

НЕКОТОРЫЕ ВОПРОСЫ ДИАГНОСТИКИ НЕИСПРАВНОСТЕЙ В ЭВМ

А.Ф. Шпагин
(Москва)

Восстановление работоспособного состояния современных систем, в частности, электронных цифровых машин (ЭВМ), все в большей степени становится зависимым от операций контроля правильности их работы и диагностики неисправностей, вызвавших их неправильную работу.

При постановке задачи диагностики неисправностей в ЭВМ весьма важно определить виды неисправностей последней. С точки зрения задач диагностики будем различать следующие неисправности ЭВМ: отказ, систематический отказ, сбой.

Отказ — неисправность, которая преобразует исходную схему в некоторую другую, описываемую новыми логическими формулами, неэквивалентными исходному описанию.

Систематический отказ — неисправность, которая вызывает неправильную работу только при выполнении определенных условий (чаще всего информационных), которыми можно управлять.

Сбой — неисправность, появлением которой нет возможности управлять и которая действует в течение некоторого (обычно малого) периода времени, а затем самоустраняется.

Необходимо отметить, что систематический отказ, причины которого неизвестны, может считаться сбоем до тех пор, пока не будут выяснены эти причины.

По нашему мнению, диагностика неисправностей должна производиться только для отказов и систематических отказов. Система

борьбы со сбоями должна включать в себя средство контроля правильности работы ЭВМ, фиксацию факта сбоя и обстановки, сопутствующей ему, и средств устранения последствий сбоев. По мере накопления данных об обстановке, сопутствующей сбоям, производится их анализ с целью выявления причин сбоя или условий их появления. Как только это выяснено, для устранения сбоев может быть использована методика диагностики систематических отказов.

Представляется необходимой выработка требований к ЭВМ со стороны системы восстановления работоспособного состояния ЭВМ. Мы предлагаем ввести количественные параметры ЭВМ (адресная и временная разрешающие способности), характеризующие ЭВМ как объект диагностики.

Адресной разрешающей способностью ЭВМ, устройства или отдельного узла (А), называется отношение числа запоминающих элементов, доступных программному опросу и управлению (m)^{*}, к общему числу запоминающих элементов, входящих в их состав (n)

$$A = \frac{m}{n}.$$

Чем ближе значение А к единице, тем легче построение диагностических программ.

Обозначим через "k" число, показывающее сколько раз меняет свое состояние элемент от установки до опроса.

Каждому элементу будет соответствовать некоторое множество чисел K, так как один и тот же элемент может изменять свое состояние по разному.

Временной разрешающей способностью элемента назовем минимальное значение K_{min} из вышеописанного множества.

Временной разрешающей способностью ЭВМ, устройства или отдельного узла В, назовем среднее арифметическое значение K_{min} элементов, входящих в них:

$$B = \frac{\sum_{i=1}^n K_{min}^i}{n},$$

где n - число запоминающих элементов ЭВМ, устройства или узла.

Имеет смысл ввести понятие максимальной временной разрешающей способности B_{max} , которое будет равно значению K_{min}

*) Элементы, отнесенные к m , наиболее полно отвечают требованиям диагностики, хотя в некоторых случаях может оказаться полезным относить к m элементы, имеющие хотя бы только возможность программного опроса, а в других случаях - программного управления.

того элемента, у которого оно наибольшее. При разработке конструкции ЭВМ или отдельных узлов и выборе системы команд необходимо стремиться к тому, чтобы их временная разрешающая способность V была ближе к единице, а V_{\max} было бы ограничено, предельное значение V_{\max} может определяться достигимым качеством средств контроля и диагностики.

Соблюдение этих условий позволит создать простые и более совершенные диагностические программы.

Имеет большое значение исследование и разработка параметров диагностических программ, которые позволили бы достаточно успешно проводить сравнение качества диагностических программ и указывали пути их дальнейшего совершенствования.

Зачастую используются характеристика контролируемых и диагностических программ (логическая полнота охвата, режимная глубина контроля, эффективность контроля, избыточность контроля). Общее их число более десяти.

Большое количество характеристик может затруднять оценку качества диагностических программ ввиду необходимости производить большое количество расчетов и обширного состава исходных данных, получение которых иногда может быть затруднительным. Целесообразно ограничиться всего тремя параметрами диагностических программ: эффективностью, временная избыточность и разрешающая способность.

Назовем кратностью неисправности число одновременно неисправных элементов.

Эффективность диагностической программы E это отношение числа неисправностей (S), локализуемых ею, к общему числу всех неисправностей, возможных в схеме ЭВМ или узле:

$$E = \frac{S}{M}.$$

Эффективность диагностической программы зависит от того, какая кратность неисправности допускается. Сравнение эффективности диагностических программ следует производить с учетом кратности неисправностей.

Если вероятности появления неисправностей равны между собой, то эффективность E можно рассматривать как вероятность локализации неисправности диагностической программой.

Не менее важной характеристикой качества диагностической программы может служить точность, с которой происходит локализация неисправного элемента. Результатом работы диагностической программы является адрес того сменного элемента, кото-

рый привел ЭВМ к неправильной работе. Однако далеко не всегда удается провести диагностику с точностью до одного неисправного элемента. Как правило, диагностическая программа имеет большое количество выходов (e). Каждому выходу соответствует некоторое подмножество N_i ; с числом элементов (N_i) исходного множества элементов N диагностируемой схемы. Среди элементов указанного подмножества и находится неисправный элемент.

Совокупность подмножеств N_i не является разбиением исходного множества N на непересекающиеся классы, подмножества N_i и N_j при $i \neq j$ могут иметь общие элементы. В общем случае

$$\sum_{i=1}^l |N_i| \geq |N|.$$

Для практики не безразлично, каково будет число элементов подмножества N_i , найденного диагностической программой. При устранении неисправности можно заменить все элементы, входящие в N_i , на исправные, а подозрительные отправить на проверку. Это приводит к необходимости иметь большой комплект запасных элементов и проводить повторную проверку исправных элементов. Другой путь состоит в замене элемента n_{i1} и проверке работоспособности машины. Если замена не помогла, то заменяется n_{i2} и т.д. И то и другое приводит к увеличению затрат.

Параметр диагностической программы, который характеризует степень локализации неисправного элемента, даваемую программой, мы называем разрешающей способностью.

Разрешающей способностью диагностической программы R называется среднее арифметическое чисел элементов подмножеств, соответствующих всем её исходам l :

$$R = \frac{\sum_{i=1}^l N_i}{l}.$$

Иногда может оказаться полезным рассмотреть максимальную R_{\max} или минимальную R_{\min} разрешающие способности, которые будут равны соответственно (N_i) того подмножества, которое имеет максимальное число элементов и (N_j) подмножества, имеющего минимальное число элементов. Чем ближе значение R к единице, тем выше степень локализации неисправного элемента.

Во многих работах вполне удовлетворительной разрешающей способностью диагностических программ считается 3-5 сменных ячеек. Если при конструировании ЭВМ будут приняты специальные меры, улучшающие их адресную и временную разрешающие способности, то разрешающую способность, на наш взгляд, можно улучшить до 1-2 сменных ячеек.

И, наконец, необходимо ввести параметр, который указывал бы на затраты, связанные с диагностикой. Для схемной и программной диагностики, имеет место аппаратурная, или, соответственно, временная избыточность. Временная избыточность для разных неисправностей является случайной величиной t . Если считать, что все неисправности взаимонезависимы, то временная избыточность диагностической программы T будет равна математическому ожиданию этой случайной величины:

$$T = Mt = \sum_{i=1}^S t_i p_i,$$

где p_i - вероятность появления неисправности i ,

t_i - время локализации неисправности i ,

S - количество неисправностей, локализуемых диагностической программой.

При $p_i = p_j$, $1 \leq i \leq S$; $1 \leq j \leq S$:

$$T = \frac{\sum_{i=1}^S t_i}{S}.$$

Преимущество программных методов диагностики перед схемными состоит в том, что они дают существенную экономию средств, не требуя дополнительного оборудования. Особенно ощутимая экономия может быть достигнута для ЭВМ, выпускаемых образовательно большими сериями, так как затраты на создание программных средств диагностики не зависят от количества ЭВМ данной серии. Затраты на схемные методы диагностики растут прямо пропорционально с ростом количества выпускаемых ЭВМ.

Тем не менее программные методы не смогли вытеснить схемных, так как некоторые неисправности ЭВМ делают их программную диагностику невозможной. В частности диагностические программы требуют в качестве условия их использования наличия некоторой части ЭВМ, которая должна заведомо работать исправно.

В связи с этим возникает задача определения минимального объема ЭВМ, который достаточен для построения диагностических программ, не накладывает слишком больших ограничений на программы и в то же время требует небольших аппаратных средств для своей проверки и диагностики.

Заслуживают также внимание вопросы разработки способов выдачи информации диагностической программой о результатах своей работы.

В силу того, что структура ЭВМ весьма сложна, целесообразно рассматривать ЭВМ, как систему, состоящую из подсистем. Причем, может быть несколько уровней иерархии. Для целей диагностики достаточно выделить три уровня иерархии; ЭВМ, устройство, блок.

При иерархической интерпретации системы, элементы в подсистеме надо объединять таким образом, чтобы связи и влияние элементов одной подсистемы на элементы другой подсистемы были незначительны по сравнению с влиянием друг на друга элементов одной подсистемы. Это уменьшает число различных состояний системы и её внутренних связей.

Для использования на разных уровнях иерархии предлагаются следующие методы диагностики неисправностей.

Первый метод. Будем считать, что ячейки a_i диагностируемого объекта являются элементами некоторого множества A .

Пусть диагностируемый объект может выполнять некоторый набор операций *) $b_k \in B$ ($k = 1, \dots, n$).

Тогда множество $a_i \in A$ относительно операции b_k разбивается на два непересекающихся класса: класс A элементов a_i , участвующих в формировании результата данной операции, и класс A_k элементов, не участвующих в формировании результата этой операции.

Допустим также, что операции некоторого подмножества

$$\bar{B} \in B,$$

где

$$b_k \in \bar{B},$$

*) Под словом "операция" здесь и дальше будет пониматься какая-либо обобщенная операция, состоящая из одной или нескольких команд ЭЦВМ.

если

$$k \in k_{\bar{B}},$$

выполняется неверно. В частности, может иметь место $\bar{B} = \emptyset$.

Тогда очевидно, что неправильно функционирующие элементы $a_i \in A$ удовлетворяют условию:

$$a_i \in \bigcap_{k \in k_{\bar{B}}} A_k \mid \bigcup_{k \in N - k_{\bar{B}}} A_k, \quad (I)$$

где

$$N = 1, 2, \dots, n.$$

Здесь $\bigcap_{k \in k_{\bar{B}}} A_k$ означает подмножество множества A , каждый элемент которого участвует в формировании результата всех операций $b_k \in \bar{B}$.

Объединение $\bigcup_{k \in N - k_{\bar{B}}} A_k$ означает подмножество множества A , каждый элемент которого не участвует в формировании ни одной из операций

$$b_k \in B/\bar{B}.$$

В связи с тем, что множества B конечны и рассматриваются только два исхода выполнения каждой операции (правильно и неправильно) последовательное выполнение диагностируемым объектом всех операций

$$b_k \in B.$$

также приводит к конечному множеству исходов, т.е. к конечному числу комбинаций правильного и неправильного выполнения различных операций. В общем случае число таких комбинаций

$$N = 2^n.$$

На практике же их число значительно меньше. Каждой такой комбинации, очевидно, и соответствует некоторое множество элементов, удовлетворяющих (I).

Перенумеруем все элементы множества B . Каждому классу A_i соотнесем двоичное число

$$\alpha_i = \alpha_1^i \dots \alpha_p^i,$$

где $p = |B|$

$$\alpha_{ik} = \begin{cases} 1, & \text{если } a \text{ } 0 \text{ } b_k \\ 0, & \text{если } a \text{ } \bar{0} \text{ } b_k, \end{cases}$$

при $a \in A_i$.

Будем считать, что неисправная ячейка (или группа ячеек) приведет к неправильному результату всех операций, в формировании результата которых она участвует и не отразится на остальных. Тогда диагностика неисправностей строится следующим образом.

Производится выполнение всех операций.

$$b \in B$$

в соответствии с установленным порядком. Выполнение каждой операции отмечается "1" в случае неправильного результата и "0" — в случае правильного.

Полученное двоичное число соотносится к классу A_1 , определяющему группу элементов

$$a \in A_1,$$

среди которых находится неисправный.

Этот метод может быть использован на всех уровнях иерархии.

Основным недостатком этого метода является то, что в случае возникновения кратных или перемешающихся ошибок он может дать неправильный результат.

Для устранения влияния кратных ошибок на результат диагностики может быть предложен другой метод.

II метод. Имеем множество A и B , описанные выше. Каждому элементу b_k соотнесем подмножество A_k , состоящее из тех и только тех элементов, которые участвуют в формировании результата операции b_k .

Расположим теперь полученные подмножества в порядке возрастания числа их элементов $|A_k|$, причем при выборе каждого последующего подмножества учитываются не все элементы, принадлежащие ему, а только те, которые не входят ни в одно предыдущее подмножество.

В соответствии с полученным порядком расположения подмножеств строим диагностическую программу, в которую входят все операции $b \in B$. Номер неправильно выполненной операции указывает на соответствующее подмножество A_k .

Наисправные ячейки определяются по элементам множества A со следующими характеристиками

$$a \in A_k \mid \bigcup_{n=1}^{k-1} A_n.$$

Метод может быть использован на всех уровнях иерархии.

К недостаткам этого метода следует отнести то, что неустойчивая или перемежающаяся ошибка может привести к неправильному результату диагностики.

Влияние перемежающейся ошибки на правильность результата диагностики связано с тем, что приходится делать анализ на основе результата нескольких операций.

III метод применяется для некоторого класса схем (к ним могут быть отнесены сумматоры, регистры, счетчики и т.п.), на результаты которых перемежающаяся ошибка влияет лишь частично. Исключается возможность ложного указания на исправный элемент, так как анализ производится на основании результата одной операции. Метод применим на третьем уровне иерархии. Однако, если элемент с перемежающейся ошибкой в момент проведения диагностики исправен, то он не будет указан. Эти неисправности можно обнаружить, квалифицируя их как сбои, и применять к ним систему борьбы со сбоями.

Устройства или блоки ЭЦВМ можно представить в виде некоторых ориентированных графов, вершинами которых будут элементы устройств или блоков, а связи между последними — дуги, ориентированные в направлении распространения сигнала. Элемент (вершина) может сработать от помехи или не сработать от полезного сигнала по одному или нескольким входам. Такие неисправности элементов будем рассматривать как неправильную работу одной или нескольких его входных связей (дуг). Неправильное определение состояния элемента будем рассматривать как неправильную работу его выходных связей.

Будем рассматривать схемы, графы для которых имеют вид, представленный на рис. 1 — 5.

α_1 — элемент, имеющий два состояния,

l_1 — внешняя выходная связь,

g_1 — внешняя входная связь,

f_1 — внутренняя связь: два воздействия по связи f на элемент α_1 эквивалентны одному воздействию на элемент α_{1-1} .

t — внутренняя связь, воздействие по которой осуществляется только при одном из двух состояний элемента α_0 до или после воздействия всех остальных связей.

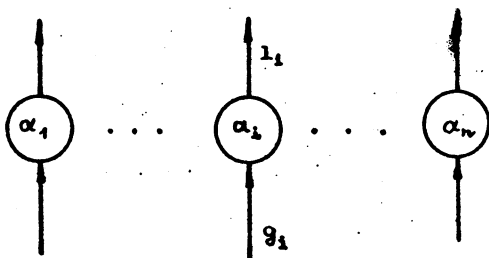


Рис. 1

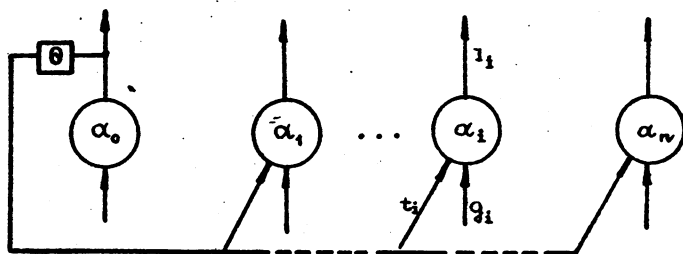
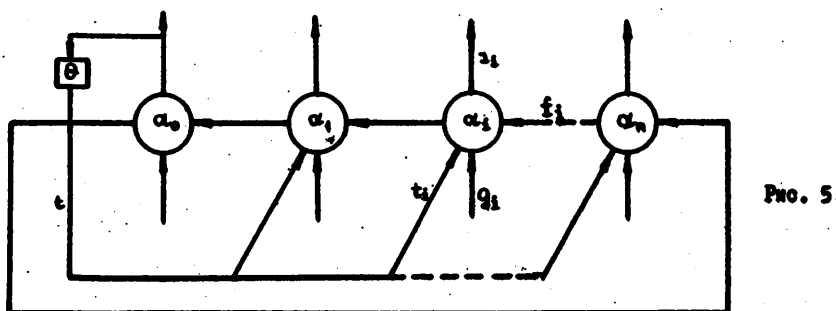
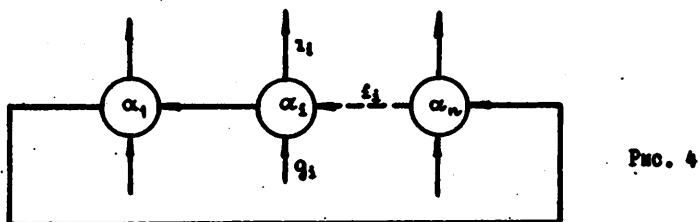
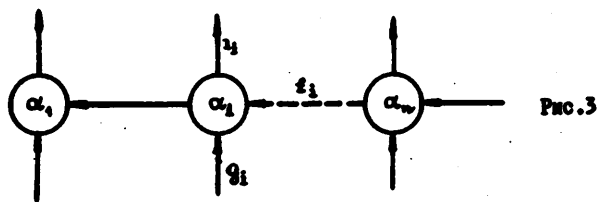


Рис. 2



θ - элемент, не входящий в рассматриваемую схему, анализирует состояние элемента α_0 и оказывает воздействие на остальные элементы, в том числе, если α_0 находится в единичном состоянии.

Рассмотрим метод выбора одного элемента из множества других, который затем применим для диагностики неисправностей в приведенных графах.

Пусть имеется множество A с элементами a_i ($i = 1, \dots, n$). Число элементов этого множества

$$|A| = n.$$

Каждый элемент a_i может иметь одно из двух возможных состояний, т.е. :

если $a_i \neq \alpha$, то $a_i = \beta$

и если $a_i \neq \beta$, то $a_i = \alpha$.

Выражение $a_i = a_j$ следует понимать так, что:

если $a_i = \alpha$, то $a_j = \alpha$

и если $a_i \neq \alpha$, то $a_j \neq \alpha$.

Необходимо найти элемент

$$a_1 \neq a_{i-1}$$

за минимальное количество проверок.

Будем считать, что существует, по крайней мере, одна пара таких элементов, и назовем её границей, которая находится между элементами a_1 и a_{i-1} .

Если воспользоваться методом простого перебора, то минимальное количество проверок, которое потребуется

$$C_{\max} = n.$$

Предполагаемый метод позволит снизить число проверок до

$$C_{\max} = \log_2 n,$$

если в нашем распоряжении имеется процедура, которая позволяет делить данное множество на два подмножества с одинаковым количеством элементов^{*)}, определять, имеется ли в одном из подмно-

*) Считаем, что вероятность того, что элемент находится в состоянии α

$$P_\alpha = P_\beta = 0,5.$$

хотят хотя бы один элемент, не равный всем остальным α , если такого элемента нет, то фиксировать состояние, в котором находятся элементы проверяемого подмножества.

Применим описанную процедуру к данному множеству A и его подмножеству A_1 , элементы которого a_i , где

$$\frac{n}{2} < i \leq n.$$

Возможны три случая:

1. Существует $a_i \neq a_j$ при $\frac{n}{2} < i \leq n$; $\frac{n}{2} < j \leq n$; $i \neq j$.
2. $a_i = \alpha$ при $\frac{n}{2} < i \leq n$.
3. $a_i = \beta$ при $\frac{n}{2} < i \leq n$.

Если имеет место случай 1, то искомый элемент будет находиться среди элементов подмножества A_1 , к которому процедура применяется также, как и к множеству A .

В случае 2 процедуру применяем к подмножеству A_2 с элементами a_i , где

$$1 \leq i \leq \frac{n}{2},$$

и к его подмножеству A_3 с элементами a_i , где

$$\frac{n}{4} < i \leq \frac{n}{2}.$$

Здесь возможны три случая:

- 1'. Существует $a_i \neq a_j$ при $\frac{n}{4} < i \leq \frac{n}{2}$; $\frac{n}{4} < j \leq \frac{n}{2}$, $i \neq j$;
- 2'. $a_i = \alpha$ при $\frac{n}{4} < i \leq \frac{n}{2}$;
- 3'. $a_i = \beta$ при $\frac{n}{4} < i \leq \frac{n}{2}$.

Случай 1' аналогичен 1.

Случай 2' аналогичен 2.

В случае 3' искомым элементом является a_i , у которого

$$\frac{n}{2} < i \leq \frac{n}{2} + 1.$$

Если имеет место случай 3, то после применения процедуры к подмножествам A_2 и A_3 получим три случая 1', 2', 3' (См. случай 2).

Случай 1' аналогичен 1.

В случае 2' искомым элементом является a_1 , у которого.

$$\frac{n}{2} < i \leq \frac{n}{2} + 1.$$

Случай 3' аналогичен 3.

Максимальное количество шагов, которое требуется для указания одного элемента, $C = \log_2 n$. В случаях 2 - 3' или 3 - 2'

количество шагов $C < \log_2 n$.

Для применения этого метода к анализу схем рис.1+5 введем дополнительные определения. Будем называть дефектом связи отсутствие требуемого воздействия тогда, когда оно должно быть в наличии. Кратностью дефекта - P назовем количество лишних или отсутствующих воздействий *).

Граф рис.1 позволяет обнаружить дефекты с $P \neq 2k$; $k = 1, 2, \dots$

Все обнаруженные дефекты можно локализовать. Если по всем связям g_1 подать одинаковые воздействия, то состояние всех связей l_1 должно быть также одинаковым. Связь l_1 , имеющую неправильное состояние, можно найти по описанному методу за максимальное количество проверок

$$C_{\max} = k \log_2 n,$$

где k - количество неисправных связей.

Граф рис. 2 позволяет обнаруживать дефекты с

$$P \neq 2k, \quad k = 1, 2, \dots$$

После подачи одинакового воздействия по всем связям g_1 все связи l_1 должны иметь одинаковые состояния. Диагноз необходимо начинать с определения состояния связи l_0 с тем, чтобы при дальнейшем диагнозе учесть воздействие связи t . Далее также, как и для графа рис.2а. Необходимое количество проверок

$$C_{\max} = k(1 + \log_2 n).$$

Граф рис. 3 позволяет обнаружить дефекты с кратностью P , определяемой следующим соотношением:

*) . Переход элемента в одно стабильное состояние (любое) рас-
считывается как дефект связи с $P = 1$.

$$\sum_{i=1}^n 2^{n-i} P_i \neq k 2^n, \quad k = 1, 2, \dots$$

Дефект связей элемента α_n можно локализовать с

$$P \neq 2k, \quad k = 1, 2, \dots$$

Дефект связей остальных элементов нельзя локализовать с

так как дефект связи элемента α_1 с $P=1$ аналогичен семейству дефектов α_{1+k} с $P = 2^k$ ($k = 1, 2, \dots; 1 + k = n$). Диагностика проводится также, как и для графа рис. 1. Количество проверок

$$C_{\max} = k \log_2 n.$$

Граф рис. 4 позволяет обнаружить дефекты с кратностью P , определяемой следующим соотношением:

$$\sum_{i=1}^n 2^{n-i} P_i \neq k(2^n - 1); \quad k = 1, 2, \dots$$

Нельзя локализовать дефект связей с $P > 1$ по тем же причинам, что и в предыдущем случае. Диагностика проводится также, как и для графа рис. 1. Количество проверок

$$C_{\max} = k \log_2 n.$$

Граф рис. 5 позволяет обнаружить дефекты по связям t с кратностью

$$P_t \neq 2k; \quad k = 1, 2, \dots,$$

а по всем остальным связям с кратностью, определяемой соотношением

$$\sum_{i=1}^n 2^{n-i} P_i \neq k(2^n - 1); \quad k = 1, 2, \dots$$

Дефект связи можно локализовать с

$$P_t \neq 2k; \quad k = 1, 2, \dots$$

Дефект по остальным связям нельзя локализовать с $P > 1$ по тем же причинам, что и в случае графа рис. 3.

Диагностика проводится также, как и для графа рис. 2. Количество проверок

$$C_{\max} = k(1 + \log_2 n).$$

При построении ДП большое значение имеет рациональный выбор набора операций. Количество операций, входящих в набор, определяет временную избыточность ДП (Т). Целесообразно рассмотреть ЭВМ или устройство как граф, а блок, для которого строится ДП, как его подграф, вершинами которого являются элементы блока, а дугами, ориентированными в направлении распространения сигнала, — связи между элементами.

Также, как и при рассмотрении третьего метода, будем рассматривать неправильную работу вершины как неисправность инцидентных ей дуг. Неисправность может быть двоякого рода. Связь может оказывать воздействие тогда, когда его не должно быть, или не оказывать тогда, когда оно должно быть. Каждая операция проверяет правильность функционирования некоторого числа дуг. Причем одни дуги проверяются на отсутствие неисправностей обоих видов, а другие — только одного из видов. Если неисправность такова, что дуга не оказывает воздействие, и при данной операции воздействия по этой дуге не должно быть, то результат операции будет верным. Несмотря на наличие в графе неисправной дуги. Судить об исправности дуги можно только тогда, когда установлено отсутствие неисправностей обоих видов.

Правильность функционирования дуг графа обычно можно определить только непосредственно по конечному результату операции, если считать, что неисправными могут быть только дуги рассматриваемого подграфа. Подграф оказывает воздействие на работу графа посредством дуг, начала которых принадлежат подграфу, а концы нет. Будем называть такую дугу выходной или, короче, выходом. По аналогии дугу, начало которой не принадлежит подграфу, а конец принадлежит, назовем входной или входом. Назовем вершину "развилкой", если из неё исходит более одной дуги, и "стыком", если в неё заходит более одной дуги. Как правило, на результат одной отдельно взятой операции влияют не все выходы графа. Дуги, составляющие путь от входов до тех выходов, которые не влияют на результаты операции, также не влияют на результаты этой операции. Если такому пути принадлежит развилка, начиная от которой существует путь к выходу, влияющему на результат операции, то дуги от входов до развилки влияют на результат операции.

Обычно каждый подграф участвует в формировании результата многих операций. Каждая ¹-ая операция разбивает множество дуг подграфа на два подмножества, элементами одного из которых

являются дуги, влияющие на результат данной операции, а элементами другого A_1 - дуги, не влияющие на результат этой операции. На количество элементов, попавших в то или иное подмножество, влияет не только вид операции, но и те коды, с которыми она работает. Вместе с тем, различные операции могут приводить к адекватным разбиениям множества дуг подграфа. При чем адекватность соблюдается не только относительно числа элементов подмножеств, но и относительно конкретных дуг. Совокупность видов операций и кодов, с которыми они работают, образуют набор, используемый для диагностики. Одна отдельно взятая операция вместе с кодом, который она использует, дает компоненту набора. С точки зрения уменьшения временной избыточности Т набор не должен содержать компонент, приводящих к попарно адекватным разбиениям. Для построения ДП, как правило, требовались наборы, число компонент которых было равно примерно десяти.

На основе изложенного составляются списки дуг, попавших в подмножества A_1 и \bar{A}_1 . Чтобы определить эффективность создаваемой ДП, необходимо отмечать каждую дугу, попавшую в подмножество A_1 . Дуга относится в подмножество A_1 только в том случае, если по правильному выполнению операции можно сделать заключение об её исправности. Если же при i -той операции дуга проверяется только в одном режиме (отсутствия воздействия или наличие воздействия), то дуга будет представлена двумя элементами: в подмножестве A_1 с пометкой проверенного режима и в подмножестве A_1 с пометкой непроверенного режима.

Если обозначить исход каждой компоненты набора с помощью двоичной цифры, где нуль означает правильный результат, а единица - неправильный, то комбинациям правильных и неправильных исходов всех компонент, входящих в набор, будут соответствовать двоичные числа. Разрядность этих чисел будет равна m , где m - число компонент, а число комбинаций в общем случае может быть равным 2^m . На практике число используемых комбинаций значительно меньше.

При использовании первого метода каждому двоичному числу будет соответствовать некоторый список подозрительных дуг, среди которых находится неисправная. Как видно из описания этого метода, в каждый список попадут дуги, участвующие в формировании результата операций, которым соответствуют единицы двоичного числа для данного списка, и не участвуют в формировании результата операций, которым соответствуют нули.

По второму методу после неправильно выполненной компоненты происходит остановка ЭВМ с указанием этой компоненты. Номер компоненты определяет список подозрительных дуг.

Для удобства пользования дуги целесообразно заменить вершинами, которым они инцидентны. Если в списке находится несколько дуг, инцидентных одной и той же вершине, то все они заменяются одной этой вершиной.

Рассмотрим практическое использование описанной методики. Ввиду ограниченности места будем строить ДП не для блока в целом, а для фрагмента, который выбран с таким расчетом, чтобы показать особенности построения ДП. В качестве примера используем фрагмент блока выработки импульсов сдвига арифметического устройства ЭВМ "Урал-4", функциональная схема которого представлена на рис. 6, а граф, соответствующий этой схеме, — на рис. 7.

Как видно из рис. 7, граф имеет 25 вершин и 55 дуг, из которых семь являются выходами (1, 2, 3, 4, 5, 6, 7). Среди 25 вершин пять являются развилками (1, 2, 3, 4, 5, 20, 25), четырнадцать — стыками (6, 7, 8, 9, 10, 12, 13, 14, 15, 17, 19, 21, 22, 23, 24) и три — развилками и стыками одновременно (11, 16, 18).

Так как число выходов равно семи, то максимальное количество операций, различным образом использующих эти выходы, теоретически равно 2^7 (каждый выход используется или нет). На практике число таких комбинаций меньше. Мы используем всего три, и, как будет показано ниже, использование большего числа операций практически ничего не дает. При использовании трех операций количество комбинаций, соответствующих различным случаям правильного и неправильного выполнения каждой из операций, равно восьми, из которых одно соответствует правильному выполнению всех операций. Таким образом, средняя разрешающая способность теоретически не может быть лучше, чем $25 : 7 = 3,6$. Фактически, как мы увидим, она будет хуже. Для того, чтобы обнулить все однократные отказы, необходимо выбрать такие операции, которые в совокупности использовали бы все выходы. Возьмем в качестве операций следующие: I операция использует выходы 1, 2, 5; II операция — выходы 5, 6, 7; III операция — выходы 3, 4. Для каждой из операций составим разбиение множества дуг и вершин графа на два подмножества A_i и \bar{A}_i .

При составлении разбиений учитываем то обстоятельство, что дуга i участвует в формировании выхода j в том случае, если существует путь, которому принадлежат обе дуги i и j . Если такого пути нет, то дуга i не участвует в формировании выхода.

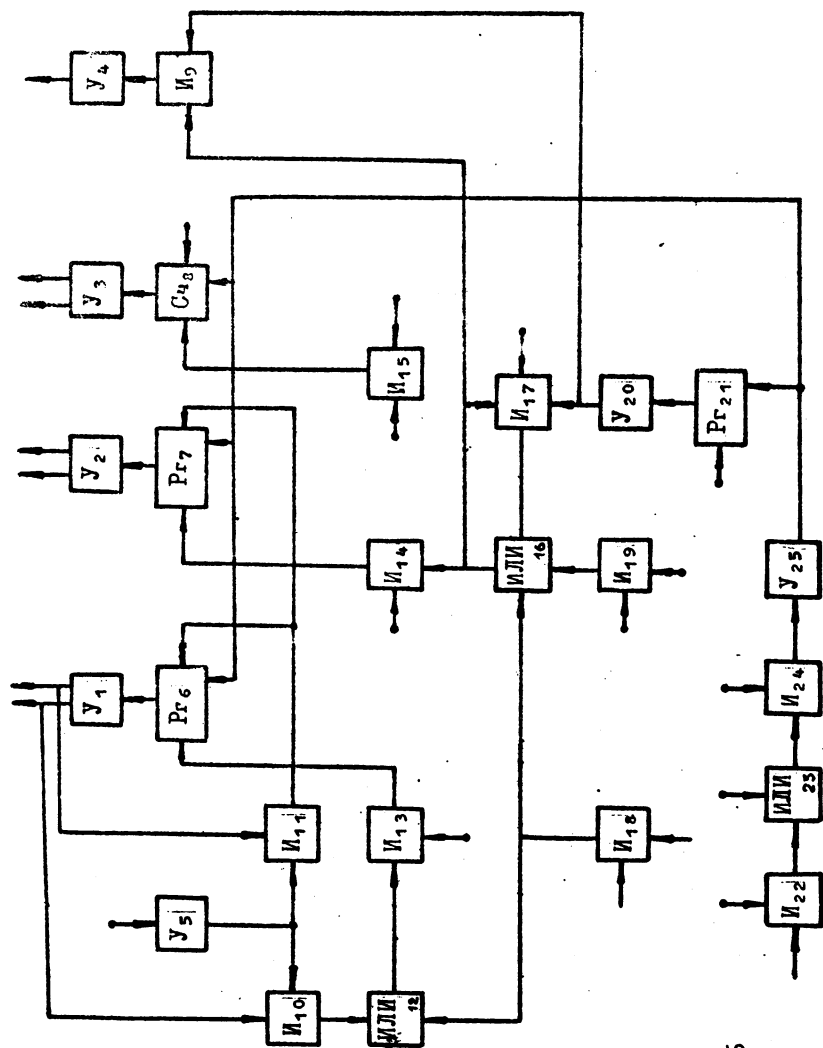


Рис. 6

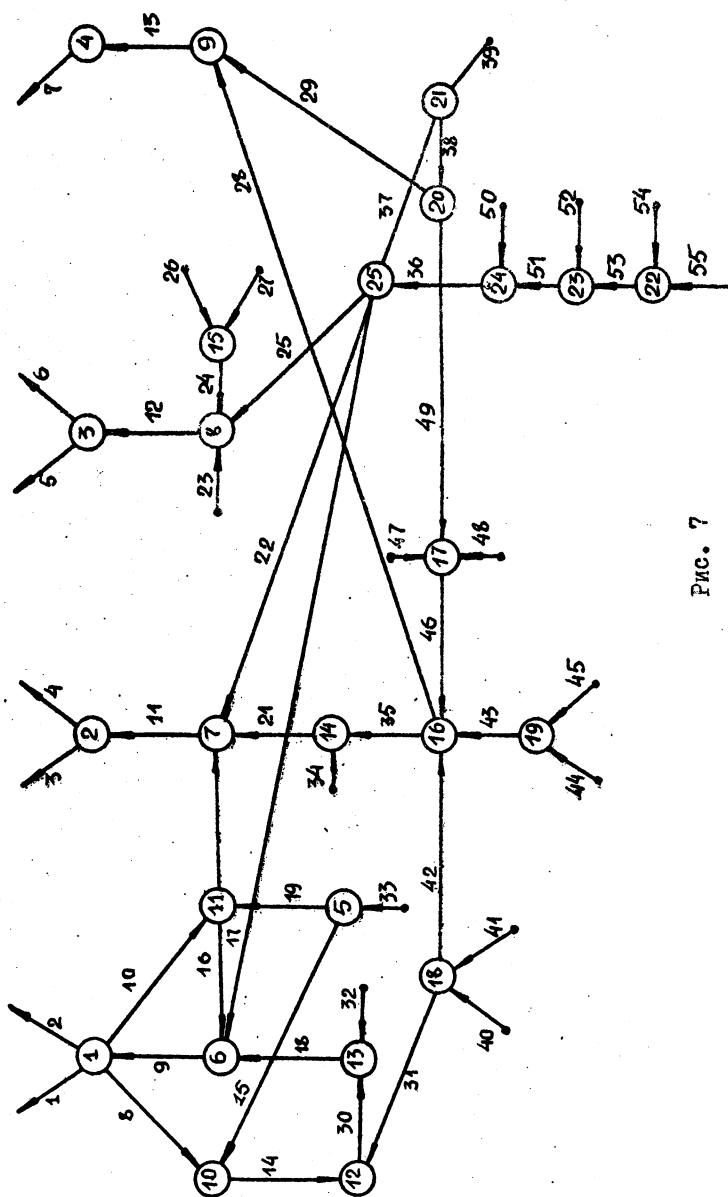


Рис. 7

В рассматриваемом фрагменте объединены следующие функциональные элементы: 22,23,24;16,17,19;10,12,13,18. С учетом этих объединений количество подозрительных сменных ячеек в каждой группе будет равно:

00I - 5;	0II - 5;	10I - 5;	III - 2.
010 - 5;	100 - 1;	110 - 4;	

Для схемы со сложными функциональными связями между элементами такую разрешающую способность можно считать допустимой.

С целью улучшения разрешающей способности увеличим число операций, входящих в набор, до четырех, добавив IV операцию, на которую влияет выход 7. Подмножество A_4 состоит из элементов 1,9,18,32,30,14,8,15,33,31,17,16,19,10,2,3,11,20,22,1,34,35,4,5,12,23,24,26,27,25.

В таблице 3 приведены списки подозрительных дуг и вершины, когда длина набора равна четырем операциям. При составлении таблицы можно учесть, что

$$A_1 \cap A_2 \cap A_3 \cap A_4 = (A_1 \cap A_2 \cap A_3) \cap A_4,$$

и пересечения для трех операций уже имеются.

Из таблицы 3 видно, что улучшение разрешающей способности не произошло. Для того, чтобы присоединение к набору дополнительной операции привело к улучшению разрешающей способности ДП, необходимо, чтобы пересечение подмножеств, соответствующих этой операции, хотя бы с некоторыми пересечениями, получаемыми без этой операции, не были бы ни пустыми, ни собственными. Подобрать такую операцию для выбранного фрагмента невозможно.

В некоторых строках таблицы появились прочерки. Это означает, что такое сочетание правильно и неправильно выполненных операций невозможно. Появление таких строк объясняет тот факт, что количество различных комбинаций будет меньше, чем 2^m (m - число операций в наборе).

При $m \leq 5$ построение диагностических таблиц не вызывает особых трудностей. Однако с увеличением числа m резко возрастает количество различных комбинаций. В том случае, если количество комбинаций превышает число дуг, можно использовать другой метод построения диагностических таблиц, который состоит в следующем:

На результат I операции влияют выходы I, 2, 5. Дуги, не влияющие на формирование ни одного из этих выходов, имеют номера 3, II, 20, 2I, 34, 35, 42, 43, 44, 45, 46, 47, 48, 49, 38, 39, 37, 22, 4, 6, 7, 13, 28, 29. Эти и только эти дуги являются элементами подмножества A_1 . Так как

$$A_1 \cup \bar{A}_1 = A \quad \text{и} \quad A_1 \cap \bar{A}_1 = \Phi,$$

то дуги, не попавшие в этот список, являются элементами подмножества A_1 .

Аналогично элементами подмножества A_2 являются дуги I, 9, 18, 32, 30, 14, 8, 15, 33, 3I, 17, 16, 19, 10, 2, 3, II, 20, 2I, 34, 35, 22, 4.

Элементы подмножества A_3 - I, 2, 5, 12, 23, 24, 26, 27, 25, 6, 7, 13, 28, 29.

Если I и II операции выполняются верно, а III операция неверно, то подозрительные дуги, среди которых находится неисправная, должны быть элементами пересечения трех подмножеств $\bar{A}_1 \cap \bar{A}_2 \cap A_3$. Если обозначить правильное выполнение операции нулем, а неправильное - единицей, то двоичное число, соответствующее данному случаю, будет равно 001. В таблице 2 приведены списки подозрительных дуг и вершин для всех двоичных чисел при выбранной длине набора (три операции).

Т а б л и ц а 2

Двоичное число	Пересечение	Дуги	Вершины
001	$\bar{A}_1 \cap \bar{A}_2 \cap A_3$	3, 4, II, 20, 22, 34, 35	7, II, 14, 16, 25
010	$\bar{A}_1 \cap A_2 \cap \bar{A}_3$	6, 7, 13, 28, 29	3, 4, 9, 16, 20
011	$\bar{A}_1 \cap A_2 \cap A_3$	37-39, 42-49	16, 17, 18-21, 25
100	$A_1 \cap \bar{A}_2 \cap \bar{A}_3$	I, 2	I
101	$A_1 \cap \bar{A}_2 \cap A_3$	8-10, 14-19, 30-33	I, 5, 6, 10-13, 18, 25
110	$A_1 \cap A_2 \cap \bar{A}_3$	5, 12, 23-27	3, 8, 15, 25
111	$A_1 \cap A_2 \cap A_3$	36, 40, 41, 50-55	22, 23, 25, 24

Максимальное количество подозрительных вершин в группе равно девяти. Если считать количество подозрительных ячеек ЭМ, то их число будет меньше, так как несколько функциональных элементов ЭМ могут быть объединены в одну сменную ячейку.

Таблица 3

Двоичное число	Пересечение	Подозрительные элементы	
		дуги	вершины
0001	$\bar{A}_1 \cap \bar{A}_2 \cap \bar{A}_3 \cap A_4$	-	-
0010	$\bar{A}_1 \cap \bar{A}_2 \cap A_3 \cap \bar{A}_4$	1, 3, 4, 11, 20, 22, 84, 85	7, 11, 14, 16, 25
0011	$\bar{A}_1 \cap \bar{A}_2 \cap A_3 \cap A_4$	-	-
0100	$\bar{A}_1 \cap A_2 \cap \bar{A}_3 \cap \bar{A}_4$	-	-
0101	$\bar{A}_1 \cap A_2 \cap \bar{A}_3 \cap A_4$	6, 7, 18, 28, 29	8, 4, 9, 16, 20
0110	$\bar{A}_1 \cap A_2 \cap A_3 \cap \bar{A}_4$	37-39, 42-49	16, 17, 18, 19, 20, 21, 25
0111	$\bar{A}_1 \cap A_2 \cap A_3 \cap A_4$	-	-
1000	$A_1 \cap \bar{A}_2 \cap \bar{A}_3 \cap \bar{A}_4$	1, 2	1
1001	$A_1 \cap \bar{A}_2 \cap \bar{A}_3 \cap A_4$	-	-
1010	$A_1 \cap \bar{A}_2 \cap A_3 \cap \bar{A}_4$	8-10, 14-19, 80-83,	1, 5, 6, 10-19
1011	$A_1 \cap \bar{A}_2 \cap A_3 \cap A_4$	-	-
1100	$A_1 \cap A_2 \cap \bar{A}_3 \cap \bar{A}_4$	5, 12, 23-27	8, 8, 15, 25
1101	$A_1 \cap A_2 \cap \bar{A}_3 \cap A_4$	-	-
1110	$A_1 \cap A_2 \cap A_3 \cap \bar{A}_4$	-	-
1111	$A_1 \cap A_2 \cap A_3 \cap A_4$	36, 40, 41, 50, 55	22, 23, 24, 25

Берется первая дуга и проверяется, в какие из подмножеств она попала. Если дуга попала в A_1 подмножество, то 1-тая цифра двоичного числа, соответствующего этой дуге, будет равна 1, если в \bar{A}_1 , то - нулю. Таким образом, для каждой дуги будет получено n -разрядное двоичное число. Среди этих чисел могут оказаться равные. В таком случае необходимо оставить из всех равных между собой чисел одно, а все дуги, соответствующие им, приписать этому числу.

При использовании II метода методики построения подмножеств A_1 и \bar{A}_1 остается этой же. Период расположения подмножеств A_1 изложен выше.

Для того, чтобы ДП были полным тестом относительно множества дуг, необходимо выполнение следующего соотношения:

$$\bigcup_{i=1}^n A_i = A$$

где n - длина набора.

По описанной методике составление диагностических таблиц можно проводить не только ручным способом но и автоматизированным. В настоящее время разработана программа, которая на основании топологии схемы и задания операций с указанием выходов, влияющих на результат операций, составляет по каждой операции списки элементов, попавших в подмножества A_1 и \bar{A}_1 .

После этого моделируется работа блока, для которого составляется ДП. моделирование производится для всех операций, вошедших в набор. На основании результатов моделирования производится корректировка подмножеств A_1 и \bar{A}_1 . На заключительном этапе программа составляет списки элементов, попавших в соответствующие пересечения.

В основе построения двухступенчатой иерархической системы диагностических программ, разработанной в качестве примера в НИИ МИЭИ в 1964-66 гг для арифметического устройства ЭВМ "Урал-4", лежат три описанных выше метода, используемые для построения ДП на уровнях устройства и блока. Кроме того, на уровне устройства используется система профилактического контроля ЭВМ "Урал-4",.

Система профконтроля разбивает все арифметическое устройство на 15 участков. На каждый участок в среднем приходится 40-50 ламповых ячеек. Диагностика неисправностей на уровне устройства с точностью до блока производится программой-диспетчером с использованием системы профконтроля. Для автоматического управления системой профконтроля в состав системы команд ЭВМ введена команда переключения режимов профконтроля, для реализации которой разработан, изготовлен и отлажен блок автоматического переключения режимов профконтроля. Блок разработан на основе типовых ячеек ЭВМ "Урал-4", в его состав входят 32 ламповых, 29 безламповых ячеек и 48 реле типа РСМ-1, РСМ-2, РСМ-3.

Программа-диспетчер печатает информацию о работоспособности каждого блока, отыскивает блок, в состав которого входит неисправный элемент или элемент, имеющий пониженный запас надежности, и вызывает в оперативную память машины локальную программу поиска неисправного элемента внутри найденного блока.

Локальные программы поиска неисправного элемента построены исходя из того, что все неисправные элементы сосредоточены внутри заданного блока, а все остальные блоки работают нормально. Средняя разрешающая способность ДП равна 3-4 сменным

ячейкам, минимальная — одной ячейке и максимальная — десяти ячейкам. Длина ДП вместе с используемой ими информацией не превышает тысячи ячеек МЗУ. Программа построена с применением двойного счета. Время работы программы составляет доли секунды. Информация о неисправных или подозреваемых ячейках выдается или на сумматор в виде некоторой кодовой комбинации, или на АЦПУ в виде номеров ячеек.

Как показала практика применения ДП, однократные отказы диагностируются во всех случаях, а систематические отказы примерно в 60% случаев.