ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ
1968 г. Выпуск 31

УДК 681.142.019.3.

О ЛОГИЧЕСКОЙ НАДЕЖНОСТИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

А.К. Олефир

Организация контроля и диагностики неисправностей является одной из основных проблем при разработке вычислительных систем [1]. В вычислительных системах (ВС) особенно остро ставится вопрос полной автоматизации контроля и диагностики. В данной работе вводится понятие логической надежности, которое характеризует способность ЭВМ и ВС осуществлять самоконтроль и самодиагностику при наличии неисправностей. Статья посвящена выработке подхода к проектированию ВС, обеспечивающего их высокую логическую надежность.

Вначале определяется логическая надежность отдельно взятой ЭВМ. Затем показано, что при объединении таких ЭВМ в ВС логическая надежность повышается. Для дальнейшего увеличения логической надежности ВС разработан формальный метод выбора соединений между машинами.

I.

Под логической надежностью ЭВМ или ВС будем понимать способность их к самовосстановлению в случае отказа.

Восстановление работоспособности вычислителя (В) ЭВМ или ВС складывается из поиска неисправности и замены неисправного блока исправным. Каждая ЭВМ и ВС снабжается аппаратными или программными средствами локализации неисправностей. Так как замена неисправного блока в настоящее время не является проблемой, будем рассматривать только способность В к автоматической локализации неисправности. Будем считать, что в В возникают одиночные устойчивые отказы, а затем обобщим результаты на одиночные неустойчивые отказы.

Логическую надежность будем оценивать двумя величинами: 1) коэффициентом полноты контроля R_1 ; 2) коэффициентом разрешающей способности R_2 .

При составлении контрольных и диагностических тестов различают основные команды, при выполнении которых работают элементы контролируемой схемы, и вспомогательные команды, с помощью которых построена тест-программа [2]. Для расширяющихся тестов (РТ) введено понятие базы [3, 4]. В состав базы входят элементы, которые участвуют в выполнении вспомогательных команд, с помощью которых можно начать построение расширяющейся тест-программы. Минимальная база является характеристикой конструкции В с точки зрения программной локализации неисправностей.

Будем называть [5] коэффициентом полноты контроля отношение числа элементов, локализуемых расширяющейся тест-программой в случае их неисправности, ко всему числу N элементов В:

$$R_1 = \frac{N - N_{баз}}{N} ,$$

где $N_{баз}$ – количество базовых элементов в В.

Для количественной оценки разрешающей способности обратимся к известному методу локализации неисправностей [2,6], который состоит в том, что блоки ЭВМ проверяются целиком с помощью коротких контрольных тестов, а диагностическая программа для данного блока начинает работу только после установления факта его неисправности. Этот метод, который можно назвать локализацией по уровням сложности, обобщим для всей структуры вычислительной системы. Если на уровне сложности I контролируется целиком исправность В, то диагностикой будем называть локализацию элементов уровня 2, из которых состоит В. Так, при контроле исправности ВС диагностикой будем считать локализацию неисправной элементарной машины (ЭМ) данной ВС. Таблица I иллюстрирует один из способов разбивки В по уровням сложности.

Пусть В уровня k состоит из N_0 объектов уровня $k+1$, N_i количество этих объектов в i -ой локализованной группе, $N_{гр}$ количество групп. Тогда коэффициент разрешающей способности будет равен: $R_2 = 1 - \frac{1}{N_0} \sum_{i=1}^{N_{гр}} (N_i - 1)$. Коэффициент разрешающей способности R_2 = I, если существует возможность локализации с точностью до одного элемента следующего уровня.

Таким образом, понятие логической надежности В вклю-

себя информацию о точности автоматической локализации неисправности и полноте охвата В контролем.

Таблица I

Уровни локализации	Объект контроля	Объект диагностики	Комментарии
Уровень I Уровень 2 Уровень 3	ВС ЭМ команда	ЭМ команда устройство	команда выполняется двумя устройствами: управляющим и операционным
Уровень 4	устройство	блок,вырабатывающий микрооперацию	
Уровень 5	блок,вырабатывающий микрооперацию		

Рассмотрим один тип ВС, систему команд, а также конструкцию ЭМ, из которых построена ВС.

Предполагается, что развитие вычислительных систем будет идти по пути создания однородных ВС [I,7], образованных многократным повторением одного и того же элемента (ЭМ) и его связей. Рассмотрим одномерную двустороннюю ВС с переменной структурой, построенную на основе существующих ЭВМ. В этой системе ЭМ связаны друг с другом таким образом, что к каждой машине подходят два и выходят только два канала связи (рис. I). Любая ЭМ в такой системе может быть определена с помощью одной координаты (номер ЭМ) и может принимать и передавать информацию как в одном, так и в другом направлении. Переменная структура создается за счет того, что элементарные машины объединяются в подсистемы [I].

В систему команд ЭМ вводятся дополнительные (системные) команды. Эти команды опишем, пользуясь символикой, близкой к предложенной в [8].

Для обозначения регистров будем пользоваться общепринятыми сокращениями. Угловыми скобками будем обозначать избирательную схему. Наименование, стоящее перед скобками, относится к устройству, в состав которого входит регистр, заключенный в скобки. Передачу между двумя регистрами будем обозначать стрелкой

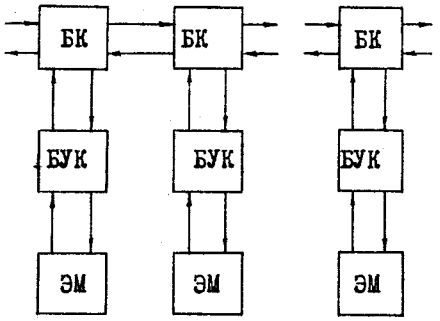

Рис. I. Блок-схема одномерной двусторонней ВС:БК – блок коммутатора; БУК – блок управления коммутатором; ЭМ – элементарная ЭВМ.

($\rightarrow$). Слева от стрелки располагается источник информации, справа – получатель.Действия отделяются друг от друга запятой. Окончание операций, выполнявшихся по некоторому условию, отмечается точкой с запятой. Указание о времени или условии передачи отделяется от содержания передачи чертой (|).

I. Групповая команда выдачи ГКВ, по которой из ЭМ выдается последовательно,начиная с адреса $A_{ОЗУ}$, L_M слов:

$$ГКВ \mid ОЗУ(\langle A\rangle) \rightarrow БК;\ A=1,2,\ldots,L_M.$$

2. Групповая команда приема ГКП, по которой ЭМ принимает последовательно, начиная с адреса $A_{ОЗУ}$, L_M слов:

$$ГКП \mid (БК) \rightarrow ОЗУ(\langle A\rangle);\ A=1,2,\ldots,L_M.$$

3. Команда обобщенного условного перехода ОУП, которая служит для изменения хода вычислений в зависимости от результатов, полученных в N' машинах:

$$ОУП \wedge ЭМ_1(\omega) \wedge ЭМ_2(\omega) \wedge \ldots \wedge ЭМ_{N'}(\omega)=1 \mid ЭМ_i(РК) \rightarrow ЭМ_j(СчК);$$

где ω – признак, на который реагирует команда ОУП;

РК – регистр команд;

СчК – счетчик команд.

4. Команда обобщенного безусловного перехода ОБП.

$$ОБП \mid ЭМ_i(РК) \rightarrow ЭМ_j(РК);$$

или

$$ОБП \mid ЭМ_i(РК) \rightarrow ЭМ_j(СчК);$$

где i и j принимают ряд значений в зависимости от настройки системы, и в общем случае $i \neq j$.

5. Команда настройки КН, которая присваивает элементарным машинам признаки для разбиения ВС на подсистемы. Наличие неисправной ЭМ в ВС устанавливается с помощью тестового или программно-логического контроля [9]. Управляющая программа выделяет в подсистему неисправную машину $ЭМ_i$ и соседнюю с ней (рис. 2): $ЭМ_{i-1}$ и $ЭМ_i$ или $ЭМ_i$ и $ЭМ_{i+1}$. В этой подсистеме из двух ЭМ поиск отказавшего объекта в неисправной ЭМ будет осуществляться с помощью машины-партнера. Исправная ЭМ проверяет базу неисправной ЭМ,и в случае исправной базы дальнейшая локализация

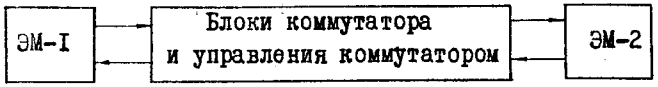

Рис. 2. Подсистема из двух ЭМ : исправной и неисправной.

производится самой неисправной ЭМ. Поэтому, не нарушая общности, во всех дальнейших рассуждениях будем рассматривать ВС, состоящую только из двух ЭМ. На рис. 3 изображена структурная схема одной ЭМ (трехадресной). Будем считать, что ЭМ не имеет

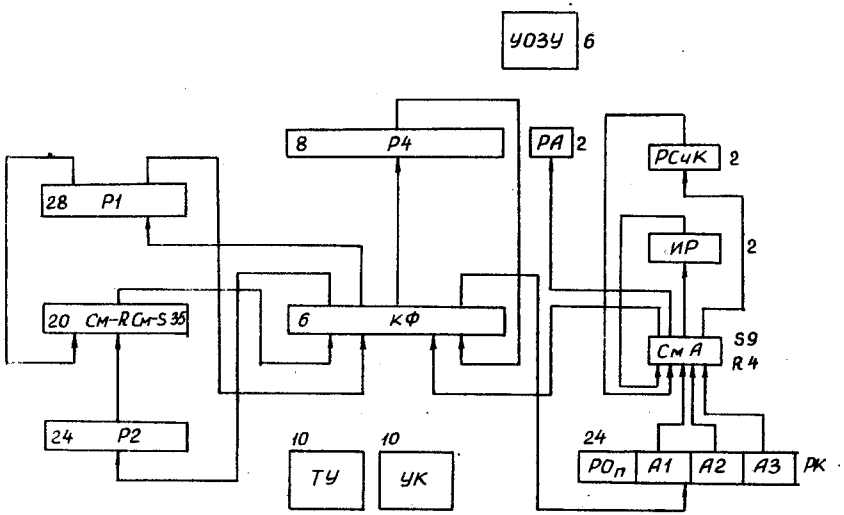

Рис.3. Структурная схема ЭМ . РI, Р2 - приемные регистры АУ; См- R - регистр сумматора; См- S - цепи поразрядного сложения и сквозного переноса; РЧ, РА - регистры числа и адреса ОЗУ; УОЗУ - управление ОЗУ; КФ - кодовые формирователи; РСчК - регистр счетчика команд; ИР - индекс-регистр ; СмА- R - регистр сумматора адреса; СмА- S - цепи поразрядного сложения и сквозного переноса; РК - регистр команд; РОп - регистр операций; АI, А2, А3 - регистры адресов первого, второго и третьего; ТУ - тактирующее устройство; УК - управление командами.

внешних устройств и команда выполняется двумя типами устройств: управляющим и операционными (регистрами, сумматорами). Цифра, проставленная рядом с устройством, означает относительный вес устройства - по отношению к общему весу всех элементов ЭМ. Такое представление ЭМ позволяет выделить I5 (для структурной схемы на рис. 3) устройств с общим весом I90. Будем считать также, что управляющие схемы, специфичные для отдельных команд, сосредоточены в устройстве УК (управление командами).

Рассмотрим вопрос о выборе базы для отдельной ЭВМ без учета связи ее с другими машинами.

База состоит из двух принципиально различных частей:

$\mathcal{N}'_{баз}$ - элементы минимального набора вспомогательных команд, с которых начинает работу РТ. Состав этих вспомогательных команд можно изменять, поэтому можно изменять и набор элементов, которые должны быть исправными перед работой РТ.

$\mathcal{N}''_{баз}$ - элементы тех устройств, которые работают при выполнении любых команд - счетчик команд, регистр команд, цепи синхронизации, управление оперативной памятью и др.

Интуитивно ясно, что наиболее сложные команды машины (например, многотактные команды - умножение, извлечение квадратного корня)-не войдут в состав базы, поэтому будем искать базу из состава следующих простых команд.

I. Перепись кода (ПК).
2. Поразрядное сложение (ПС).
3. Условный переход (УП).
4. Безусловный переход (БП).
5. Команда останова (Ост).
6. Изменение содержимого индекс-регистра (ИИР).
7. Циклическое сложение (ЦС).
8. Команда сдвигов (Сдв).
9. Логическое умножение (ЛУ).
I0. Логическое сложение (ЛС).
II. Условный переход после цикла (ПЦ).

В связи с тем, что для дальнейших рассуждений понадобится описание некоторых команд, приведем их в использованной выше символике.

I. Перепись кода

$$\begin{aligned} ПК \mid\ & A1 \to СмА, \\ & СмА \to РА, \\ & ОЗУ(<РА>) \to Р1, \\ & А3 \to СмА, \\ & СмА \to РА, \\ & Р1 \to ОЗУ(<РА>); \end{aligned}$$

2. Сравнение кодов - поразрядное сложение - с выработкой признака $\omega = I$ в случае равенства кодов.

$$\begin{aligned} ПС \mid\ & A1 \to СмА, \\ & СмА \to РА, \\ & ОЗУ(<РА>) \to Р1, \\ & А2 \to СмА,\ Р1 \to См, \\ & СмА \to РА, \end{aligned}$$

$$ОЗУ(\langle РА\rangle) \rightarrow Р2,$$
$$АЗ \rightarrow СмА,\ См \oplus Р2 \rightarrow См;$$
$$ПС \wedge См = 0 \mid \text{„}1\text{“} \rightarrow Рег\ \text{„}\omega\text{“};$$
$$ПС \mid СмА \rightarrow РА,$$
$$См \rightarrow ОЗУ(\langle РА\rangle);$$

Здесь знаком $\oplus$ обозначено поразрядное сложение.

3. Условный переход по $А2$, если ω = I.

$$УП \mid А2 \rightarrow СмА,$$
$$УП \wedge \omega{=}1 \mid СмА \rightarrow СчК;$$

4. Безусловный переход по $А2$.

$$БП \mid А2 \rightarrow СмА,$$
$$СмА \rightarrow СчК;$$

5. Команда останова.

$$Ост \mid А1 \rightarrow СмА,$$
$$СмА \rightarrow РА,$$
$$ОЗУ(\langle РА\rangle) \rightarrow Р1,$$
$$А2 \rightarrow СмА,$$
$$СмА \rightarrow РА,$$
$$ОЗУ(\langle РА\rangle) \rightarrow Р2;$$

Построим ориентированный граф (рис. 4) $\mathcal{G}_K(K,\Gamma)$, поставив в соответствие каждой команде вершину $к_i \in K$ и считая, что $\Gamma^{-1}_{к_i}$ составляет набор вспомогательных команд, если $к_i$ – основная.

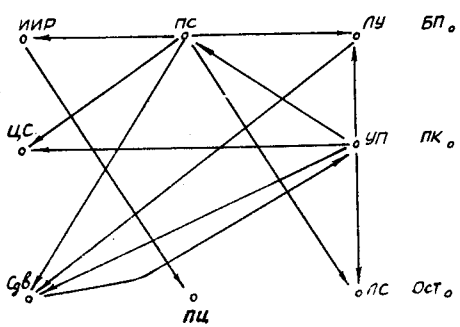

Рис. 4. Зависимость между основными и вспомогательными командами.

Вершины БП, ПК и Ост должны быть соединены со всеми другими вершинами, но эти соединения не показаны, чтобы не загромождать рисунок. Легко убедиться, что, выбрав в качестве базовых команды БП, ПК, Ост, УП, можно обойти (проверить) все другие вершины (команды). В состав базы войдут также РК, РСчК, КФ, РЧ, РА, УОЗУ, ТУ, УК.

Оценим логическую надежность отдельной ЭВМ.

I. Коэффициент полноты контроля R_1

$$R_1 = \frac{N - N_{баз}}{N} = 0{,}47,$$

где $N_{баз}$ – относительный вес устройств РК, РСчК, КФ, РЧ, РА, УОЗУ, ТУ и элементов, участвующих в выполнении команд БП, ПК, Ост, УП (УК, РП, СмА–R).

2. Коэффициент разрешающей способности R_2.

а) для РТ, локализующего неисправную команду:

$$R_2' = 1 - \frac{1}{N_0} \sum_{i=1}^{N_{ир}} (N_i - 1) = 0{,}9,$$

где N_0 = 30 – число команд ЭМ без модификаций,

N_i = 4 – число базовых команд;

б) для РТ, локализующего неисправное устройство:

$$R_2'' = 1 - \frac{1}{N_0} \sum_{i=1}^{N_{ир}} (N_i - 1) = 0{,}4,$$

где N_0 = I5 – количество устройств ЭМ;

N_i = I0 – количество базовых устройств.

Считается, что на данном уровне диагностики РТ локализует с точностью до команды или до устройства. Заметим, что выбор базы в отдельной ЭВМ сводится к выбору минимального набора базовых команд, поскольку определенная часть базы ($N''_{баз}$) не допускает варьирования, а целиком определяется конструкцией ЭВМ.

4.

Наличие связей между ЭМ в ВС позволяет увеличить логическую надежность всей ВС. Рассмотрим возможности диагностики базы одной ЭМ, считая, что ВС состоит из двух ЭМ – исправной и неисправной.

Не показывая промежуточных устройств – коммутатора и согласующих – установим следующие массовые каналы между двумя ЭМ:

$$ЭМ\text{–}I(КФ) \rightarrow ЭМ\text{–}II(РЧ),$$
$$ЭМ\text{–}II(КФ) \rightarrow ЭМ\text{–}I(РЧ).$$

Примем, что относительный вес оборудования, необходимого для реализации этих вариантов, будет равен сумме весов соединяемых устройств. В данном случае он составит I4. Задачей диагностики, которую решает машина – партнер по отношению к неисправной ЭМ, является локализация неисправного устройства на том уровне, на котором производится его замена. В табл. 2 показана последовательность испытаний, проводимых в неисправной ЭМ, и набор устройств, отделяемых от всей ЭМ при данном испытании (N_i). В этом РТ сначала проверяются команды системы, а затем – базовые команды ВС. Оценим логическую надежность этой ВС.

I) Коэффициент полноты контроля R_1.

$$R_1 = \frac{N - N_{баз}}{N} = 0{,}72,$$

Таблица 2

	Испытания	Локализуемые устройства	Комментарии
I.	КН, ОБП, ГКВ	УК, РА, УОЗУ, РК, СмА-R, ТУ	Производится выдача константы из неисправной ЭМ.
2.	КН, БП, ГКВ	КФ, РЧ	Проверка разрядов на константах
3.	ОБП, ГКП, ГКВ	СчК	
4.	ПК	РI	
5.	БП	управление	
6.	УП	управление	

где N - вес устройств ЭМ с учетом каналов связи;

$N_{баз}$ - вес устройств, с которых начинает работу РТ (УК, РА, УОЗУ, РК, СмА-R, ТУ).

2. Коэффициент разрешающей способности R_2''.

$$R_2'' = 1 - \frac{1}{15}(5+1) = 0{,}6,$$

так как $N_1 = 6$ (УК, РА, УОЗУ, РК, СмА-R, ТУ);

$N_2 = 2$ (КФ, РЧ).

Увеличение логической надежности ВС по сравнению с отдельно взятой ЭВМ достигается за счет того, что при испытании команд системы производится разделение устройств, входящих в базу ЭВМ.

5.

Подход к выбору базы, рассмотренной в §§ 3,4, основан на эвристических соображениях: берутся устройства ($N''_{баз}$), общие для всех команд, и оптимизируется набор вспомогательных команд для построения расширяющегося теста. В отдельной ЭВМ база проверяется вручную или с помощью дополнительных приборов и устройств, в ВС - с помощью машины-партнера. При выбранной базе и установленных каналах между ЭМ увеличение разрешающей способности РТ может быть достигнуто за счет дополнительных связей между ЭМ. Эти связи позволяют разделить неисправности в самой базе.

Ниже предлагается другой метод, при котором по критерию минимального веса оптимизируется набор связей между ЭМ, обеспечивающий заданную разрешающую способность РТ.

Метод состоит в следующем. Составляется формальное описание ЭВМ и группы команд, из которых должна быть построена база. Составляется также список всех устройств ЭВМ на уровне детализации, который считается достаточным для построения базы. Команды упорядочиваются по возрастанию суммарного веса устройств, которые участвуют в выполнении этих команд. Для каждого устройства списка находится команда (или группа команд), с помощью которой обеспечивается проверка этого устройства. При этом фиксируются все вспомогательные устройства. Результаты сводятся в таблицу (таблица контроля), с помощью которой строится минимальная база.

Дадим определение операции контроля, уточнив известное разделение испытания на основные и вспомогательные действия или команды [2]. Будем считать, что основные действия состоят из переписи информации из запоминающего устройства (ОЗУ или регистра) в контролируемое, выполнения операции (микрооперации) в контролируемом устройстве, переписи полученной информации в запоминающее устройство; вспомогательные действия - из анализа информации, находящейся в запоминающем устройстве.

Таким образом, операцию контроля можно определить как совокупность действий по созданию канала для основных и вспомогательных пересылок информации. Эти пересылки можно записать упрощенно в виде последовательных передач между регистрами (устройствами). Так, проверка работы устройства РI записывается следующим образом:

основные пересылки:

$$ОЗУ(\langle A^1\rangle) \to РЧ \to КФ \to Р1 \to См \to R;$$

вспомогательные пересылки:

$$ОЗУ(\langle A^1\rangle) \to РЧ \to КФ \to Р2 \oplus См \to См \to R;$$
$$См = 0|1 \to Рег\ \omega'';$$

Выполнение этих пересылок возможно, например, с помощью команды поразрядного сложения ПС.

Такая формальная запись операции контроля позволяет легко построить в виде таблицы или ориентированного графа зависимость между основными и вспомогательными устройствами. В табл.3 строки соответствуют проверяемым устройствам, а столбцы - вспомогательным. Элемент таблицы b_{ij} = I тогда и только тогда, когда j-ое устройство является вспомогательным для i-го устройства. Теперь можно отвлечься от конструкции ЭВМ и оперировать только с табл. 3. Задача выбора минимальной базы содержательно формулируется следующим образом: выбрать совокупность устройств с минимальным весом, начиная с которой можно обойти операцией контроля все остальные устройства.

Таблица 3

Веса устройств	28	20	35	24	6	24	2	I6	2	4	9	Общий вес вспом. устройств
вспомог. / Осн. устр.	PI	СМ-R	СМ-S	Р2	КФ	РК	ИР	ОЗУ	РСчК	СмА-R	СмА-S	
PI		I		I	I	I		I		I		94
См-R	I		I	I	I	I		I		I		I37
См-S	I	I		I	I	I		I		I		I22
Р2	I	I			I	I		I		I		98
КФ	I	I		I		I		I		I		II6
РК					I		I	I	I	I	I	63
ИР						I			I	I		30
ОЗУ	I	I		I	I	I			I	I		I08
РСчК	I				I	I		I		I	I	87
СмА-R	I	I		I	I	I		I				II8
СмА-S	I	I		I	I	I	I	I		I		I24

6.

Рассмотрим алгоритм выбора базы при объединении ЭВМ в ВС.

Построим ориентированный конечный граф $\mathcal{L} = (X, \Gamma)$, поставив в соответствие каждому устройству ЭВМ вершину $a_i \in X$ ($i = 1, 2, \ldots, n$) с ее весом, выраженным вещественным положительным числом. Будем считать, что контролируется устройство a_i с помощью устройств $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$, если $\Gamma^{-1}_{a_i} = \{a_{i_1}, a_{i_2}, \ldots, a_{i_k}\}$. В этом случае Γ_{a_i} и $\Gamma^{-1}_{a_i}$ при любом i – непустые множества.

Дадим следующее определение. Вершина a_j ($a_j \notin A$) называется присоединимой к набору вершин $A = \{a_{i_1}; a_{i_2}; \ldots; a_{i_k}\}$, если $\Gamma^{-1}_{a_j} \subset A$. Построим вспомогательный граф $\mathcal{L}_1 = \{X_1, \Gamma_1\}$, где $X_1 = \{A_1, A_2, \ldots, A_\tau\}$ – множество всех возможных непустых наборов вершин a_i (легко видеть, что $\tau = 2^n - 1$), и вершина A_j тогда и только тогда принадлежит множеству $\Gamma_1 A_\ell$, когда набор A_j можно получить из набора A_ℓ добавлением какой-то одной присоединимой к нему вершины a_i.

Для удобства будем считать, что $A_\tau = \{a_1, a_2, \ldots, a_n\} = X$. Весом вершины A_j ($j = 1, 2, \ldots, \tau$) будем считать суммарный вес всех вершин a_i, входящих в набор A_j. Вершину A_j назовем полной, если $A_\tau \in \Gamma_1 A_j$, т.е. если в графе $\mathcal{L}_1$ существует хотя бы один путь из вершины A_j в вершину A_τ.

Требуется найти полную вершину графа $\mathcal{L}_1$, имеющую минимальный вес.

Алгоритм. Дадим описание алгоритма, наиболее удобного для практического применения.

Пусть $m = \min |\Gamma^{-1}_{a_i}|$, где через $|A|$ обозначена мощность множества A [I0]. Очевидно, что любой набор, содержащий менее чем m вершин графа $\mathcal{L}_1$, будет изолированной вершиной графа $\mathcal{L}_1$.

I этап. Рассмотрим множество $\mathcal{Y} = \{\Gamma^{-1}_{a_1}; \Gamma^{-1}_{a_2}; \ldots; \Gamma^{-1}_{a_n}\}$. Выберем из него все элементы, представляющие собой наборы из m вершин графа $\mathcal{L}$. Получаем множество

$$\mathcal{Z} = \{\Gamma^{-1}_{a_{i_1}}; \Gamma^{-1}_{a_{i_2}}; \ldots; \Gamma^{-1}_{a_{i_\gamma}}\}.$$

Каждый элемент множества $\mathcal{Z}$ – это некоторая вершина графа $\mathcal{L}_1$. Выбираем из этих вершин вершину с наименьшим весом. Пусть это будет вершина A_{j_1}. Из множества $\Gamma_1 A_{j_1}$ выбираем произвольную вершину A_{j_2}, затем из множества $\Gamma_1 A_{j_2}$ выбираем произвольную вершину A_{j_3} и т.д., до тех пор, пока не получим вершину A_{j_s}, для которой $\Gamma_1 A_{j_s} = \Phi$. Тогда, как легко видеть, если $A_{j_s} = A_\tau = X$, то вершина A_{j_1} будет полной, а если $A_{j_s} \neq A_\tau$ – неполной. В первом из этих двух случаев вершину A_{j_1} назовем отмеченной. Если у нас уже имеется отмеченная вершина, то другая вершина может стать отмеченной вместо нее лишь тогда, когда она, во-первых, полная, а во-вторых, имеет меньший вес. Поэтому, если вершина A_{j_1} отмечена, то в множестве $\mathcal{Z}$ больше нельзя отметить ни одну вершину. Легко видеть, что целью нашего алгоритма является отыскание такой отмеченной вершины, для которой уже нельзя найти вершину-заменитель.

Вернемся теперь к тому случаю, когда вершина A_{j_1} неполная, а значит, и неотмеченная. Если множество $\mathcal{Z} \setminus A_{j_1}$ непустое, то выбираем из него вершину A_{j_2} с наименьшим весом и рассуждаем относительно нее так же, как мы рассуждали выше относительно вершины A_{j_1}. Снова имеем две возможности: I) вершина A_{j_2} становится отмеченной, и можно переходить ко второму этапу применения нашего алгоритма и 2) вершину A_{j_2} нельзя сделать отмеченной, и тогда необходимо рассмотреть множество $\mathcal{Z} \setminus \{A_{j_1}; A_{j_2}\}$, извлечь из него, если оно непустое, вершину A_{j_3} с наименьшим весом и т.д. В конце концов мы либо исчерпаем все множество $\mathcal{Z}$, либо найдем в нем отмеченную вершину.

II этап. На предыдущем этапе мы либо найдем отмеченную вершину A_ε, либо докажем, что в множестве $\mathcal{Z}$ нельзя выбрать отмеченную вершину.

Рассмотрим теперь множество Y_1 всех таких вершин графа $\mathcal{L}_1$, каждая из которых представляет собой набор из $(m+1)$ вершин графа $\mathcal{L}$.

Если у нас уже имелась отмеченная вершина A_ε, то теперь в множестве Y_1 выбираем подмножество Z_1 всех таких вершин, каждая из которых имеет вес, меньший, чем вес вершины A_ε. Если Z_1 пусто, то вершина A_ε и будет, очевидно, искомой полной вершиной с минимальным весом. Если же Z_1 непусто, то выбираем в Z_1 вершину $A_{j'}$ с наименьшим весом и исследуем ее на полноту. Если вершина $A_{j'}$ окажется полной, то мы сделаем ее отмеченной вместо вершины A_ε и перейдем к третьему этапу алгоритма. В противном случае нужно из множества $Z_1 \setminus A_{j'}$, если оно еще не пустое, выбрать вершину с наименьшим весом, исследовать ее на полноту и т.д., до тех пор, пока мы либо заменим вершину A_ε новой отмеченной вершиной, либо исчерпаем все множество Z_1.

Если же на предыдущем этапе алгоритма мы не получили отмеченную вершину, то на рассматриваемом этапе нужно искать полную вершину в множестве Y_1, извлекая из него последовательно вершины с наименьшим весом.

Второй этап закончится получением одной из следующих четырех ситуаций:

1) на первом этапе была отмечена вершина A_ε и на втором этапе ее не удалось заменить (в этом случае, если множество Z_1 было пустым, процесс применения алгоритма заканчивается, т.к. вершина A_ε является искомой, если же Z_1 было непусто, то нужно перейти к третьему этапу алгоритма);

2) вершина A_ε, отмеченная на первом этапе, на втором этапе оказалась замещенной другой отмеченной вершиной;

3) ни на первом, ни на втором этапе не удалось найти отмеченную вершину;

4) на первом этапе отмеченной вершины не было, но на втором этапе она появилась.

В каждой из трех последних ситуаций необходимо перейти к третьему этапу алгоритма.

<u>III этап</u> алгоритма аналогичен второму этапу, только начинается он с рассмотрения множества Y_2 всех таких вершин графа $\mathcal{L}_1$, каждая из которых представляет собой набор из $(m+2)$ вершин графа $\mathcal{L}$.

Работа алгоритма закончится тогда, когда на некотором этапе будет отмечена такая вершина, которую нельзя заменить на следующих этапах, причем, что особенно важно, эти следующие этапы нужно рассматривать лишь до тех пор, пока на одном из них множество Z_i не окажется пустым.

7.

В качестве примера рассмотрим работу описанного алгоритма для оптимизации связей между ЭМ, используя таблицу 3. В этой таблице РА, РЧ и УОЗУ объединены в одно устройство ОЗУ с суммарным весом этих устройств, а УК и ТУ не включены, так как локализация неисправностей будет проводиться только для операционных устройств.

<u>I этап.</u> Выбираем вершины A_j в порядке возрастания веса и проверяем на полноту. Нетрудно убедиться, что вершины $A_1 = \{РК, РСчК, СмА-R\}$ (вес 30)

$A_2 = \{КФ, ИР, ОЗУ, РСчК, СмА-R, СмА-S\}$ (вес 63)

$A_3 = \{Р1, КФ, РК, ОЗУ, СмА-R, СмА-S\}$ (вес 87)

окажутся неполными. Вершина $A_4 = \{См-R, Р2, КФ, РК, ОЗУ, СмА-R\}$ (вес 94, $m = 6$) является полной, так как $\Gamma_1 A_4 = A_5$, где A_5 получается присоединением вершины $Р1$ к A_4; $\Gamma_1 A_5 = A_6$, где A_6 получается присоединением вершины $См-S$ к A_5 и т.д.

Последовательным присоединением вершин ИР, СмА-S, РСчК получаем A_7, A_8, A_9.

<u>II этап.</u> Отбирается множество вершин $A_{10}, A_{11}, \ldots$, вес которых меньше, чем вес A_4, а $m > m\,|A_4|$.

В данном примере эти множества оказываются неполными, поэтому вершину A_4 следует считать отмеченной.

Итак, между ЭМ необходимо организовать связи, позволяющие вызывать информацию непосредственно с устройств $См-R$, $Р2$, $КФ$, $РК$, $ОЗУ$, $СмА-R$ (вес 102). Тогда локализация неисправности будет производиться с точностью до операционного устройства.

Оценим логическую надежность ВС с этими связями.

1. Коэффициент полноты контроля R_1.

$$R_1 = \frac{N - N_{баз}}{N} = 0{,}93,$$

где N – вес устройств и связей; $N_{баз}$ – вес ТУ и УК.

2. Коэффициент разрешающей способности R_2''.

$$R_2'' = 1 - \frac{1}{N_0} \sum_{i=1}^{N_{гр}} (N_i - 1) = 0{,}8,$$

так как $N_1 = 3$ (УОЗУ, РА, РЧ); $N_2 = 2$ (ТУ, УК).

Однако представляется целесообразным пойти на некоторое ухудшение коэффициента R_2'' в пользу сокращения веса, а, стало быть, и стоимости оборудования связей. Установим следующие связи (для одной ЭМ):

ЭМ-I (КФ) — ЭМ-II (РК)	(вес 24)
ЭМ-II (См А-Р) — ЭМ-I (РЧ)	(вес 4)
ЭМ-II (РК) — ЭМ-II (РЧ)	(вес 32)

Здесь остаются неразделенными неисправности в группах: РЧ, РА, УОЗУ; ТУ, УК; Р2, См-Р. В этом случае коэффициент разрешающей способности R_2'' уменьшится:

$$R_2'' = 1 - \frac{1}{N_0} \sum_{i=1}^{N_{гр}} (N_i - 1) = 0{,}73 ,$$

так как $N_1 = 3$ (УОЗУ, РА, РЧ), $N_2 = 2$ (ТУ, УК), $N_3 = 2$ (Р2, См-Р).

При этом вес соединяемых устройств снизился со 102 до 60 (24 + 4 + 32). Последний вариант связи представляет особый интерес, так как высокая логическая надежность достигнута при организации информационного канала через управляющую часть ЭВМ (РК). Можно сделать вывод, что в вычислительной системе с высокой логической надежностью информационные связи между ЭМ должны осуществляться через их управляющие части.

Заключение

1. В работе предложен формальный метод оптимизации структуры связей между ЭМ с целью повышения логической надежности ВС. Вначале составляется описание команд ЭМ и фиксируются контрольные операции. Затем строится так называемая таблица контроля. Таблица контроля является исходной для работы алгоритма выбора минимальной базы.

2. Результаты оптимизации, реализованные в конструкции ВС, обеспечивают увеличение логической надежности, что видно из сводной таблицы 4.

Дальнейшее увеличение коэффициента полноты контроля возможно за счет дублирования базовых команд вместе с тактирующими и управляющими устройствами в каждой ЭМ.

3. Алгоритм выбора минимальной базы применен на уровне устройств ЭВМ. В случае необходимости этот же алгоритм можно применить и для более точной детализации.

4. В работе рассмотрен один класс неисправностей - устойчивые отказы. Так как выбор каналов не связывался с характеристикой системы [13], результаты могут быть обобщены на более широкий класс неисправностей - перемежающиеся отказы. В этом случае задача испытательной программы состоит в воспроизведении перемежающихся отказов при различных кодовых комбинациях.

Таблица 4

Вычислитель	R_1	R_2''
I. ЭВМ (§ 3)	0,47	0,4
2. ВС без специальных связей (§ 4)	0,72	0,6
3. ЭВМ со специальными схемами [II]	0,85	
4. ВС с дополнительными каналами (§ 7)	0,93	0,8-0,73
5. ВС со специальным оборудованием в центральном процессоре (ЭВМ-360, см [I2])	0,9-0,97	

5. В работе показан формальный выбор базы для однородной ВС, но результаты применимы для оптимизации связей и в системах из разнородных машин. В этом случае база определяется раздельно для каждого типа ЭВМ.

Литература

I. Э.В. Евреинов, Ю.Г. Косарев. Однородные универсальные вычислительные системы высокой производительности. Изд-во "Наука", Новосибирск, 1966г.

2. Г.А. Миронов. Испытательные программы для контроля электронных цифровых машин. М., Изд-во "Наука", 1964.

3. А.К. Олефир, А.И. Гракин, Г.А. Сенькина. О возможности программного поиска неисправностей в ЭЦВМ.- Материалы конференции молодых ученых СО АН СССР и специалистов Новосибирска и области, Новосибирск, 1962.

4. А.К. Олефир. О поиске неисправных элементов в системе из двух ЭВМ- Вычислительные системы, Новосибирск, 1963, выпуск 6, стр. 21-31.

5. А.К. Олефир. Об эффективности программного контроля в вычислительной системе "Минск-222".- Труды Симпозиума "Вычислительные системы", Новосибирск, 1967.

6. А.К. Олефир. Организация тестового контроля для системы из двух ЭВМ -Вычислительные системы, Новосибирск, 1964, выпуск 13, стр. 58-66.

7. Э.В. Евреинов, Г.П. Лопато. Универсальная вычислительная система "Минск-222".- Вычислительные системы, Новосибирск, 1966., выпуск 23, стр. 13-20.

8. G.P. Dinneen, I.L. Lebow, I.S. Reed. The logical design of CG-24. Proc. EJCC, Boston, Dec. 1959, p. 91-94.

9. Э.И. Клямко. Схемный и тестовый контроль автоматических цифровых вычислительных машин. М., Изд-во "Сов. радио", 1963.

I0. К. Берж. Теория графов и ее применение. М., ИЛ, 1962.

9. Э.И.Клямко. Схемный и тестовый контроль автоматических цифровых вычислительных машин. М., изд-во "Сов.радио",1963.

10 К.Берж. Теория графов и ее применение. М., ИЛ., 1962.

11. K.Maling,E.L.Allen . A computer organization and programming system for automated maintenance.IEEE Trans.,EC-12, 1963,№6,p.887-895.

12. W.C.Carter, H.C.Montgomery,R.I.Preiss,H.I.Reinheimer. Design of serviceability features for the IBM System/360. IBM J.Res. and Dev.,1964,8,№2,p.115-126.

13. А.М.Половко. Основы теории надежности. М.,Изд-во "Наука", 1964.

Поступила в редакцию
29.IУ.1967г.