

УЧЕННЫЕ ЗАПИСКИ

КИБЕРНЕТИЗАЦИЯ
НАУЧНОГО
ЭКСПЕРИМЕНТА

5

Министерство высшего и среднего специального образования
Латвийской ССР
Латвийский ордена Трудового Красного Знамени
государственный университет имени Петра Стучки
Проблемная лаборатория физики полупроводников

Ученые записки
Латвийского государственного университета
имени Петра Стучки
том 196

КИБЕРНЕТИЗАЦИЯ НАУЧНОГО ЭКСПЕРИМЕНТА

Выпуск 5

Редакционно-издательский отдел ЛГУ им. Петра Стучки
Рига 1973

А

844480028

75
96

УДК 007.5:539.2

КИБЕРНЕТИЗАЦИЯ НАУЧНОГО ЭКСПЕРИМЕНТА

Выпуск 5. 1973.

В сборнике рассматриваются вопросы программного обеспечения эксперимента, моделирования потоков информации при построении системы разделения времени между несколькими экспериментами, обслуживаемыми одной ЦВМ, сообщается о базисном ассемблере для программирования на ЭЦВМ "Днепр-21" задач обработки информации и задач управления экспериментами в реальном масштабе времени, о новом рекуррентном алгоритме экспоненциального сглаживания, а также о системном приборе для машинного управления тепловыми потоками мощностью до 320 квт.

Ряд статей сборника посвящен вопросам построения универсального программного обеспечения эксперимента, которое обладает высокой степенью независимости от аппаратуры и методики эксперимента и позволяет проводить поисковые исследования в режиме диалога.

Сборник предназначен для научно-технических работников и студентов, интересующихся практическими вопросами применения вычислительной техники для целей непосредственного управления научными и техническими экспериментами средней сложности.

Редакционная коллегия: О.Аболиньш, А.Бернуп-Бернхоф (зам. главного редактора), Ю.Кузьмин (главный редактор), В.Полмане, Я.Страумен, И.Тале, Э.Тарденак, Я.Цирулис, М.Штерн.

Заказы на сборники "КИБЕРНЕТИЗАЦИЯ НАУЧНОГО ЭКСПЕРИМЕНТА" просим направлять по адресу: Латвийская ССР, г.Рига-50, бульв.Райниса, 19, ЛГУ им.П.Стучки. Проблемная лаборатория физики полупроводников, редколлегия сборника "Кибернетизация научного эксперимента".

© Редакционно-издательский отдел ЛГУ им.П.Стучки, 1973 г.

К 3-3-14-096у 420-73
М 812(II)-73



200023428 ✓

Ю.Я.Кузьмин

ОБ ОДНОЙ АЛЬТЕРНАТИВЕ ПРОГРАММИРОВАНИЯ ЭКСПЕРИМЕНТА

Использование цифровой техники получения измерительной информации и управления экспериментом порождает проблему программирования эксперимента, так как практически каждое действие, элементарное с точки зрения экспериментатора, при использовании цифровой техники реализуется в виде длинной последовательности операций, произведенных над входами и выходами цифровых приборов. Например, поиск одного из максимумов сигнала, воспринимаемого аналоговой системой, состоящей из датчика, индикатора и устройства возбуждения сигнала, которое имеет ручку сканирования, распадается на два элементарных действия: наблюдение за индикаторами и вращение ручки сканирования, пока показания индикатора от возрастающих значений сигнала перейдут к уменьшающимся. В цифровой системе элементарных действий будет значительно больше. Например, одному вращению ручки сканирования в пределах 360° может соответствовать последовательный набор на клавиатуре свыше тысячи кодов и тысяча нажатий на клавишу "Пуск".

Аналогично обстоит дело и с другими цифровыми приборами, а общее число операций для указанной выше задачи поиска одного из максимумов сигнала в случае цифровой системы может достигать десятков и сотен тысяч.

Известный выход из подобной ситуации состоит в программированном управлении цифровыми системами, что позволяет экспериментатору значительно снизить число своих действий в процессе общения с системой.

Однако эффективность этого подхода непосредственно зависит от выбора способа программирования цифровой системы.

В настоящее время наиболее распространена альтернатива "последовательного" программирования (A1): разрабатывается жесткий алгоритм конкретного эксперимента, по которому в кодах ЦВМ составляется программа с максимальным использованием имеющегося программного обеспечения ЦВМ.

Менее распространена альтернатива использования трансляторов на основе специальных языков управления цифровыми системами (A2) [1, 2].

Наконец, наименее всего известна альтернатива применения комплекса универсальных групповых программ, настраиваемых на конфигурацию конкретной цифровой системы (A3) [3].

Нами получены результаты экспертных оценок при анализе коэффициента относительного преимущества указанных альтернатив: A1 - 0,04; A2 - 0,36; A3 - 0,60.

При оценке альтернатив учитывались следующие факторы:

- 1) относительные затраты на программирование каждого эксперимента;
- 2) зависимость альтернатив от методики эксперимента;
- 3) простота обслуживания программного обеспечения, соответствующего альтернативе;
- 4) простота организации режима диалога в эксперименте;
- 5) зависимость альтернативы от изменений аппаратуры цифровой системы;
- 6) относительные затраты на базовое программное обеспечение, необходимое при реализации альтернативы.

ОРГАНИЗАЦИЯ ГРУППОВЫХ ПРОГРАММ

Ниже будет рассмотрена альтернатива, получившая высшую оценку.

Центральной идеей альтернативы является разбиение процесса экспериментирования на конечное множество состояний системы $S(S_1, \dots, S_n)$. В этом случае эксперимент мож-

но описать некоторым графом переходов, зависящим от состояний, измерительной информации, уставок, расчетных параметров и некоторого множества условий; тогда программное обеспечение эксперимента можно представить в виде отдельных универсальных программ, реализующих основные функции системы экспериментирования.

Для сравнительно широкого класса экспериментов необходим следующий набор функций [4 - 7]:

- 1) сбор измерительной информации,
- 2) управление аппаратурой,
- 3) контроль состояний системы,
- 4) связь с оператором,
- 5) первичная обработка измерений,
- 6) регистрация результатов обработки,
- 7) реакция системы на аварийные ситуации.

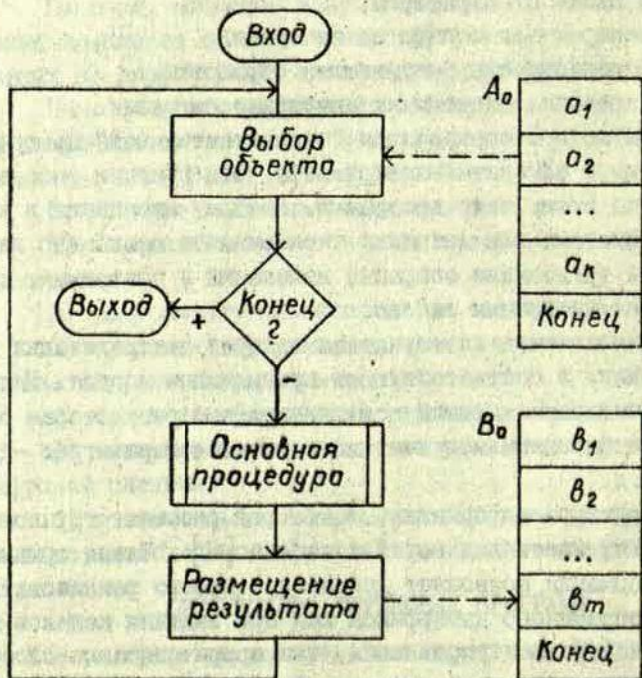
Обычно при составлении "последовательной" программы, реализующей алгоритм эксперимента, эти функции рассредотачивают по всему телу программы. Поэтому изменения в аппаратуре системы или методике эксперимента неизбежно влекут за собой трудоемкие операции изменений в программе, либо написания программы заново.

Предлагаемая альтернатива требует централизации каждой функции в соответствующем программном модуле. Например, вычисление заданий всем дискретным регуляторам осуществляется одним модулем, линеаризация параметров — другим и т.д.

Структура информации, с которой работают групповые программы, имеет вид сетей и таблиц [8]. Такая организация информации позволяет достаточно просто реализовать идею программного интерфейса как при задании каналов измерений, контроля и управления, так и при описании способа функционирования всей системы. В этом случае изменения в технической части системы или методике эксперимента не затрагивают ядра программного обеспечения эксперимента, а требуют лишь настройки его интерфейсной части.

Все групповые программы организованы следующим образом (рис. I). Выбирается очередной элемент (узел сети таблиц, измерительный параметр управления или т.п.), прове-

руется, не последний ли он; если он не является последним, то подготавливаются данные для программы обработки выбранного элемента. Далее программа обрабатывает элемент, например, определяет задание i -му регулятору, либо опрашивает датчик, либо преобразует изменение и т.п. Полученный результат размещается в заданное место. Затем последовательность повторяется, пока не будут исчерпаны все элементы.

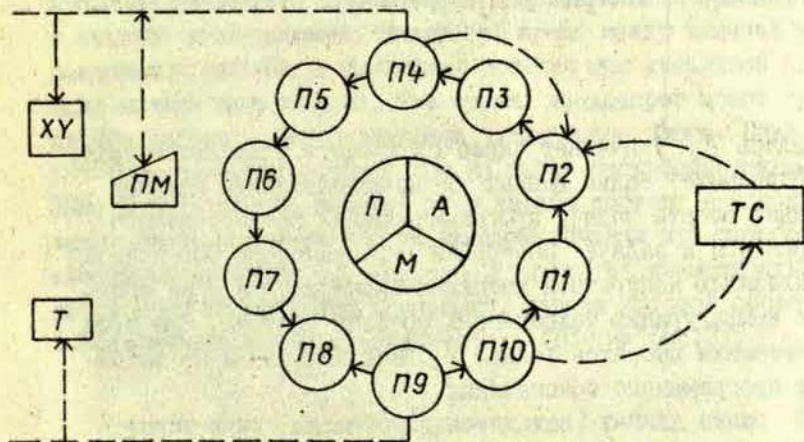


Р и с. 1. Структура групповой программы.

ОРГАНИЗАЦИЯ "КОЛЬЦА" ГРУППОВЫХ ПРОГРАММ

В качестве примера реализации указанной альтернативы приведем программное обеспечение системы для физико-технических экспериментов, разработанное на основе УВМ "Днепр-I" в Проблемной лаборатории физики полупроводников ЛГУ им. П. Стучки [9].

Принцип групповых программ предъявляет определенные требования к формату представления информации. Это связано с тем, что одна и та же основная программа должна быть использована при работе с группой элементов. Нами использована табличная форма представления, при которой каждый элемент (входной канал, выходной канал, вычислительный



Р и с. 2. Программное обеспечение эксперимента, выполненное в виде "кольца" групповых программ.

ХУ - графопостроитель; ПМ - печатающий механизм;
Т - табло; ТС - технические средства системы;
П1-П10 - программы; ПМА - банк данных.

параметр и т.п.) имеет свой ключ N_i (например, номер канала) и следующую за ключом соответствующую информацию X_i . Конец таблицы помечается специально (единицы в

полном машинном слове):

$$\begin{cases} N_1, & X_1 ; \\ N_2, & X_2 ; \\ \dots & \dots \\ N_n, & X_n ; \end{cases}$$

Конец.

Такое представление информации позволяет учитывать изменения эксперимента путем добавления новых строк без переделки групповых программ. Более универсальной формой представления информации являются списки [см. 8]; при этом возможна, например, следующая их интерпретация:

$$\begin{cases} \dots \dots \dots \dots \dots \dots \\ N_{j-1}, & X_{j-1}, & A_j \\ \dots \dots \dots \dots \dots \dots \\ \text{Конец} \end{cases}$$

Здесь A_j указывает адрес следующего элемента списка.

Выделение банка данных в самостоятельную единицу позволяет свести задачу программирования каждого нового эксперимента к задаче "настройки" программного обеспечения на реализацию конкретной методики эксперимента и на конкретную конфигурацию технических средств системы. При этом все изменения вносятся только в банк данных и не затрагивают программного обеспечения.

В банке данных находится информация двух видов:

I) информация для настройки программного обеспечения (интерфейс) (A, M):

- таблицы соответствия адресов физических каналов (входных и выходных) ключам (именам) каналов, с которыми работают программы;
- логические функции, описывающие действие выходных каналов;
- диагностические таблицы;
- таблица аварийных ситуаций, по которым нужна реакция;
- уставки для контроля;

- константы нормализации сигналов и приведения к физическим единицам измерения;

- таблица элементарных состояний системы;

- сеть, описывающая граф перехода от одних состояний к другим в ходе эксперимента и т.п.;

2) информация, генерируемая в ходе эксперимента (П):

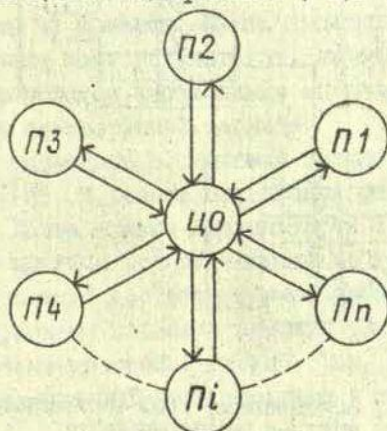
- таблицы текущих значений параметров измерительных и контрольных каналов;

- таблица текущих значений каналов управления;

- вычислительные параметры и т.п.

Теперь перейдем к рассмотрению программ "кольца".

Объединение групповых программ в "кольцо" возможно, если цикл его - время обхода всех программ - согласован с временем физических процессов в исполнительных органах и в объекте исследования. Например, время между выдачей управляющего сигнала и принятием решения при аварийной ситуации должно быть меньше, чем время, за которое может произойти авария в контролируемом техническом блоке. Цикл может быть сокращен с переходом на более быстродействующую УВМ. Другой способ решения этой задачи состоит в использовании системы прерывания по внешним каналам при отработке аварийных ситуаций. И, наконец, существует возможность разработки специальной программы, задача которой состоит в оптимизации цикла эксперимента (ЦО на рис.3).



Р и с. 3. Структура программного обеспечения эксперимента с использованием программы ОПТИМИЗАТОР (ЦО).

Возвращаясь к варианту программного обеспечения, изображенному на рис.2, отметим, что циклическая последовательность выполнения групповых программ может быть нарушена лишь программой КОНТРОЛЬ и РЕАКЦИЯ при наличии аварийных ситуаций.

В зависимости от модификации системы перечень программ может быть уменьшен (например, если управление аппаратурой не требуется), либо дополнен.

Как уже сообщалось выше, эксперимент описывается графом переходов от одних состояний к другим. Функцию определения текущего состояния выполняет программа П I, имеющая в своем распоряжении таблицу элементарных состояний

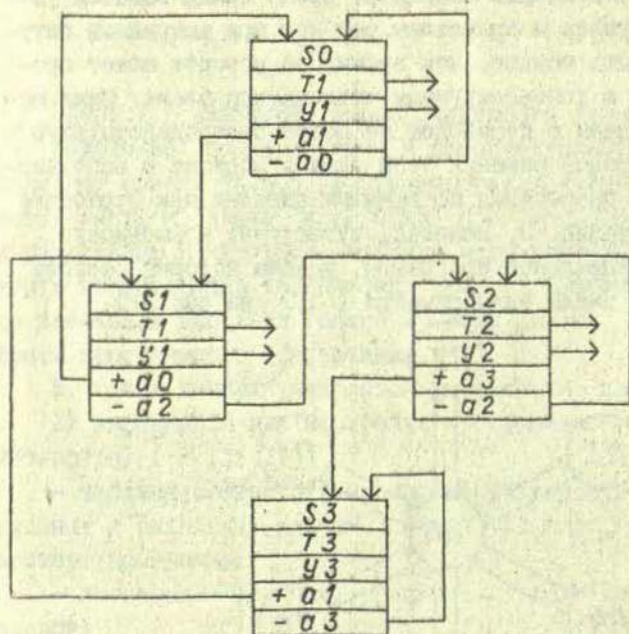


Табл.Т

T1
T2
T3
T4
T5
T6
T7
...
...
Конец

Табл.У

Y1
Y2
Y3
Y4
Y5
Y6
Y7
...
...
Конец

Р и с. 4. Сеть, описывающая граф переходов в одном из научных экспериментов.

(из которых составляется граф эксперимента) и таблицу условий переходов. Обе таблицы расположены в банке данных системы. Результатом работы этой программы будет индекс состояния, которое должна принять система в данном цикле эксперимента.

Однако более общим приемом программирования графа переходов эксперимента следует признать программу, обрабатывающую сеть переходов (рис. 4).

Программа согласно указателю S должна сообщить системе индекс состояния, которое необходимо отработать другим групповым программам; по указателю T выдать ключ текста оператору (при работе в диалоговом режиме); по указателю Ч найти и вычислить список условий перехода к следующему узлу сети (адрес узла +a - если условия выполнены, и -a - в противном случае).

"Программирование" переходов системы из одних состояний в другие состоит в этом случае в задании списка условий переходов, списка состояний, списка текстов-сообщений оператору и, наконец, самой сети переходов.

Групповая программа СБОР (П2) производит сбор измерительной и контрольной информации. Последовательно спрашиваются все датчики согласно специальной интерфейсной таблице, в которой указано соответствие между физическими адресами каналов и соответствующим местом в памяти, закрепленным за каналом. Далее происходит последовательное усреднение всех результатов, линеаризация, масштабирование и переход к необходимым единицам измерения согласно меткам в интерфейсной таблице.

Диалог оператора с системой организует программа ОПЕРАТОР (П3). В режиме прерывания она принимает приказ оператора. Далее приказ сравнивается с набором приказов, по которым имеются обрабатывающие программы, и УВМ уходит на выполнение соответствующей программы. Групповой характер программ позволил сделать реализацию программы ОПЕРАТОР очень простой и гибкой. Так, например, в режиме диалога доступно 30% всех программ и свыше 60% банка данных.

Большинство приказов оператора связано с просьбой сообщить состояние отдельных каналов системы. Это выполняется специальной программой Вывод (П 4); используя систему прерывания, она организует печать содержимого каналов параллельно с выполнением остальных групповых программ. Эта же программа управляет выводом графиков на два самописца ЦДС.

Групповая программа Выход (П 5) вычисляет параметры выходных каналов системы. Вычисляются дизъюнктивные нормальные формы (ДНФ), которыми описываются выходные дискретные каналы. ДНФ задаются специальной таблицей и зависят от контрольных каналов, уставок, остальных выходных каналов и состояния системы. ДНФ есть универсальный способ записи логических условий; это позволяет описывать каналы очень высокой сложности. Значительно сложнее обстоит с небулевыми каналами управления, поскольку для них не существует такого универсального способа представления, как ДНФ. Групповая программа в этом случае может быть построена на основе использования полиномов определенного вида, описывающих каналы, либо аппроксимированными режим таблицами.

Изменение состояния системы во времени приводит к необходимости изменения способа контроля, т.е. в одном случае контрольный параметр должен быть, например, больше уставки, в другом - меньше, иногда требуется сменить уставку. Эту функцию выполняет программа Закон Контроля (П 6). Она вычисляет трехзначную переменную α , условия для которой заданы в виде ДНФ. Эта переменная используется далее программой Контроль (П 7).

Программа Контроль осуществляет сравнение параметров (X) с уставками (Y) согласно значению α (при $\alpha = 0$ проверяется условие $X \leq Y$, при $\alpha = 1$ - условие $X > Y$ при $\alpha = 1/2$ сравнение не делается); В результате дейст-

вия программы вырабатывается аварийное сообщение, в котором отмечены те каналы, где условие оказалось невыполненным. Эта же программа осуществляет проверку ЦВМ по несложному алгоритму.

Программа ДИАГНОСТИКА (П 8) сравнивает аварийное сообщение с таблицей возможных аварий технических узлов и определяет узел, по чьей вине создалась аварийная ситуация.

Это же аварийное сообщение сравнивается программой РЕАКЦИЯ (П 9) с набором аварий, при которых УВМ принимает решение независимо от оператора, например, выключает систему.

Последняя групповая программа ОТРАБОТКА ВЫХОДНЫХ ПАРАМЕТРОВ (П 10) согласно интерфейсной таблице (или алгоритму сопряжения) осуществляет выдачу выходных параметров конкретным выходным каналам, а также преобразование параллельного кода в последовательный, размножение кодов для синхронно работающих каналов и пр. П 10 блокирует канал в случае требования программы РЕАКЦИЯ, а также блокирует для программы ВЫХОД те каналы, которыми управляет оператор в режиме диалога с пульта.

ПОСЛЕДОВАТЕЛЬНОСТЬ НАСТРОЙКИ ГРУППОВЫХ ПРОГРАММ

Основное преимущество написания программного обеспечения методом групповых программ состоит в том, что появляется возможность отделить переменную часть обеспечения от постоянной, следовательно, упростить проблему программирования эксперимента.

Создание постоянной части программного обеспечения, как показал опыт, возможно практически независимо от количества датчиков, исполнительных механизмов

и алгоритма эксперимента.

Настройка программного обеспечения сводится к заполнению таблиц соответствия физических адресов входных и выходных каналов программным именам каналов, а также к заполнению других таблиц и занесению сетей, описывающих, например, контур управления. Настройку можно упростить, разработав для этой цели специальный язык настройки, по форме напоминающий, например, [1, 2] .

Когда уже известна точная конфигурация системы, необходимо:

1) задать таблицу соответствия адресов физических каналов программным именам. Конец такой таблицы помечается специальным символом, который известен групповой программе;

2) задаются уставки для контроля, задается сеть для описания программных выходных каналов. Определяются выходные каналы, которые нельзя описать как программные. Эти каналы объединяются в группы, каждой из которых может соответствовать один индекс состояния;

3) определяется полное число состояний системы; для этого к числу состояний по пункту 2 добавляются те, которые будут связаны с графом переходов эксперимента;

4) задается граф переходов эксперимента в виде сети (рис.3) и связанный с ним список условий перехода, а также список текстов, если предполагается работать в режиме диалога;

5) задаются уставки для контроля, сеть для вычисления закона контроля по уставкам, таблица возможных аварий в системе, таблица аварий, по которым нужна реакция системы;

6) задаются таблицы для линеаризации величин, приведения их к физическим единицам (град, атм, сек).

В качестве примера временных затрат при реализации предлагаемой альтернативы приведем следующие данные. Научно-исследовательская система с машинным управлением для автоматизации тепловых испытаний, в которой применялась данная альтернатива, имеет 43 контрольных и измерительных входных каналов, 22 контура регулирования, в том числе контур управления источником тепловых воздействий мощностью до 320 квт. Требование максимальной безаварийности работы ограничивает цикл выполнения всех групповых программ (рис.2) временем порядка 1 секунды.

Разработка программного обеспечения ("Днепр-1") велась независимо и параллельно с разработкой технической части и потребовала около 5000 чел/час. Настройка обеспечения на систему была осуществлена менее чем за 200 чел/час. Отладка эксперимента заняла около двух месяцев. Все групповые программы были размещены в оперативной памяти емкостью 2048 ячеек, 26 - разрядных слов.

Почти все перечисленные групповые программы описаны более подробно в следующих статьях сборника. В этой статье была сделана попытка дать описание действия "кольца" групповых программ и частично затронута проблематика организации его.

Л И Т Е Р А Т У Р А

1. Ицкович Э.Л., Трахтенгерц Э.А. Алгоритмы централизованного контроля и управления производством. М., 1967.352 с.

2. Щербашин Ю.Д., Иванова А.И. Язык описания систем контроля и управления технологическими процессами. Труды НИИ управляющих вычислительных машин, № 4. Северодонецк, 1971.

3. Кузьмин Ю.Я., Тале И.А., Плаудис А.А. Системный подход в задачах автоматизации физического эксперимента. - В кн.: Тезисы конференции "Автоматизация научных исследований". Л., 1972.
4. Кибернетизация научного эксперимента. Рига, 1968, 203 с.
5. Ученые записки ЛГУ им.П.Стучки, т.144. Кибернетизация научного эксперимента, вып. 2, Рига, 1971, 170 с.
6. Ученые записки ЛГУ им.П.Стучки, т.160. Кибернетизация научного эксперимента, вып. 3. Рига, 1972. 129 с.
7. Ученые записки ЛГУ им.П.Стучки, т.170. Кибернетизация научного эксперимента, вып. 4. Рига, 1972. 259 с.
8. Холгуд Ф. Методы компиляции. М., 1972. 160 с.
9. См.настоящий сборник, с.17-54.

К РЕАЛИЗАЦИИ ГРУППОВОГО АЛГОРИТМА СБОРА И КОНТРОЛЯ ИЗМЕРИТЕЛЬНОЙ ИНФОРМАЦИИ

ВВЕДЕНИЕ

При управлении объектом с помощью управляющей вычислительной машины (УВМ) значительное место в программном обеспечении отводится задачам сбора, первичной обработки и контроля информации.

В работах [1, 4, 5] описаны некоторые общие стороны данного вопроса, предложены различные решения.

В статье будут рассмотрены некоторые алгоритмы [3], реализованные, исходя из принципа наибольшей универсальности как по отношению к управляемому объекту, так и к УВМ, которая осуществляет управление.

В общем случае проблема создания универсального алгоритма сбора информации разбивается на ряд подзадач:

- выделение из алгоритма некоторых функционально независимых блоков;
- организация хранения и обмена информацией между блоками, исходя из принципа универсальности алгоритма;
- создание общей структуры программы.

Ниже будут обсуждаться подробности каждой из перечисленных подзадач.

ФУНКЦИОНАЛЬНЫЕ ЭЛЕМЕНТЫ ПРОГРАММЫ

Простейшим способом реализации функций сбора, первичной обработки и контроля информации является так называемый "последовательный" алгоритм. Сущность его в данном



случае заключается в следующем: производится опрос i -го канала, обработка результата опроса и сравнение с эталоном. Далее производится выбор соответствующей альтернативы управления и повторяется описанная процедура для $(i+1)$ -го канала. Процедура закилевается.

Подобный алгоритм обладает существенными недостатками. Отметим некоторые из них:

- алгоритм не приспособлен для изменения технической части системы; при увеличении числа опрашиваемых каналов требуется написание дополнительных программ, т.е. программа не получается универсальной по отношению к числу каналов;

- при функционировании системы, использующей "последовательный" алгоритм, ЭВМ не имеет полной информации о состоянии управляемого объекта, что затрудняет принятие решения в сложных ситуациях;

- вследствие своей зависимости от числа каналов алгоритм в случае многоканальных систем требует много памяти.

На наш взгляд кажется целесообразным собрать все функции, разбросанные в разных местах, и построить групповой алгоритм [3], состоящий из ряда функционально независимых блоков, которые производят операции сразу над группой каналов:

- блок сбора информации;
- блок усреднения;
- блок перевода в физические величины;
- блок сравнения с уставками (контроль);
- блок опознания причин аварийных ситуаций и реакции на них.

ОРГАНИЗАЦИЯ ОБМЕНА И ХРАНЕНИЯ ИНФОРМАЦИИ

Реализация группового алгоритма, в который входят в некотором смысле независимые блоки, требует стандартизации обмена и хранения информации. В качестве ос-

новой структуры организации информации были приняты таблицы [2] как наиболее отвечающие принципу универсальности алгоритма. Действительно, каждый блок работает лишь с таблицей, которая в свою очередь не обязана быть фиксированной. Процесс расширения или модернизации управляемого объекта не влечет за собой изменения программы, а лишь корректировку информационных таблиц, что в общем случае является более простой операцией и может не требовать участия разработчиков программного обеспечения. Кроме того, применение таблиц позволяет более экономно использовать память УВМ.

ОПИСАНИЕ ФУНКЦИОНАЛЬНОЙ СХЕМЫ АЛГОРИТМА

Прежде чем перейти к описанию собственно алгоритма и его реализации, хотелось бы отметить, что выбранная в данном случае конкретная форма таблицы не накладывает особых ограничений на алгоритм и существенна лишь для реализующей его программы.

В общем случае таблица должна содержать следующие три элемента:

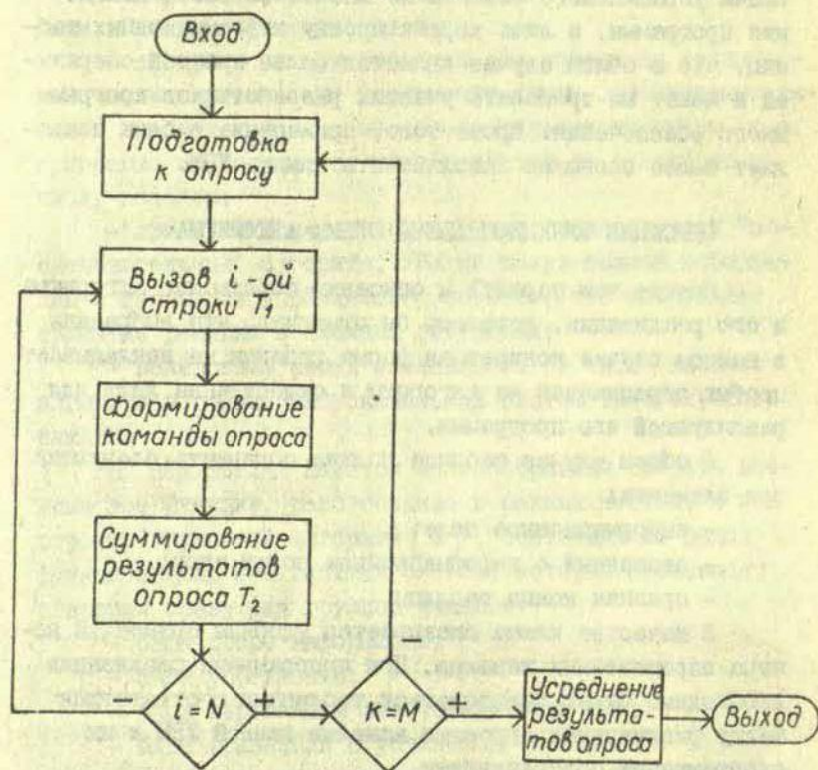
- информационное поле;
- связанный с информационным полем ключ;
- признак конца таблицы.

В качестве ключа оказывается удобным применять номера опрашиваемых каналов. При программной реализации необходимо будет пользоваться таблицами соответствия между физическими адресами каналов данной УВМ и их программными обозначениями.

Таблицы будут обозначаться $\{N, X_i\}$, где N - номер канала, а X_i - соответствующие информационные поля.

Блок сбора и усреднения информации (рис. I). Исходной информацией для этого блока является последовательность номеров опрашиваемых каналов. С помощью таблиц соответствия программа формирует команду (или макроко-

манду) опроса данного канала и заполняет нужную строку таблицы $\{N, \Sigma X\}$. Для получения возможности усреднения опрашиваемых величин процесс опроса каналов заиклевается K раз, а результаты опроса суммируются в таблице $\{N, \Sigma x\}$.

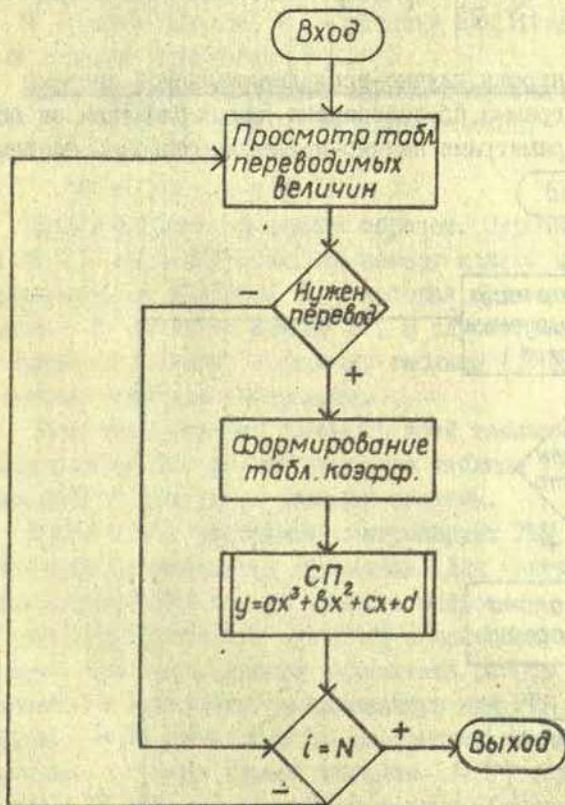


Р и с . 1. Блок-схема сбора измерительной информации.

Далее происходит усреднение снятой информации и формирование таблицы $\{N, \bar{X}\}$. В этом же блоке можно предусмотреть также некоторые дополнительные действия для повышения достоверности обрабатываемых величин, в част-

ности экспоненциальное сглаживание.

Блок перевода в физические величины (рис.2). В ряде случаев бывает необходимо часть усредненных значений перевести в соответствующие физические величины.



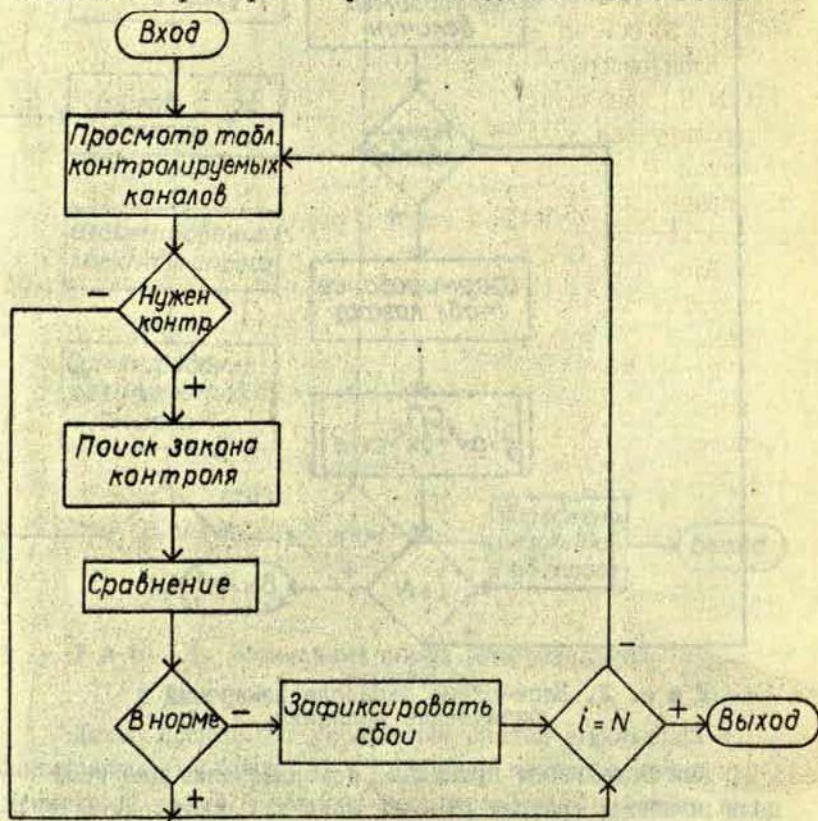
Р и с. 2. Блок-схема перевода измерений в физические величины.

Как показывает практика, использование для этой цели полинома третьей степени является достаточным. Следовательно, блок перевода должен осуществлять под-

готовку для работы стандартной программы просчета значения кубической параболы в данной точке. Для этого программа реализует просмотр таблицы $\{N_i, A_i\}$ переводимых величин, где N_i - номера каналов, требующих перевода, A_i - адреса коэффициентов полинома.

По номеру N_i осуществляется поиск исходного значения полинома в таблице $\{N, \bar{x}\}$. Результат записывается в ту же таблицу $\{N, \bar{x}\}$

Блок контроля научно-исследовательской системы (рис.3). Программа предназначена для наблюдения за обрабатываемыми параметрами научно-исследовательской системы



Р и с. 3. Блок-схема контроля НИС.

(НИС) и формирования аварийного слова при выходе их за допустимые границы.

К моменту начала работы данного блока в таблице $\{N, \bar{X}\}$ имеется полная информация о состоянии НИС и, следовательно, можно осуществить контроль.

В этой процедуре используется таблица $\{\alpha, N, Y\}$ где N - номер канала, Y - уставка для данного канала, α - закон сравнения [6].

Нами выбраны следующие законы сравнения:

1) $\alpha = 0$: \bar{X} и Y не сравниваются;

2) $\alpha = 1$: $\bar{X} < Y$;

3) $\alpha = 2$: $\bar{X} \geq Y$.

Блок работает следующим образом. Строчка таблицы $\{\alpha, N, Y\}$ анализируется, по номеру канала находится контролируемая величина \bar{X} , которая сравнивается с уставкой Y согласно закону α . В случае, если результат сравнения ложен, в строчку таблицы $\{N, S\}$ заносится соответствующая информация.

Блок осуществляет просмотр всей таблицы $\{\alpha, N, Y\}$ и результатом его работы является таблица $\{N, S\}$ с информацией об аварийных каналах системы.

Кроме того, программа контролирует УВМ и таблицы, по которым производится сравнения. Для контроля работоспособности УВМ в цикл включен микротест, выполняющий ряд арифметических операций с псевдослучайными числами. При несовпадении результата работы теста с эталоном в некоторую фиксированную для УВМ строчку таблицы $\{N, S\}$ записывается информация об аварийной ситуации. Еще одна строка таблицы $\{N, S\}$ служит для фиксации аварийного состояния таблиц. Основанием для фиксации сбоя, связанного с таблицами, является обнаружение программой испорченного номера канала N . Это происходит путем сравнения двух разных таблиц, содержащих однако одинаковый перечень номеров каналов $\{N\}$.

Таким образом, в таблице {N,S} мы имеем практически полную информацию как об аппаратуре НИС, так и об УЗМ.

Блок определения причин аварийных ситуаций и реакций на них (рис.4). Блок предназначен для диагностики неисправных узлов НИС по аварийному слову {N,S}, индикации аварий на табло и автоматической реакции на некоторые особо опасные ситуации.



Р и с. 4. Блок-схема диагностики и реакций.

Блок диагностики и реакций, в отличие от всех рассмотренных блоков, не поддается столь явной стандартизации. Очевидно, что в конкретных случаях при реализации он может сильно различаться в зависимости от характера управляемого объекта. Ниже приводится простейшая схема реакций на аварийные ситуации с помощью сигнализации на световом табло.

Для этого результаты сравнения, записанные в таблице $\{N, S\}$, последовательно сравниваются с данными набора аварийных причин $\{N, S\}$. В случае совпадения (или несовпадения) $\{N, S\}$ с $\{N, S\}_{i=\overline{1, n}}$ соответствующие сигнальные лампочки на табло включаются (или выключаются). Кроме того, результаты сравнения могут передаваться в другие блоки путем заполнения специальных программных регистров.

Такая организация оповещения оператора об авариях в НИС обладает рядом преимуществ. Обслуживающий персонал имеет возможность в каждом конкретном случае решать вопрос о целесообразности продолжения эксперимента в условиях аварийности объекта. Кроме того, отпадает проблема автоматического анализа ряда причин, которые связаны с ложными сигналами аварий, например, при несоответствии времен срабатывания различных устройств НИС.

Предусмотрено также, что при некоторых авариях вследствие их опасности блок должен реагировать определенным образом с целью предотвращения выхода из строя объекта управления. При конкретной реализации данного алгоритма в подобных ситуациях были предусмотрены следующие автоматические действия:

- отключение некоторого блока, задающего режимы во времени (режим ожидания);
- блокировка выдачи управляющих воздействий;
- возврат системы в исходное состояние.

В то же время сбор информации, контроль и связь с оператором не прекращаются.

Итак, мы рассмотрели в общих чертах все блоки, связанные со сбором информации, контролем и реакцией. Как видно, достаточной информацией для работоспособности этих программ является наличие таблиц $\{N\}$, $\{N, A\}$, $\{\alpha, N, U\}$, $\{N, S\}$ в памяти ЭВМ. Следовательно, процесс изменения условий управления объектом со стороны сбора информации сводится к генерации новых таблиц, что является разовым действием и никак не связан с изменением действующего программного обеспечения.

Л И Т Е Р А Т У Р А

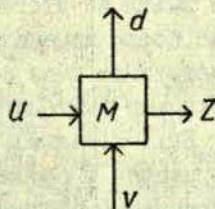
1. Ицкович Э.Л., Трахтенгерц Э.А. Алгоритмы централизованного контроля и управления производством. М., 1967. 352 с.
2. Хопгуд Ф. Методы компиляции. М., 1972. 160 с.
3. См. настоящий сборник, с. 3-16.
4. Колин К.К., Липаев В.В. Проектирование алгоритмов управляющих ЦЕМ. М., 1970. 343 с.
5. Щербашин Ю.Д., Иванова А.И., Язык описания систем контроля и управления технологическими процессами. Тр. НИИ управляющих вычислительных машин, № 4. Северодонецк, 1971.
6. См. настоящий сборник, с. 27-31.

СПОСОБ ФОРМИРОВАНИЯ ЗАКОНА КОНТРОЛЯ В АВТОМАТИЗИРОВАННОЙ СИСТЕМЕ

Достоверность информации, полученной в эксперименте, зависит прежде всего от состояния научно-исследовательской системы (НИС) и от правильности реализации отдельных процедур с помощью управляющих вычислительных машин (УВМ). Отсюда вытекает функция контроля НИС. Контролю подлежат параметры управления, а также измерительные, специальные контрольные и вычислительные параметры.

Реально состояние большинства НИС меняется во времени, и значение параметра, которое было нормальным для одного состояния, может стать критическим для другого и недоступным для третьего состояния. Это значит, что закон контроля (сравнения параметров с уставками) должен быть переменным.

Рассмотрим подробнее, от чего и как может меняться закон контроля. Прежде всего заметим, что любая НИС состоит из элементов, каждый из которых может быть четырехполюсником, реализующим некоторую функцию:



Р и с. I. Элемент НИС.

u — информационный вход; z — соответствующий ему информационный выход; d — вход управления состоянием элемента M ; v — контрольный канал (например, ответ о срабатывании элемента).

Контроль может охватывать как выход z , так и выход v . При этом существуют две возможности:

1. Контроль состоит в определении в любой момент времени и независимо от входов u и d , лежат ли значения выхода элемента в заданных границах или вышли за них. Например, температура в азотном криостате не может быть меньше 70°K и больше 600°K ; всякий выход за эти пределы НИС должна рассматривать как аварийную ситуацию (либо перегрев криостата, либо выход из строя канала контроля температуры).

2. Контроль зависит, например, от управляющего входа d . При этом в одном состоянии выход v примет значение v_1 , а в другом состоянии — значение v_2 . Контроль в данном случае состоит в определении того, появится ли на выходе сигнал v_1 в ответ на сигнал управления d_1 или же v_2 в ответ на d_2 ; в противном случае должна быть отмечена аварийная ситуация.

Чаще всего дело осложняется при инерции элемента, когда после появления на входе d сигнала управления элементу может понадобиться определенное время τ для перехода из одного состояния в другое. В этом случае указанная выше процедура контроля, т.е. определение того, соответствует ли сигнал v_i сигналу d_i в каждый момент времени, будет непригодна, поскольку во время переходного процесса τ будут вырабатываться псевдоаварийные ситуации. Простейший выход из этого — блокировка контроля на время τ тотчас после изменения значения входа d .

Рассмотренная выше ситуация соответствует случаю, когда необходимо вычислить закон контроля во время эксперимента. Ниже показано, как это можно сделать, а также будет приведен и обоснован подход, позволяющий разрабатывать групповые программы вычисления закона контроля, независимые от конкретных особенностей НИС и экспериментов.

Введем теперь одну весьма общую математическую модель процесса контроля, охватывающую ситуацию, описанную выше. Мы будем иметь дело с набором переменных, называемых кана-

лами; предполагается, что для каждого канала определено множество возможных его значений и что в дискретные моменты времени $t \in \{0, 1, \dots\}$ он принимает то или другое значение из этого множества. Мы будем считать, что каждый канал может находиться в одном из следующих состояний: 0 ("Канал выключен"), 1 ("Канал включен"), 1/2 ("Переходное состояние"). С каждым каналом связана его управляющая переменная со значениями 0 и 1. Их можно интерпретировать следующим образом: изменение значения управляющей переменной d_i с 0 на 1 означает команду перевести канал K_i в состояние 1 ("включить"), а с 1 на 0 - в состояние 0 ("выключить"). Обычно канал срабатывает не мгновенно, кроме того, он может такую команду вообще не выполнить.

Однако любой канал K_i , если он исправен, должен сработать не позже, чем через τ_i тактов при условии, что его управляющая переменная d_i за это время не изменяется. Чаще всего точная информация об истинном состоянии неизвестна, поэтому вместо состояния канала K_i введем в рассмотрение переменную a_i - "предполагаемое состояние канала K_i " с теми же значениями - 0, 1, 1/2 и в дальнейшем под состоянием канала будем понимать значение этой переменной. Для текущего момента t значение a_i определяется следующим образом:

$$a_i(t) = \begin{cases} 0, & \text{если } \max d_i(t-t') = 0 \text{ при } 1 \leq t' \leq \tau_i, \\ 1, & \text{если } \min d_i(t-t') = 1 \text{ при } 1 \leq t' \leq \tau_i, \\ 1/2 & \text{в остальных случаях.} \end{cases}$$

Далее, с каждым каналом K_i связано некоторое множество $U_i = \{u_{i1}, u_{i2}, \dots, u_{is_i}\}$ условий, которым реально может подчиняться текущее значение канала. В простейшем случае можно считать, что U_i - подмножество значений канала, а условие заключается в том, что значение его должно принадлежать u_{ij} . Контроль канала сводится к тому, что в различные моменты времени проверяется выполнение или невыполнение некоторых условий из U_i ; такая проверка не должна осуществляться, если K_i находится

в переходном состоянии, т.е. если α_i в этот момент равно $1/2$. Чтобы описать это, пронумеруем все условия, а затем введем для каждого u_n переменную α_n со значениями $0, 1, 1/2$, интерпретируемые следующим образом:

$$\alpha_n(t) \begin{cases} 0 & \text{— условие } u_n \text{ в момент } t \text{ проверяется на невыполнение,} \\ 1 & \text{— условие } u_n \text{ в момент } t \text{ проверяется на выполнение,} \\ 1/2 & \text{— условие } u_n \text{ в момент } t \text{ не проверяется.} \end{cases}$$

Назовем общим термином "параметры контроля совокупность всех состояний каналов вместе с видами контроля. На основе текущих значений этих параметров определяются следующие значения переменных α_n (при этом результаты проверки условий непосредственно не используются, но они учитываются при определении будущих значений управляющих переменных). Мы ограничимся рассмотрением случая, когда

$$\alpha_n(t+1) = f_n(P_{n1}(t), \dots, P_{ns_n}(t)),$$

где f_n — фиксированная для каждого α_n и не меняющаяся со временем функция ("закон контроля"), $(P_{n1}, \dots, P_{ns_n})$ — также фиксированный набор параметров контроля. На закон контроля наложим следующие требования: эта функция принимает значение $1/2$ тогда и только тогда, если хотя бы один из ее аргументов принимает это значение.

На параметры контроля можно смотреть как на своеобразные трехзначные логические переменные. Назовем отрицанием, дизъюнкцией, конъюнкцией соответственно операции \sim, \vee, \wedge , определяемые на множестве $0, 1, 1/2$ следующей таблицей:

P_1	P_2	$\sim P_1$	$P_1 \vee P_2$	$P_1 \wedge P_2$
0	0	1	0	0
1/2	0	1/2	1/2	1/2
1	0	1	1	0
0	1/2	1	1/2	1/2
1/2	1/2	1/2	1/2	1/2
1	1/2	0	1/2	1/2
0	1	1	1	0
1/2	1	1/2	1/2	1/2
1	1	0	1	1

Пусть, далее, F - класс всевозможных законов контроля, т.е. функций указанного выше вида. Легко видеть, что отрицание, дизъюнкция и конъюнкция принадлежат этому классу и что он замкнут относительно суперпозиций. Мы покажем, что любую функцию из F можно представить в стандартном виде с помощью операций \sim, \vee, \wedge .

Назовем простой конъюнкцией всякую конъюнкцию переменных (т.е. параметров контроля) или их отрицаний (не исключая при этом случай "одночленной конъюнкцией": p или $\sim p$); дизъюнктивной нормальной формой (ДНФ) - всякую дизъюнкцию простых конъюнкций (тоже, быть может, состоящую из одной простой конъюнкцией). Тогда имеет место следующее

ПРЕДЛОЖЕНИЕ. функция принадлежит классу F тогда и только тогда, если она имеет равносильную ДНФ.

Очевидно, всякая ДНФ принадлежит классу F . Пусть, наоборот, f - некоторая функция из F . Если в таблице ее значений рассматривать только строки, в которых аргументы (а тем самым и сама f) не принимают значения $1/2$, то она превращается в обычную булеву функцию, для которой известными средствами можно найти ДНФ. Итак, в рассматриваемых строках значения f и найденной ДНФ совпадают. Но они очевидно совпадают и в остальных строках (так как принимают там значение $1/2$). Таким образом, найденная ДНФ действительно представляет функцию f .

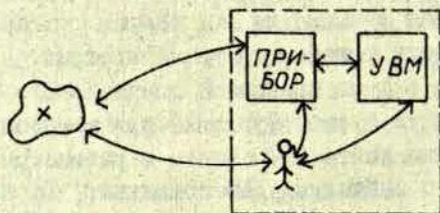
В заключение отметим, что рассмотренный здесь способ формирования закона контроля использован для задания параметра α в программе, описанной С.В.Гвоздевым в настоящем сборнике (см.с.17-26).

ОСОБЕННОСТИ РЕЖИМА ДИАЛОГА В АВТОМАТИЗИРОВАННОМ ЭКСПЕРИМЕНТЕ

К ОЦЕНКЕ ОПТИМАЛЬНОЙ АЛЬТЕРНАТИВЫ

Режимом диалога будем называть взаимодействие оператора и управляющей вычислительной машины (УВМ) в процессе решения задачи в режиме реального времени.

Для выяснения причин, которые могут привести к целесообразности осуществления режима диалога в автоматизированном эксперименте, рассмотрим схему взаимодействия оператора с научно-исследовательской системой (НИС).



Р и с. 1. Структура связей в НИС диалогового типа.

Заметим, что целью всякой НИС является максимально быстрое и качественное получение интересующей исследователя информации о процессах, происходящих в объекте исследования, либо о его механических, кинематических и других свойствах.

Перечислим ситуации, при которых НИС не может достигнуть этой цели, действуя полностью автоматически.

1. Ввод информации об объекте, недоступной датчи-

кам, имеющим непосредственную связь с НИС.

2. Установка, перемещение, замена и контроль датчиков во время эксперимента, не предусмотренное автоматикой НИС.

3. Осуществление управления состоянием объекта, недоступное исполнительным механизмам НИС.

4. Установка, перемещение, замена и контроль исполнительных механизмов, не предусмотренное автоматикой НИС.

5. Проведение общего контроля НИС, недоступного автоматическим тестам НИС.

6. Выбор регистрируемых параметров; задание режимов во время эксперимента, не предусмотренное заранее программой.

7. Принятие решения в альтернативной ситуации, которое нельзя выразить алгоритмически, (если требуется использовать опыт оператора или отсутствует схема однозначных реакций).

8. Осуществление поискового эксперимента, когда операции обрабатываются в ходе самого эксперимента.

9. При отладке программ эксперимента и отдельных каналов НИС.

Во всех этих случаях достижение цели возможно лишь совместными действиями УВМ и оператора во время эксперимента. Упомянутые выше ситуации порождают 512 комбинаций, которые могут встретиться на практике.

Важно отметить, что диалог может быть активен как со стороны УВМ, которая задает вопросы человеку и он отвечает, либо УВМ отдает приказы, а человек выполняет их, так и со стороны оператора, который задает вопросы или приказывает, а УВМ отвечает на вопросы или выполняет приказы.

В настоящее время известно большое количество способов организации режима диалога, каждый из которых связан с определенными техническими средствами - телетайп, дисплей и различные комбинации пультов с табло, ЭЛТ, электрические печатающие машинки (ЭПМ) и другие - и соответствующим программным обеспечением.

Выбор оптимальной альтернативы начинается с определения целесообразности режима диалога. Для этого определяется, какая из 9 названных выше ситуаций (или их комбинация) имеет место в конкретном случае. Далее, имеющиеся альтернативы сравниваются с уточненным набором ситуаций; те из них, которые заведомо не удовлетворяют набору, отбрасываются.

По оставшимся проводится системный анализ [1, 2]. Для этого определяется перечень факторов, которым должны удовлетворять альтернативы, устанавливается по каждому из них относительное предпочтение альтернатив и, наконец, определяется альтернатива, имеющая лучший показатель по всему перечню факторов.

Перечень факторов зависит от конкретной ситуации, однако следует выделить достаточно общие и важные факторы, которые необходимо учитывать в подобных случаях. В качестве одного из них в [3] предлагается использовать коэффициент оперативности диалога, т.е. отношение времени подготовки сообщения ко времени передачи сообщения. Однако, на наш взгляд, этот коэффициент не охватывает другие стороны проблемы, кроме того, он не позволяет отличить систему, имеющую оба одинаково плохих временных показателя, от системы, имеющей хорошие показатели, но такое же их отношение. Нам кажется, что более приемлемы для оценки разных альтернатив следующие факторы.

1. **П о л н о т а** диалога, т.е. насколько диалог позволяет управлять основными программами и какой объем информации, хранящейся в памяти ЭИС, доступен в режиме диалога. Чтобы оценить полноту диалога, можно, например ввести следующие коэффициенты:

$$K_1 = \frac{N_{\text{вд}}}{N_{\text{зд}}} , \quad K_2 = \frac{N_{\text{зд}}}{N_{\text{зд}}} , \quad K = \frac{N_{\text{пп}}}{N_{\text{лп}}}$$

$N_{\text{вд}}$ - количество управляющих входов программ ЭИС, к которым имеется доступ в режиме диалога.

$N_{\text{зд}}$ - количество всех управляющих входов программ ЭИС.

$N_{\text{зд}}$ - количество параметров, к которым имеется доступ в режиме диалога.

$N_{\text{лп}}$ - количество всех параметров ЭИС, хранящихся в ее памяти.

$N_{\text{об}}$ - количество программ НИС, охваченных диалогом.

$N_{\text{зп}}$ - количество всех программ НИС.

Последний коэффициент может быть использован при грубой оценке степени полноты диалога.

2. Простота диалога, т.е. количество манипуляций при реализации запроса (приказа) в режиме диалога. Под манипуляциями будем понимать набор отдельных пультавых, телетайпных и тому подобных клавишей, в том числе цифр на клавиатурах числа и адреса.

3. Быстрота диалога, т.е. время с момента ввода (вывода) приказа или запроса до момента начала выполнения приказа или получения ответа. Этот фактор зависит от простоты диалога и от его наглядности.

При разработке диалоговых систем часто встает вопрос об оптимальном количестве приказов (запросов). Всякий сложный приказ (запрос) можно расчленить на ряд простых и, наоборот, последние можно объединить в сложные приказы (запросы). В этом случае, на наш взгляд, в качестве дополнительного фактора нужно учитывать частоту употребления приказа (запроса) в реальной или предполагаемой обстановке. Если частота оказывается высокой и если налицо выигрыш по скорости и простоте диалога, то есть смысл ввести подобный новый приказ (запрос) в систему.

ПРАКТИЧЕСКИЙ ПРИМЕР ДИАЛОГА

Рассмотрим режим диалога, организованный для работы с системой групповых программ [4].

Основные причины введения режима диалога.

1. Принятие решений в альтернативной ситуации. Количество программ для принятия решений при различных комбинациях аварийных ситуаций значительно превышает память машины, поэтому программно охвачены лишь наиболее опасные аварии, а решения по другим авариям предоставляются оператору НИС.

2. Количество измеряемых и контролируемых параметров (в конкретной системе их 43) значительно превышает количество выводных устройств (их 3), поэтому коммутация

"Параметр - Выходное устройство" должна быстро осуществляться оператором по ходу эксперимента.

3. Обычно вмешательство в программы эксперимента доступно лишь высококвалифицированным программистам, составившим эти программы. Режим диалога позволяет вносить изменения в ходе эксперимента на языке, который более близок к естественному, чем коды машинных операций. Изучение подобного языка доступно и непрограммистам.

Программа связи с оператором, о которой идет речь, составлена из трех подпрограмм, осуществляющих:

- ввод символов с электрической печатающей машинки;
- расшифровку операторских приказов, которые составлены из символов, введенных с ЭПМ;
- печать ответов оператору или подготовку программ НИС для отработки приказа оператора.

Техническая часть представляет собой ЭПМ, подключенную через систему прерывания к устройству связи с объектом (УСО) "Днепр-1" (рис.4), и индикационное табло (рис.5), на которое выводится динамическая служебная информация.

Форматов сообщения (приказов, запросов) три:

- /идентификатор/+
- /идентификатор/.. /восьмеричное 4-разрядное число/+
- /идентификатор/.. /восьмеричное 4-разрядное число/..
- /десятичное 5-разрядное число/+

Идентификатором являются первые три буквы сообщения оператора, которое само может быть любым текстом, содержащим не более одного пробела и не имеющим символов "+" и "-". Такая структура сообщения позволяет давать вместе с наименованием приказа и его обширный комментарий.

Пример приказа с комментарием:

"ВК+ выключить, поскольку обнаружена авария в блоке охлаждения+"

Машина по символам ВК+ найдет подпрограмму выключения приборов НИС, выполнит ее и сообщит об этом оператору. На бумаге ЭПМ останется объяснение причины выключе-

ния (комментарий).

Для введения этого приказа достаточно во время эксперимента напечатать его с помощью ЭПМ " Днепр-1".

Ниже будут изложены некоторые детали рассматриваемого примера реализации режима диалога во время эксперимента.

Программное обеспечение. Большое влияние на построение программного обеспечения режима диалога оказал способ организации информации в памяти УВМ [4]. Поскольку все контролируемые величины непосредственно связаны с номерами контролируемых каналов, то запрос на вывод любого канала, интересующего оператора, сводится к указанию в запросе номера канала, например:

"Печатай 23+"

Для удобства программирования оказалось целесообразным ввести ряд программных регистров, т.е. выделить специальные ячейки памяти, в которых отмечаются состояния устройств вывода (занято, свободно) и запросы программы на устройства.

Среди этих регистров имеются следующие:

- регистры запроса на вывод δ , δ' , δ'' ;
- регистр окончания печати α .

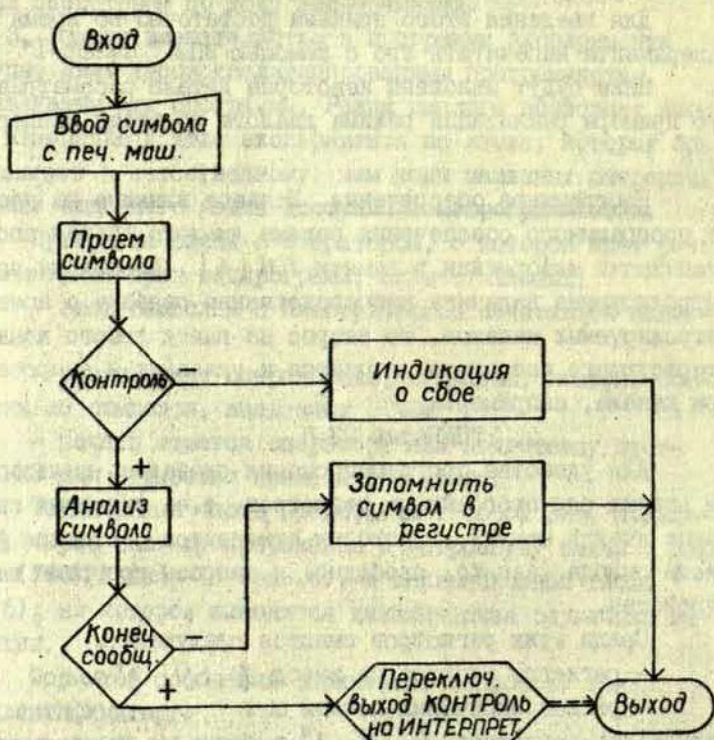
Разрядность регистров δ , δ' , δ'' выбрана по числу устройств вывода. Наличие "1" в соответствующем разряде указывает на необходимость вывода на данное устройство.

Печать информации в предусматриваемых форматах производится в несколько приемов, отсюда появляется необходимость указать программе на то, что устройство печати освободилось и готово к работе. Эту роль выполняет одноразрядный регистр α .

Рассмотрим ряд блоков программного обеспечения. Программа приема приказов оператора обслуживает одно из наиболее медленных внешних "устройств" УВМ - человека. Чтобы скомпенсировать затраты времени на контакт человека с программой, ввод символов происходит с использованием системы прерывания.

При нажатии клавиши ЭПМ происходит прерывание ос-

новой программы и управление передается блоку приема символа.



Р и с. 2. Блок приема приказов оператора.

Код символа снимается с дискретных входов устройства связи УВМ с объектом. Совместно с шестью разрядами кода принимается и контрольный седьмой бит, равный нулю, если система прерывания возбудилась от помех.

При задании приказов используется три специальных символа:

"-" - отказ от приказа, если оператор заметил ошибку в процессе набора приказа;

"+" - признак конца приказа, по нему УВМ переходит к анализу и выполнению приказа;

" " - пробел, вспомогательный символ.

Два отпечатанных символа пробела подряд используются для

отделения различных частей приказа (идентификатор, восьмеричное число, десятичное число).

Для хранения принятой информации используются три регистра (ячейки) - B_0, B_1, B_2 . Символы, поступающие в УВМ, анализируются следующим образом:

1. В случае равенства нулю контрольного разряда кода, на табло выдается сообщение о наличии сбоя устройства ввода, а принятая в этот момент информация игнорируется.

2. Если принятый код не является одним из служебных знаков, то код в компактном виде помещается в один из регистров, т.е. B_0, B_1, B_2 .

3. Символ "-" используется для возврата программы приема приказов оператора в исходное состояние, при этом очищаются все программные регистры, связанные с вводом.

4. Знак "+" завершает ввод любого приказа. По этому символу выдается сообщение основной программе о необходимости ответить на принятый приказ.

Несмотря на простоту и некоторую очевидность данной программы, она является довольно гибкой и не зависит от других программ. При выполнении этой программы на прием и обработку символа требуется в среднем около 1,5 мсек, что является пренебрежимо малым по сравнению с полным временем цикла программы управления экспериментом (в данном случае около 1 сек).

Блок расшифровки и реализации операторских приказов напоминает интерпретатор. Назначение блока ясно из его названия. Вход в блок организуется в заранее установленном месте программы эксперимента и активизируется только после получения символа "+". Сам блок состоит из двух довольно независимых частей:

- программа сравнения принятого идентификатора (B_0) с набором наименований подпрограмм для отработки приказа;
- библиотека подпрограмм, необходимых для отработки приказов оператора.

Разделение блока на вышеперечисленные части позволяет достаточно просто вносить новые или исключать старые подпрограммы отработки приказов из библиотеки подпрограмм НИС.

В библиотеку входят подпрограммы двух типов:

- подпрограммы управления устройствами вывода;
- подпрограммы оперативного внесения изменений в эксперимент.

Ниже приводятся операторские сообщения, используемые в одной из действующих разработанных систем. Дана полная и сокращенная форма приказа, а также содержание каждого приказа.

1. НАЧИНАЙ ЭКСПЕРИМЕНТ+ (НАЧ+) - запускается программа эксперимента.

2. ВЫКЛОЧИТЬСЯ+ (ВЫК+) - система переходит в исходное состояние.

3. ЖДИ УКАЗАНИЙ+ (ЖДИ+) - система переходит в режим стабилизации заданных на момент приказа параметров.

4. ИДИ ДАЛЬШЕ+ (ИДИ+) - система выходит из режима ожидания.

5. ОСТАНОВИСЬ+ (ОСТ+) - система приходит в исходное состояние и выключает все каналы управления. Оператору сообщается общее число аварийных ситуаций за время эксперимента, перечисляются все каналы, в которых наблюдались аварийные ситуации.

6. ВЫДАЙ КАНАЛУ..К..М+ (ВМД..К..М+) - управление каналом, имеющим номер "К"; в канале устанавливается значение параметра, равное числу "М".

7. БЕТА..М+ (БЕТ..М+) - система переходит в состояние с номером "М".

8. СОСТОЯНИЕ СООБЩИ+ (СОС+) - печатается содержимое всех каналов НИС.

9. АВАРИИ СООБЩИ+ (АВА+) - печатается содержимое каналов НИС, вышедших за уставки.

10. СПРОСИ КАНАЛ..К+ (СПР..К+) - однократно печатается содержимое данного канала.

11. ПЕЧАТАЙ КАНАЛ..К+ (ПЕЧ..К+) - печатается содержимое данного канала за каждый цикл эксперимента.

12. ПРЕКРАТИ ПЕЧАТЬ+ (ПРЕ+) - прекращается вывод информации на печать.

13. ОЪ+ - на двухкоординатном самописце рисуется ось X.

14. ПО X РИСУЙ КАНАЛ..K+ (ПО X..K+) - по X-оси самописца в каждый цикл эксперимента выводится содержимое канала "K".

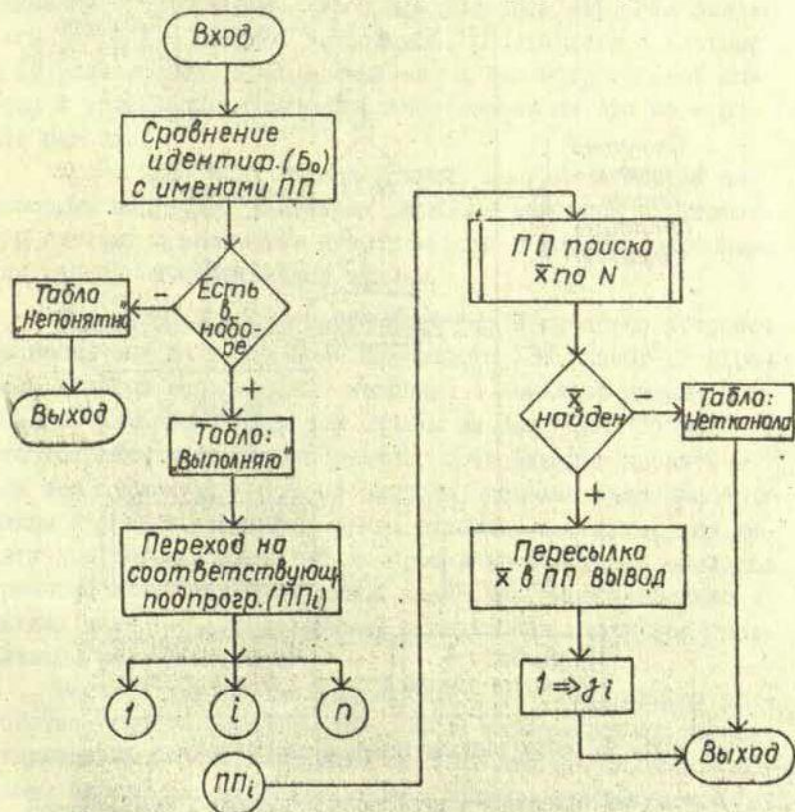
15. ПО У РИСУЙ КАНАЛ..K+ (ПО У..K+).

16. X НЕ РИСУЙ+ (X НЕ+) - прекращается вывод по X-оси.

17. У НЕ РИСУЙ+ (У НЕ+).

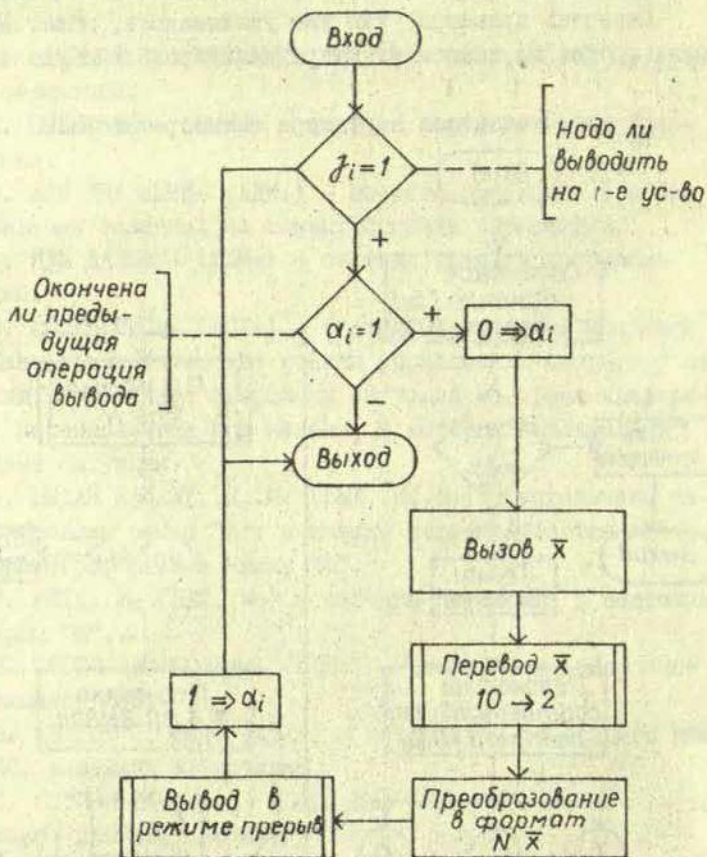
Перечень приказов, как уже указывалось, может быть развит; одно из возможных направлений развития указано в [5].

На рис.3 показана структура интерпретатора.



Р и с. 3. Блок расшифровки операторских приказов (интерпретатор).

На рис. 4 показана подпрограмма обработки запроса на вывод содержимого канала K на одно из устройств вывода. Она осуществляет поиск информации в таблице $\{N, \bar{X}\}$ и пересылку ее в один из программных регистров подпрограммы вывода. Далее в соответствующем разряде регистра запрос на вывод γ устанавливается "1".



Р и с. 4. Блок вывода информации в режиме диалога.

Исходя из принципа блочного построения, оказывается целесообразным все функции вывода объединить в одной программе. Со осуществление этого принципа становится возможным.

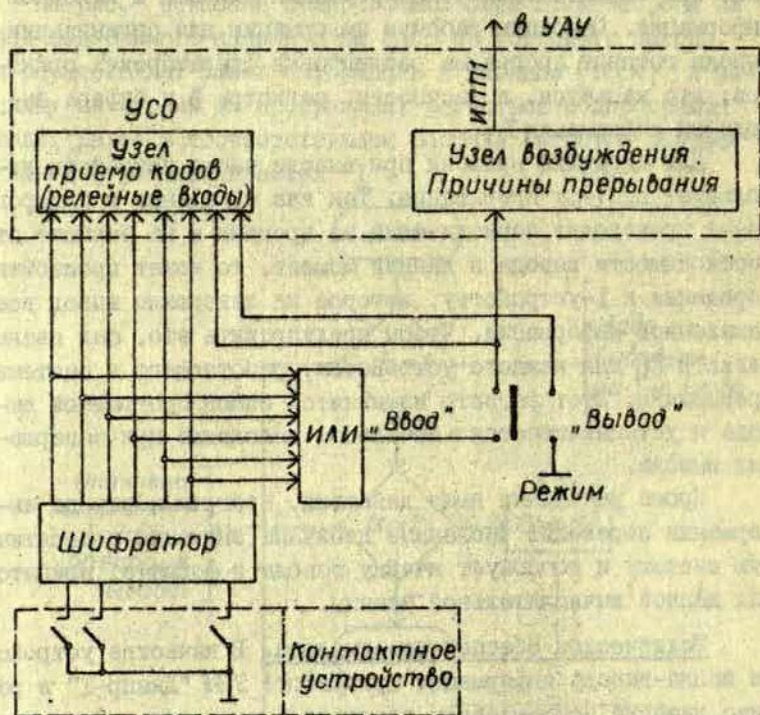
лишь при обязательной стандартизации входной и выходной информации. Основную рабочую информацию для организации вывода готовит программа расшифровки операторских приказов; это касается, в частности, регистра δ и буфера выводимой информации A ;

Для экономии времени при выводе часть устройств использует систему прерывания. Так как обращение к подпрограмме происходит периодически во времени и не зависит от необходимости вывода в данный момент, то может произойти обращение к i -устройству, которое не завершило вывод всей заказанной информации. Чтобы предупредить это, был введен регистр α_i для каждого устройства, работающего в системе прерывания. Этот регистр изменяется самой программой вывода и устанавливается в начальное положение при завершении вывода.

Кроме указанных выше действий, программа вывода информации переводит численные двоичные значения в десятичную систему и компоует ячейку вывода в формате, принятом для данной вычислительной машины.

Техническое обеспечение диалога. В качестве устройства ввода-вывода информации при работе УВМ "Днепр-1" в режиме диалога использовано входящее в комплект машины устройство алфавитно-цифровой печати на базе ЭПМ "Scemtron". Это устройство дополнено узлами, позволяющими вводить в УВМ все алфавитно-цифровые символы, используемые при программной печати, т.е. все буквы русского алфавита, все десятичные цифры, знаки "-" и пробел, кодируемые согласно принятой в УВМ шестизрядной кодировке. Дополнительно введен знак "+", используемый в служебных целях при организации режима ввода.

Вводное устройство подключено к релейным входам устройства связи с объектом УВМ. Прием информации от соответствующих входов и размещение ее в памяти производится согласно программе отработки первой причины прерывания. Структура вводного устройства изображена на рис. 5 и состоит из следующих узлов:



Р и с. 5. Схема устройства ввода знаков в УВМ.

1) контактное устройство; контакты размещены под штоками клавиш и подключены ко входам шифратора, их количество соответствует числу вводимых знаков. Нажатием клавиши ЭТМ находящийся под клавишей контакт замыкается на корпус;

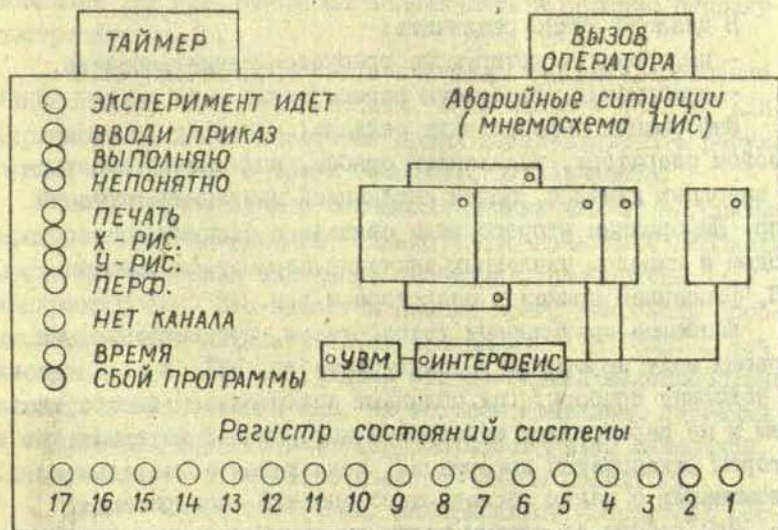
2) двухступенчатый шифратор, вырабатывающий шестизначный код знака и подключенный к релейным входам УВМ;

3) схема ИЛИ, ко входам которой подключены выходные шины шифратора. Выходной сигнал этой схемы служит для возбуждения системы прерывания, а также для контроля наличия информации на выходных шинах шифратора, так как с целью упрощения конструкции вспомогательный регистр для хранения кода очередного вводимого символа не применен;

4) узел возбуждения причины прерывания, представляющий собой один из входов аварийного слова УСО, перестроенный с целью получения однократного возбуждения первой причины прерывания при нажатии любой клавиши ЭПМ.

Режим работы печатающего устройства определяется положением переключателя. В положении "Ввод" выход схемы ИЛИ подключается к релейному входу и к узлу возбуждения прерывания. В положении "Вывод" выход схемы ИЛИ отключается, что блокирует возбуждение причины прерывания при программной печати; дополнительно в УСО выдается признак режима вывода, необходимый для организации работы устройства.

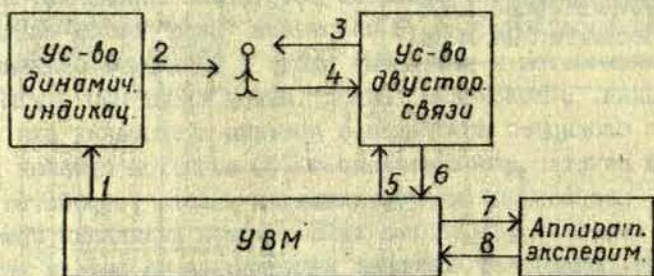
В качестве устройства динамической индикации применено табло. Включение лампочек накаливания на панели индикации табло производится релейными выходами УВМ. Вид этой панели приведен на рис.6.



Р и с. 6. Вид панели индикации табло связи с оператором.

ИСПОЛЬЗОВАНИЕ ТАБЛО В ПРОЦЕССЕ ДИАЛОГА

На рис. 7 показаны основные информационные связи, которые устанавливаются с УВМ во время эксперимента в процессе диалога.



Р и с. 7. Информационные связи НИС и оператора в процессе диалога.

В диалоге можно различить:

- информацию, которую не требуется регистрировать,
- информацию, требующую регистрации.

Информация первого вида связана, например, с общим вызовом оператора, индикацией ошибок, индикацией занятости выводных каналов, общей индикацией аварийных ситуаций и пр. Информация второго вида связана с заданием паспорта, вводом и выводом численных значений параметров каналов НИС, фиксацией приказов оператора и т.п.

Наиболее приемлемыми устройствами, соответствующими первому виду информации, являются табло, ЭЛТ, и тому подобные приборы. Они обладают значительным быстродействием и не перегружают избыточной информацией материал, на котором фиксируется информация, подлежащая обязательному запоминанию с целью последующего анализа эксперимента.

Для работы со вторым видом информации используются телетайпы, ЭПМ с вводом символов в УВМ и пр.

Нами выбран комплекс устройств, включающий табло (см. рис. 5), ЭПМ с вводом в УВМ и двухкоординатный самописец ЦС-021М, подключенный к блоку аналоговых выходов (многоканальный цифро-аналоговый преобразователь).

Взаимодействие оператора с этими устройствами происходит следующим образом.

При наличии сбоя в процессе ввода любого приказа на табло выводится сообщение "НЕПОНЯТНО"; это же сообщение возникает, если среди подпрограмм системы не имеется требующейся для отработки приказа. Если набран номер канала, который отсутствует в системе, появляется сообщение "НЕТ КАНАЛА". О принятом к исполнению приказе сообщается "ИЗПОЛНЯЮ".

При занятых выводных устройствах на табло высвечиваются слова "X-САМОПИСЕЦ", "У-САМОПИСЕЦ", "ПЕЧАТЬ", "ПЕРЕОРАТОР".

Обнаружение сбоев в интерфейсных таблицах высвечивается "ИНТЕРФЕЙС". Прохождение полного цикла программы можно заметить по миганию лампочки табло с надписью "ТАЙМЕР". Эта лампочка остается включенной (выключенной), если остановилась УВМ или программа заиклилась в заранее непредусмотренном месте.

Во время эксперимента высвечивается также сообщение "ЭКСПЕРИМЕНТ ИДЕТ". Оно гаснет, если набрать приказ "ИДИ" (при необходимости некоторое время стабилизировать регулирующие параметры) и после окончания эксперимента.

Выше перечисленные сообщения независимы от алгоритма конкретного эксперимента, однако часто возникает необходимость динамической индикации сообщений, специфичных для конкретной ИИС. Это касается прежде всего аварийных сообщений, которые вырабатываются программой КОНТРОЛЬ. С точки зрения простоты восприятия эти сообщения целесообразно связать с шемосхемой ИИС, на которой аварийная ситуация в отдельном узле отображается соответствующей лампочкой. Одновременно с этим высвечивается "ВЫЗОВ ОПЕРАТОРА".

Другим видом сообщений, зависящих от конкретных особенностей ИИС, является отображение состояния дискретных выходов системы (вентили подачи газа и жидкости, заслонки, выключатели). Индикация этих сообщений особенно ценна при отладке и проверке ИИС. Имея в виду неопределенность в количестве каналов ИИС, связанную с возможностью

развития последней, целесообразно не включать индикацию состояния каналов в мнемосхему, а выполнить их в виде отдельного регистра лампочек, который соответствует имеющемуся набору каналов с дополнительным резервом на случай расширения числа каналов. Соответствие между каналами и регистром задается с помощью специальной программной таблицы, что облегчает внесение любых изменений.

Рассмотрим далее действия оператора при наличии аварийной ситуации. Эффективна следующая схема действий. Сообщение о наличии аварийной ситуации появляется в 1-м табло (связь 2, рис.7). Оператор может игнорировать это сообщение (например, если он знает, что ничего плохого не произойдет), либо среагировать на него. В последнем случае он может выяснить, в каком канале произошла авария. Для этого он введет вопрос "АВАРИЯ+" (связь 4, рис.7), на что получит ответ (связь 3, рис.7) в виде упорядоченных сообщений о номерах контрольных каналов и соответствующих значениях параметров (связь 8, рис.7). Зная, с какими управляемыми каналами связаны датчики контрольных каналов, оператор может заблокировать управляющие каналы (связь 7, рис.7), либо примет решение об остановке эксперимента.

Здесь важно то, что оператор может сравнительно легко и быстро выполнить некоторые незапрограммированные ранее процедуры и "спасти" эксперимент.

ОЦЕНКА ПРЕДЛАГАЕМОЙ АЛЬТЕРНАТИВЫ ДИАЛОГА

В заключение приведем таблицу оценки трех альтернатив диалога: А1 - приведенный выше пример, А2 - диалог с использованием записи приказа с клавиатурн пульта УВМ, но для программного обеспечения указанного примера, А3 - диалог, аналогичный А2, но с традиционной организацией программы, т.е. с пульта заносятся все изменения в соответствующие программы, с которыми взаимодействует оператор. Приводится среднее число манипуляций, которые должен проделать оператор, чтобы реализовать одну из трех форм приказа (вопроса).

Т а б л и ц а

№	Форма приказа	А1	А2	А3
1	< > +	4	12	104
2	< > · < > +	9	28	143
3	< > · < > · < > +	12	30	136

Как видим, А1 в среднем более 2,5 раза эффективнее, чем А2 и в 15 раз, чем А3. Следует отметить, что при этом не оценивалась простота и скорость анализа информации, поступающей от УВМ к оператору в режиме диалога при разных альтернативах. Следует также отметить уменьшение эффективности альтернатив с усложнением формы приказа.

Л И Т Е Р А Т У Р А

1. Оптнер С.Л. Системный анализ для решения деловых и промышленных проблем. М., 1969. 216 с.
2. Диксон Дж. Проектирование систем. М., 1969. 440 с.
3. Математическое и информационное прогнозирование наукой. Киев, 1971. 346 с.
4. См. настоящий сборник, с. 3-16.
5. См. настоящий сборник, с. 50-54.

РЕАЛИЗАЦИЯ ФРАЗ В РЕЖИМЕ ДИАЛОГА

Как отмечалось в [1], программирование эксперимента является одной из важных проблем автоматизации исследований. Аналогично поставленному в [3] вопросу о взаимоотношении программиста и ЦММ положение экспериментатора, пользующегося цифровой аппаратурой управления эксперимента, можно сравнить с взаимодействием "исследователь-лаборант". Если во втором случае исследователь дает указания, которые интерпретируются лаборантом и реализуются в виде последовательности элементарных действий, то в первом случае экспериментатор обычно должен задать всю последовательность элементарных действий в виде программы. Это приводит к возникновению своеобразного барьера между исследователем и цифровой техникой эксперимента, который особенно велик при поисковых исследованиях, когда программа формируется в ходе эксперимента.

Один из эффективных способов снижения величины указанного барьера состоит во включении диалоговых программ в программное обеспечение эксперимента. В свою очередь метод групповых программ [1] сильно упрощает реализацию режима диалога, поскольку управление одной программой, в которой централизована определенная функция, гораздо легче, чем управление несколькими программами, по которым эта функция "размазана".

В [2] описана система диалога, с помощью которой выполняются односложные приказы в режиме интерпретации. При этом необходимо заранее выбрать достаточно полный набор приказов (вопросов), позволяющих оператору во время эксперимента вмешиваться в работу групповых программ. Однако

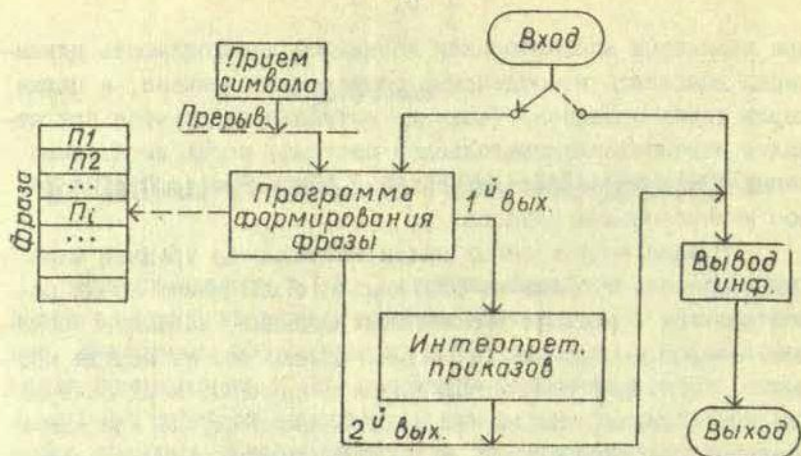
при поисковых исследованиях возникает необходимость циклически выполнять не отдельные односложные приказы, а целые серии таких приказов. Такая же ситуация образуется при отладке научно-исследовательской системы, когда необходимо заиклевать управление отдельными узлами системы или опрос измерительных каналов.

Решение поставленных задач возможно по крайней мере двумя способами. Один из них связан с введением новых более сложных приказов, выполняющих функцию, заданную последовательностью простых приказов. Однако это не всегда возможно, поскольку количество сложных приказов и их содержание часто неизвестно заранее, особенно в случае поисковых исследований. Кроме того, этот путь делает режим диалога громоздким, требует большого объема памяти для хранения соответствующих программ, заставляет оператора запоминать значительное количество приказов.

Другая альтернатива состоит в том, что в программное обеспечение вводятся только две программы и несколько специальных приказов. Одна из программ должна позволить формировать и запоминать фразы, т.е. последовательности простых приказов, которые необходимо в дальнейшем заиклевать.

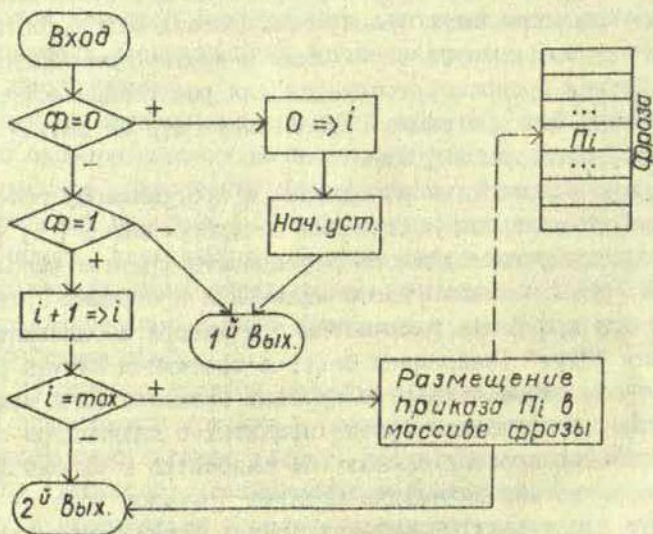
Другая программа требуется для реализации сформированной фразы. Эта программа обрабатывает шаг за шагом простые приказы, входящие в фразу.

Нами разработаны дополнения к программному обеспечению, общие принципы построения которого даны в [1], что позволяет в режиме диалога формировать фразы и выполнять их. На рис. 1 показано место включения программы приема фраз. Эта программа включается при наборе на клавиатуре приказа "КФ+" (заносится $\Phi=1$), в противном случае работает обычная схема отработки простых приказов (I-й выход). В режиме формирования фразы принятый с клавиатуры печатающей машинки простой приказ (П) заносится в массив фразы. Последовательное занесение простых приказов (без их немедленного выполнения) формирует фразу. Признаком начала и конца послышки фразы является приказ "ФРАЗА+" (заносится $\Phi=0$).



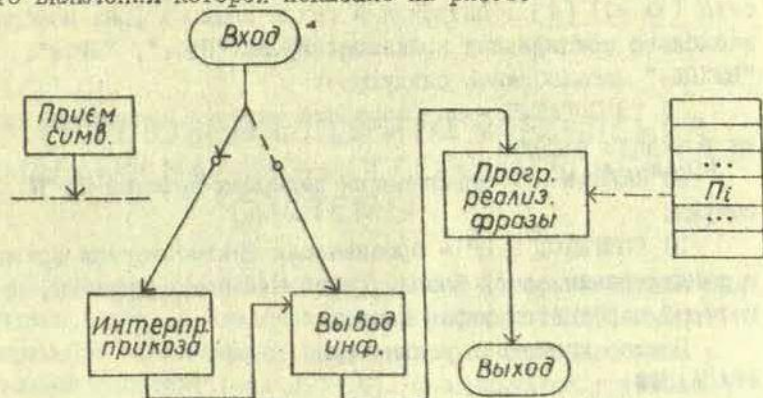
Р и с. 1. Структура программы связи с оператором для формирования фраз в режиме диалога.

Блок-схема алгоритма формирования фразы дана на рис.2. Б₁, 2, 3 – регистры, на которых хранится принятое сообщение оператора.



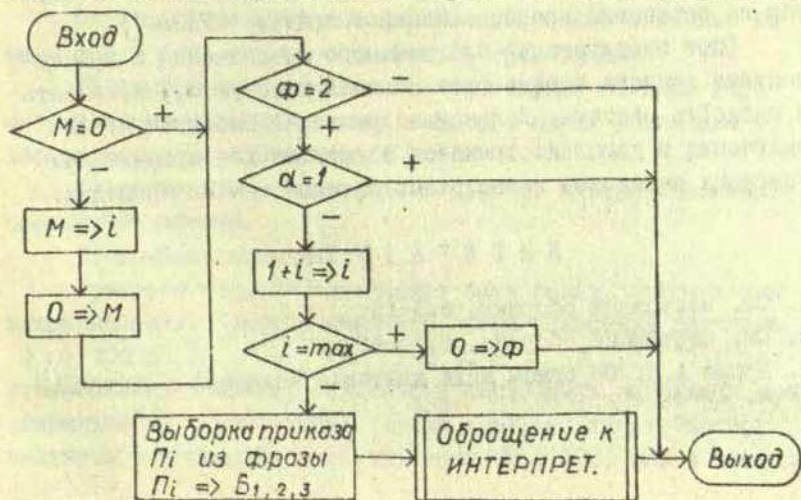
Р и с. 2. Алгоритм формирования фразы.

Реализация фразы осуществляется другой программой, место включения которой показано на рис.3.



Р и с. 3. Структура программы связи с оператором при реализации фразы,

Программа реализации фразы с периодом цикла эксперимента делает выборку очередного приказа из фразы, выдает его в регистры $B_{1,2,3}$ и однократно обращается к ИНТЕРПРЕТАТОРУ ПРИКАЗОВ, который обеспечивает выполнение приказа. Блок-схема алгоритма реализации фраз приведена на рис.4. Вызов программы осуществляется приказом "ПФ+" (заноcится $\Phi=2$).



Р и с. 4. Алгоритм реализации фразы,

На рис. 4 α - регистр признака занятости печатающего устройства ($\alpha = 1$) [2]. Введение в режим диалога фраз породило несколько специфичных приказов. Кроме "ПРФ+", "ПР2+", "ВЫХОД+", понадобились следующие:

1) "ВОЗВРАТ+" - для возврата каретки печатающей машинки в начало строки;

2) "ПАУЗА N+" - организация задержек времени на N секунд;

3) "ПЕРЕХОД К M" - организация циклов внутри фразы и заиклевание самой фразы. Здесь M - номер приказа, на который передается управление.

Пример диалога в режиме фраз:

ФРАЗА+ПРФ+

А. ВОЗВРАТ+ПАУЗА 2+ОПРОСИ ВРЕМЯ+ВЫДАЙ КАНАЛУ 2 I+
ОПРОСИ КАНАЛ 22+ОПРОСИ I42+ СООБЩИ АВАРИИ+

Б. ВОЗВРАТ+ ПАУЗА 10+ВЫДАЙ 2 0+ВЫДАЙ I4 I+ ВЫДАЙ 25
0+ПО X РИСУЙ I2+ПО У РИСУЙ I16+ОПРОСИ ВРЕМЯ+

В. ПАУЗА I6+ВЫДАЙ 2 I+ВЫДАЙ I4 0+ВЫДАЙ 25 I+ПО X
РИСУЙ I3+ПО У РИСУЙ 242+ОПРОСИ I24+ПЕРЕХОД 9+

ПРФ+ФРАЗА+ далее следует выполнение фразы, причем А выполнится один раз, а Б и В заиклятся до тех пор, пока оператор не остановит процесс (наберет приказ "ФРАЗА+").

Опыт эксплуатации программного обеспечения с возможностями диалога в виде фраз показал высокую эффективность и гибкость системы. Дальнейшее развитие мыслится путем включения в комплект приказов элементов для организации условных переходов и несложных вычислительных процедур.

Л И Т Е Р А Т У Р А

1. См. настоящий сборник, с.3-16.

2. См. настоящий сборник, с.32-49.

3. Ершов А.П. Об одном виде контакта человека с машиной. Киев, 1963.

В. Я. Кузьмин

ГРУППОВОЙ АЛГОРИТМ УПРАВЛЕНИЯ КАНАЛАМИ НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ СИСТЕМЫ

Программа, реализующая групповой алгоритм управления каналами научно-исследовательской системы (НИС), входит в программное обеспечение, общее описание которого дано в настоящем сборнике (см. с. 3-16). Она является групповой, так как значения параметров на выходах всех каналов НИС вычисляются централизованно. Это позволяет упростить задачу программирования НИС, использовать унифицированные программы, применить наиболее универсальные методы вычисления, а также существенно упростить отладку и проверку НИС.

Вычисление параметров для выходных каналов системы осуществляется в несколько стадий. Прежде чем описать их, рассмотрим важный вопрос о типах выходных каналов системы.

С точки зрения функционирования системы выходные каналы могут быть условно отнесены в трех типам:

1) программные - зависящие от текущих входных параметров системы и вычисляемых величин, но не зависящие от набора основных состояний;

2) связанные - зависящие только от набора основных состояний системы;

3) комбинированные.

Примером первого типа может быть канал, описываемый высказыванием: "затвор включить, если показания датчиков

$X_1 \geq 100$ и $X_3 \geq X_2$ или $X_2 < 10$ и $X_5 - X_1 \geq \epsilon$ ". Канал второго типа: "затвор включить в состояниях нагрева и стабилизации температуры". Канал третьего типа: "затвор включить в состоянии нагрева, если $X_1 \geq 100$, или в состоя-

ним охлажденца, если $X_3 \geq X_2$ и $X_5 - X_1 \geq \varepsilon$ "

Опишем, как можно построить групповую программу для вычисления параметров дискретных выходов системы.

На первой стадии определяются параметры всех каналов, зависящих только от состояния системы, согласно таблице связи состояние-канал. Это может быть сделано следующим образом. Задается таблица, в которой ключом является число (символ β_i), однозначно соответствующее состоянию НИС, а информация, связанная с ключом, должна представлять перечень значений для всех каналов управления $\{d_k\}$, которые должны быть приняты последними, т.е.:

$$\begin{aligned} & (\beta_1; d_1^1, d_2^1, \dots, d_N^1), \\ & (\beta_2; d_1^2, d_2^2, \dots, d_N^2), \\ & \dots \dots \dots \\ & (\beta_M; d_1^M, d_2^M, \dots, d_N^M). \end{aligned}$$

Таким образом алгоритм определения параметров на первой стадии очень прост и состоит в поиске строки таблицы $\{\beta_i; d_1^i, \dots, d_N^i\}$ для соответствующего β_i . Следует сделать два замечания:

1) параметры тех каналов, которые являются "программными", можно задавать произвольно (например, все $d_j^i = 0$), либо руководствуясь другими соображениями;

2) вместо таблицы информация может быть задана в форме списка.

На второй стадии осуществляется переход от действительных переменных к булевским. В большинстве случаев для этого годится формула

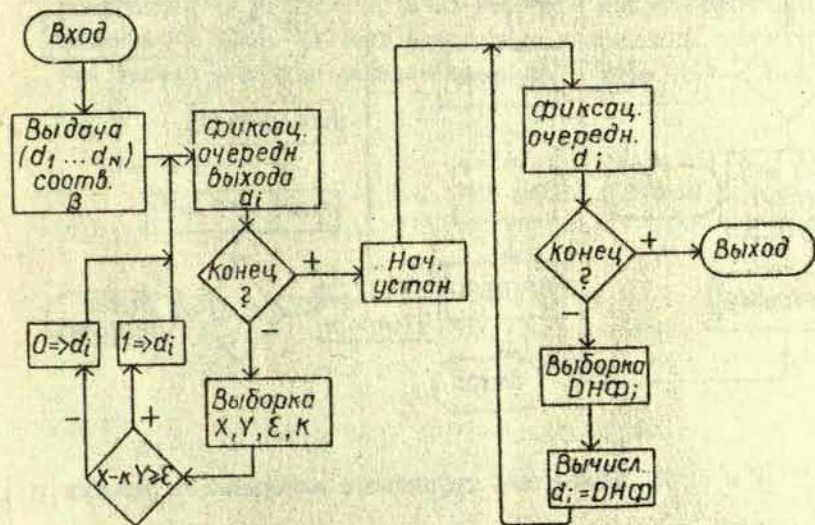
$$d_i (x_i - K_i y_i \geq \varepsilon_i), \quad (I)$$

где X_i - показания датчика или другой параметр, y_i - уставка или другой параметр, K_i - нормирующий множитель,

ε_i - параметр отклонения, d_i - булевский параметр. Далее вычисляется дизъюнктивная нормальная форма (ДНФ) согласно сети, описывающей условия функционирования каналов:

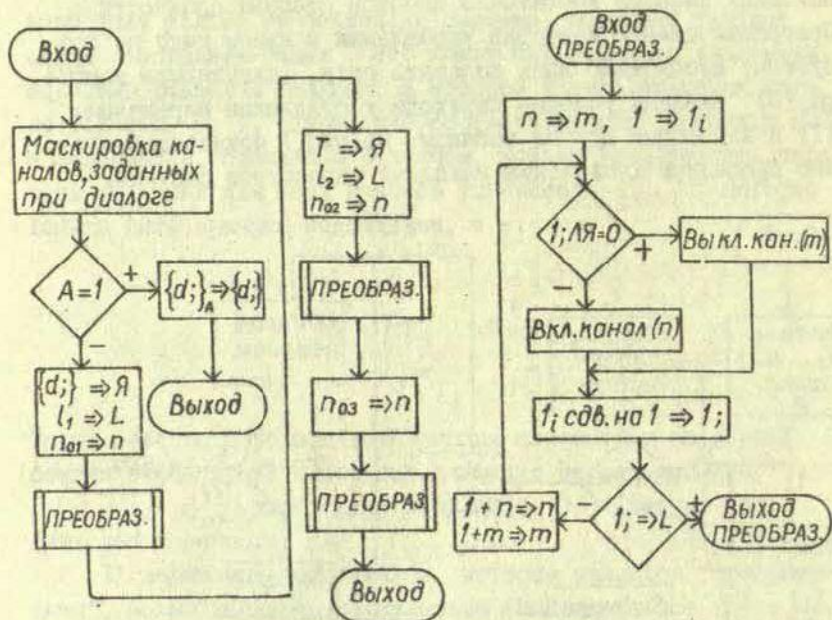
$$d_k = \text{ДНФ} [\{ d_i^{\varphi} \}, \{ d_i \}, S], \quad (2)$$

Такой подход позволяет задавать условия включения выходных каналов достаточно высокой степени сложности. Программирование контуров управления в явном виде не требуется, необходимо лишь задавать сети, описывающие контуры (2), таблицу условий перехода к булевым переменным (I) и некоторые другие таблицы. На рис. I показаны основные процедуры вычисления выходных параметров системы.



Р и с. I. Блок-схема группового алгоритма вычисления выходных параметров дискретных каналов НИС.

Обработка управляющих каналов производится также централизованно одной групповой программой [I] согласно d_k и правилу, либо таблице соответствия d_k с физическими выходами системы (код реле). Эта же программа производит преобразование параллельного кода в последовательный и размножение кода на несколько выходов. На рис. 2 показана блок-схема обработки d_k .



Р и с. 2. Блок-схема группового алгоритма обработки $\{d_i\}$.

Здесь: $A=1$, если была аварийная ситуация, по которой нужно блокировать выходы системы; $D = \{d_i\}$ - кортеж дискретных выходов, который нужно обработать; l - код с единицей на месте, следующем за последним элементом кортежа; n_0 - начальный адрес группы физических выходов; l_i - код с единицей в i -элементе кортежа выходов, например: $l_3 = 000\ 000\ 100$, где $i = 3$.

Обрабатываемый выходной канал может быть замаскирован операторским призывом в режиме диалога. При этом, вместо программно заданного d_i заносится значение, указанное оператором. Чтобы демаскировать канал, оператор

отдает приказ:

"Выдай.. d.. программный"

К сожалению, небулевские выходы не могут быть описаны универсальной формой, подобной ДНФ. Однако и в этом случае достаточно широкий круг задач может быть охвачен групповыми программами вычисления, например, степенных полиномов, описывающих поведение выходов, а также вычисления параметров по аппроксимирующим таблицам. В первом случае при описании системы необходимо задать таблицу коэффициентов полиномов, а во втором - аппроксимирующие таблицы; в обоих случаях необходимо установить соответствие таблиц выходным каналам системы.

МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ОБРАБОТКИ ИНФОРМАЦИИ ВЫЧИСЛИТЕЛЬНО- УПРАВЛЯЮЩЕГО КОМПЛЕКСА

В процессе проектирования вычислительно-управляющих комплексов, в том числе мультипроцессорных научно-исследовательских систем [1] необходимо оценить большое число альтернатив как в аспекте выбора связей аппаратурных подсистем, так и в смысле разработки структуры внутреннего программного обеспечения. Рассматриваемые системы работают в реальном масштабе времени, поэтому на их временные характеристики налагаются более жесткие ограничения, чем на обычные ЭЦМ. Проектировщики должны своевременно определить такие параметры как коэффициент нагрузки центральной ЭЦМ при известном числе периферийных процессоров, время ожидания в очереди программы, имеющей определенный приоритет, степень нагрузки каналов связи и другие.

Интервалы времени между исследуемыми системными событиями настолько малы, что в их пределах явно наблюдается эффект последствия, а иногда нарушается также свойство стационарности. Поэтому для выяснения временных характеристик системы нельзя пользоваться математическими моделями марковских процессов [2].

Если исследуемые последовательности событий подобно [3] описаны имитационными моделями, можно применять метод вероятностного моделирования на ЭВМ. Известно, что для моделирования параллельных процессов на базе "Алгол-60" разработан универсальный язык программирования "Симула". Объем его специального транслятора 5000 инструкций машины "UNIVAC-1107" [4]. Однако, если соответствующий трансля-

тор недоступен, задача может быть решена относительно простым методом, рассматриваемым в настоящей статье.

В процессе моделирования внутреннего программного обеспечения системы, работающей в режиме разделения времени, существенно учесть зависимость интенсивности ее входного потока от скорости обслуживания заказов. Последняя, как известно, зависит от числа пользователей и особенностей системы. Эти факторы в известной ситуации не позволяют пользоваться моделью [5], в которой предусмотрен автономный блок имитации интенсивности входного потока.

Рассматриваемый метод предполагает, что разработчику моделей доступны программы пользователя, временные параметры аппаратуры и альтернативы внутреннего программного обеспечения системы. Имитируемые программы рассматриваются как неавтономные кусочно-непрерывные агрегаты [6] с заданными приоритетами реализации.

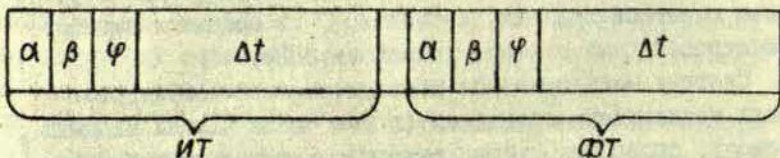
Модель входного потока описывается последовательностью событий, обрываемой операцией обращения процесса пользователя к исследуемой подсистеме, например, к супервизору. Интервалы времени Δt между этими событиями определяются при предположении, что процессор, выполняющий соответствующую программу пользователя, работает в однопрограммном режиме. В случае, когда это происходит в реальном масштабе времени, значения Δt в основном зависят от быстрогодействия внешних устройств ЭЦМ.

Систему моделирования образуют четыре компонента: модели исследуемых процессов (в том числе модели входных потоков), программа управления процессом имитации и накопления статистических данных (ПУН), программа организации диалога с исследователем (ДИАЛОГ) и программы, выполняющие расчет временных характеристик исследуемого объекта. Для формализации моделей вводится следующая терминология. Процессором названо устройство, способное некоторое время самостоятельно выполнять программу. Процессом Q_j обозначается последовательность событий, реализуемая j -ым процессором. Агрегатом q_j именуется программа с

определенным приоритетом реализации на некотором процессоре. Агрегат может находиться в следующих основных состояниях: запрошенном ($\beta = 1$), активизированном ($\alpha_i = 1$), пассивном ($\alpha_i = \beta_i = 0$). Каждому агрегату q_i задается множество входных сигналов $\{X_i\}$, которые определяют разветвления его алгоритма, а также множество его выходных сигналов $\{y_i\}$. Дополнительной координатой Δt_i состояния агрегата обозначен текущий интервал времени между моментами, в которых q_i изменяет выходные сигналы $\{y_i\}$. Отметим, что элементы X_i, α_i, β_i могут принадлежать множеству $\{U_k\}$, где $k \neq i$, а элементы X_i, β_i - также множеству $\{Y_i\}$.

Внесение любых изменений в множества $\{\alpha_i\}, \{\beta_i\}, \{X_i\}$ рассматривается как событие.

Процесс моделирования фиксируется при помощи исходной таблицы (ИТ) и формируемой таблицы (ФТ), в которых для каждого агрегата отведена одна строка (рис. 1). Агрегаты в таблице группированы по соответствующим процессам и для каждого процессора расположены в порядке убывания приоритетов. Высший приоритет в каждом процессе выделяется для q^0 , который имитирует систему прерывания программ процессора. Кроме исходной и формируемой таблицы, имеется общая для всех процессов таблица значений $\{X_i\}$



Р и с. 1. α, β - признаки активизации и запроса агрегата; φ - признак начала описания процесса в таблице.

Следует указать на отличие между алгоритмом [7] и принципами построения рассматриваемой системы моделирования. Известно, что координата состояния Δt_i меняется во времени в зависимости от активности более приоритетных

агрегатов соответствующего процессора. Поэтому состояния

q должны кодироваться переменными $\alpha_i, \beta_i, \{X_i\}, \{\alpha_k\}$, где $k \neq i$. Однако число состояний, получаемых таким способом, слишком большое, чтобы руководствуясь методом разработки модели [7], их перечислить, анализировать и программировать переходы между ними. Поэтому функция генерации кодов состояний возлагается на моделирующую ЭЦМ и распределяется между моделями и ПУН. В описаниях моделей $\{q_i\}$ указываются только значения переменных $\{\alpha_i\}, \{\beta_i\}, \{\Delta t_i\}, \{X_i\}$, а не коды состояний. ПУН осуществляет анализ состояний и планирование событий во времени.

Модель каждого агрегата q_i в общем случае задается в форме программы путем m -кратного применения кортежа с операторами A_{rs} четырех типов. Если r номер типа оператора и s порядковый номер применения его в описании, то обобщенная модель агрегата q приобретает вид: $A_{11}, A_{21}, A_{31}, A_{41}, A_{12}, A_{22}, A_{32}, A_{42}, \dots, A_{1m}, A_{2m}, A_{3m}, A_{4m}, A_{rs}$ -оператор разветвления имитирующей программы по результату проверки некоторого логического условия, заданного над значениями переменных подмножеств $\{\alpha\}, \{\beta\}, \{X_i\}$. Оператор A_{1s} может быть и пустым.

A_{2s} - оператор присвоения новых значений некоторым переменным множества $\{y\}$. Значения переменных могут быть заданы априорно или же вычисляться во время имитации по некоторому закону распределения вероятностей.

A_{3s} - оператор фиксации текущего интервала времени Δt_{is} до следующего события в агрегате q_i . К числу $\{A_{3s}\}$ относятся как операторы внесения заранее заданных значений Δt_{is} в формируемую таблицу, так и операторы, которые значения Δt_{is} вычисляют по некоторой функции.

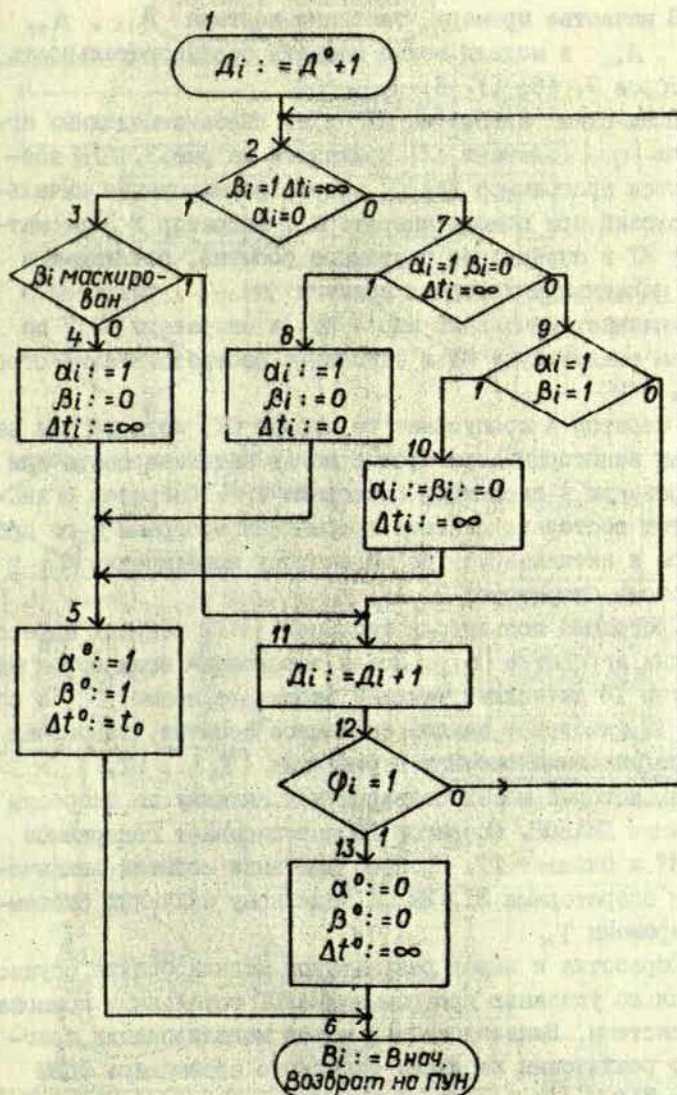
A_{4s} - оператор формирования возврата на модель и передачи управления на ПУН. Кроме перечисленных операторов, модель может включать операторы вспомогательного характера. Согласно определению кусочно-непрерывного агрегата сумма значений Δt_i в некоторой ветви q_i является собственным временем его определения [6].

Разработка моделей q_i связана с некоторыми особенностями. Запрос возбуждения агрегата q_i осуществляется путем формирования для него в ФТ значений $\alpha_i = 0, \beta_i = 1, 0 \leq \Delta t_i \leq \infty$. Символом ∞ в данном случае обозначен код машинной бесконечности. Если агрегат самовозбуждающийся, то в момент перехода q_i в пассивное состояние эту операцию осуществляют его операторы A_{2s}, A_{3s} . Агрегат q_i , в котором не предусмотрено самовозбуждение, заканчивает фазу активности с занесением в ФТ значений $\alpha_i = 1, \beta_i = 1, \Delta t_i = \infty$. Перевод любого агрегата в пассивное состояние обязательно осуществляется активизацией модели q_i системы прерывания программ соответствующего процессора. Это осуществляется путем занесения в ФТ значений $\alpha^0 = 1;$

$\Delta t^0 = 0$, если $\beta^0 = 0$ или же $\alpha^0 = 1, \Delta t^0 = \Delta t^0 - \Delta t_{\min}$ при $\beta^0 = 1$. Важно отметить, что операция перевода агрегата в состояние пассивности без дальнейшего самовозбуждения применяется в модели входного потока, если текущий интервал времени между заявками на обслуживание зависит от внутреннего программного обеспечения исследуемой системы.

Агрегат q^0 , имитирующий систему прерывания программ отдельного процессора (рис.2), несколько отличен от остальных моделей. После его активизации оператор 1 устанавливает адрес строки ИТ первого агрегата, подчиненного q^0 . Далее по строкам анализируется ИТ и вносятся изменения в ФТ. Операторы 2-6 отмечают в ФТ время работы системы прерывания программ исследуемой ЭВМ в режиме приема запроса на активизацию i -го агрегата, а операторы 2, 7, 8, 5, 6 фиксируют время самого процесса активизации.

Последовательность операторов 2, 7, 9, 10, 5, 6 фиксирует задержку процессора, возникающую по запросу перевода в пассивное состояние q_i . Операторы 11, 12, 13, 6, заканчивают активную фазу q^0 при условии, что все запросы j -го процессора рассмотрены.



Р и с. 2. t_0 - время срабатывания системы прерывания программы процессора; $V_{нач}$ - начальный адрес программы, имитирующей q^0 ; V_i - текущий адрес программы, имитирующей q_i ; D_i - адрес анализируемой строки исходной таблицы; D^0 - адрес строки агрегата q^0 в исходной таблице.

В качестве примера типового кортежа A_{15} , A_{25} , A_{35} , A_{45} в модели можно назвать последовательность операторов 7, (8, 5), 6.

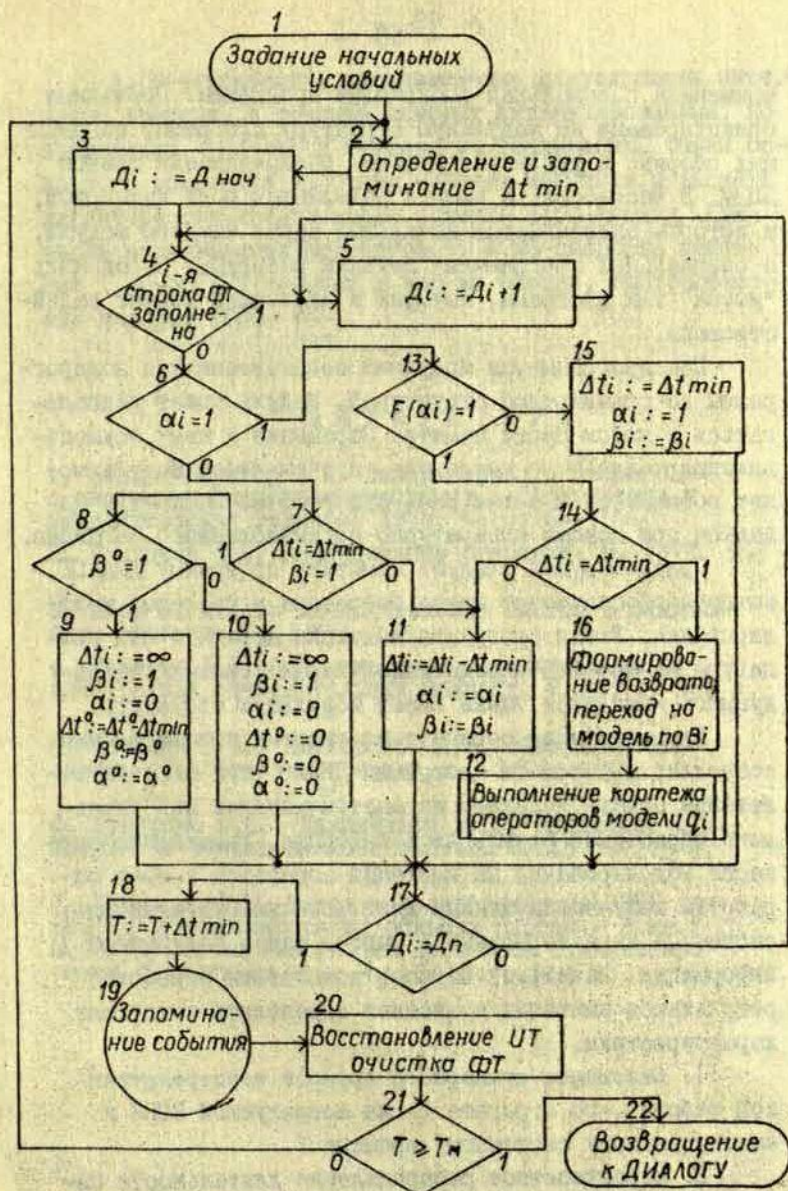
Блок-схема алгоритма ПУН и ее связь с моделями агрегатов $\{q_i\}$ (элемент 12) приведена на рис.3. ПУН возбуждается программой ДИАЛОГ. После формирования начальных условий при помощи оператора 1 оператор 2 пересматривает ИТ и определяет системное событие, реализуемое через наименьший интервал времени Δt_{\min} . Оператор 3 устанавливает начальный адрес ИТ, а операторы 4-17 по строкам анализируют ИТ и заполняют соответствующие строки ФТ.

Оператор 4 пропускает те строки ФТ, которые уже заполнены некоторым агрегатом с целью активизировать или же перевести в пассивное состояние q_i . Оператор 8 анализирует состояние системы прерывания программ j -го процессора и активизирует ее по запросу возбуждения $\{q_j\}$ при помощи операторов 9, 10.

Программа пользуется таблицей $\{B_i\}$ текущих адресов имитации агрегатов $\{q_i\}$. После заполнения всех строк ФТ оператор 18 вычисляет текущее системное время T , а оператор 19 фиксирует каждое системное событие, запоминая текущие значения таблицы переменных $\{X_i\}$, ИТ, T в массив, который может подвергаться анализу по запросам программы ДИАЛОГ. Оператор 20 переписывает содержимое ФТ в ИТ и очищает ФТ. Процесс имитации событий заканчивается операторами 21, 22 по заданному значению системного времени T_n .

Обработка и вывод результатов моделирования осуществляется по указанию программы ДИАЛОГ остальными компонентами системы. Вышеизложенный метод моделирования программно реализован на языке базисного ассемблера ЭЦМ "Днепр-21".

Для того, чтобы было удобно собрать систему моделирования из частей, добавить, изъять или заменить отдельные ее компоненты, а также упростить процесс отладки,



Р и с. 3. $F(\alpha_i)$ – булева функция, которая равна единице, если рассматриваемый Q_i в данный момент имеет высший приоритет среди активизированных агрегатов соответствующего процессора; n – число строк в таблице; T_M – требуемое системное время моделирования; T – текущее системное время.

применена специальная собирающая программа. Последняя ориентирована на модульную структуру программ, подлежащих сборке, и реализует функцию распределения памяти ЭЦМ. В частности, в памяти выделяется поле паспортов, в котором записывается начальный адрес каждого модуля, а специальный программный магазин позволяет определить "имена" тех программ, которые в настоящий момент задействованы.

Под программными модулями подразумеваются подпрограммы со стандартной структурой. Модуль может располагаться в любом месте памяти. Обращение к нему осуществляется через поле паспортов, причем любой модуль может обращаться к остальным. Его рабочие параметры задаются при помощи аппаратурно организованного магазина.

Среди модулей следует отметить программу ДИАЛОГ, которая обеспечивает связь оператора с системой моделирования. Любая программа имитации агрегата или процедура вычисления определенной характеристики исследуемого комплекса также имеет модульную структуру.

Рассмотренная организация системы моделирования позволяет при помощи программы ДИАЛОГ посредством телетайпа задать исходные параметры моделей или управлять обработкой результатов имитации. Разделение процесса моделирования на имитацию агрегатов и этап обработки полученных данных дает возможность вычислять системные характеристики с учетом ранее полученной информации. Начальный вариант подсистемы обработки результатов имитации позволяет определить следующие характеристики.

1. Отношение суммарного времени непосредственной работы i -го агрегата t_i на исследуемой ЭЦМ к имитированному системному времени T .
2. Вероятностное распределение длительности интервала непрерывной работы i -го агрегата.
3. Вероятностное распределение времени ожидания в очереди на активизацию запрошенного агрегата q_i

4. Вероятностное распределение длительности интервалов времени, в течение которых активизированный, но прерванный Q_i стоит в очереди на реализацию. Объем основного комплекта программ системы имитации и расчета характеристик исследуемого объекта 1100 команд. При числе моделируемых агрегатов $n = 20$ скорость имитации $(40-50)10^3$ событий в час на машине, быстродействие которой $10000 \frac{\text{оп}}{\text{сек}}$.

Л И Т Е Р А Т У Р А

1. Бернуп-Берикоф А.А. Выбор варианта связи ЭЦМ с системами экспериментирования. - "Автоматика и вычислительная техника", 1973, 2, с. 64-70.
2. Вентцель Е.С. Исследование операций. М., 1972. 551 с.
3. Шерр А. Анализ вычислительных систем с разделением времени. М., 1970. 135 с.
4. Поляков Ю.Г. Вероятностное моделирование на электронных вычислительных машинах. М., 1971. 400 с.
5. Мачулин В.В., Пятибратов А.П. Эффективность систем обработки информации, М., 1972. 280 с.
Гл. 3, с.100-145.
6. Бусленко Н.П., Калашников В.В., Коваленко И.Н. Лекции по теории сложных систем. М., 1973. 273 с.
7. Генделев Г.М. Метод моделирования на ЭЦМ процессов функционирования больших систем. - В кн.: Методология моделирования процессов функционирования больших систем. М., 1968, с.38-51.

БАЗИСНЫЙ АССЕМБЛЕР ЭЦВМ „ДНЕПР-21“

Задача кибернетизации научного эксперимента в Проблемной лаборатории физики полупроводников связана с большим объемом работ по программированию на ЭЦВМ "Днепр-21", притом на языке, близком к машинному языку. Существующие языки программирования ЭЦВМ "Днепр-21" для этого либо неэффективны (ЧКЦ - числовой код), либо не позволяют использовать все возможности ЭЦВМ (АКЦ - автокод). Поэтому была разработана система программирования, дающая программисту возможность:

- использовать символические коды операций,
- использовать символические адреса наряду с численными,
- вводить программы и исправления как с перфоленты, так и с телетайпа,
- писать и транслировать программу по частям,
- собрать из транслированных частей рабочую программу,
- хранить транслированные и нетранслированные программы и их части на магнитной ленте,
- вести работу с системой в режиме диалога.

Система программирования состоит из языка программирования, библиотеки программ, программы диалога, собирающей программы и сервисных программ. Система получила название базисного ассемблера (БА), так как она может быть положена в основу программного обеспечения.

Чтобы работать в БА, необходимо знать структуру памяти ЭЦВМ, способы представления информации в памяти и систему машинных команд. Напомним некоторые сведения о структуре памяти.

Следует различать физическую и логическую структуры памяти. Физическая структура памяти задана аппаратно. Основной единицей физической структуры памяти является символ. Логическая структура памяти задана программно. Основной единицей логической структуры памяти является машинное слово (в терминологии [1] - слово). Следующий уровень логической структуры памяти - команды и значения, состоящие из группы машинных слов, в частности из одного машинного слова. Более высокие логические структуры памяти в этой работе не будут рассмотрены.

ОПИСАНИЕ ЯЗЫКА БАЗИСНОГО АССЕМБЛЕРА

Все конструкции языка строятся из следующих знаков:

0 1 2 3 4 5 6 7 8 9 A B V Г Д Е Ж З И К Л

М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Э Ю Я

F G I J L N Q R S U V W Z + - / , . () = : ? _

Знак _ обозначает пробел.

Основная конструкция языка БА - это БА-слово (в дальнейшем просто слово), т.е. последовательность БА-знаков, содержащая знак _ только в конце.

Слова подразделяются на основные и вспомогательные. Слово называется основным, если в транслированной программе ему соответствует одно или несколько последовательных машинных слов. Основные слова - коды операций, значения, адреса значений. Вспомогательные слова не имеют соответствующих машинных слов. Эти "лишние" слова являются платой за удобства, предоставляемые ассемблером. В большинстве случаев замена БА-слов машинными словами происходит двумя путями:

- перегруппировкой двоичных разрядов БА-слова (целые, числа, численные адреса, коды);
- заменой согласно таблицам - постоянным и вырабатываемым в процессе трансляции (символические адреса, коды операций).

Следующая важная конструкция языка БА - строка, т.е. последовательность слов. Строки разделяются (между

собой) словом разделителем , — (запятая и пробел). Знак , (запятая) резервируется только для разделения строк. Первое слово строки вспомогательное — номер строки — это десятичное число без знака, не большее 9999 и записанное не более чем четырьмя цифрами, за которыми может стоять одна буква русского алфавита. Ассемблер рассматривает строки по возрастанию номеров, независимо от их действительного порядка. Возрастающий порядок номеров строк:

1, 1А, 1Б, ..., 1Я, 2, 2А, 2Б, ..., 2Я, ..., 9999, ..., 9999Я

Необязательно использовать все номера подряд. Если есть несколько строк с одинаковыми номерами, то учитывается та, которая написана последней.

Строки бывают трех типов: команды, константы и псевдокоманды (описания). Слово, следующее за номером строки, определяет тип строки. Если это слово К или К., то это константа; если Р, Р., Э, Н, СГМГ, СГА, ИС, ПСТ, КОМ — псевдокоманда; в любом другом случае строка будет рассмотрена как команда. Перед рассмотрением различных типов строк познакомимся со способами записи адресов и значений.

В БА адреса бывают трех видов — численные, символические и номера строк. Запись численных адресов: математические адреса записываются как восьмеричный номер ячейки, за которым следует точка и номер символа; запись относительных адресов от математических отличается только буквой Р после адреса. Символический адрес записывается как последовательность букв и цифр длиной не более шести и начинающаяся с буквы. Каждому символическому адресу соответствует некоторый математический или относительный адрес. Запись номеров строк уже рассмотрена; заметим, что номер строки в качестве адреса значения — основное слово. Каждому номеру строки соответствует некоторый математический адрес.

Адреса могут быть модифицированы при помощи десятичного целого. Это целое прибавляется к соответствующему численному адресу. Пример записи:

201.1 +3, ФТ-2, IC+9

Запись значений. Различаются следующие типы значений: целые, числа, коды, адреса и посимвольные значения.

Целые и числа бывают десятичными и восьмеричными. Перед восьмеричными целыми и числами ставится точка.

Ц е л о е. Перед целым должен быть знак.

Примеры: -0 , $+031$ - десятичные целые,

$.+0$, $.-1172$ - восьмеричные целые.

Если в записи целого нет незначащих нулей, то целое занимает наименьшее возможное количество символов. Каждый незначащий нуль удлиняет целое на один символ.

Пример: целое $.-00177$ в памяти будет представлено в виде $10000000\ 00000000\ 0.111111$.

Ч и с л о. Числа бывают с фиксированной или плавающей запятой. Для чисел с фиксированной запятой обязателен знак, после которого стоит точка.

Примеры: $-.3128$, $+.400$ - десятичные числа с фиксированной запятой,

$.-.570$, - восьмеричное число с фиксированной запятой.

После трансляции каждая полная или неполная пара цифр занимает один символ.

Запись числа с плавающей запятой состоит из двух частей - порядка и мантиисы. Порядок записывается как целое, мантииса как число с фиксированной запятой, только точка после знака опускается:

Примеры: $+2+1290$ - десятичное число с плавающей запятой,

$.+2-300472$ - восьмеричное число с плавающей запятой.

К о д. Содержимое каждого символа значения записывается двоичными или восьмеричными цифрами. После записи содержания каждого символа ставится M, если этот символ с маркером, или H, если без маркера. Перед кодом ставится точка.

Примеры: 1. кодн $.00101100M$ и $.054M$ эквивалентны.

2. $.031H\ 01110100M214H$.

Адрес. Перед адресом ставится двоеточие.

Примеры: :200.I, :ALFA+10, :103.

Посимвольное значение. Содержимое каждого символа значения записывается одним знаком, согласно приложению 2 в [1]. Посимвольное значение должно быть словом; в его записи не должно быть запятых. Перед посимвольным значением пишется знак равенства.

Примеры: =2ABC, =D+/8I3.

Рассмотрим псевдокоманды. Все слова, входящие в запись псевдокоманд - вспомогательные.

Псевдокоманда PCT. Форма:

$\langle N \rangle _ PCT _ .$

Здесь и в дальнейшем $\langle N \rangle$ будет обозначать номер строки. Псевдокоманда PCT удаляет строку с номером N. Используется для исправления БА-программы.

Псевдокоманда КОМ. Форма:

$\langle N \rangle _ КОМ _ \langle \text{произвольная последовательность знаков, не содержащая запятую} \rangle _ .$

Используется для записи комментариев.

Псевдокоманда Э. Это самая универсальная псевдокоманда для описания символических адресов.

Форма:

$\langle N \rangle _ Э _ \langle A \rangle _ \langle B \rangle .$

Псевдокоманда указывает на эквивалентность символического адреса $\langle A \rangle$ адресу $\langle B \rangle$. Адрес $\langle B \rangle$ может быть любого типа, простой или модифицированный.

Псевдокоманды P и P. Используются для присвоения символических адресов данным в рабочем поле. Форма:

$\langle N \rangle _ P _ \langle A \rangle _ n _ ,$
 $\langle N \rangle _ P. _ \langle A \rangle _ n _ ,$

В рабочем поле резервируется n символов; самому левому символу присваивается символический адрес $\langle A \rangle$. n - десятичное целое без знака. Зоны, резервируемые псевдоко-

мандой P, размещаются с начала рабочего поля одна за другой в порядке их описания. В случае псевдокоманды P, зона начинается всегда с начала ячейки.

Использование остальных псевдокоманд связано с сегментацией. БА-сегментом называется последовательность строк, транслируемых отдельно. Результатом трансляции БА-сегмента является машинный сегмент или просто сегмент. Программа может состоять из одного или нескольких сегментов. Транслированные сегменты объединяются в рабочую программу собирающей программой. Так как сегменты транслируются независимо один от другого, встает вопрос о связях между ними - об использовании данных, находящихся в других сегментах и взаимных передачах управления. Оба случая сводятся к использованию общих адресов. Эти адреса могут быть численными или символическими, так как номера строк создаются для каждого сегмента независимо от других. Символические адреса, соответствующие относительным адресам, действуют только в том сегменте, в котором они описаны. Любой символический адрес, которому соответствует математический адрес, становится известным во всей программе (глобальный адрес). Каждый глобальный адрес должен быть описан только в одном сегменте. В других сегментах он используется без какого-либо описания.

П с е в д о к о м а н д а Н. Форма:

$\langle N \rangle _ N _ \langle A \rangle$.

$\langle A \rangle$ - математический адрес - определяет начальный адрес всей программы, а также сегмента, содержащего псевдокоманду N; последняя может присутствовать не более чем в одном БА-сегменте программы. Если ни один из БА-сегментов программы не содержит псевдокоманду N, то все сегменты должны быть написаны так, чтобы их работа не зависела от места их расположения в памяти.

П с е в д о к о м а н д ы CGA и CGMT определяют названия БА-сегмента и транслированного сегмента соответ-

ственно. Формы:

$\langle N \rangle _ _ \text{СТА} _ _ \langle \text{НС} \rangle _ _$

$\langle N \rangle _ _ \text{СТМТ} _ _ \langle \text{НС} \rangle _ _$

$\langle \text{НС} \rangle$ - название сегмента - последовательность не более восьми БА-знаков, за исключением $_$ и $,$ (запятой).

Псевдокоманда ИС содержит названия тех сегментов, в которых описаны глобальные адреса, используемые в сегменте, содержащем псевдокоманду ИС. Форма:

$\langle N \rangle _ _ \text{ИС} _ _ \langle \text{НС} \rangle _ _ \langle \text{НС}' \rangle _ _ \langle \text{НС}'' \rangle _ _ \dots$

Запись констант. Формы записи:

$\langle N \rangle _ _ \text{K} _ _ \langle \text{значение} \rangle _ _ \dots _ _ \langle \text{значение} \rangle _ _ ,$

$\langle N \rangle _ _ \text{K} _ _ \langle \text{симв.адрес} \rangle _ _ \langle \text{значение} \rangle _ _ \dots _ _ \langle \text{значение} \rangle _ _ .$

В результате трансляций БА-константы получается последовательность значений. Самому левому символу первого значения можно присвоить символический адрес; заметим, что этот символ имеет еще один адрес - номер строки. В БА-константе возможен еще один тип значений - циклическое значение - перед каким-либо значением можно поставить в скобках целое, обозначающее количество повторений значения. Например, циклическое значение $(+3)=\text{AB}$ является более удобной формой записи значения $=\text{ABABAB}$. Если после номера строки вместо K написать K. , то последовательность значений после трансляции размещается с начала ячейки.

Запись команд. Форма записи команд:

$\langle \text{номер строки} \rangle _ _ \langle \text{КОП} \rangle _ _ \langle \text{I операнд} \rangle _ _ \dots _ _ \langle \text{n-ый операнд} \rangle$

Адреса второго ранга ставятся в скобки; адреса второго ранга с индексацией ставятся в скобки и после них пишется И.

Примеры: I2I.0 - адрес первого ранга,

(П+4) - адрес второго ранга,

(A)И - адрес второго ранга с индексацией.

Запись значений непосредственно заданных операндов такая же, как запись значений в константах.

Для сокращения записи форм команд будут использованы следующие обозначения:

- < N > - номер строки,
- < A1 > - адрес первого ранга,
- < A2 > - адрес второго ранга,
- < A2i > - адрес второго ранга с индексацией,
- < A > - общее обозначение для < A1 > , < A2 > , < A2i > ,
- < H3 > - непосредственно заданный операнд.

Команды арифметических операций с целыми и числами и команды логических операций имеют три операнда: первые два суть собственно операнды, третий показывает, куда должен быть записан результат операции. В этих командах для обозначения сумматора будет использован знак / , для обозначения магазина - знак H.

Коды операций следующие:

- для целых:

- + сложение,
- вычитание,
- умножение;

- для чисел:

- +Д - сложение,
- Д - вычитание,
- Д - умножение,
- /Д - деление;

логических:

- +Л - логическое сложение,
- Л - логическое умножение,
- +П - поразрядное сложение,
- +Т - поразрядное сложение с остановом.

Форма команды для этих операций без применения магазина:

< N > ⊂ < КОП > ⊂ $\left[\begin{array}{c} / \\ \langle A \rangle \end{array} \right] \left[\begin{array}{c} \langle A \rangle \end{array} \right] \left[\begin{array}{c} / \\ \langle H3 \rangle \end{array} \right] \left[\begin{array}{c} \langle H3 \rangle \end{array} \right] \left[\begin{array}{c} / \\ \langle A \rangle \end{array} \right]$

где $\langle \text{КОП} \rangle$ — один из вышеприведенных кодов операций; запись типа $\left[\begin{array}{c} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{array} \right]$ здесь и в дальнейшем означает выбор одной из конструкций, заключенных в квадратные скобки. Пример: $29 \text{ П} / .007\text{М} (\text{ПМТ})\text{И}$.

Для этих операций с применением магазина имеется три формы (КОПы те же):

1) $\langle N \rangle \text{ П} \langle \text{КОП} \rangle / \text{ П} \text{ П} /$.

Содержимое регистра магазина уменьшается,

2) $\langle N \rangle \text{ П} \langle \text{КОП} \rangle / \text{ П} \text{ П}$.

Содержимое регистра магазина не изменяется,

3) $\langle N \rangle \text{ П} \langle \text{КОП} \rangle \left[\begin{array}{c} / \\ \langle A \rangle \end{array} \right] \left[\begin{array}{c} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{array} \right] \text{ П}$.

Содержимое регистра магазина увеличивается.

Команда для пересылки целых и чисел имеет две формы:

1) $\langle N \rangle \text{ П} \langle \text{КОП} \rangle \left[\begin{array}{c} / \\ \langle A \rangle \end{array} \right] \left[\begin{array}{c} \langle A \rangle \\ \text{П} \end{array} \right]$

2) $\langle N \rangle \text{ П} \langle \text{КОП} \rangle \left[\begin{array}{c} \text{П} \\ \langle A \rangle \end{array} \right] \left[\begin{array}{c} \langle A \rangle \\ / \end{array} \right]$

$\langle \text{КОП} \rangle$ для пересылки целых П, для чисел ЦД.

Пример: $30 \text{ П} .+377\text{М} /$.

В команде П вместо / можно записать любой другой регистр, доступный программисту. Форма записи названий регистров,

::РАК, ::РБ, ::РМ, ::РН.

Пример: $31 \text{ П} \text{ ЯЯЯ} ::\text{РБ}$.

Команда сдвига:

$\langle N \rangle \text{ П} \langle \text{КОП} \rangle \left[\begin{array}{c} / \\ \langle A \rangle \end{array} \right] \left[\begin{array}{c} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{array} \right] \left[\begin{array}{c} / \\ \langle A \rangle \end{array} \right]$

<КОП> для сдвига арифметического СДА, для сдвига арифметического с маркером СДАМ, для сдвига логического СДЛ.

Первый операнд сдвигаемый, второй - константа сдвига, третий показывает, куда записать результат.

Нормализация:

$$\langle N \rangle \text{ — НРМ — } \left[\begin{array}{c} / \\ \langle A \rangle \\ \langle НЗ \rangle \end{array} \right] \text{ — } \langle A \rangle \text{ — } \left[\begin{array}{c} / \\ \langle A \rangle \end{array} \right],$$

первый операнд - нормализуемое число,

второй - куда записать результат,

третий - куда записать число сдвига.

Вычитание модулей:

$$\langle N \rangle \text{ — М — } \left[\begin{array}{c} / \\ \langle A \rangle \\ \langle НЗ \rangle \end{array} \right] \text{ — } \left[\begin{array}{c} \langle A \rangle \\ \langle НЗ \rangle \end{array} \right] \text{ — } \left[\begin{array}{c} / \\ \langle A \rangle \end{array} \right],$$

операнды те же, что и при вычитании чисел.

Присвоение знака:

$$\langle N \rangle \text{ — ПЗН — } \left[\begin{array}{c} \langle A \rangle \\ \langle НЗ \rangle \end{array} \right] \text{ — } \left[\begin{array}{c} / \\ \langle A \rangle \\ \langle НЗ \rangle \end{array} \right] \text{ — } \left[\begin{array}{c} / \\ \langle A \rangle \end{array} \right],$$

первый операнд - число, знак которого берется,

второй - число, которому знак присваивается,

третий - куда записать результат.

Формирование числа:

$$1) \langle N \rangle \text{ — ФД — } \left[\begin{array}{c} \langle A \rangle \\ \langle НЗ \rangle \end{array} \right] \text{ — } / \text{ — } \langle A \rangle,$$

$$2) \langle N \rangle \text{ — ФД — } \left[\begin{array}{c} \langle A \rangle \\ \langle НЗ \rangle \end{array} \right] \text{ — } \left[\begin{array}{c} \langle A \rangle \\ \langle НЗ \rangle \end{array} \right] \text{ — } \left[\begin{array}{c} / \\ \langle A \rangle \end{array} \right];$$

первый операнд - длина формируемого числа,

второй - исходное число,

третий - куда записать сформированное число.

В дальнейшем описание операндов будет даваться только в том случае, когда количество или порядок операндов иной, чем в машинных командах.

Команды преобразования буквенно-цифровой информации.

Пересылка:

$$\langle N \rangle \lrcorner \langle \text{КОП} \rangle \lrcorner \left[\begin{array}{c} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{array} \right] \lrcorner \langle A \rangle ,$$

где $\langle \text{КОП} \rangle$ - ПСЕ - пересылка символа единичная,
 ПСЛ - пересылка слова.

Пересылка символов групповая:

$$\langle N \rangle \lrcorner \text{ПСГ} \lrcorner \left[\begin{array}{c} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{array} \right] \lrcorner \left[\begin{array}{c} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{array} \right] \lrcorner \langle A \rangle ;$$

первый операнд - число пересылаемых символов,
 второй - указывает, что пересылать,
 третий - указывает, куда пересылать.

Сравнение

$$\langle N \rangle \lrcorner \langle \text{КОП} \rangle \lrcorner \left[\begin{array}{c} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{array} \right] \lrcorner \left[\begin{array}{c} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{array} \right] ,$$

где $\langle \text{КОП} \rangle$ - ССЕ - сравнение символов единичное,
 ССЛ - сравнение слов.

Сравнение символов групповое:

$$\langle N \rangle \lrcorner \text{ССТ} \lrcorner \left[\begin{array}{c} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{array} \right] \lrcorner \left[\begin{array}{c} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{array} \right] \lrcorner \left[\begin{array}{c} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{array} \right] ;$$

первый операнд - длина сравниваемых операндов;
 второй операнд сравнивается с третьим.

Пересылка слова программы:

$$\langle N \rangle \lrcorner \text{ПСЛП} \lrcorner \langle \text{АГ} \rangle \lrcorner \langle A \rangle ,$$

Слово из первого операнда пересылается по адресу, задаваемому вторым операндом. Адрес в первом операнде математический или соответствующий математическому.

Формирование слова:

$$\langle N \rangle \sqcup \langle \text{КОП} \rangle \sqcup \begin{bmatrix} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{bmatrix} \sqcup \begin{bmatrix} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{bmatrix} \sqcup \langle A \rangle ,$$

$\langle \text{КОП} \rangle$ - ФСЛ1 - формирование слова 1,

ФЛ2 - формирование слова 2.

Первый операнд - длина формируемого слова,
второй и третий соответствуют первому и второму машинным
операндам.

Маркерный нуль:

$$\langle N \rangle \sqcup \text{МН} \sqcup \langle A \rangle .$$

Маркерный нуль групповой:

$$\langle N \rangle \sqcup \text{МНГ} \sqcup \begin{bmatrix} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{bmatrix} \sqcup \langle A \rangle .$$

Первый операнд - количество символов,
второй соответствует машинному операнду.

Маркерная единица:

$$1. \langle N \rangle \sqcup \text{МЕ} \sqcup \langle A \rangle ,$$

$$2. \langle N \rangle \sqcup \text{МЕ} \sqcup \begin{bmatrix} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{bmatrix} \sqcup \langle A \rangle .$$

Первый операнд задает номер того символа второго операнда,
в котором проставляется маркер.

Маркерная единица групповая:

$$\langle N \rangle \sqcup \text{МЕГ} \sqcup \begin{bmatrix} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{bmatrix} \sqcup \langle A \rangle .$$

Первый операнд - количество символов,
второй - соответствует машинному операнду.

Счет числа символов:

$$\langle N \rangle \sqcup \text{СС} \sqcup \begin{bmatrix} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{bmatrix} \begin{bmatrix} \langle A \rangle \\ \text{Н} \end{bmatrix} .$$

Команды управления. Команды условных переходов:

< КОП >	Условие перехода
УПР	[СМ] = 0
УПНР	[СМ] ≠ 0
УПМ	[СМ] < 0
УПМР	[СМ] ≤ 0
УПБ	[СМ] > 0
УПБР	[СМ] ≥ 0
УПН	По переполнению
УПНП	По непереполнению
УПНО	Состояние Тг команды УО равно единице
УПНЗ	По несовпадению знаков

Условный переход по маркерной единице:

$$\langle N \rangle \text{ — УПМЕ — } \left[\begin{array}{c} \langle A \rangle \\ \langle HЗ \rangle \end{array} \right] \text{ — } \langle AI \rangle .$$

Условный переход по маркерному нулю:

$$\langle N \rangle \text{ — УПМН — } \left[\begin{array}{c} \langle A \rangle \\ \langle HЗ \rangle \end{array} \right] \text{ — } \langle AI \rangle .$$

Безусловный переход:

$$\langle N \rangle \text{ — БП — } \langle AI \rangle .$$

Во всех названных командах управления AI должен соответствовать математический адрес.

Безусловный переход по адресу:

$$\langle N \rangle \text{ — БПА — } \left[\begin{array}{c} \langle A \rangle \\ \langle HЗ \rangle \end{array} \right] .$$

Нет операций:

$$\langle N \rangle \text{ — НОП .}$$

Индексация 1:

$\langle N \rangle \text{ — ИНД1 — } \begin{bmatrix} \langle A1 \rangle \\ \langle A2 \rangle \end{bmatrix} .$

Индексация 2:

$\langle N \rangle \text{ — ИНД2 — } \begin{bmatrix} \langle A1 \rangle \\ \langle A2 \rangle \end{bmatrix} \text{ — } \begin{bmatrix} \langle A \rangle \\ \langle H3 \rangle \end{bmatrix} .$

Конец цикла:

$\langle N \rangle \text{ — } \langle \text{КОП} \rangle \text{ — } \begin{bmatrix} A1 \\ A2 \end{bmatrix} \text{ — } \langle A1 \rangle ,$

где $\langle \text{КОП} \rangle$ — КЦ1 — конец цикла 1,
КЦ2 — конец цикла 2.

Команды режимов:

$\langle N \rangle \text{ — } \langle \text{КОП} \rangle ,$

где $\langle \text{КОП} \rangle$ — РПЗ — режим плавающей запятой,
РФЗ — режим фиксированной запятой,
РПП — режим прерывания по переполнению,
РППП — режим непрерывания по переполнению,
РПС — режим правого сумматора.

Обращение к диспетчеру (останов):

$\langle N \rangle \text{ — ОД} .$

Пересылка аргументов подпрограммы:

$\langle N \rangle \text{ — ПАП — } \begin{bmatrix} \langle A \rangle \\ \langle H3 \rangle \end{bmatrix} \text{ — } \dots \text{ — } \begin{bmatrix} \langle A \rangle \\ \langle H3 \rangle \end{bmatrix} .$

Эта команда засылает также количество операндов в счетчик.

Поднять регистр базы:

$\langle N \rangle \text{ — ПРБ — } \begin{bmatrix} \langle A \rangle \\ \langle H3 \rangle \end{bmatrix} .$

Опустить регистр базы:

$\langle N \rangle \text{ — ОРБ — } \begin{bmatrix} \langle A \rangle \\ \langle H3 \rangle \end{bmatrix} ,$

Безусловный переход на подпрограмму:
 $\langle N \rangle \rightarrow \text{БП} \rightarrow \langle AI \rangle$.

AI должен соответствовать математический адрес.

Безусловный переход на подпрограмму по адресу:

$$\langle N \rangle \rightarrow \text{БПА} \rightarrow \left[\begin{array}{l} \langle \text{НЗ} \rangle \\ \langle A \rangle \end{array} \right].$$

Выход из подпрограммы:

$$\langle N \rangle \rightarrow \text{ВП}.$$

Посылка адреса рабочего поля:

$$\langle N \rangle \rightarrow \text{ПАДР} \rightarrow \langle A \rangle \rightarrow \langle A \rangle.$$

Посылка адреса программы:

$$\langle N \rangle \rightarrow \text{ПАДП} \rightarrow \langle AI \rangle \rightarrow \left[\begin{array}{l} \langle A \rangle \\ \text{H} \end{array} \right].$$

$\langle AI \rangle$ должен соответствовать математический адрес.

Поднять регистр магазина:

$$\langle N \rangle \rightarrow \text{ПРМ} \rightarrow \left[\begin{array}{l} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{array} \right].$$

Опустить регистр магазина:

$$\langle N \rangle \rightarrow \text{ОРМ} \rightarrow \left[\begin{array}{l} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{array} \right].$$

Посылка в счетчик:

$$\langle N \rangle \rightarrow \text{С} \rightarrow \left[\begin{array}{l} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{array} \right].$$

Пуск ввода-вывода:

$$\langle N \rangle \rightarrow \text{УО}.$$

Команды обмена с системой прерывания:

1) $\langle N \rangle \rightarrow \langle \text{КОП} \rangle \rightarrow \left[\begin{array}{l} \langle A \rangle \\ \langle \text{НЗ} \rangle \end{array} \right];$

2) $\langle N \rangle \rightarrow \langle \text{КОП} \rangle \rightarrow \langle \text{НЗ} \rangle \rightarrow \langle \text{НЗ} \rangle,$

где <КОП> - ЗАЖ - зажечь СПР,
ГАС - гасить СПР,
МСК - маскировать СПР,
ДМСК - демаскировать СПР.

Вторая форма команды для СПР второго рода.

<N> ⌋ <КОП> ,

где <КОП> - ЕП - разрешить прерывание,
НП - запретить прерывание,
КНЦ - конец прерывающей программы.

РАБОТА С БАЗИСНЫМ АССЕМБЛЕРОМ

Работа ведется в режиме диалога: имеется набор слов-приказов; каждому приказу соответствует некоторая программа, выполняющая этот приказ. Приказы вводятся с телетайпа. Разрешение на ввод первого приказа дается после запуска ассемблера. После отработки приказа дается разрешение на ввод следующего приказа. Ниже следуют перечень приказов и описание программ их отработки.

- СГ - установка некоторого начального состояния. Этот приказ обязателен при переходе с одного сегмента на другой;
- КОНЕЦ - завершение работы в ассемблере;
- ВТТ - ввод БА-сегмента с телетайпа;
- ВПЛ - ввод БА-сегмента с перфоленты;
- ВМЛ ⌋ <название БА-сегмента> - ввод БА-сегмента с магнитной ленты;

Информация, введенная по любому из последних трех приказов в ходе работы с одним сегментом, накапливается; позже введенная информация рассматривается как продолжение ранее введенной;

- ЛИСТ - вывод текста БА-сегмента на АЦПУ;
- Т - трансляция БА-сегмента; на телетайп выводится название сегмента. В случае обнаружения ошибок на АЦПУ могут быть выведены следующие сообщения:

- МНОГОКРАТНО \lfloor <симв.адрес> - символический адрес описан в данном сегменте несколько раз;
- <N> \lfloor ФОРМАТ \lfloor - в строке с заданным номером количество или тип операндов не соответствуют коду операции;
- <N> \lfloor ОШИБКА \lfloor N \lfloor <номер операнда> - в строке с данным номером неправильно записан операнд с данным номером; ОШИБКА \lfloor N \lfloor 0 выводится, если использован несуществующий код операции;
- ПРА - перфорация БА-сегмента;
- ПРП - перфорация транслированного сегмента;
- ЗП \lfloor <назв.сегмента> - запись на магнитную ленту; можно записать как БА-сегмент, так и транслированный;
- ПКАТ - печать каталога магнитной ленты; на АЦПУ выводится таблица со следующими столбцами:
 - название массива,
 - объем массива,
 - номер маркера,
 - номер зоны;
- ПТР - вывод на АЦПУ таблицы символических адресов в рабочем поле;
- ПТП - вывод на АЦПУ таблицы со столбцами:
 - номер БА-строки,
 - соответствующий ему математический адрес;
- СОБР \lfloor <название сегмента> \lfloor - формирование рабочей программы. Из сегмента, название которого дано, берется адрес начала программы. Если в этом сегменте не было псевдокоманды Н, рабочая программа формируется с адреса 0.0. Все сегменты, входящие в данную программу, должны быть записаны на магнитной ленте. Собранный программу можно вывести на перфоленту приказом ПРП.

В заключение некоторые сведения о трансляторе. Транслятор состоит из блоков, которые хранятся на магнитной ленте. При запуске ассемблера особая организующая программа считывает блоки транслятора с МЛ и образует из них транслятор. Транслятор занимает около 5000 ячеек. Максимальная величина БА-сегмента - около 1300 строк.

Л И Т Е Р А Т У Р А

1. Управляющая система "Днепр-2" комплекса "Днепр-21". Математическое обеспечение, ч. I. Киев, 1970. 243 с.

РЕКУРРЕНТНЫЙ АЛГОРИТМ ЭКСПОНЕНЦИАЛЬНОГО СГЛАЖИВАНИЯ

Рассмотрим задачу сглаживания случайной функции $y(t)$ заданной в виде суммы детерминированной функции, и случайной погрешности $\varepsilon(t)$, где $\varepsilon(t) \sim N(0, \sigma)$, а t пробегает конечное множество значений $t = 0, n$:

$$y(t) = x(t) + \varepsilon(t) \quad (1)$$

Обычный путь решения такой задачи заключается в том, что вводится допущение $x(t) = F(a_0, a_1, \dots, a_m, t)$, где параметры a_k неизвестны. В этом случае задача сглаживания сводится к задаче оценки значений параметров a_k , которая может быть решена методом наименьших квадратов. Другими словами, для определения оценок \tilde{a}_k требуется решить систему нормальных уравнений, что является достаточно трудоемким процессом при больших m . При условии, что $F(a_0, a_1, \dots, a_m, t)$ есть полином, в настоящей статье для сглаживания $y(t)$ рассматривается простой алгоритм, основанный на рекуррентном соотношении. В результате применения этого алгоритма получается та же сглаженная функция $\tilde{y}(t)$, что и найденная по методу экспоненциально взвешенных наименьших квадратов.

Пусть $x(t)$ на интервале $[0, i]$, $i = \overline{1, n}$ описывается полиномом m -ой степени:

$$P(t) = \sum_{k=0}^m \frac{a_k}{k!} (t-i)^k \quad (2)$$

Значение $x(t)$ в точке $t=i$ будет равно $x(i) = P(i) = a_i^0$. В силу присутствия $\varepsilon(t)$ можно вычислить лишь некоторую оценку \hat{a}_i^0 . Тогда сглаженное значение $\tilde{y}(t)$ в точке $t=i$ определяется как

$$\tilde{y}(i) = \tilde{a}_i^0 \quad (3)$$

Оценка величины $\tilde{\alpha}_i^0$ вычисляется по методу экспоненциально взвешенных наименьших квадратов [1] на интервале $[0, i]$ т.е. из условия, что

$$\sum_{t=0}^i \alpha (1-\alpha)^t [y(t) - P(t)]^2 = \min \quad (4)$$

Назовем оператором экспоненциального сглаживания порядка l оператор $S^{(l)}$, определяемый соотношением

$$S_i^{(l)}(y) = \sum_{j=0}^i \alpha^l (1-\alpha)^j \frac{(l-1+j)!}{j! (l-1)!} y(t-j), \quad (5)$$

где $S_t^{(l)}(y) = S^{(l)}[y(t)]$. Фундаментальная теорема экспоненциального сглаживания [1] утверждает, что оценки коэффициентов $\tilde{\alpha}_k$ полинома (2) могут быть представлены в виде линейной комбинации величин, полученных в результате применения операторов $S^{(l)}$ $l = 1, \overline{m+1}$ к $y(i)$, причем

$$\tilde{\alpha}_i^0 = [E - (E-S)^{m+1}]_i(y), \quad E_i(y) = y(i), \quad (6)$$

Непосредственный способ получения этих коэффициентов дает теорема D'Esopo [2], согласно которой эти оценки $\tilde{\alpha}_i^k$ являются оптимальными в смысле минимума среднеквадратичной ошибки, т.е. минимизируют (4). Решение системы нормальных уравнений относительно $\tilde{\alpha}_i^k$ в явном виде для $m > 2$ нигде не приводится из-за своего громоздкого вида.

Рассмотрим рекуррентное соотношение

$$\tilde{y}_k = \tilde{y}_{k-1} + S(y - \tilde{y}_{k-1}) \quad (7)$$

Здесь y_k означает сглаженную функцию, полученную на k -ом шаге. Пусть далее $\tilde{y}_0 = 0$, тогда $\tilde{y}_1 = S(y)$. Покажем, что функция, получаемая на $m+1$ -ом шаге, согласно соотношению (7) совпадает со сглаженной функцией (3), полученной в результате скользящего сглаживания полиномом степени m , оценки коэффициентов которого вычислялись по методу экспоненциально взвешенных наименьших квадратов.

Перешлем (7) следующим образом:

$$\tilde{y}_k = (E - S) \cdot \tilde{y}_{k-1} + S(y) \quad (8)$$

Легко выразить \tilde{y}_n через исходную функцию :

$$\tilde{y}_n = [(E \cdot S)^{n-1} \cdot S + (E \cdot S)^{n-2} \cdot S + \dots + (E \cdot S)^0 \cdot S] \cdot y. \quad (9)$$

Справедливость формулы (9) проверяется по методу математической индукции. Выражение в квадратных скобках формулы (9) есть геометрическая прогрессия. Вычислив ее сумму, получим

$$\tilde{y}_n = [E - (E \cdot S)^n] y. \quad (10)$$

Сравнивая (10) с (3) и (6), можно заметить, что выражение для \tilde{y}_i^0 , где i произвольная точка, в которой вычисляется значение сглаженной функции, полностью совпадает с выражением (10) для вычисления сглаженной функции в той же точке.

Увеличение шагов сглаживания по формуле (7) равносильно повышению степени полинома (2), описывающего исходную функцию. Во избежание ошибок округления, зависящих от от числа слагаемых, сами непосредственные вычисления рекомендуется проводить по формуле (7), а не по (9).

В начале статьи было сделано предположение, что степень сглаживающего полинома известна, однако, при сглаживании экспериментально полученных зависимостей она все же редко бывает известной. В подобных случаях степень полинома задается приближенно на основании априорной информации или же исходя из внешнего вида экспериментальной зависимости.

Рассмотрим случай, когда степень полинома заранее неизвестна. Допустим, что "качество сглаживания" можно охарактеризовать суммой квадратов ошибок σ_n^2 :

$$\sigma_n^2 = \sum_{t=0}^n [y(t) - \tilde{y}_n(t)]^2 \quad (11)$$

Эта сумма будет уменьшаться с увеличением числа шагов сглаживания. Вычисляя на каждом шагу σ_n^2 , сглаживание осуществляем до тех пор, пока

$$[\sigma_n^2 - \sigma_{n-1}^2] < \xi, \xi > 0. \quad (12)$$

σ_n^2 можно выразить через σ_{n-1}^2 :

$$\sigma_n^2 = \sigma_{n-1}^2 - \sum_{t=0}^n S_t (y - \tilde{y}_{n-1}) [2(y(t) - \tilde{y}_{n-1}(t)) - S_t(y - \tilde{y}_{n-1})] \quad (13)$$

Разности $y(t) - \tilde{y}_{n-1}(t)$ и $S_t(y - \tilde{y}_{n-1})$ для всех точек $t = \overline{0, n}$ определяются уже при нахождении сглаженной функции. Таким образом, вычисление σ_n^2 можно вести в процессе сглаживания.

ЛИТЕРАТУРА

1. Brown R.G. Smoothing, Forecasting and Prediction of Discrete Time Series. Prentice-Hall, Englewood Cliffs, 1963. 468 p.
2. Д'Евора Р.А. - "Operations Research", 1963, 11, 3, p.686-687.
3. Кузьмина Л.М. Исследование статистической и динамической ошибки линейных методов сглаживания - Уч. зап. ЛГУ П. Стучки, т.144. Кибернетизация научного эксперимента, вып. 2. Рига, 1971, с.114-149.

АВТОМАТИЗАЦИЯ ТЕСТОВ УВМ

При управлении процессами, в частности физическим экспериментом, с помощью управляющих вычислительных машин (УВМ) предъявляются повышенные требования к достоверности информации, получаемой УВМ, перерабатываемой и выдаваемой в виде управляющих воздействий. Выход из строя УВМ может привести не только к неверным данным о протекании процессов, но также к приостановке его и даже к выводу из строя объекта управления. Следовательно, целесообразно проводить оперативную проверку основных узлов системы непосредственно перед пуском программы управления, предварительно сделав эту проверку доступной оператору, работающему с системой, без привлечения специального персонала, обслуживающего вычислительную машину. В статье будут рассмотрены вопросы проверки только УВМ, хотя принципы, изложенные ниже, применимы и для организации оперативной проверки узлов воздействия и сбора информации, относящихся к объекту управления.

Применение тестов, предусмотренных для профилактических работ инструкциями по эксплуатации УВМ, требует большого количества ручных операций при последовательном вводе тестов и при их выполнении; результаты проверки представляются в недостаточно удобной форме [1]. Автоматизируемая система тестов для оперативной проверки должна по возможности исключить ручные операции и представлять результаты проверки более наглядно.

Под автоматизацией проверки УВМ будем подразумевать организацию тестов в режиме диалога, где смену тестов и вывод указаний как на ручные операции, не поддающиеся

автоматизации, так и сообщений об обнаруженных сбоях производит специализированная программа, а оператор лишь выполняет указания этой программы.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

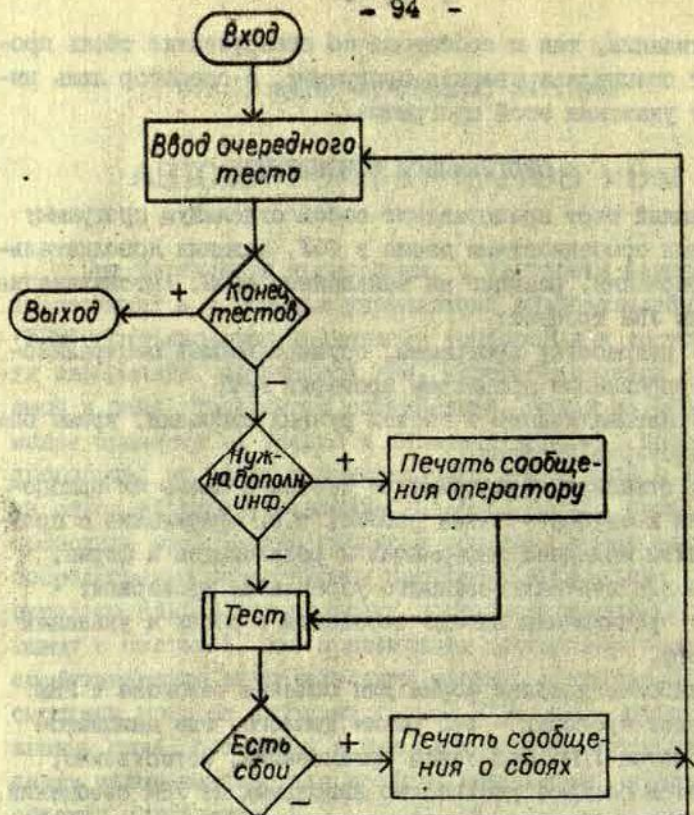
Каждый тест представляет собой отдельную программу со своими особенностями ввода в ОЗУ, задания дополнительной информации, реакции на выявленные сбои. Автоматизация проверки УВМ требует:

- 1) разработку программы, осуществляющей централизованное управление процессом проверки УВМ;
- 2) автоматизацию в тестах ручных операций, кроме операций проверки пультов;
- 3) стандартное оформление тестов с целью их приспособления к автоматической работе, т.е. оформление с представлением исходной информации и результатов в форме, удобной для централизованного управления проверкой;
- 4) организацию вывода сообщений о сбоях и указаний оператору.

Наиболее удобная форма для общения человека с УВМ в процессе проверки - это режим диалога, где инициатором является УВМ. При такой организации, естественно, появляется большое количество выводимых из УВМ сообщений, для формирования и печати которых целесообразно ввести некоторую специализированную программу.

Общая структура программы, осуществляющей централизованное управление процессом проверки УВМ, определена возложенными на нее функциями (рис. 1):

- ввод очередного теста в ОЗУ и контроль за качеством ввода;
- передача управления очередному тесту и указания о количестве циклов работы теста;
- анализ причин обращения теста к программе централизованного управления и подготовка информации для подпрограммы вывода.



Р и с. 1. Организация программного обеспечения системы автоматизированных тестов.

АВТОМАТИЗИРОВАННАЯ ПРОВЕРКА УСТРОЙСТВА СВЯЗИ С ОБЪЕКТОМ

Целью оперативной проверки устройства связи с объектом (УСО), как уже упоминалось, является проверка общих узлов УСО, а не каждого задействованного канала в отдельности, проверку которых целесообразно отнести к проверке объекта. Известный опыт автоматизации проверки УВМ накоплен при разработке системы управления экспериментом с применением УВМ "Днепр-1", имеющей блок УСО (входные каналы для опроса двухпозиционных датчиков и датчиков непрерывно-

го сигнала), блок Р (релейные выходы), блок А (аналоговые выходы). Для такой комплектности в кросс-шкафу [2] необходимо произвести некоторые соединения с применением соответствующих устройств:

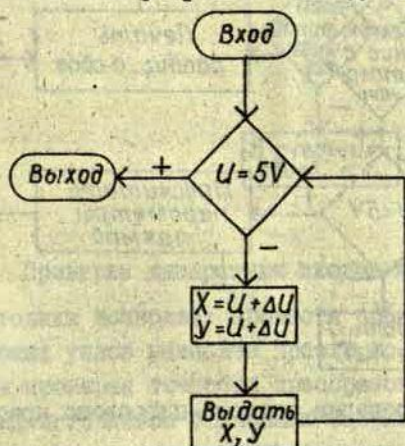
- подключить к двум выходам блока А устройства графического представления данных (в конкретном случае был применен двухкоординатный самопишущий потенциометр);

- подключить один выход блока А к аналоговому входу УСО;

- подать на один из входов опроса двухпозиционных датчиков УСО сигнал, принимающий значение 0V и -12 V в зависимости от состояния одного из реле блока Р .

Проверка блока А . На цифро-аналоговые преобразователи, к которым подключено устройство графического представления данных, выдаются коды, соответствующие линейно растущему напряжению на выходах блока А от нуля до максимального значения. Качество прямой $Y=X$ оператор проверяет визуально (рис.2).

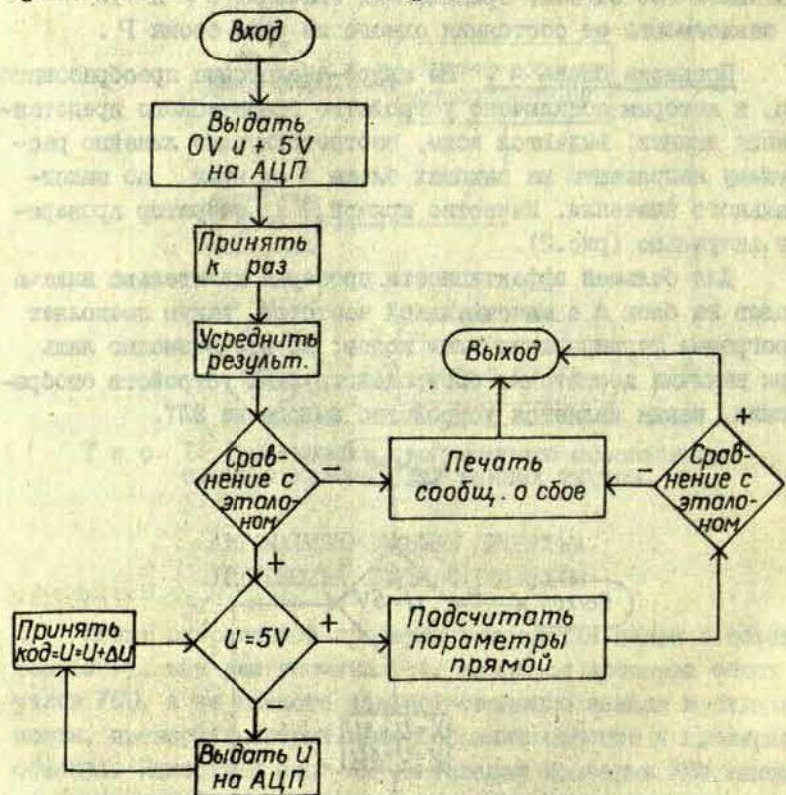
Для большей эффективности проверки желательна выдача кодов на блок А с максимальной частотой, какую позволяет программа формирования этих кодов; но это возможно лишь при наличии достаточно быстродействующих устройств отображения, каким является устройство вывода на ЭЛТ.



Р и с . 2. Проверка блока аналоговых выходов.

ПРОВЕРКА АНАЛОГО-ЦИФРОВОГО ПРЕОБРАЗОВАТЕЛЯ УСО

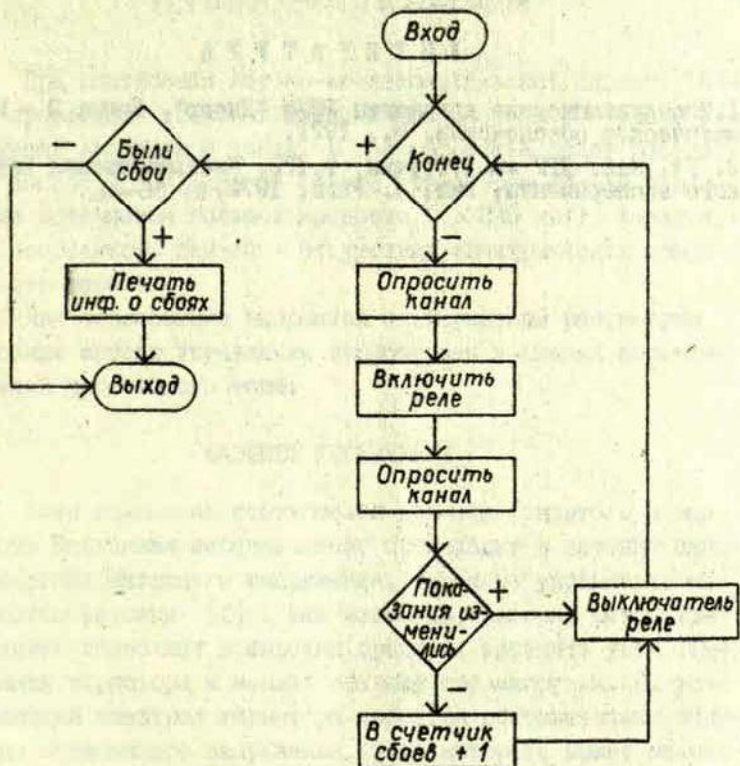
Проверка производится подачей от блока А на вход аналого-цифрового преобразователя (АЦП) предельных значений напряжения ($0V$ и $+5V$) и сравнением результатов преобразования с эталонными кодами. Далее проверяется линейность преобразования по всему рабочему диапазону на прямой $Y=X$. При этом проверяется как форма кривой, получаемой в результате преобразований (в качестве параметра проверки используется ее длина), так и отклонения от исходной прямой, выдаваемой в блок А (рис.3).



Р и с. 3. Проверка аналого-цифрового преобразователя.

ПРОВЕРКА ДИСКРЕТНЫХ КАНАЛОВ

Заданное количество раз производится переключение реле, управляющего сигналом, подключенным ко входу опроса двухпозиционных датчиков УСО, с последующим опросом этого входа. Появление сбоев и их число фиксируется (рис. 4).



Р и с. 4. Проверка дискретных входов и выходов.

Такая методика позволяет провести проверку УСО на функционирование общих узлов наиболее просто, но с пониженными требованиями к проверке точности преобразований. Применение соответствующих источников эталонных напряжений позволяет решать и этот вопрос достаточно достоверно.

Предложенный метод приемлем только как мера обнаружения внезапно появившихся неисправностей с целью предотвращения аварий или ошибок в информации во время работы и может быть эффективен только при систематическом проведении профилактических работ с обязательным устранением всех, даже самых мелких, замеченных отклонений от нормы.

Л И Т Е Р А Т У Р А

1. Эксплуатационные документы ЭВМ "Днепр". Книга 3 - Математическое обеспечение. М., 1971.
2. Уч. зап. ДГУ им. П. Стучки, т. 170. Кибернетизация научного эксперимента, вып. 4. Рига, 1972, с. 66-94.

СИСТЕМНЫЙ КАНАЛ УПРАВЛЕНИЯ МОЩНЫМ ТЕПЛОВЫМ ПОТОКОМ

При построении научно-исследовательской системы (НИС) для проведения тепловых исследований возникает задача управления источником тепла. В данной статье будет рассмотрен вопрос о создании на основе тиристоров канала управления источником большой мощности (до 320 квт). Характерная особенность канала - отсутствие электрических помех при его работе.

Для обоснования выбранной альтернативы рассмотрим основные методы управления тиристорами в схемах переключателей переменного тока.

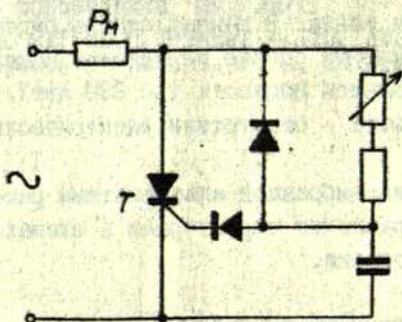
ФАЗОВОЕ УПРАВЛЕНИЕ

Если изменение соотношения времени открытого и закрытого состояния переключателя происходит в течение одного периода питающего напряжения, то такое управление называется фазовым [1]. Как известно, фазовый метод управления позволяет в широких пределах изменять угол открывания тиристора и менять средний ток нагрузки. На управляющий электрод тиристора подается положительная полуволна переменного напряжения, фаза которого может меняться по отношению к фазе питающего анодного напряжения (рис.1). При этом изменяется момент открывания тиристора. Начиная с этого момента до конца положительной полуволны анодного напряжения тиристор находится в открытом состоянии. При совпадении фаз напряжений на управляющем электроде и на аноде время протекания тока через тиристор приблизительно равно $T/2$, где T - период питающего переменного

го напряжения. При сдвиге фазы напряжения на управляющем электроде относительно анодного на угол α происходит задержка момента отпирания тиристора. В однофазном переключателе угол отпирания α может меняться от 0 до 180° , и среднее значение тока через нагрузку будет

$$J_\varphi = \frac{J_m}{\pi} \int_\alpha^\pi \sin \omega t dt = \frac{J_m}{\pi} (1 + \cos \alpha) = J_0 \frac{1 + \cos \alpha}{2},$$

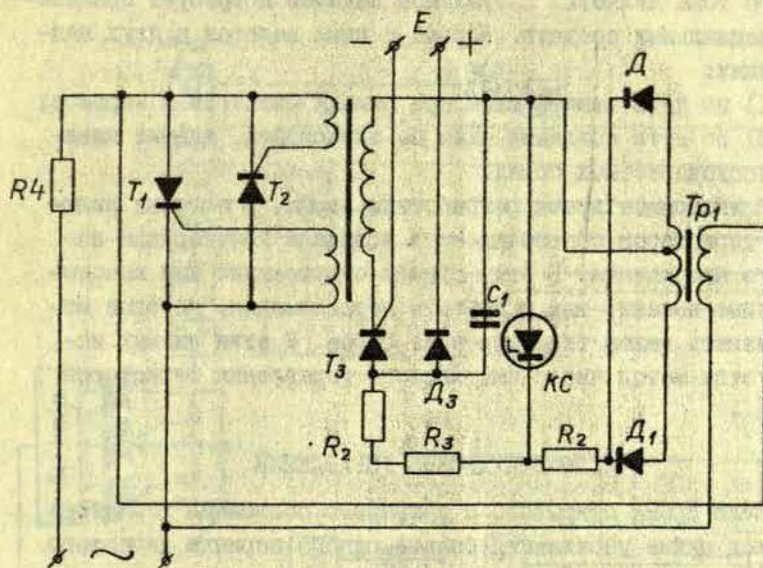
где J_m - амплитуда тока нагрузки, J_0 - среднее значение тока, протекающего через нагрузку при $\alpha = 0$.



Р и с. 1. Схема фазового управления тиристором.

При амплитудном методе управления в качестве управляющего напряжения может быть использовано напряжение анода тиристора. Через регулируемое сопротивление управляющий электрод подключается к аноду тиристора. Меняя величину этого сопротивления, можно изменять угол включения тиристора от 0 до 90° [2]. Однако управляющее напряжение тиристора при амплитудном методе - величина непостоянная, поэтому и угол включения тиристора будет нестабильным. Отсюда ясно, почему метод фазового управления применяется редко.

Импульсно-фазовое управление обеспечивает большую точность включения тиристов. При этом методе на управляющий электрод подается импульс с крутым фронтом. Меняя время появления этого импульса по отношению к фазе анодного напряжения, получается необходимый угол отпирания (рис.2).



Р и с. 2. Схема импульсно-фазового управления тиристором.

Как видно из вышеизложенного, фазовый метод управления тиристорами сравнительно прост. Однако он имеет существенный недостаток, ограничивающий его применение при больших токах нагрузки: большие скачки тока в момент включения тиристора, при которых из-за наличия в сети распределенных индуктивностей и емкостей возникает большой уровень помех [1]; это оказывает влияние на различные чувствительные датчики, а также на цепи управления и питания других тиристоров. Наблюдаются два вида помех:

- а) напряжение помех, которое проявляется в виде паразитной составляющей напряжения на проводах питающей сети;
- б) поле помех, которое определяется уровнем поля радиопомех на заданном расстоянии от источника (например, тиристора, управляемого фазовым методом). Предельно допустимые нормы помех даны в [3], методика их измерения приведена в [1].

Уменьшение помех при работе тиристоров в цепях пере-

менного тока является актуальной задачей и требует применения специальных средств. Борьба с ними ведется в двух направлениях:

- 1) по пути защиты схем при помощи фильтров и экранов;
- 2) по пути создания схем на тиристорах, дающих минимум высокочастотных помех.

В настоящее время разработаны схемы, в которых включение тиристоров обеспечивается в начале полупериода питающего напряжения. В этих схемах отсутствуют как высокочастотные помехи, так и эффекты переключения, которые могут вызвать выход тиристоров из строя. В этих схемах используется метод число-импульсного управления тиристором [1, 4].

ЧИСЛО-ИМПУЛЬСНОЕ УПРАВЛЕНИЕ

Если время открытого и закрытого состояния тиристора в период цикла управления больше одного периода питающего напряжения и кратно ему, то имеет место число-импульсное управление. На рис.3 изображена схема многопозиционного тиристорного переключателя переменного тока, в которой соотношение времени открытого и закрытого состояний тиристора задается дискретно. Шаг дискретности равен или кратен периоду питающего напряжения. Временные интервалы открыто-

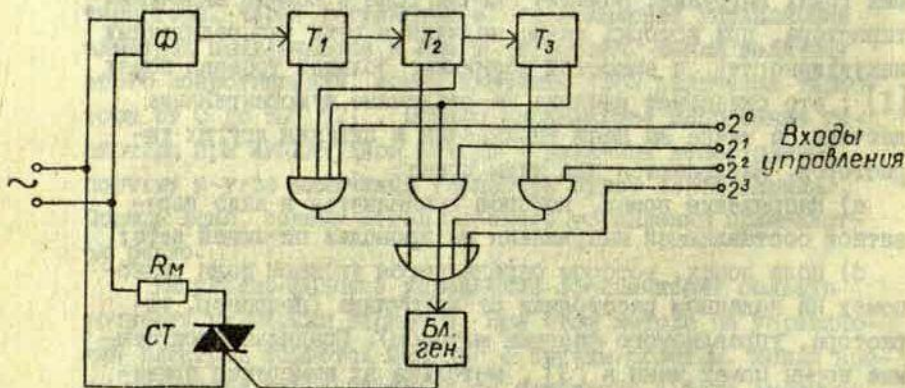
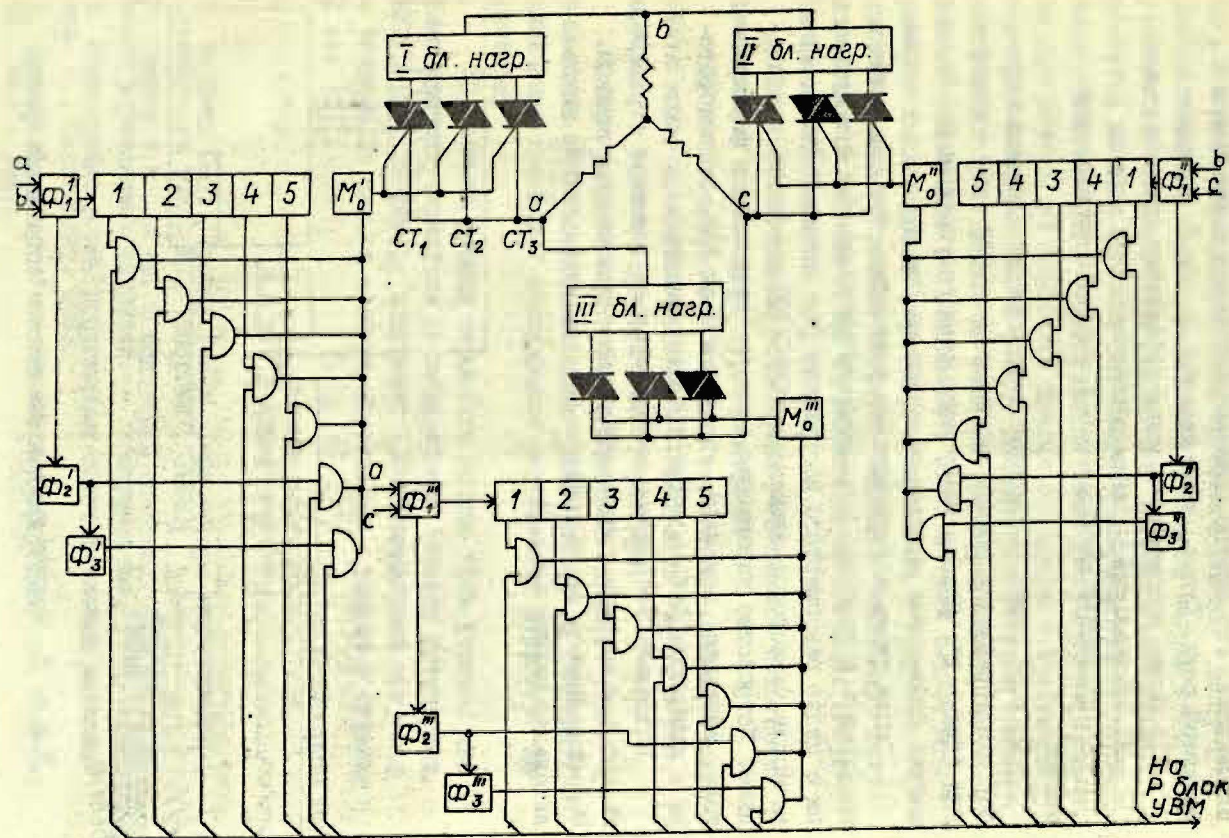


Рис. 3. Восьмипозиционный переключатель переменного тока.



Р и с. 4. Шестидесятичетырехпозиционный переключатель переменного тока.

го и закрытого состояний тиристора задаются логическими элементами. Устройство работает синхронно с напряжением питающей сети. Схема, изображенная на рис.3, обладает:

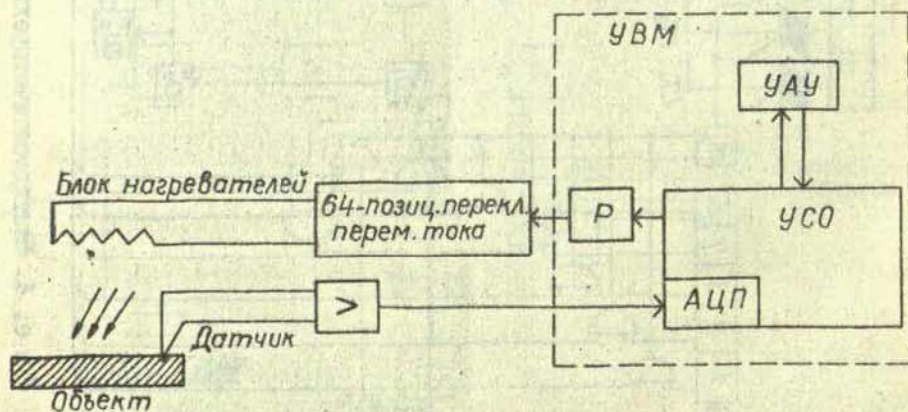
- 1) более высокой точностью управления средним током;
- 2) минимальным уровнем высокочастотных помех;
- 3) простотой согласования с выходами управляющей машины.

машины.

По этой схеме разработан и изготовлен системный прибор для управления мощным нагревателем (320 квт) с помощью УВМ "Днепр-1". Прибор имеет вход ручного управления. Упрощенная блок-схема прибора показана на рис.4.

Общее число нагревательных элементов, которые можно подключить к прибору, 63. Элементы разделены на три группы с целью равномерного нагружения фаз питающей линии. В качестве тиристоров применены ВКЛУС-100 с водяным охлаждением. Устройство управления состоит из входного формирователя, счетчика, вентилях, управляемых от УВМ, моновибратора и выходного развязывающего трансформатора.

Описанное устройство выполняет роль элемента обработки кодов в контуре управления мощным тепловым источником. Основным звеном этого контура является управляющая вычислительная машина (рис.5).



Р и с. 5. Контур управления мощным источником тепла.

Входы устройства подключены к выходам реле, находящимся в блоке Р управляющей вычислительной машины "Днепр-1". Датчиком служит платинородиевая термopара, сигнал которой усиливается операционным усилителем IУТ53I. Далее сигнал подается на вход аналого-цифрового преобразователя, находящегося в УСО УВМ. Регулирование температуры осуществляется программно, на основании данных о текущем и требуемом значении температуры, а также предыдущем и текущем состоянии регулятора. Данные по температуре усредняются, что повышает надежность регулирования; этой же цели служит программное согласование периода регулирования с градиентом кривой заданного режима нагрева. Диапазон температур регулирования от +20 до +1500°C, максимальная скорость нагрева около 20°C/сек.

При проектировании устройства учитывались некоторые особенности тиристорных переключателей:

1) из-за дискретной скорости нарастания анодного напряжения может произойти превышение параметра $\frac{dU_a}{dt}$, а это приводит к ложному срабатыванию тиристора;

2) при больших скоростях нарастания анодного тока, превышающих $\frac{dI_a}{dt_{\text{ток}}}$, материал может разогреться до температуры плавления кремния с последующим разрушением структуры тиристора;

3) режимом работы тиристора должно быть учтено, что симметричные тиристоры не выдерживают больших перегрузок, ВКЛУС-100, например, допускает 1,25 номинальной нагрузки в течение 30 сек, а двойную перегрузку - только в течение 1 сек [1].

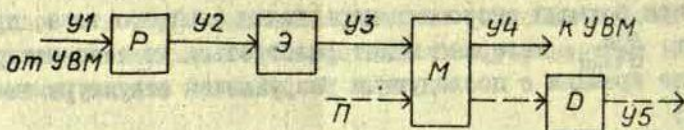
ЛИТЕРАТУРА

1. Бруфман С.С., Трофимов Н.А. Тиристорные переключатели переменного тока. М., "Энергия", 1969.
2. Грехов И.В. и др. О некоторых способах применения силовых управляемых вентилях. - Электричество, 1965, 2.
3. Общесоюзные нормы допускаемых промышленных радиопомех. М., Связьиздат, 1963.
4. Джентри Ф. и др. Управляемые полупроводниковые вентиля. М., "Мир", 1967.

ДИСКРЕТНЫЕ КАНАЛЫ НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ СИСТЕМЫ ДЛЯ УПРАВЛЕНИЯ ПОДАЧЕЙ ЖИДКОСТИ И ГАЗОВ

При построении научно-исследовательских систем (НИС), в контуре которых находится управляющая вычислительная машина (УВМ), часто возникает задача регулировать подачу различных потоков (свет, жидкость, газ и т.п.). Это означает, что элементная база НИС должна содержать соответствующие программно-управляемые приборы, которые должны удовлетворять таким основным факторам, как быстрота срабатывания, надежность, отсутствие электрических помех и простота управления.

Всякий подобный прибор построен по следующей схеме.

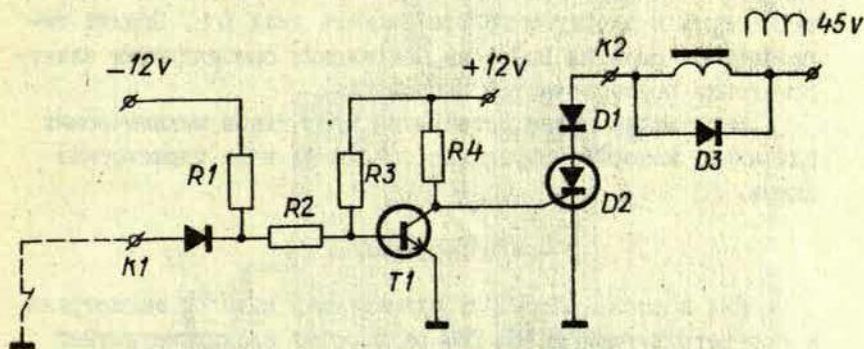


Р и с. I. Схема включения механизма М в цепь УВМ.

Управление потоком Π (газ или жидкость) осуществляется путем подачи от УВМ сигнала "У1" (Включить-выключить); по этой команде замыкается реле в блоке Р, что служит сигналом "У2" на входе электронного элемента Э, который формирует ток (сигнал "У3") величины, достаточной для приведения в действие электромагнита регулирующего клапана М; последний прерывает (либо возобновляет) поток Π . О срабатывании механизма М УВМ судит по величине контрольного

сигнала "М". В случае, если нет возможности идентифицировать состояние механизма М непосредственно, то ставится датчик D потока, по сигналу которого ("У5") можно контролировать действие механизма М.

Для управления электромагнитными клапанами от УВМ необходимы устройства, способные коммутировать сильные токи, поскольку релейные контакты выхода УВМ для прямого коммутирования больших токов непригодны; для этой цели были использованы тиристорные ключи.



Р и с. 2. Принципиальная схема тиристорного ключа.

Тиристорный ключ (рис.2) состоит из формирователя пускового потенциала для тиристора (транзистор T1) и самого тиристора (D2). Схема работает следующим образом. Для управления работой тиристорных ключей используются нормально замкнутые (на землю) контакты входных реле УВМ. При необходимости включения определенного электромагнитного клапана должно сработать соответствующее реле блока Р управляющей вычислительной машины, а его контакты - разомкнуться. В этом случае на входе тиристорного ключа (контакт K1) пропадет нулевой потенциал. Сопротивления R1, R2, R3 подобраны таким образом, чтобы в этом случае транзистор T1 был закрыт. Тогда на управляющий электрод тиристора D2 будет подан положительный потенциал. При этом тиристор D2 откроется и сработает электромагнитный клапан, который пи-

тается положительными полупериодами синусоидального тока с амплитудой 45 в.

При необходимости закрыть электромагнитный клапан нужно выключить соответствующее реле выхода УВМ (замкнуть контакты на землю). Тогда транзистор Т1 откроется до насыщения и положительный потенциал на управляющем электроде тиристора D2 станет небольшим, но сам тиристор останется в включенном состоянии до того момента, пока импульс питающего напряжения упадет до нуля. Чтобы скомпенсировать эффект индуктивности, приводящий к сглаживанию импульса, между тиристором и электромагнитом включен диод D1. Защита тиристора от больших выбросов напряжения самоиндукции электромагнита осуществляется диодом D3

Рассмотрим далее устройство трех типов механических клапанов, которыми управляют описанные выше тиристорные ключи.

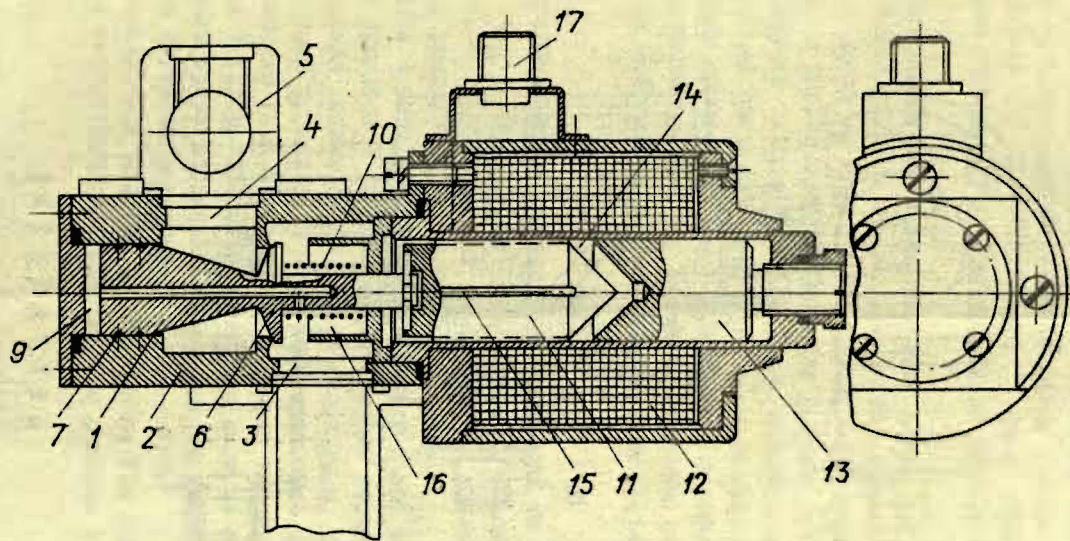
ВОДЯНОЙ КЛАПАН

Для подачи, например, охлаждающей воды от магистрали к соответствующим узлам НИС разработан электромагнитный клапан с проходным диаметром 25 мм.

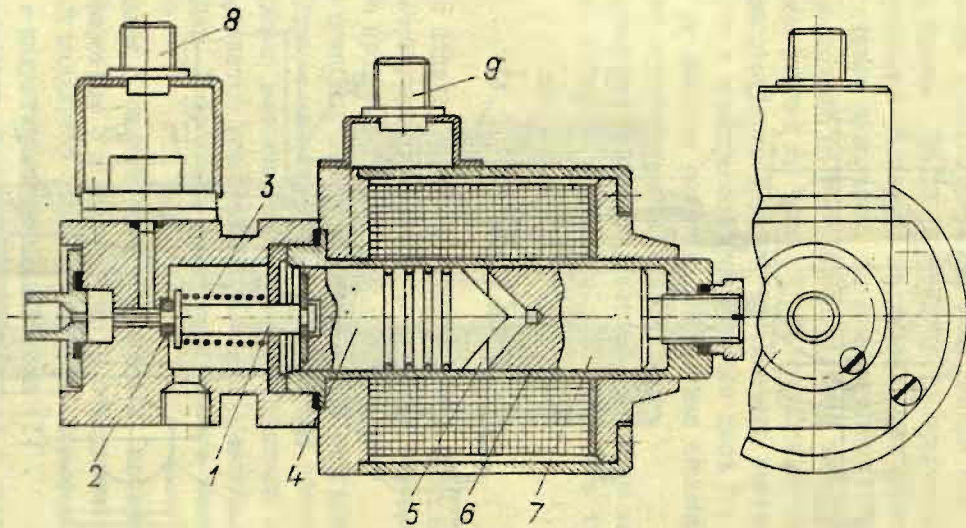
По принципу действия этот клапан разгруженного типа, с устройством мягкого переключения, чем обеспечивается срабатывание клапана независимо от перепада давления и предотвращается возникновение гидравлических ударов в трубопроводах. Устройство электромагнитного клапана показано на рис.3.

Клапан I смонтирован в корпусе 2, который подсоединяется отверстием 3 к магистрали; к отверстию 4 через промежуточный коллектор 5 в свою очередь присоединяются потребители и индуктивные датчики потока [I].

Перекрытие прохода от отверстия 3 к отверстию 4 осуществляется тарелкой 6 клапана I. Аксиальные силы, возникающие от разности давления рабочей среды по обеим сторонам тарелки 6, компенсируются поршнем 7, с этой целью рабочая среда от магистрали подается по калиброванному отверстию 8 в полость 9. Уплотнение клапана I осуществляется пруж-



Р и с. 3. Водяной клапан с электрическим приводом.



Р и с. 4. Газовый клапан.

жиной 10, открытке - якорем 11 электромагнита 12. Ход клапана 1 регулируется сердечником 13. Предотвращение гидравлического удара, возникающего при открывании клапана, осуществляется дросселированием рабочей среды (якорь 11), заполняющей полость 14. Для этого якорь 11 имеет шлицы 15, сечение которых выбирается в зависимости от вязкости рабочей среды.

Предотвращением гидравлического удара при закрывании клапана осуществляется дросселированием выхода (отверстие 8) рабочей среды из полости 9; диаметр отверстия 8 выбирается также в зависимости от вязкости рабочей среды.

Разгрузка клапана от аксиальных динамических усилий потока рабочей среды достигнута выбором соответствующей конфигурации самого клапана и уплотняемой кромки корпуса, а также наличием втулки 16. Для подсоединения кабеля управления от УВМ клапан снабжен штепсельным разъемом 17.

ГАЗОВЫЙ КЛАПАН

Электромагнитный клапан для управления от УВМ подачи газа (рис. 4) может быть использован до давлений порядка 100 кг/см². Электромагнит этого клапана аналогичен электромагниту вышеописанного водяного клапана.

Клапан 1 уплотняет проход (диаметр 6 мм) и снабжен резиновой прокладкой 2; уплотняющее усилие создает пружина 3. Для гашения механических ударов необходимо, чтобы якорь 4 был подогнан по втулке 5 электромагнита. Это обеспечит сжатие рабочего газа в полости 6 и мягкое касание якорем 4 сердечника 7 (якорь дросселирует выход газа из полости 6). Для контроля срабатывания электромагнитный клапан снабжен мембранным датчиком давления.

Присоединение кабелей управления от УВМ осуществляется через соответствующие штепсельные разъемы 8 и 9.

ЛИТЕРАТУРА

Г. Уч. зап. ЛГУ им. П. Стучки, т. 170. Кибернетизация научного эксперимента, вып. 4. Рига, 1972. 259 с.

СО Д Е Р Ж А Н И Е

1.	КУЗЬМИН Ю.Я. Об одной альтернативе программирования эксперимента	3
2.	ГВОЗДЕВ С.В. К реализации группового алгоритма сбора и контроля измерительной информации	17
3.	КУЗЬМИН Ю.Я., ЦИРУЛИС Я.П. Способ формирования закона контроля в автоматизированной системе	27
4.	КУЗЬМИН Ю.Я., ГВОЗДЕВ С.В., БАНГА А.Я., КОТМИН Е.А. Особенности режима диалога в автоматизированном эксперименте	32
5.	КУЗЬМИН Ю.Я. Реализация фраз в режиме диалога	50
6.	КУЗЬМИН Ю.Я. Групповой алгоритм управления каналами научно-исследовательской системы	55
7.	БЕРНУЦ-БЕРНХОФ А.А., КРАЗЕ-КРУЗЕ Х.Р., ГУТАНС Я.Я. Моделирование процессов обработки информации вычислительно-управляющего комплекса	60
8.	КРАЗЕ-КРУЗЕ Х.Р. Базисный ассемблер ЭЦВМ "Днепр-21"	70
9.	КУЗЬМИНА Л.М. Рекуррентный алгоритм экспоненциального сглаживания	88
10.	ГВОЗДЕВ С.В., БАНГА А.Я., ПУЦЕ Л.Р. Автоматизация тестов УВМ	92
11.	ЗАРИНЫШ М.Я. Системный канал управления мощным тепловым потоком	99
12.	СТРАУМЕН Я.Я., ПАСПАРНЕ А.К. Дискретные каналы научно-исследовательской системы для управления подачей жидкости и газов	106

Ученые записки, том 196

КИБЕРНЕТИЗАЦИЯ НАУЧНОГО ЭКСПЕРИМЕНТА

Выпуск 5

Редактор Э.Тарденак
Технический редактор М.Штерн
Корректор В.Полмане

Редакционно-издательский отдел ЛГУ им. Петра Стучки
Рига 1973

Подписано к печати 22.08.1973 ЯТ 06319 Зак. № 812.
Ф/б 60x84/16. Бумага №1. Физ.п.л. 7,3. Уч.-и.л. 5,2
Тираж 500 экз. Цена 52 к.
Отпечатано на ротационте, Рига-50, ул.Вейденбаума,5
Латвийский государственный университет им. П.Стучки

LU bibliotēka



200023728

PT-75

196

VU
2

Цена 52 коп.

Учен. зап. (ЛГУ им. П.Стучки), 1973 г., т. 196, 1-112