

МЕТОДИКА ПОСТРОЕНИЯ ПОЛНЫХ КОНТРОЛИРУЮЩИХ И ДИАГНОСТИЧЕСКИХ ТЕСТОВ

Г.А. Миронов, Д.Э. Федотова
(Москва)

В настоящее время все большее распространение получают дискретные автоматические системы. По мере их усложнения возникают специфические задачи, от решения которых зависит успех применения таких систем и перспективы их дальнейшего совершенствования. К ним относятся задачи повышения надежности и эффективности использования автоматических систем.

Один из путей решения этих задач связан с вопросами организации контроля и диагностики в сложных автоматических системах.

В настоящее время наряду со схемными методами широко используются методы программного контроля, предусматривающие составление контролирующих и диагностических тестов. Процесс составления теста для сложной системы является весьма трудоемким. В целях экономии времени и затрат труда его желательно автоматизировать. Для этого необходимо разработать по возможности формальную методику составления теста, которая была бы применима к объектам различной физической природы. Теоретические основы достаточно общего подхода к этой проблеме заложены в [1], однако сегодня нет приемлемого практического решения поставленных задач.

В работе, по которой составлен настоящий доклад, делается попытка восполнить этот пробел. Показана принципиальная возможность формализации процесса составления тестов и предложены алгоритмы, с помощью которых можно получать исходные данные для организации эксперимента в сложных системах, а также рассмотрены вопросы минимизации исходных данных и применения разработанных алгоритмов для составления тестов электронных цифровых машин.

1. Постановка задачи

Задача разработки доступного для практической реализации достаточно общего метода построения тестов, применимого к различным классам схем является актуальной [12]. Метод построения тестов должен удовлетворять двум требованиям:

1) Получаемый тест (набор тестов) должен обеспечить полный контроль всей логики проверяемой системы. Применительно к ЭЦМ это означает, что логическая полнота охвата машины равна единице [2].

2) Процесс нахождения теста должен быть полностью формализован. Это обеспечивает возможность применения автомата (электронной цифровой машины) для составления теста. Необходимость применения автомата диктуется тем, что для существующих сложных систем, характеризуемых большим числом элементов и многочисленными связями между ними, составление теста вручную было бы весьма трудоемким и длительным процессом.

Ниже коротко описан формальный метод построения тестов для дискретных автоматических систем, удовлетворяющий указанным требованиям. Этот метод позволяет автоматизировать процесс составления тестов для широкого класса сложных систем, независимо от их физической структуры. Идея метода одновременно и независимо выдвинута в [2] и [13].

При постановке задачи выделяются два этапа, требующие последовательного решения.

1) Разработка методов построения тестов для комбинационных схем. При этом учитывается широкое распространение схем комбинационного типа, которые входят в состав других, более сложных схем. Вопросы, связанные с построением таких тестов, исчерпаны не до конца, несмотря на большое число работ, посвященных этому вопросу.

2) Применение и модификация методов, разработанных для комбинационных схем, к схемам более общего вида — вентильным автоматам. Вопросы разработки тестов для таких схем в литературе почти не освещались.

Поскольку одной из наиболее сложных по своей логической структуре дискретных автоматических систем являются современные ЭЦМ, метод составления тестов должен быть в первую очередь применен к устройствам и блокам ЭЦМ.

Вопросы формализации и упорядочивания тестов имеют самостоятельный интерес и в данной работе не рассматриваются.

Формальный метод построения теста, основанный на анализе логической структуры системы, требует некоторого достаточно формального описания этой структуры. Наиболее распространенные способы описания в виде: а) физических (принципиальных) схем; б) функциональных (логических) схем; в) аналитической записи — не являются формальными, так как алгоритм выполнения для них, то есть моделирование работы системы, осуществляет человек. Полностью формализованное описание структуры системы и законов ее функционирования может быть получено с помощью некоторого алгоритмического языка. Возможности построения алгоритмических языков весьма многообразны. Наиболее удобными для нашего рассмотрения являются алгоритмические языки, построенные на базе функциональных схем и аналитической записи.

II. Тесты для комбинационных схем

Пусть имеется некоторый алгоритмический язык, с помощью которого могут быть описаны асинхронные (однотактные) и синхронные (многотактные) комбинационные схемы, удовлетворяющие следующим ограничениям: а) максимальное число входов у каждого элемента схемы равно двум (схемы такого вида мы называем равносильными некоторым исходным схемам, имеющим элементы с числом входов больше двух; требование равносильности не ограничивает общности рассмотрения, но упрощает анализ); б) схема не содержит замкнутых цепей в пределах одного мгновенного режима (под мгновенным режимом понимается интервал времени, в течение которого все входные и промежуточные сигналы схемы сохраняют свое значение); в) в схеме возможны только одиночные неисправности. Кроме перечисленных требований, ставится

специфическое условие правильности схемы, заключающееся в том, что при наличии в схеме разветвления ни один из выходных сигналов не может являться функцией от сигналов в более чем одной ветви разветвления.

Сначала может быть поставлена задача: найти общий метод (алгоритм), позволяющий составлять полные контролирующие тесты для асинхронных комбинационных схем, удовлетворяющих изложенным выше требованиям. Полнота контролирующего теста совпадает с логической полнотой охвата схемы для контролирующей программы [2].

Для полной проверки схемы необходимо осуществить полную проверку каждого ее элемента, что можно сделать двумя способами: 1) подавать контролирующие сигналы непосредственно на входы элементов и проверять правильность работы непосредственно на входы элементов и проверять правильность работы непосредственно на выходе элемента; 2) подавать контролирующие сигналы через те элементы схемы, которые соединяют входы контролируемого элемента с входами схемы, и проверять правильность выходного сигнала этого элемента по значениям выходных сигналов всей этой схемы.

Первый способ практически неприменим, второй требует разработки алгоритма, осуществляющего подбор таких комбинаций сигналов на входах схемы, которые для каждого из контролируемых элементов обеспечивают: 1) последовательное поступление на входы этого элемента всех комбинаций для его полной проверки и 2) изменение хотя бы одного из выходных сигналов схемы, если контролируемый элемент содержит неисправность. Второе условие равносильно уравнению

$$((\alpha \& \&) + (\bar{\alpha} \& \&)) = 1 \quad (\text{mod } 2), \quad (I)$$

где α — подформула, в которой проверяется главная операция;

$f(x_1, x_2, \dots, x_n) = (\alpha \& \&)$ — аналитическая запись исследуемой системы.

Уравнение (I) преобразуется к виду $\& = 1$. Если аналитическая запись исследуемой системы имеет вид

$f(x_1, x_2, \dots, x_n) = (\alpha \vee \&)$, то второе условие равносильно уравнению

$$(\alpha \vee \&) + (\bar{\alpha} \vee \&) = 1 \quad (\text{mod } 2), \quad (2)$$

которое преобразуется к виду $\mathcal{Z} = 0$.

Подбор на входах некоторого элемента каждой комбинации для его полной проверки обеспечивается методом спуска от проверяемого элемента ко входам схемы. Спуск осуществляется путем перехода от элемента, на который следует подать комбинацию для его контроля, ко всем тем элементам, с которыми соединены его входы. Для каждого из этих элементов подбираются значения входных сигналов, обеспечивающие такие значения их выходных сигналов, которые образуют требуемую комбинацию для контролируемого элемента. Для каждого из элементов, непосредственно соединенных с контролируемым элементом, повторяется та же процедура и так далее, вплоть до тех элементов, входы которых являются входами схемы. При этом могут возникнуть неоднозначности, которые решаются в пользу варианта, обеспечивающего максимальную надежность выполнения теста.

Метод подбора оправдывает себя лишь в том случае, если на входах схемы можно задать 2^n различных комбинаций, то есть если любая из найденных при подборе комбинаций переменных окажется осуществимой. В противном случае либо тест будет неполным, либо в пределе метод подбора сведется к полному перебору 2^n вариантов, что практически невыполнимо.

Выражения, аналогичные (2), могут быть получены для схем, построенных над разными базисами.

Авторами построен алгоритм, реализующий подбор контролируемых комбинаций на входах асинхронной схемы, описываемой наложением деревьев. Для удобства все элементы схемы разбиваются на группы (ярусы), что позволяет установить порядок просмотра деревьев в порядке возрастания ярусов из выходных вершин. Алгоритм прекращает работу, когда просмотрены все деревья, выходы которых являются вершинами последнего яруса. Анализ каждого дерева начинается с вершины, лежащей в его корне. Работа алгоритма подбора для отдельного элемента, описываемого некоторой вершиной дерева, состоит из двух этапов: спуск к действующим входам и подъем до ближайшего выхода. Спуск ведется вплоть до элемента первого яруса. Подъем состоит в отыскании ближайшего к данному элементу выхода схемы.

Исходные данные алгоритма: кодированная запись исследуемой схемы с указанием типов элементов (по реализуемым ими логическим функциям), соединений элементов друг с другом, перечня входов схемы. Результаты работы алгоритма: набор комбина-

ций сигналов на действующих входах схемы, который соответствует полной проверке всех элементов схемы. Логическая схема алгоритма приведена в [5]. Зависимость продолжительности работы составленного алгоритма от количества элементов в схеме выражается соотношением

$$an^2 + bn, \quad (3)$$

где n - число элементов в схеме;

a, b - некоторые константы.

Алгоритм был запрограммирован на ЭЦМ "Урал-2". Полное время нахождения теста для схемы из 70 элементов составило ≈ 10 мин, что согласуется с оценкой (3). При быстроедействии ЭЦМ $2 \cdot 10^5$ оп/сек полный тест для схемы, содержащей 10^4 элементов, может быть рассчитан приблизительно за 60 часов машинного времени.

Анализ синхронных (многотактных) комбинационных схем показал, что рассмотрение такой схемы в течение L тактов можно заменить рассмотрением L одноктактных комбинационных схем. Это значит, что методика построения тестов для асинхронных схем распространяется и на синхронные схемы, которые при этом разбиваются на термы (такие подсхемы, быть может, пересекающиеся, что каждая подсхема является идеальной одноктактной схемой).

Имеется алгоритм разбиения схемы на термы [9], применимый к записи схемы в виде системы нормальных уравнений [2]. Под термом понимается совокупность элементов, работающая в некотором мгновенном режиме.

Алгоритм подбора контролирующих комбинаций для синхронной схемы состоит из нескольких этапов: а) спуск ко входам данного терма; б) спуск ко всем термам вплоть до первого; в) подъем к выходам данного терма; подъем по всем термам вплоть до последнего. Это вытекает из того, что каждая подсхема (терм) работает в один момент времени (такт), а исходными данными для ее работы служат сигналы, полученные в предыдущий момент.

Представляет интерес качество тестов, составленных по изложенной методике. Класс логических неисправностей ограничен неисправностями, которые вызываются изменением элементарной логической функции отдельного элемента. Рассматриваются равносильные схемы.

Число одиночных логических неисправностей выделенного

класса определяется соотношением

$$N = 3n_1 + 15n_2, \quad (4)$$

где N - число элементов схемы;

n_1 - число элементов с одним входом;

n_2 - число элементов с двумя входами.

Для оценки качества тестов рассматриваются результаты эксперимента с логической схемой. Каждая неисправность из числа N приводит к изменению некоторых результатов эксперимента. Если схема может содержать N одиночных неисправностей, то возможны $N+1$ результатов эксперимента. Из них только один результат отвечает правильной работе схемы. Он назван правильным ответом.

Анализ результатов эксперимента для каждой из неисправностей, возможных в схеме для схем, построенных над базисом КДО, выявляет все случаи совпадения функции и классов неисправностей. В результате анализа получены следующие соотношения.

Число совпадающих функций неисправностей в произвольной схеме с одним выходом, построенной над базисом КДО, равно

$$P_f = (5n_2 - 4) + 3n_1, \quad (5)$$

или

$$P_f = (5n_2 - 2) + 3n_1, \quad (6)$$

если на выходе схемы стоит элемент, реализующий отрицание [6]. Коэффициент неразличимости функций одиночных неисправностей для схем над базисом КДО

$$K_p = \frac{P_f}{15n_2 + 3n_1} = \frac{5n_2 - 4 + 3n_1}{15n_2 + 3n_1}. \quad (7)$$

Исследован предел этого выражения для случаев: 1) n_2 ограничено $0 \leq n_1 \leq \infty, n_2 \rightarrow \infty$; 2) $n_1 \rightarrow \infty, 0 \leq n_2 < \infty$; 3) $n_1 \rightarrow \infty, n_2 \rightarrow \infty$.

Установлены верхняя и нижняя оценки для коэффициента неразличимости функций одиночных неисправностей

$$\frac{1}{3} \leq \lim K_p \leq 1. \quad (8)$$

На практике число элементов отрицания бывает ограничено и значительно меньше числа элементов, реализующих конъюнкцию и дизъюнкцию переменных. Элементы (отрицания) или "активные" элементы схемы зачастую играют чисто вспомогательную роль развяз-

ки и усиления сигналов. Поэтому можно утверждать, что число различных результатов эксперимента в схемах из функциональных элементов составляет приблизительно $2/3$ от общего числа результатов.

Может быть доказано предложение [6], утверждающее, что если тест является полным для каждого элемента схемы, то он является полным и для всей схемы. На основании этого предложения устанавливается полнота контролирующего теста, получаемого в соответствии с методикой подбора, для выделенного класса логических неисправностей. Доказана диагностическая способность этих тестов для одиночных различных неисправностей [6]. Выводы, полученные для равносильных схем, могут быть распространены на схемы более общего вида. Если число входов k логических элементов произвольно и достаточно велико (например, $k = 10$), то построение полного теста станет затруднительным в силу большого перебора (2^{10}). Приемлемые с точки зрения качества работы алгоритма подбора значения величины k ограничены числами порядка $4 + 5$. Это значение следовало бы учитывать при разработке новых систем.

Существуют некоторые условия, при которых полный тест равносильной схемы может служить полным контролирующим тестом для соответствующей реальной схемы. Эти условия выполняются, если в схеме возможны только такие одиночные неисправности, которые проявляются как изменение значений входных переменных.

Длина полного теста (число комбинаций переменных, входящих в тест) допускает сокращения. В соответствии с алгоритмом подбора комбинаций длина полного теста должна составлять

$$L = 4n_2 \quad (9)$$

Этот тест не является минимальным, так как содержит совпадающие примеры. Длина сокращенного полного теста, полученного удалением совпадающих примеров, составляет

$$L = 4 + 2(n_2 - 1) = 2(n_2 + 1). \quad (10)$$

Если схема описывается наложением деревьев, то можно добиться сокращения длины общего теста за счет объединения двух и более примеров теста в один. Такое объединение примеров названо склейкой. Имеют место многочисленные возможности склеивания тестов на отдельных примерах. Длина сокращенного теста, кото-

ную можно получить, если отказаться от требования максимальной надежности выполнения теста, выражается соотношением:

$$L = 4 + (n_2 - 1) = n_2 + 3. \quad (11)$$

Для частного случая многовыходных схем, описываемых вхождением деревьев, максимально возможная относительная экономия в длине полного теста $\frac{\Delta L}{L}$ при числе выходов $b = 2$ равна $\frac{n-1}{2n}$, где n — число элементов, реализующих двуместные операции.

Длина полного теста для однородных схем, состоящих из какого-либо одного типа элементов, находится в пределах

$$2n \geq L \geq Я + 3, \quad (12)$$

где $Я$ — число ярусов схемы, а число выходов схемы лежит в пределах $1 \leq b \leq n-1$. Длина полного теста для неоднородных схем при тех же условиях находится в пределах

$$2n \geq L \geq 5. \quad (13)$$

Минимальная длина теста схемы, описываемой как пересечение q деревьев, определяется значением

$$L_q = \max_q E + \max_q D, \quad (14)$$

где E, D подсчитываются для элемента схемы, описываемого вершиной пересечения деревьев. Они представляют собой соответственно число единиц и число нулей в каждом из слов, являющихся правильными ответами на полный тест всей схемы, наблюдаемыми на выходах того же элемента.

Исследована возможность применения метода подбора контролирующих комбинаций для построения тестов схем в случае, когда на входах схемы осуществимы не все 2^n комбинаций. Функции, описывающие работу таких схем, названы неполностью определенными. Применение метода подбора в этом случае приводит к полному перебору.

Введенные информационные оценки тестов (информационная нагрузка отдельной проверки в данном тесте; разрешающая способность теста) позволили доказать, что для целого ряда реальных случаев, имеющих место на практике, можно обойтись предполными тестами, для которых полный перебор окажется значительно меньше 2^n .

Если применить метод подбора к неправильным схемам, то построение полного теста оказывается невозможным. Попытка построить тест приводит к полному перебору. Авторами построен алгоритм формального обнаружения неправильных схем. На вопросах об оценке соотношения правильных и неправильных схем в реальных устройствах авторы не останавливались.

Ш. Тесты для накапливающих схем

Метод подбора контролирующих комбинаций, описанный для комбинационных схем, может быть применен и к схемам накапливающего типа, которые описываются в терминах теории автоматов. В рассмотрение включены такие элементарные автоматы, как элемент задержки, триггер (рассмотрен обобщенный вариант триггера, имеющего раздельные и счетный входы).

Схемы накапливающего типа являются многотактными и потому разбиваются на термы. Входами терма могут быть входы схемы, входы элементарных автоматов; выходами терма — выходы схемы, выходы элементарных автоматов. Чтобы применить методику подбора, необходимо задать описание исследуемой схемы на некотором алгоритмическом языке, то есть математическую модель работы системы. Такие модели могут быть составлены без учета переходных процессов в элементарных автоматах. Однако это представление недостаточно точно описывает существующие схемы устройств накапливающего типа. Оно применимо только к тем схемам, которые не содержат элементов формирования сигнала. Для них действующими значениями переменных являются значения сигналов, соответствующих переходным процессам в других элементах.

Мы анализируем наиболее общий случай, отвечающий схемам с потенциально-импульсными связями, которые содержат элементы формирования сигнала. В связи с этим нами используются материалы [3], в которых содержится описание математической модели работы временной логической схемы с учетом переходных процессов в элементарных автоматах. Сделан переход от двухбуквенного кодирования сигнала в алфавите $(0, 1)$ к многобуквенному кодированию. Показано, что элементы схемы, реализующие элементарные логические функции (например, конъюнкции и дизъюнкции переменных), описываются в многобуквенном алфавите с неполной областью определения. Ранее было показано, что для схем, опи-

сываемых функцией с неполной областью определения, применение метода подбора приводит к полному перебору. Во избежание перебора предлагается задавать множество возможных состояний элементов схемы в отдельные моменты времени (термы) прямым перечислением этих состояний. Такой процесс назван составлением временных диаграмм. Введено понятие режима работы схемы: это множество всевозможных состояний схемы, определяемых множеством значений переменных на независимых входах при некоторых фиксированных комбинациях значений переменных на зависимых входах. Число режимов определяется числом допустимых комбинаций значений зависимых переменных. Для некоторых схем ЭЦМ режимы могут определяться машинными операциями. Временные диаграммы работы отдельных элементов схемы составляются в пределах каждого элемента. Существует алгоритм составления временных диаграмм, использующий математическую модель работы схемы, данную в [3].

Составление временных диаграмм необходимо для последующей реализации алгоритма подбора контролируемых комбинаций, однако может представлять и самостоятельный интерес при моделировании работы проектируемых устройств. Исходными данными работы алгоритма помимо задания математической модели является задание временных диаграмм на входах схемы для всех возможных режимов. Режимы, которым соответствуют одинаковые временные диаграммы, объединяются в группы. Представляют интерес оценки качества составленного алгоритма, то есть зависимость продолжительности его работы от количества элементов в схеме. При анализе времени работы алгоритма комбинационными логическими элементами пренебрегаем, что дает верхнюю оценку. Анализ качества работы алгоритма проведен на различных вариантах схем:

1) Для схемы, состоящей из m независимых каналов, каждый из которых имеет два устойчивых состояния, время работы алгоритма

$$T_1 = 2t \sum_{i=1}^m k_i, \quad (15)$$

где k_i - число групп режимов по i -му входу схемы;

t - продолжительность работы алгоритма в процессе нахождения временной диаграммы для одного элемента схемы в одном режиме по известным входным диаграммам.

2) Для схемы, состоящей из m независимых однотипных каналов, каждый из которых имеет 2^n устойчивых состояний, время работы алгоритма

$$T_2 = n^2(n-1) + \sum_{i=1}^m k_i \quad (16)$$

3) Для схемы, состоящей из m каналов, связанных между собой по схеме последовательного счетчика, где каждый канал имеет два устойчивых состояния, время работы алгоритма

$$T_3 = (2 \sum_{i=1}^m k_i + \sum_{i=1}^m k_i \sum_{\substack{j=1 \\ j \neq i}}^{n-1} k_j) t \quad (17)$$

4) Для схемы, состоящей из m однотипных каналов, связанных между собой по схеме сумматора с циклическим переносом, где каждый канал имеет два устойчивых состояния, время работы алгоритма

$$T_4 = mt(2 \sum_{i=1}^m k_i + \sum_{i=1}^m [k_i \sum_{\substack{j=1 \\ j \neq i}}^{n-1} k_j]) \quad (18)$$

5) Для схемы, объединяющей рассмотренные четыре случая, время работы алгоритма

$$T_{\text{полное}} = T_1 + T_2 + T_3 + T_4 \quad (19)$$

где T_1, T_2, T_3, T_4 - рассчитываются в соответствии с формулами (15), (16), (17), (18).

Рассмотренные пять моделей охватывают практически все схемы реальных автоматов. В качестве примера рассчитано время работы алгоритма для схемы пятиразрядного накапливающего сумматора вместе с некоторой схемой управления (схема полностью соответствует сумматору ЭЦМ "Урал-2" за исключением числа разрядов). На примере этой схемы были проверены правильность работы алгоритма составления временных диаграмм на ЭЦМ "Урал-4". Схема содержит около 200 элементов, в том числе свыше 20 элементарных автоматов. Согласно формулам (15) + (19) полное время работы алгоритма должно составлять приблизительно 3 часа, что отвечает результатам эксперимента.

Алгоритм составления временных диаграмм является вспомогательным. Основным алгоритм определения теста основан на опи-

санном ранее методе подбора, но теперь подбор ведется с учетом составленных временных диаграмм. Авторами сформулированы условия, эквивалентные уравнениям типа (1) или (2) для случая многобуквенного кодирования сигналов, указано множество необходимых элементарных проверок для различных типов элементов. Алгоритм подбора складывается из трех частей: А - управляющий алгоритм, U - алгоритм подъема, D - алгоритм спуска. Исходные данные для работы алгоритма подъема U: а) выбран элемент i , для которого строится тест; б) выбрана проверка j , которую желательно осуществить; в) выписаны все режимы, в которых может осуществляться проверка j для элемента i , и все термы. Эти данные получаются в процессе работы алгоритма А. Алгоритм подъема находит перечень элементов, принадлежащих одной ветви, которая соединяет выход анализируемого элемента i с выходом схемы или входом триггера, который в данном терме меняет свое состояние. Кроме того, алгоритм U сужает множество допустимых режимов и термов. Это новое множество входит в состав исходных данных для работы алгоритма D. Спуск производится в несколько этапов, на каждом из которых происходит спуск до ближайшего яруса триггеров (или входов схемы). Оценка качества работы алгоритма построения теста для накапливающих устройств может быть сделана на основе соотношения (3). Алгоритмы подробно описаны в [10].

Чтобы разработанную методику можно было широко применять на практике, необходимо предъявить специальные требования к машинам, на которых будет осуществляться работа алгоритмов построения тестов. Проведенный эксперимент показал, что применение ЭЦМ типа "Урал-2" или "Урал-4" оказывается малоэффективным в силу их низкого быстродействия и малого объема памяти - как оперативной, так и промежуточной (на магнитных барабанах). Использование магнитных лент сильно увеличивает время решения задачи (на магнитных лентах приходится записывать временные диаграммы ввиду их большого объема, на магнитных барабанах - математическую модель работы схемы). Приведенные оценки качества предложенных алгоритмов позволяют сделать вывод, что приемлемыми для такого класса задач являются машины с быстродействием порядка 10^5 - 10^6 оп/сек и объемом оперативной памяти порядка $30 + 60$ тысяч кодов. Из отечественных машин такого класса может быть использована машина БЭСМ-6.

IV. Построение тестов для ЭЦМ

Методика составления тестов для дискретных автоматических систем, предлагаемая в настоящей работе, является достаточно универсальной, поскольку применима к объектам самой различной физической природы. Однако использование ее в конкретных ситуациях может потребовать уточнений, связанных с особенностями структуры той или иной системы.

Целесообразно рассмотреть применение методики к электронным цифровым машинам. Выбор для иллюстрации методики именно такой дискретной системы неслучаен. Современные ЭЦМ представляют собой, пожалуй, наиболее сложные технические объекты как по числу составляющих их элементов, так и по логической структуре. Кроме того, составление тестов для ЭЦМ является одной из важнейших проблем повышения надежности их использования, особенно насущной в связи с возрастающей сложностью и увеличением объема выпуска ЭЦМ.

Моделировать схему ЭЦМ в целом в настоящее время не представляется возможным в силу недостаточности объема оперативной памяти современных ЭЦМ. Поэтому моделирование ЭЦМ и составление тестов для ЭЦМ нужно производить поблочно. Существуют различные способы разбиения ЭЦМ на блоки. Отметим некоторые из них: 1) разбиение схемы на функциональные блоки (регистры, сумматоры чисел, порядков, счетчики и т.д.); 2) разбиение схемы на блоки машинных операций, что выполняется обычно для каждого из устройства АУ, УУ, ОЗУ в отдельности; 3) разбиение на функциональные блоки операций, по существу объединяющее первые два способа; 4) разбиение на участки по источникам питания; 5) разбиение на участки профилактического контроля, помогающее локализовать неисправность в схеме путем изменения питающих напряжений; 6) конструктивное разбиение схемы на ячейки, блоки, панели, стойки и др. Перечисленные разбиения не являются формализованными, поэтому не могут быть использованы как основа поблочного деления ЭЦМ при формальном составлении тестов. Однако мы используем эти разбиения для анализа и классификации возможных видов входных сигналов для отдельных блоков ЭЦМ. Вся информация в ЭЦМ разбивается на две группы: 1) информация, зависящая от выполняемой операции; 2) информация, зависящая от числового материала. В связи с этим входные сигналы некоторого блока ЭЦМ для самого общего

случая можно классифицировать следующим образом: 1) временные тактирующие сигналы $x(\tau)$; 2) сигналы, зависящие от выполняемой операции $x(k, \tau)$; 3) сигналы, зависящие от числового материала $x(r, \tau)$; 4) сигналы смешанного типа $x(k, r, \tau)$; 5) сигналы, не зависящие от времени x . Различные режимы работы схемы определяются сигналами типов 2 + 5. При разбивке схемы на блоки желательно, чтоб попарная корреляция входных сигналов была минимальной, число допустимых режимов, определяемых сигналами $x(k, \tau)$ и $x(k, r, \tau)$, также было мало, а число режимов, определяемых сигналами $x(r, \tau)$, стремилось к 2^n в переводе на двоичный алфавит.

Введено формальное разделение схемы ЭЦМ на блоки, реализующие элементарные машинные операции [8]. Выходными ячейками элементарных машинных операций могут быть ячейки памяти ОЗУ, счетчик команд, сумматор АУ и т.д. Исходными данными для построения алгоритма формального разбиения схемы ЭЦМ на элементарные машинные операции, являются: а) формальное описание схемы ЭЦМ; б) описание всех элементарных машинных операций; в) описание алгоритма выполнения, который в ЭЦМ реализуется некоторой управляющей схемой типа дешифратора команд. Такая схема вырабатывает сигналы, управляющие выполнением того или иного комплекса элементарных операций. Составление теста для формализованных разбиений схемы на элементарные машинные операции дает предпосылки для создания программирующей программы [II], которая могла бы строить систему испытательных программ. Испытательная программа — это автоматическое выполнение последовательности тестов, каждый из которых проверяет одну элементарную операцию. Исходными данными для программирующей программы являются: 1) наборы тестов для элементарных операций, найденные по методике данной работы; 2) описание системы команд машины с помощью элементарных машинных операций.

Л и т е р а т у р а

1. И.А. Чегис, С.В. Яблонский. Логические способы контроля работы электрических схем. Труды Математического института им. Стеклова, 1958, вып. III.
2. Г.А. Миронов. Методика построения систем испытательных программ для электронных цифровых машин, диссертация, 1961.

3. Ю.В. Гайкович, Г.А. Миронов. Универсальная модель на уровне функциональных элементов. Настоящий сборник.
4. Д.Э. Федотова. О построении одновыходных тестов для логических схем типа "деревьев" из функциональных элементов. Сб. "Применение математических методов и вычислительной техники в горном деле". - Гипроуглеавтоматизация, 1966.
5. Г.А. Миронов, Д.Э. Федотова. Составление тестов для логических сетей. Сб. "Цифровая вычислительная техника и программирование", п/р Китова А.И. Изд. "Сов. радио", вып. 3, 1968.
6. Г.А. Миронов, Д.Э. Федотова. Применение контролирующих тестов для диагностики неисправностей логических сетей. Сб. "Цифровая вычислительная техника и программирование". Изд. "Сов. радио", вып. 4, 1968.
7. Д.Э. Федотова. Информационные оценки тестов. Построение многовыходных тестов для схем типа "деревьев" из функциональных элементов. Сб. "Применение математических методов и вычислительной техники в горном деле". Гипроуглеавтоматизация, 1966.
8. Н.А. Криницкий, Г.А. Миронов, Г.Д. Фролов. Использование понятия элементарная машинная операция для анализа вычислительных систем. Настоящий сборник.
9. Г.А. Миронов, Д.Э. Федотова. Составление тестов для синхронных логических сетей. Сб. "Цифровая вычислительная техника и программирование". Изд. "Сов. радио", вып. 5, в печати.
10. Д.Э. Федотова. Формализация процесса составления тестов для дискретных автоматических систем. Диссертация на соискание ученой степени кандидата технических наук. 1967.
11. Г.А. Миронов, В.П. Битюцкий, Н.А. Комиссаров, В.П. Пинаев. О программирующей программе для построения испытательных программ. Настоящий сборник.
12. Г.А. Миронов. Основные проблемы программного контроля и диагностики дискретных автоматических систем. Настоящий сборник.
13. K.Maling, E.L.Allen. A Computer Organization and Programming System for Automated Maintenance. IEEE Trans. on Electronic Computer, 1963, EC-12, N 6. (Русский перевод "Кибернетический сборник" № 1).