

ИНСТИТУТ ФИЗИКИ ВЫСОКИХ ЭНЕРГИЙ

И Ф В Э 81-184
ОЭА

Ю.А.Белокопытов, В.М.Дикий, Л.Г.Каминский, С.В.Клименко,
В.Н.Кочин, Ю.Л.Куркин, А.И.Курносенко, В.Д.Лонгинов,
Г.Д.Некипелова, А.П.Соколов, В.Д.Юрпалов

АВТОМАТИЗИРОВАННАЯ СИСТЕМА
ОБРАБОТКИ СНИМКОВ RTFAS.
УПРАВЛЯЮЩАЯ ПРОГРАММА АБСТРАКТНОГО ПРОЕКТОРА

Ю.А.Белокопытов, В.М.Дикий, Л.Г.Каминский, С.В.Клименко,
В.Н.Кочин, Ю.Л.Куркин, А.И.Курносенко, В.Д.Лонгинов,
Г.Д.Некипелова, А.П.Соколов, В.Д.Юрпалов

АВТОМАТИЗИРОВАННАЯ СИСТЕМА
ОБРАБОТКИ СНИМКОВ RTFAS.
УПРАВЛЯЮЩАЯ ПРОГРАММА АБСТРАКТНОГО ПРОЕКТОРА

Аннотация

Белокопытов Ю.А., Дикий В.М., Каминский Л.Г., Клименко С.В., Кочин В.Н., Куркин Ю.Л., Курносенко А.И., Лонгинов В.Д., Некипелова Г.Д., Соколов А.П., Юрпалов В.Д.

Автоматизированная система обработки снимков RTFAS. Управляющая программа абстрактного проектора. Серпухов, 1982.

20 стр. с рис. (ИФВЭ ОЭА 81-184).

Библиогр. 14.

В работе предложена математическая модель управляющей программы мини-ЭВМ многомашинного комплекса, ориентированного на первичную обработку снимков с трековых детекторов. Выбор модели основывается на использовании понятия логических устройств ввода и введенном понятии виртуального устройства развертки графических объектов и обработки прерываний, что дает возможность представить модель в виде системы конечных автоматов.

Abstract

Belokopytov Yu.A., Dikiy V.M., Kaminski L.G., Klimentko S.V., Kochin V.N., Kurkin Yu.L., Kurnosenko A.I., Longinov V.D., Nekipelova G.D., Sokolov A.P., Yurpalov V.D.

Real-Time Film Analysis System. Minicomputer Steering Program. Serpukhov, 1982.

p. 20. (ИФВЭ 81-184).

Refs. 14.

A mathematical model for the steering programme of the minicomputer in the many-computer complex oriented for primary analysis of track detector films is offered. The choice for this model is based on application of the notion of logic input device as well as in the introduced concept of virtual processing devices of typical graphic objects and interrupts. This enables to represent the model as a system of finite automata.

1. ВВЕДЕНИЕ

В работе ^{/1/} были рассмотрены основные принципы построения программного обеспечения технологического уровня автоматизированной системы RTFAS ^{/2/}. Проведенная формализация процедур первичной обработки снимков, основанная на общем для них методе проверки полноты и качества поступающих измерительных данных, позволила создать единый программный процессор, обеспечивающий взаимодействие между оператором и информационной компонентой в типовых процедурах первичной обработки.

Программный процессор распределен между ЭВМ, входящими в систему. Одна его часть – организующая программа (ОРП) – действует на ЦЭВМ комплекса, а другая – сканер – составляет основу управляющей программы мини-ЭВМ проекторов системы. Для создания открытой для физика-экспериментатора системы управляющая программа должна быть инвариантной к процедурам первичной обработки, а функциональная полнота должна обеспечиваться параметрами примитивов обмена.

2. ТРЕБОВАНИЯ К УПРАВЛЯЮЩЕЙ ПРОГРАММЕ

Учитывая принятый нами общий подход к реализации различных процедур первичной обработки снимков ^{/1/}, будем далее называть измерением любую процедуру обработки. Рассмотрим характерные черты процесса измерения объектов с точки зрения требований к управляющей программе мини-ЭВМ.

Решение задачи рационального использования вычислительных ресурсов системы ЦЭВМ – мини-ЭВМ протекторов заключается в таком разделении функций контроля программного процессора по вычислительным машинам, при котором достигается минимальная частота обмена между ЭВМ. Прагматика такого разделения обсуждалась в работе ^{1/} и фактически привела к переносу на мини-ЭВМ тех операций контроля при непосредственном взаимодействии с оператором и аппаратурой, которые не связаны с использованием информационной компоненты системы. В соответствии с этим основная часть семантического контроля выполняется в ОРП на ЦЭВМ, а лексико-синтаксического – на мини-ЭВМ.

Однако задача повышения эффективности процесса измерений делает целесообразным реализацию в управляющей программе простейших семантических проверок. Для таких проверок может использоваться информация, полученная как в текущем процессе измерения (координаты точек на треке для расчета координат точки сегментации в режиме измерения маски для автоматов), так и в процессе предварительной "off-line" обработки данных (например, данных с электронных детекторов гибридных установок, перед процедурами сверки результатов двух независимых просмотров, перед физическим просмотром и т.п.), т.е. ввод информации от оператора должен быть организован в режиме уточнения, частным случаем которого является свободный ввод. С другой стороны, в системе должна быть предусмотрена возможность непосредственного взаимодействия программ ЦЭВМ с оператором в экзотических ситуациях, предполагающая выполнение лексического анализа на ЦЭВМ.

Важным требованием, предъявляемым к организации процесса измерения, является необходимость представления оператору средств изменения заданной последовательности обработки (отказ от выполнения запрашиваемых действий, отказ от некоей последовательности уже совершенных действий и т.п.). В связи с этим необходимо ввести прерывание процесса получения данных по инициативе оператора, приводящее как и получение всех запрошенных данных к завершению обработки командного примитива в мини-ЭВМ.

К прерываниям процесса измерений следует отнести и действия оператора по управлению устройствами проектора. Однако они не должны завершать обработку командного примитива, так как не требуют обращения к ОРП.

Кроме устройств ввода информации в аппаратуру проектора включены устройства отображения. Обеспечение сквозной передачи данных из ЦЭВМ на устройства отображения и организация запросов данных оператором непосредственно от программ ЦЭВМ также входит в задачу управляющей программы мини-ЭВМ.

При разработке управляющей программы необходимо учитывать специфику работы программы в реальном времени и эксплуатационные требования – гибкость, простоту в отладке и модернизации – вытекающие из необходимости обработки в системе постоянно меняющихся экспериментов на развивающемся парке проекторов. Существенным дополнительным требованием является обеспечение возможности автономной работы проекторов – без ЦЭВМ.

Определение структуры управляющей программы, удовлетворяющей всем перечисленным требованиям, возможно только при условии создания адекватной математической модели реализуемых ею задач.

Следует отметить, что задачи, решаемые управляющей программой мини-ЭВМ многомашинного комплекса, ориентированного на первичную обработку снимков, созвучны задачам мини-ЭВМ в других приложениях (например, в задачах организации ввода в системах интерактивной машинной графики). В связи с этим представляется целесообразным определить некий регулярный способ построения параметров примитивов обмена между ОРП и управляющей программой мини-ЭВМ.

3. ВЫБОР ПАРАМЕТРОВ ПРИМИТИВОВ ОБМЕНА

При организации взаимодействия ОРП с управляющей программой, используемой для решения некоторого класса задач, необходимо устранить или, по крайней мере, формализовать воздействие "физики" графических объектов,

с которыми оперируют программы обработки на ЦЭВМ, на правила контроля входной информации и на режим работы устройств ввода.

Полностью абстрагироваться от объектов измерения можно только на уровне лексического преобразования. С таких позиций решается задача организации взаимодействия в универсальных системах графического ввода, в которых используется понятие виртуального логического устройства^{13/}. Введение виртуальных логических устройств обеспечивает независимость формирования лексем от реально используемой аппаратуры ввода. Взаимодействие в таких системах основано на использовании команд явного задания режима работы логических устройств (разрешить, запретить, установить и т.п.) и последовательной передачи информации с каждого из разрешенных устройств по инициативе программ обработки. Последовательность ввода лексем, которая и реализует развертку объекта, полностью определяется в программах обработки.

Учитывая специфику систем специального назначения, в них целесообразно введение логических устройств, формирующих запрашиваемые и контролируемые характеристики объектов (в задачах первичной обработки – отсчет, проекция, квалификатор, описание и т.п.). Некоторые из них могут базироваться на виртуальных логических устройствах, используемых в графике (функциональная клавиатура, алфавитно-цифровая клавиатура, указатель, координатное и линейное устройства ввода).

Реализовать развертку всего объекта (а это необходимо для создания абстрактного проектора высокого уровня) в рамках виртуальных устройств становится возможным благодаря использованию лишь ограниченного списка обрабатываемых объектов. Если при этом ограничен набор синтаксических правил контроля развертки объектов, то можно выделить такие правила, которым подчиняется измерение некоторого класса объектов. В нашем случае мы вводим объекты двух типов – точку, под которой мы будем понимать несколько более сложный, чем это обычно принято объект: с точкой будем связывать не только координатную информацию (отсчет), но и квалификатор, описание, и линию как совокупность отсчетов (возможно, помеченных). От-

метим, что устройства обработки такого рода объектов образуют новый уровень виртуальных устройств, доступных ОРП. Однако для задания их корректной работы в командных примитивах должны передаваться параметры синтаксических и семантических правил контроля.

Синтаксис обработки зависит от конкретного объекта и текущей цели обработки. Для формализации текущей цели обработки введем понятие модальности работы логических устройств как отношение каждого устройства к запрашиваемому действию. Модальность будем задавать тремя значениями: устройство должно работать (запускается автоматически), устройство не может работать (запрещено), устройство может работать (запускается оператором). Первые два значения соответствуют инициативе выбора действия программным процессором, третье – инициативе оператора. Появление значения модальности "должно работать" вызвано желанием стандартизировать взаимодействие с аппаратурой в том случае, когда не требуется вмешательство оператора, и нужно имитировать действия оператора. Использование этой моды возможно при наличии устройства и наличии у программы информации, необходимой для его работы.

Необходимость предоставления оператору аппарата прерываний процесса формирования данных приводит к включению в примитивы обмена модальности устройств обработки прерываний, если их модальность не может быть определена однозначно в управляющей программе.

Определенные для абстрактного проектора устройства ввода, значения модальности этих устройств и примеры используемых комбинаций приведены в таблице. Измеряемыми физическими объектами являются реперные кресты, вершины взаимодействия, треки. Неупомянувшееся ранее логическое устройство "конец последовательности" формирует "длину объекта". Контролируемыми из ОРП устройствами обработки прерываний являются алфавитно-цифровая клавиатура (как устройство ввода в нестандартных ситуациях) и устройство отказа от измерения текущего объекта. Остальные устройства прерывания контролируются управляющей программой.

Таблица

Запрос	Устройство ввода						
	Отсчет	Квалификатор	Проекция	Описание	Конец	Отказ	Алфавитно-цифровая клавиатура
Начать обработку объекта	оп	авт	оп	авт/запр	запр	оп	оп
Начать обработку объекта на заданной проекции	оп	оп/авт/запр	авт	запр	оп/авт	запр	оп
Указать объект на другой проекции	оп	запр	оп	запр	авт	оп	оп
Описать объект	запр	запр	авт	авт	авт	оп	оп
Указать связь идентификацией объекта	оп	авт	оп	запр	авт	оп	оп
Ввести квалификатор	запр	авт	запр	запр	авт	запр	оп
Запрос прямого ответа	запр	запр	запр	запр	оп/авт	запр	авт

Среди семантических процедур, выполняемых виртуальными устройствами, выделим четыре типа процедур, реализующих указание измеряемого объекта (вывод координатного устройства в предполагаемую область измерения), контроль входной информации и ее преобразование в выходную, отображение оператору списка возможных команд и текстов. Эти процедуры должны отражать "физику" объектов для каждого класса решаемых задач.

Параметры семантической процедуры указания объекта, например, задают режим сопровождения измерений. Сопровождение может отсутствовать, полностью выполняться в мини-ЭВМ (экстраполяцией по параболе при измерении трека), задаваться координатами из ЦЭВМ, быть смешанным (одна координата задается из ЦЭВМ, другая рассчитывается в мини-ЭВМ). Последний режим удобен для обработки экспериментов с искровых камер. Для организации перемеров отдельных треков в событиях, не прошедших этап геометрической реконструкции после измерения на автоматах, введен дополнительный режим предварительного указания трека перед его измерением. Такой режим значительно упрощает идентификацию нужного трека и дает возможность эффективно выполнять перемеры на проекторах, не снабженных графическим дисплеем.

Таким образом, командные примитивы, определяющие запрос к управляющей программе, должны содержать тип объекта и модальность каждого логического виртуального устройства в качестве параметров синтаксиса, а также данные для семантических программ. Последовательное применение принципа виртуальных устройств упрощает реализацию симметричного взаимодействия управляющей программы с ОРП.

4. МОДЕЛЬ УПРАВЛЯЮЩЕЙ ПРОГРАММЫ В ВИДЕ СИСТЕМЫ КОНЕЧНЫХ АВТОМАТОВ

Наиболее привлекательным представлением программы, работающей в интерактивном режиме с точки зрения простоты наблюдения ее поведения и управления является конечный автомат (КА)^{/5/}. Представление системы в виде конечного автомата дает преимущества, особенно ощутимые при программировании на мини-ЭВМ:

- при реализации конечного автомата на ЭВМ обработка входного символа требует небольшого числа операций, следовательно, программа будет работать быстро;

- программная реализация конечного автомата требует фиксированного объема памяти, тем самым исключая задачу управления последней.

Кроме того конечный автомат и некоторые его модификации являются хорошо изученным математическим аппаратом широко применяемым, например, в теории построения компиляторов^{/4, 6/} и, следовательно, мы можем воспользоваться результатами этих разработок.

Проведенная аналогия между лексическими анализаторами и виртуальными логическими устройствами ввода обеспечивает возможность моделирования их работы конечными автоматами^{/4/}.

Широко используемая для задания синтаксиса нотация БНФ не отражает динамики разбора и, следовательно, не дает естественного способа представления необходимой параметризации, определяющей отработку конкретного запроса. Одной из форм более наглядного представления является диа-

грамма переходов ^{/10/} – ориентированный граф с помеченными ребрами. Разметка ребер включает в себя входной символ (в том числе и пустой) и набор процедур, выполняемых на переходе.

Какого вида диаграмма переходов будет отражать требуемую параметрическую настройку по значениям модальности устройств? Определим диаграмму конструктивно. Каждую пару состояний, переход между которыми возможен по выработке лексемы с некоторого устройства, соединим двумя однонаправленными дугами. Одну пометим входным символом, вторую пустым символом ϵ . Представим параметрическую настройку в виде функции уничтожения дуги для каждой пары. Уничтожаться могут как обе дуги (это соответствует запрету перехода ϵ , значит, невозможности работы с устройством, поставляющим такой входной символ), так и только одна из дуг. Наличие только ϵ -дуги, выходящей из некоторого состояния, соответствует запуску устройства автоматически по достижении этого состояния, в противном случае – по получении лексемы с устройства, запущенного оператором. Таким образом, аргументами функции являются модальности виртуальных устройств, передаваемые в командных примитивах.

Для реализации диаграмм переходов в рамках конечных автоматов формально задаются два требования: однозначности – в диаграмме не существует двух ребер с одинаковыми входными пометками, выходящими из одной вершины; и полноты – для любой вершины и для любого входного символа существует ребро, помеченное этим символом и выходящее из этой вершины ^{/10/}.

Если на функцию уничтожения дуг не накладывать дополнительных ограничений, то в некоторых узлах диаграммы возможно появление нескольких ϵ -переходов, что приводит к невыполнению требования однозначности. В таком случае диаграмма переходов задает недетерминированный конечный автомат (НКА). Однако класс НКА не мощнее класса КА и существуют алгоритмы построения КА по заданному НКА ^{/4/}. Более того, программировать иногда проще НКА, чем полученный из него КА ^{/7/}.

Нарушение требования полноты, происходящее, например, при уничтожении обеих дуг, отражает существование определенной взаимосвязи между внутренней структурой автомата и источником входных событий. Класс автоматов, для которых не каждый входной символ может быть подан на вход в определенном состоянии, известен как класс автоматов с ограничением на входе^{/8/}. Именно с помощью такого класса автоматов возможно решение задачи управления потоком входных событий в зависимости от состояния.

Объединяя диаграммы переходов, задающие обработку развертки объектов всех используемых типов в единую диаграмму, можно моделировать управляющую программу в виде инициально-недетерминированного автомата с ограничением на входе. Однако в рамках такой модели сложно описать прерывание процессов обработки запроса по инициативе оператора и возврат в прерванный процесс. Для этого потребуются дополнить диаграмму переходами практически из каждого состояния в абсолютно идентичные поддиаграммы, описывающие поведение системы в каждом из возможных прерываний. Это приведет к значительному увеличению таблиц переходов. Кроме того неясно, каким образом описывать возврат в прерванный процесс, если обработка прерывания требует взаимодействия с программами обработки на ЦЭВМ.

Адекватное описание этих процессов может быть получено моделированием управляющей программы в виде системы конечных автоматов (СКА), формально определяемой^{/9/} как непустое конечное множество автоматов (A_1, A_2, \dots, A_n) , обладающее следующими свойствами:

- автомат A_1 является начальным;
- каждый автомат A_i - это пятерка $(Q_i, \Sigma_i \cup Q', \delta_i, Q_{0i}, F_i)$, где Q_i - непустое конечное множество неструктурированных состояний A_i ; F_i - непустое множество заключительных состояний; Σ_i - непустое конечное множество входных символов; $\delta_i: Q_i \times (\Sigma_i \cup Q'_n) \rightarrow \mathcal{P}(Q_i)$, где $\mathcal{P}(Q_i)$ - множество всех подмножеств Q_i , а $Q'_n = \bigcup_{i=1}^n \text{об}(Q_{0i})$.

Разметочная функция $lab(x) = (x = \epsilon \rightarrow \epsilon, x \in \Sigma_i \rightarrow x, x = Q_{oi} \rightarrow Q'_{oi})$ ставит в соответствие каждому начальному состоянию Q_{oi} метку, однозначно соответствующую этому начальному состоянию, т.е. $lab^{-1}(Q'_{oi}) = Q_{oi}$. Если Q_i и Q_j — множества состояний автоматов A_i и A_j ($i \neq j$), то $Q_i \cap Q_j = \emptyset$. Если Σ_i — множество входных символов A_i , $\Sigma = \bigcup_{i=1}^n \Sigma_i$ и $Q = \bigcup_{i=1}^n Q_i$, то функция перехода автомата A_i это $\delta_i: Q \times (\Sigma \cup Q') \rightarrow P(Q)$.

В качестве начального автомата A_1 выберем автомат, принимающий запросы к управляющей программе и реализующий выбор и запуск автомата, ответственного за обработку заданного типа объекта.

Кроме автоматов для формирования данных в СКА включены автоматы отображения информации, а также автоматы осуществляющие обработку прерываний.

В отношении своего поведения СКА допускает две интерпретации:

- вся система рассматривается как единое целое, причем на ленте ввода, соответствующей системе, помещена цепочка символов из Σ^+ ;
- каждый автомат рассматривается как распознающее устройство, имеющее свою собственную ленту ввода, на которой помещена цепочка символов из $(\Sigma_i \cup Q')^+$.

В решаемой задаче вторая интерпретация более естественна, так как передача управления в начальное состояние любого автомата системы, являющаяся следствием вычисляемого перехода соответствует вызову устройства, а передача управления другому автомату по получении символа из Q' , соответствует прерыванию работы текущего автомата по инициативе оператора. Возврат в прерванный автомат осуществляется по достижении вызванным автоматом заключительного состояния (в том числе и посылке оператора, означающей отказ от запрошенного вызова).

Обращение к программе обработки на ЦЭВМ за очередным запросом происходит либо при возврате в A_1 (завершение отработки запроса), либо явным вызовом A_1 (оператор прервал выполнение запроса и ему необходима информация от ОРП). После предоставления информации оператору во втором случае может возникнуть необходимость возврата в прерванный

процесс без потери измерительной информации и без повторения предыдущего запроса. Для реализации такого возврата введем для A_1 символ "null" $\in \Sigma_1$, при получении которого A_1 будет переводиться в заключительное состояние, и, следовательно, управление возвратится в автомат, вызвавший A_1 .

Важным требованием для реализации в мини-ЭВМ является упоминавшееся требование использования фиксированной памяти. СКА при реализации требует бесконечной магазинной памяти, как следствие возможности рекурсивных обращений к автоматам системы. Поскольку рекурсии при работе оператора в режиме диалога не требуются, то мы вводим запрет рекурсий (явных или неявных) с помощью ограничения на входе.

Таким образом, в предлагаемой модели управляющей программы оператор является источником событий и прерываний. Алфавит входных символов, доступных оператору, представляется множеством символов, входящих в лексемы работающих виртуальных логических устройств ввода, вместе с символами явного вызова автоматов СКА. Использование модели в виде СКА с ограничением на входе и связанное с этим четкое определение каждого состояния программы и множества допустимых входных символов позволяет с помощью выдачи оператору описания состояния СКА (контекста) достаточно просто обеспечить его ориентацию в текущей стадии обработки.

5. РЕАЛИЗАЦИЯ УПРАВЛЯЮЩЕЙ ПРОГРАММЫ

5.1. Структурная схема управляющей программы

Переходя к программной реализации существенно решить три взаимосвязанных вопроса: как расчленив задачу, чтобы упростить реализацию и получить возможность легкой модификации; как представлять входы, состояния и переходы, описывающие СКА; как разрешить специфические проблемы, связанные с ограниченными ресурсами мини-ЭВМ.

Для задания каждого автомата СКА кроме двух стандартных автоматных таблиц, нам потребуется еще одна таблица, определяющая ограничение на входе в каждом состоянии автомата.

Наиболее естественным внутренним представлением допустимого подмножества входных символов является представление в виде двоичного вектора.

Пусть $V = \bigcup_{i=1}^n (\Sigma_i) U Q'$, тогда рассматриваемое в любом состоянии множество допустимых символов является подмножеством $S \in V$. При линейном упорядочивании V подмножество S представляется в виде вектора V_s , в котором i -й разряд равен 1 тогда, когда i -й элемент множества S принадлежит V . В этом случае определение принадлежности i -го элемента заданному подмножеству реализуется за время, независимое от размера множества, а основные операции, такие как объединение и пересечение, осуществляются как машинные операции над двоичными векторами.

При решении задачи выбора перехода по входному символу воспользуемся методом вектора перехода, при котором адреса или метки тех процедур, на которые должно передаваться управление, хранятся в виде вектора в последовательных ячейках памяти, по одной на каждый входной символ. Входной символ служит индексом, по которому выбирается элемент вектора, дающий нужный переход. Достоинством метода является скорость выбора и небольшие затраты памяти.

В силу регулярности таблиц переходов для каждого автомата из СКА (каждая таблица содержит адреса автоматов, запускаемых по прерываниям) и использования техники ограничения на входе, естественно свести все таблицы в одну, введя для процедур обработки остальных символов дополнительную индексацию по номеру работающего автомата.

Состояния автоматов проще всего хранить в явном виде. В таком случае ими также можно пользоваться как индексом для передачи управления внутри процедуры.

Такая реализация автоматов дает возможность подстройки под конкретный эксперимент простой модификацией таблиц.

Заданием в таблицах ограничения на входе СКА только команд управления устройствами проектора естественным образом реализуется управляющая программа автономной работы проектора (рис. 1).

При реализации процессора обмена необходимо учитывать фактор ограниченной памяти мини-ЭВМ, который не позволяет выделить для обмена с ЦЭВМ массив достаточно большого объема, необходимого для удовлетворения произвольных запросов (например, при работе с графическим дисплеем). Для передачи запросов с размерами, превышающими размер выделенного буфера, естественно ввести динамическую разбивку, а в протокол взаимодействия включить признак продолжения. Таким образом, примитив обмена может иметь признаки "начало", "продолжение", "конец" или "начало и конец". Однако практика показывает, что при работе редко требуются одновременно большие буфера как для входных, так и для выходных данных. Поэтому, создавая в общем массиве динамическую границу между буферами и располагая выходной буфер непосредственно за входным, можно организовать подавляющее число обменов между ЭВМ без использования цепочек буферов при ограниченных размерах буферной памяти в мини-ЭВМ.

При прерывании выполнения запроса по инициативе оператора (например, требованием изменить или повторить в другом масштабе изображение на экране графического дисплея) накопленные данные, если они существуют, передаются в ЦЭВМ без признака "конец", освобождая выходной буфер. Для сохранения информации, относящейся к прерванному запросу, запрос оператора формируется на месте уже переданных данных. Тогда для продолжения работы с точки прерывания по получении "нулевого" символа из ЦЭВМ достаточно простого возврата из процессора обмена. В случае приема нового запроса, не связанного с непосредственным отображением информации, процессор обмена реорганизует буферную память, перемещая полученную информацию в начало буферной памяти, и осуществляет передачу управления в начальное состояние требуемого автомата.

5.2. Реализация виртуальных устройств ввода

Единственным физическим устройством ввода символов от оператора про-ектора в RTFAS служит алфавитно-цифровой дисплей. В целях минимизации времени обращения оператора к системе, естественно в качестве основного механизма взаимодействия использовать функциональные клавиатуры, являю-щиеся устройствами ввода односимвольных команд. Число таких клавиатур соответствует числу введенных логических устройств. Общее число клавиш всех клавиатур достаточно большое (~ 60), но число клавиш, отражающих возможные альтернативы в выборе действия в каждом состоянии, не превы-шает двенадцати. Для имитации клавиш разных клавиатур, необходимых од-новременно, мы воспользовались программируемой функциональной клавиату-рой (ПФК)^{/11/}, реализовав ее на цифровой части клавиатуры алфавитно-цифрового дисплея. Отображением ПФК на экране дисплея является "шкала", представляющая собой две строки: верхняя - обозначения допустимых команд, нижняя - соответствующие этим командам обозначения клавиш^{/12/}.

Основная шкала ПФК составлена из набора команд, необходимых для нормальной (без ошибок) работы оператора в пределах одного кадра. Для переключения шкал введены дополнительные команды.

Параллельно с ПФК в режиме функциональной клавиатуры работает и сим-вольная клавиатура дисплея как клавиатура ввода квалификаторов. Использо-вание клавиатуры в обычной моде фразового диалога является нестандарт-ным и в случае необходимости вход в нее осуществляется нажатием кла-виши ESCAPE.

Для ввода описания объектов, представляющего ввод параметров с из-вестными границами, введено специфическое устройство, названное микро-диалогом, которое обеспечивает необходимый контроль вводимой оператором информации наряду с предоставлением удобных средств редактирования. Вход-ной информацией для устройства микродиалога является список вопросов и допустимые пределы цифровых или символьных значений ответов. При ответе оператора на каждый вопрос, вплоть до последнего, возможен возврат и ре-

дакция предыдущих ответов. Ответ на вопрос может состоять из одного или двух символов. Если ответ оператора не укладывается в рамки допустимого, оператор предупреждается об ошибке, а вопрос повторяется.

ПФК и микродиалог осуществляют ввод с помощью стандартного драйвера алфавитно-цифрового дисплея в режиме ввода заданного числа символов. Такой ввод исключает необходимость посылки оператором дополнительного символа завершения ввода.

Координатным устройством ввода являются датчики прецизионной измерительной каретки проектора, управляемой оператором с помощью трекбола. Указание выполняется программным перемещением изображения снимка на столе-экране оператора в зону измерительной марки. Для указания именованных объектов на экране графического дисплея используется программируемая функциональная клавиатура.

5.3. Отображение информации

Кроме стола-экрана оператора устройствами отображения информации в RTFAS являются экраны символьного и запоминающего дисплеев, служащие, в частности, для формирования описания и графического изображения измеряемых объектов. По характеру использования информация на экране алфавитно-цифрового дисплея может быть разделена на постоянную, оперативную и временную. В соответствии с этим экран разбит на функциональные зоны:

- 1 - описание кадра;
- 2 - описание текущего объекта (в том числе и при его формировании);
- 3 - описание шкалы ПФК, доступной оператору в текущий момент;
- 4 - локальная информация - номер проекции, кадра и т.п.;
- 5 - текущий контекст;
- 6 - ввод в режиме фразового диалога и печать диагностических сообщений об ошибках оператора или неисправностях аппаратуры.

Экран запоминающего дисплея (Tektronix-611) используется для вывода текстов достаточно большого объема и формирования графического изо-

бражения в задачах, требующих возможности указания и редактирования уже известных измерительных данных. Как следствие в этих задачах необходима привязка изображения на экране к изображению на столе оператора. Генерируемое программой на экране запоминающего дисплея перекрестие перемещается синхронно с измерительной кареткой проектора.

Набор графических примитивов ограничен строкой символов и совокупностью векторов ^{/13/}. Кроме этих примитивов управляющая программа обрабатывает команды установки коэффициентов преобразования экран/проектор и изменение таблиц, определяющих формат символов (размер и шрифт).

6. ЗАКЛЮЧЕНИЕ

Разработанная математическая модель управляющей программы мини-ЭВМ в виде системы конечных автоматов с ограничением на входе адекватно описывает поведение программы в процессе взаимодействия с оператором и аппаратурой проектора, обеспечивает наглядность представления, возможность проверки корректности работы программы, простоту модернизации при сохранении "узнаваемости" ^{/14/} программы - характеристики, существенной при длительной эксплуатации.

Определение регулярной структуры МО мини-ЭВМ значительно сокращает трудоемкость создания типовых автоматизированных систем обработки снимков, разрабатываемых в ИФВЭ для НИИЯФ МГУ, ЕрФИ, ИФ АН Азерб. ССР.

Авторы выражают благодарность А.Ф.Дунайцеву, В.Д.Жильченкову, П.В.Шляпникову, В.А.Ярбе за поддержку данной работы, А.Ю.Сычеву - за полезные обсуждения и всему коллективу, участвующему в создании и развитии RTFAS - за помощь.

ЛИТЕРАТУРА

1. Ю.А.Белокопытов и др. Препринт ИФВЭ 80-54, Серпухов, 1980.
2. Ю.А.Белокопытов и др. Препринт ИФВЭ 79-176, Серпухов, 1979.
3. Status Report of the Graphic Standards Planning Committee of ACM/SIGGRAPH. Computer Graphics. V. 11, N 3, 1977.
4. Д.Грис. Конструирование компиляторов для ЦЭВМ. М., "Мир", 1975.
5. Б.Р.Гейнс. Опыт разработки и применения интерактивных систем. ТИИЭР, 76 (1975).
6. Ф.Льюис и др. Теоретические основы проектирования компиляторов, М., "Мир", 1979.
7. А.Ахо. Построение и анализ вычислительных алгоритмов. М., "Мир", 1979.
8. А.Гилл. Введение в теорию конечных автоматов. М., "Наука", 1966.
9. А.Оллонгрен. Определение языков программирования интерпретирующими автоматами. М., "Мир", 1977.
10. Б.А.Трахтенброт, Я.М.Бардзинь. Конечные автоматы. Поведение и синтез. М., "Наука", 1970.
11. Сб. Системы с распределением времени, под ред. Карплюса. М., "Мир", 1969.
12. В.М.Дикий и др. Материалы II Всесоюзного семинара по обработке физической информации, Ереван, Арус, 1978.
13. Л.Г.Каминский и др. Препринт ИФВЭ 75-17, Серпухов, 1975.
14. Дж.Р.Дистасо. Обзор методов управления разработкой программного обеспечения. ТИИЭР, 68, 80 (1980).

Рукопись поступила в издательскую группу
10 декабря 1981 года.



Цена 12 коп.

© - Институт физики высоких энергий, 1982.

Издательская группа И Ф В Э

Заказ 1261. Тираж 270. 0,8 уч.-изд.л. Т-00261.

Январь 1982. Редактор М.Л. Фоломешкина.