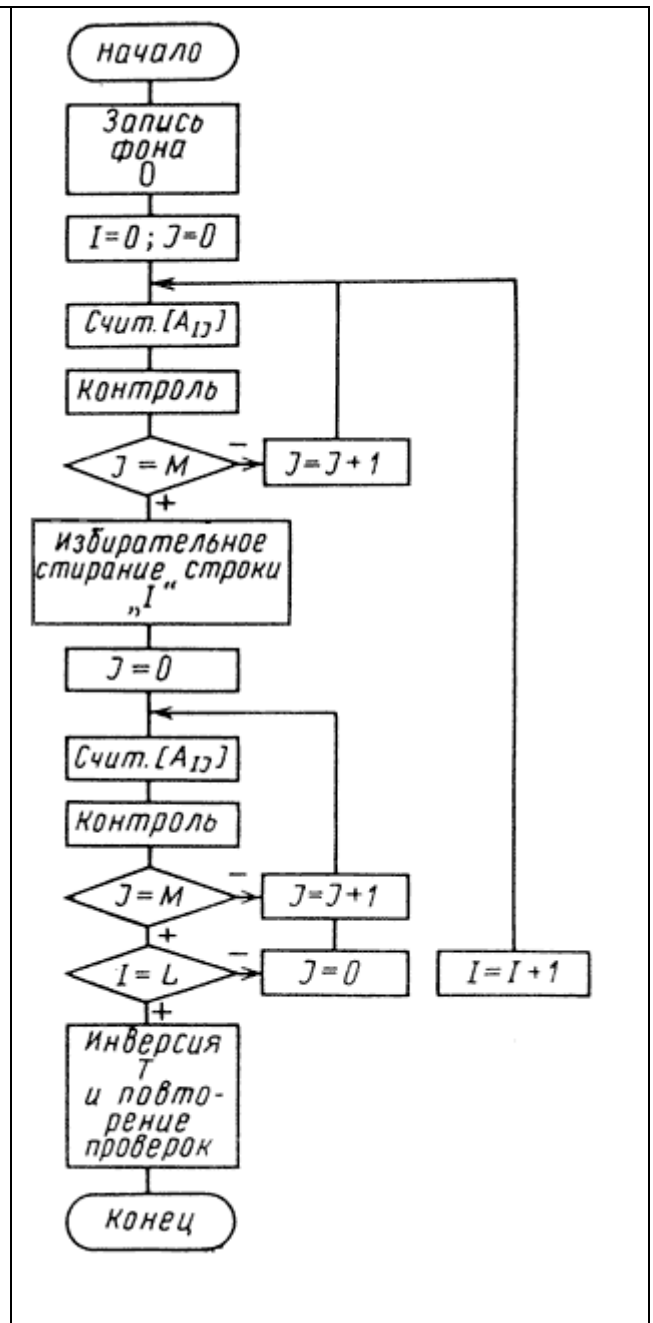


Рис. 11. Алгоритм теста «Марширующая строка».