

О ПОИСКЕ НЕИСПРАВНЫХ ЭЛЕМЕНТОВ В СИСТЕМЕ ИЗ ДВУХ ЭВМ

А.К. Олейяр

Введение

В связи с усложнением электронных вычислительных машин (ЭВМ) и недостаточной их надежностью возникает важная задача быстрого отыскания неисправных элементов. Поиск неисправного элемента требует высокой квалификации обслуживающего персонала и занимает большую часть времени на ремонт машины.

В настоящее время в вычислительной технике наметилась тенденция создавать не изолированные вычислительные машины, а комплексы машин [1], [2]. Оказывается выгодным также объединить ЭВМ, уже имеющиеся на вычислительных центрах [3], и сами вычислительные центры с помощью линии связи в одну общую сеть.

Появляется возможность поиска неисправных элементов в ЭВМ с помощью другой, исправной, посредством так называемого программного метода. Последний характеризуется высоким быстродействием. Так как за время поиска одной неисправности вероятность появления второй невелика, то можно построить программу для отыскания одного, любого, неисправного в данный момент элемента.

В статье рассматривается задача установления такой связи между двумя ЭВМ, которая позволила бы обнаруживать неисправный элемент в вышедшей из строя машине с помощью программы.

1. Методы поиска неисправных элементов

Возможны следующие два метода решения поставленной задачи:

Первый — заключается в полном контроле неисправной ЭВМ с помощью исправной (машины — партнера). Для этого между двумя машинами устанавливается связь, позволяющая считывать информацию о работе элементов в любой точке машины и передавать эту информацию машине-партнеру. Если передачи информации производить после каждого такта работы машины, то можно с большой точностью установить момент выдачи неверной информации и указать неисправный элемент. Хотя этот метод очень удобен и позволяет быстро отыскивать неисправный элемент, он требует больших затрат на дополнительное оборудование.

Второй метод заключается в том, что контролируется не вся машина, а только некоторая её часть, называемая базой. В состав ее входят элементы, которые участвуют в выполнении команд, с помощью которых можно проверить остальную часть ЭВМ без помощи машины-партнера. Осуществление контроля по этому методу также связано с некоторым введением дополнительного оборудования, однако оно значительно меньше, чем для первого метода. Далее будет рассматриваться только второй метод.

Задача связи между двумя ЭВМ для контроля и поиска неисправного элемента теперь сводится к выбору минимальной базы и установлению связи для ее проверки.

2. Выбор базы

В ЭВМ, не связанной с другими машинами, выбор минимальной базы играет большую роль при составлении тест-программы. Если база проверена и работает исправно, то оставшуюся часть ЭВМ с помощью программы нетрудно проверить и установить в ней неисправный элемент [4]. Сущность такой проверки сводится к следующему: на основе команд, выполняемых только с участием элементов базы (базовых команд), строится программа проверки одной из непроверенных команд. Проверка следующих команд основывается на базовых командах и уже проверенной команде и т.д. Таким образом получается расширяющаяся тест-программа, при выполнении которой в каждом новом шаге работа непроверенных устройств проверяется с помощью проверенных. Чем меньше элементов проверяется в каждом шаге, тем с большей точностью можно указать неисправный элемент.

Выбор базы начинается с выбора минимального числа команд, с помощью которых можно построить простейшую тест-программу (см. Программа I), позволяющую проводить сравнение (поразрядное сложение) и по результату сравнения осуществлять условный переход.

Программа I

Ячейки памяти	Операция	A-1	A-2	A-3
0001	Ср	0003	0004	-
0002	УП	-	0001	-
0003				
0004				

Условные обозначения: Ср — команда сравнения кодов, вызванных по первому и второму адресу. При несовпадении кодов происходит останов, при совпадении — вырабатывается признак $\omega = 1$.

УП — условный переход. При $\omega = 1$ управление передается по коду в разрядах второго адреса, при $\omega = 0$ — по счетчику команд. В ячейках 0003 и 0004 находятся коды для сравнения. При правильной работе сумматора происходит многократное выполнение программы.

Команду сравнения выполняют следующие устройства: регистр операций, коммутатор операций, регистры первого и второго адресов, схема выбора ячейки памяти по адресу, тактирующее устройство, схема, вырабатывающая признак ω , и сумматор. В базу достаточно включить один (любой) разряд сумматора, так как правильная работа одного разряда сумматора позволяет судить об исправности устройства управления поразрядным сложением.

Команду условного перехода осуществляют устройства: регистр операций и коммутатор операций, разряды второго адреса и схема передачи кода из A2 в счетчик команд, тактирующее устройство и устройство, воспринимающее признак ω . Кроме перечисленных устройств, в базу необходимо включить счетчик команд, четыре ячейки запоминающего устройства и те разряды регистра кодовых формирователей, которые считывают из запоминающего устройства код операции и 3 младших разряда A1 и A2. Эти устройства должны быть проверены с помощью машины-партнера.

Рассмотрение больших ЭВМ и выбор базы для них показывает, что элементы, составляющие базу, охватывают около 20-30% всего количества элементов машины.

В соответствии с идеей расширяющегося теста с помощью констант, засылаемых в ячейки 0003 и 0004, проверяется работа всех остальных разрядов сумматора.

После проверки сумматора необходимо проверить правильность записи кодов по третьему адресу команды, что осуществляется по программе II.

Программа II

Ячейки памяти	Операция	A1	A2	A3
0001	Ср	0003	-	0004
0002	Ср	0003	0004	-
0003	УП	-	0001	-
0004				

По команде из ячейки 0001 происходит перепись кода, а по команде из ячейки 0002 - сравнение считанного кода с кодом в ячейке 0003, который принят за эталон.

При помощи устройств, проверенных I и II программами, строится программа III, которая служит для проверки части запоминающего устройства.

Программа III

Ячейки памяти	Операция	A1	A2	A3
0001	Ср	0004	-	A
0002	Ср	0004	A	-
0003	УП	-	0001	-
0004				

В ячейке 0004 находится эталонный код, а адрес A принимает значения номеров тех ячеек запоминающего устройства, которые необходимо проверить.

После того, как проверены сумматор и часть ячеек памяти,

в проверенную часть запоминающего устройства вводится основная тест-программа для всех непроверенных команд.

Идея расширяющегося теста требует последовательной работы небольших по объему тест-программ, а ввод их в проверяемую машину занимает много времени. Однако это время можно значительно сократить, если выполнять эти программы с помощью машины-партнера. Машина-партнер засылает программу I для проверки сумматора, изменяя после каждого цикла коды для сравнения в ячейках 0003-0004. Программа II выполняется без перемены кодов, а для программы III машина-партнер засылает коды в ячейку 0004 и команды в ячейки 0001 и 0002 (для модификации адреса A).

Таким образом, для достижения выигрыша во времени, кроме базы, необходимо проверить еще часть устройств неисправной машины. Хотя для этой проверки не требуется дополнительных устройств, выбор их оптимального количества представляет собой самостоятельную задачу, которая должна быть рассмотрена отдельно.

Минимальное количество базовых команд равно единице. Если в трехадресной машине выделена одна базовая команда, то она должна выполнять:

1. Сравнение кодов, вызываемых по первому и второму адресу.
2. Вызов следующей команды по счетчику команд (при равенстве кодов).
3. Останов и последующую передачу управления по третьему адресу (при неравенстве кодов).

С помощью такой универсальной команды можно составить программу и для проверки почти всех других команд. При обнаружении неисправности происходит останов, а чтобы повторить (для проверки) неверно выполненную команду, используется передача управления.

Варьируя разные наборы базовых команд, можно выбрать базу с минимальным количеством оборудования и на ее основе составить тест-программу.

3. Проверка базы

Рассмотрим проверку базы одной машины с помощью другой. Будем считать, что любая из машин может оказаться неисправной и тогда исправная ЭВМ должна проверить базу неисправной и указать неисправный элемент.

Установить правильность работы базовых команд несложно: составляется программа, при выполнении которой каждый из элементов базы должен сработать хотя бы один раз. В результате работы этой программы вырабатывается несколько кодов, которые пересылаются в память машины-партнера.

Правильность выполнения базовых команд устанавливается путем сравнения полученных кодов с заранее установленными константами.

При несовпадении кодов неисправность нужно искать в базе. Трудность поиска неисправного элемента в базе заключается в том, что конечный результат может оказаться неверным из-за неверного выполнения любой из команд программы. Даже если удастся определить такую команду, ее невозможно выполнить одну, вне всей программы. Кроме того, одна и та же неисправность может влиять на правильность выполнения нескольких базовых команд.

Введение универсальной команды не расширяет возможности программы для указания неисправного элемента. Оказывается, в выполнении этой команды последовательно участвует несколько блоков машины, и неисправность в любом из них искажает результат.

Для установления неисправного элемента необходимо считать получаемые из проверяемой ЭВМ промежуточные результаты, что достигается следующими двумя способами.

1. Вводятся дополнительные команды, которые выполняются устройствами, не участвовавшими в выполнении базовых команд. Если установлена неисправность базовых команд, то программа, составленная из дополнительных команд, может проверить каждую из базовых команд по отдельности и указать неисправность.

2. В каждую машину вводится контрольный регистр, элементы которого запоминают управляющие сигналы устройств, участвующих в выполнении базовых команд. Машина-партнер может обратиться к этому регистру по специальной команде. Теперь отпадает необходимость составлять программу теста из следующих друг за другом базовых команд. После выполнения в неисправной машине одной команды машина-партнер считывает показания контрольного регистра неисправной машины. Анализ полученных данных не представляет трудностей.

Недостатком этого способа является то, что команда обращения к контрольному регистру имеет узкую область применения и не может быть использована в рабочей программе.

Общим недостатком для 1 и 2 способа является введение дополнительного оборудования. Попытка уменьшить дополнительное оборудование привела к следующему решению.

В машинах с жестким управлением управляющие точки имеют постоянное соединение с центральным устройством управления операциями [5]. При выполнении команды сигналы в этих точках появляются в последовательности, определяемой временной диаграммой работы данной команды. Неисправность элемента проявляется в том, что сигнал в одной из управляющих точек либо отсутствует в момент, определяемый временной диаграммой, либо появляется ложный сигнал.

Задача тест-программы состоит в том, чтобы по анализу полученных результатов указать неверно функционирующую управляющую точку и неисправный элемент (неисправности в разрядах регистров обнаруживаются сравнительно просто).

В этой работе предлагается 3-й способ отыскания неисправного элемента.

В каждую машину вводится специальная команда. Разряды адресной части этой команды отводятся для управления сигналами, идущими к тем управляющим точкам машины-партнера, которые выполняют свои базовые команды. С помощью специальной команды каждую базовую команду можно выполнить не по её коду, принятому на регистр команд, а по программе машины-партнера.

При составлении тест-программы базовые команды удобно выполнять по частям, а в исправной машине проверять результаты их действия. Например, можно выполнить команду, начиная с последнего сигнала управления, затем с предпоследнего и т.д.

Этот способ открывает следующие возможности:

1. Составляя соответствующим образом программу выполнения испытываемых команд, можно выполнять каждую команду в любой последовательности, удобной для обнаружения неисправного элемента.

2. Для обнаружения работающего со сбоями элемента можно создать тяжелые режимы для отдельных блоков путем многократного повторения одного и того же сигнала. Этот программный способ удобно применять совместно с методом предельных испытаний.

3. Имеется возможность проверки команд комбинацией сигналов, входящих в разные команды. Например, удобно комбинировать проверяемые сигналы, используемые для передачи управления, и сигналы, участвующие в пересылке кода в исправную машину.

4. Изменения в тест-программе не приводят к необходимости конструктивных изменений или дополнений.

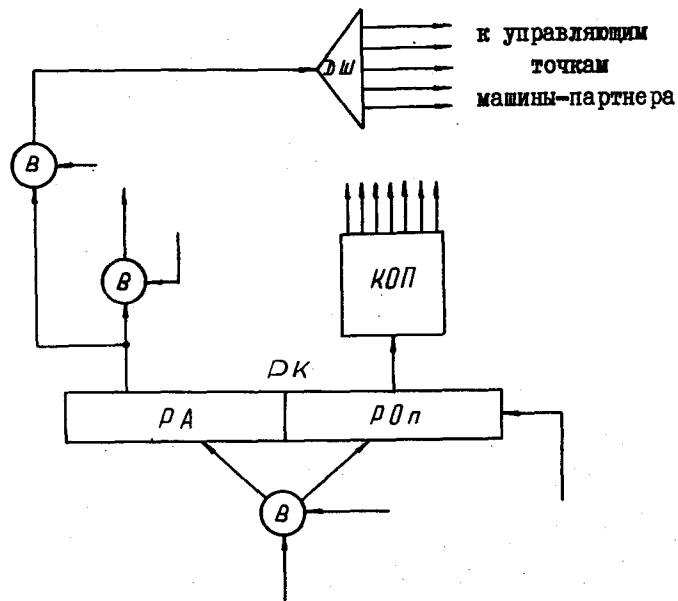


Рис. 1. Схема подключения дополнительного оборудования: ДШ - дешифратор; В - вентили; РК - регистр команды; КОП - коммутатор операций; РА - регистр адреса; РОП - регистр операций.

Рассмотрим устройства, которые составляют дополнительное оборудование (Рис.1). Последнее состоит из вентильного регистра и дешифратора. На вход ДШ поступают сигналы из адресной части регистра команды. Выходные линии ДШ подключаются к управляющим точкам машины-партнера.

Применение этого способа для системы из 2 ЭВМ с микропрограммным управлением связано с введением меньшего количества оборудования, чем с жестким управлением [6].

Одна из машин системы с микропрограммным управлением в упрощенном виде изображена на рис. 2. Особенностью такой ЭВМ является способность данной ЭВМ выполнять команду по микропрограмме, которая хранится в постоянной памяти, так называе-

мой коммутационной матрице (КМ). Изменяя содержимое КМ, можно изменять систему команд.

Микропрограмма состоит из микроопераций, которые задаются с помощью соединений в коммутационной матрице. Горизонтальные шины КМ идут ко всем управляющим точкам машины, а вертикальные подходят к дешифратору (ДШ). Каждая вертикальная шина имеет свой порядковый номер (адрес). По коду в разрядах АМП дешифратор выбирает соответствующую вертикальную шину. В соответствии с соединением в КМ на горизонтальных шинах появляются сигналы для выполнения первой микрооперации. Тактирующее устройство, вырабатывая последовательность импульсов, включает вертикальные шины по порядку, пока не выполнится данная микропрограмма.

Устройство связи воспринимает управляющие сигналы машины-партнера. Для проверки базовых команд (в данном случае базовых микропрограмм) горизонтальные шины с сигналами специальной микропрограммы из КМ машины-партнера подключаются к управляющим точкам проверяемой машины.

Таким образом, всё дополнительное оборудование свелось к нескольким горизонтальным и соединительным шинам.

Выводы

Применение программных методов для отыскания неисправностей в отдельной ЭВМ возможно только в 70-80% случаев выхода машины из строя.

При объединении в систему двух ЭВМ можно осуществить отыскание неисправностей без применения ручного труда. Наилучшие результаты при наименьших затратах на дополнительное оборудование получаются для машин с микропрограммным управлением.

Осуществление программного поиска неисправных элементов позволит уменьшить простояние ЭВМ при ремонте.

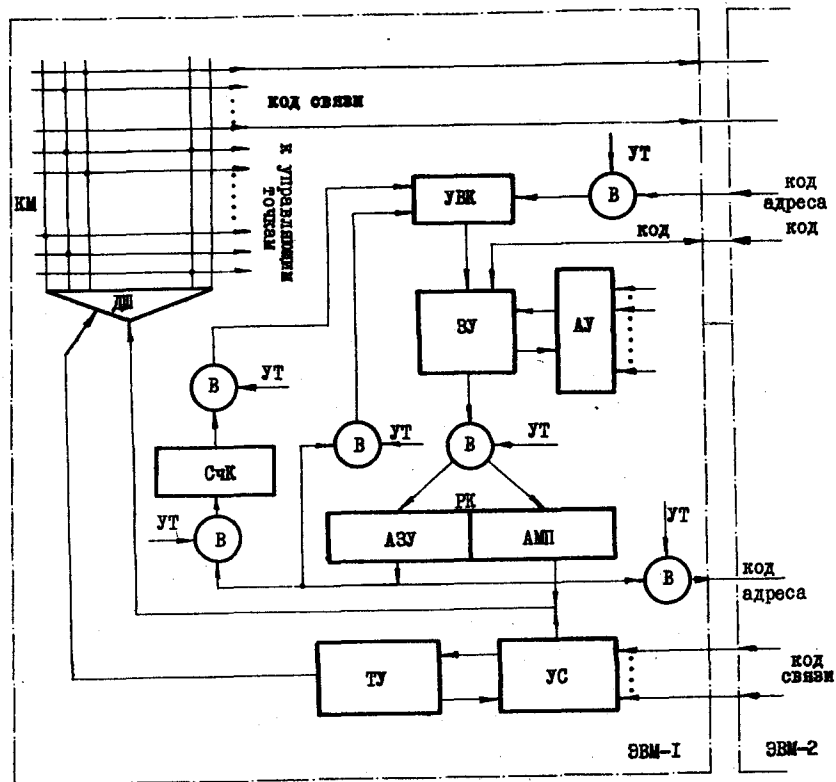


Рис. 2. Блок-схема ЭВМ с микропрограммным управлением:

- управляющая точка; РК - регистр команд; АЗУ - адрес запоминающего устройства (ЗУ), по которому производится вызов кода или передача управления; АМП - адрес микропрограммы; КМ - коммутационная матрица; УС - устройство связи; ТУ - тактирующее устройство; УБК - устройство выбора кода по адресу; СчК - счетчик команд; В - вентиль; ДШ - дешифратор.

Литература

1. Евреинов Э.В., Косарев Ю.Г. О возможности построения вычислительных систем высокой производительности. Изд-во СО АН СССР, Новосибирск, 1962.
2. Курс программирования для ГАММА-60. М., ИЛ, 1962.
3. Карцев М.А. Автоматизация вычислений и развитие вычислительной техники. Вестник АН СССР, № II, 1962.
4. Миронов Г.А. Диагностическая подпрограмма. Испытательная программа. Испытательных программ система. Энциклопедия "Автоматизация производства и промышленная электроника", том I, 1962.
5. Китов А.И., Крицкий Н.А. Электронные цифровые машины и программирование. М., Физматгиз, 1961.
6. Robert J. Mercer., Micro-programming. Journal of the ACM, 1957, 4, N 2.