

5. Bash P. A., Pattabiraman N., Huang C. e. a. Van der Waals surfaces in molecular modeling: implementation with real-time computer graphics.— Science, 1983, v. 222, p. 1325.
6. Одежко Б. Н., Нигматуллин Р. С. Построение объемных молекулярных моделей с помощью вычислительной машины.— ЖСХ, 1977, т. 17, № 1.
7. Васильев А. А. Метод доступа для цветной растровой дисплейной системы PERICOLOR.— Владивосток, 1981. (Препринт/ДВНЦ АН СССР, ИАПУ; 27).

Поступило в редакцию 29 января 1985 г.

УДК 681.3.06

А. И. ОСТРОВНОЙ  
(Дубна Московской)

### МЕТОДИКА ПРОГРАММИРОВАНИЯ СИСТЕМ АВТОМАТИЗАЦИИ ЭКСПЕРИМЕНТОВ НА ЯЗЫКЕ ПАСКАЛЬ

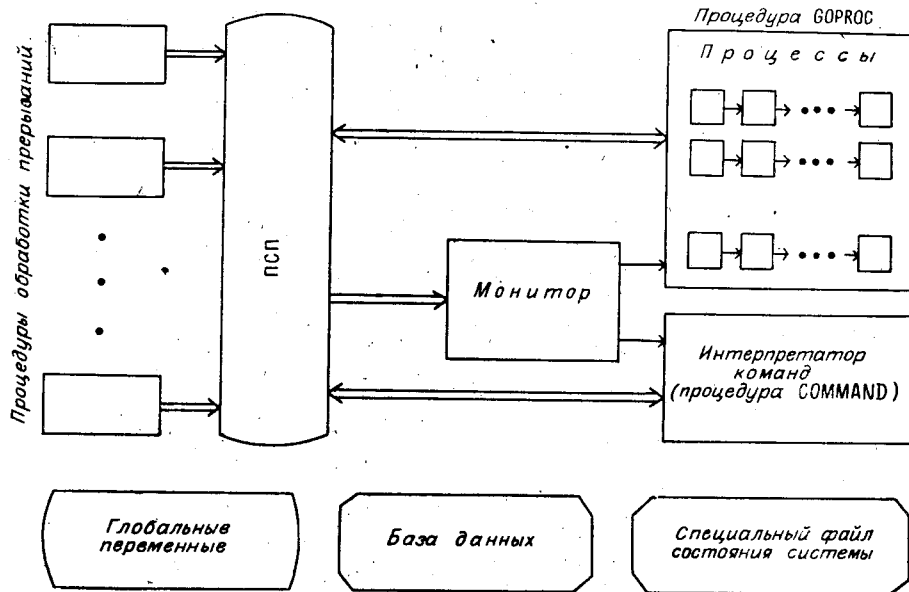
В работе описывается методика программирования систем автоматизации спектральных экспериментов на языке Паскаль, основанная на использовании набора типовых решений для поставленных перед программистами задач. В качестве таких типовых решений предлагается: 1) использовать унифицированную организацию программного обеспечения систем автоматизации; 2) представлять алгоритмы обработки в виде параллельных процессов; 3) программировать операции с аппаратурой КАМАК с помощью набора процедур в соответствии с предложениями комитета ESONE [1]; 4) для архивизации данных и интеграции программ накопления с программами обработки использовать простую базу данных [2].

Перечисленные типовые решения коротко описаны ниже. Они определяют лишь форму решения задач и позволяют концентрировать внимание и усилия на прикладных аспектах проблемы разработки программного обеспечения конкретных экспериментов.

Унифицированная организация программных систем (рисунок) включает процедуры обработки прерываний; алгоритмы обработки, представленные в виде параллельных процессов; таблицу переменных состояния процессов (ПСЦ); интерпретатор интерактивных приказов и монитор. Все программные компоненты имеют доступ к глобальным переменным, к базе данных и специальному файлу, в котором хранится информация о текущем состоянии аппаратуры, программной системы и проводимых измерений.

Процедуры обработки прерываний обеспечивают передачу информации между оперативной памятью (набором глобальных переменных) и аппаратурой КАМАК. Они выполняют минимально необходимые действия, а если требуется дальнейшая обработка поступившей информации, то устанавливается флаг готовности в переменной, отражающей состояние нужного процесса. Монитор, просматривая таблицу ПСЦ, обнаруживает готовый к работе процесс и инициирует его. Одновременно монитор обеспечивает прием приказов с терминала пользователя, и если приказ введен, то инициирует интерпретатор.

Специальный файл, содержащий информацию о состоянии системы, автоматически создается в момент первого запуска программной системы. В дальнейшем ра-



бота программной системы начинается с копирования содержимого этого файла в оперативную память; так определяются значения переменной состояния системы (ПСС). При изменении состояния программа изменяет значение нужного поля ПСС и обновляет содержимое специального файла. Периодически записывая в этот файл таблицу ПСП, можно в случае сбоя в работе программы или аппаратуры восстановить состояние программной системы и продолжить ее работу с момента, который был зафиксирован на магнитном диске.

**Параллельные процессы** удобно использовать для представления алгоритмов обработки нескольких потоков информации. В рамках данной методики предлагается простейший способ программирования параллельных процессов, не требующий использования часов ЭВМ или операционной системы с мультипрограммным режимом работы. С точки зрения организации все процессы выполняются одной процедурой GOPROS. Она состоит из оператора выбора CASE, каждая альтернатива которого реализует один процесс. В свою очередь, процесс с помощью оператора CASE делится на этапы. Каждому процессу в таблице ПСП соответствует переменная, содержащая номер очередного этапа и информацию о готовности процесса к работе, наличии или отсутствии флага блокировки, ожидании завершения другого процесса.

Функционирование процессов осуществляется следующим способом. Монитор в цикле проверяет таблицу ПСП, и если какой-либо процесс готов к работе, не блокирован и не ожидает завершения другого процесса, то производится обращение к процедуре GOPROC с параметром, равным номеру выбранного процесса. GOPROC выполняет очередной этап указанного процесса и возвращает управление в монитор, где продолжается ожидание готового к работе процесса.

Такая организация параллельных процессов позволяет реализовать простые, но удобные средства отладки программных систем. Для этого используются интерактивные приказы, позволяющие блокировать исполнение процессов, разблокировать, вывести на терминал информацию о состоянии всех процессов, выполнять алгоритмы, представленные в виде параллельных процессов, по шагам. Каждый шаг соответствует выполнению одного этапа выбранного процесса.

**Программирование операций с аппаратурой КАМАК** обеспечивается с помощью набора процедур, аналогичных набору подпрограмм на Фортране [3]. Он включает процедуры вычисления значений переменных, описывающих регистры и каналы; обеспечения связи вектора прерываний с соответствующей процедурой; выполнения одиночных и блочных операций обмена с модулями КАМАК; разрешения и запрещения работы отдельного канала и всего крейта.

Созданные средства обеспечивают бесконфликтную работу параллельно функционирующих программ обработки прерываний путем изменения приоритета процессора ЭВМ на время исполнения одной операции КАМАК. Процедура обработки прерываний является обычной процедурой баз параметров; она выполняет минимально необходимую обработку и в случае потребности устанавливает флаг готовности в таблице ПСП для последующего инициирования заранее заготовленного процесса обработки. Разработанные программные средства ориентированы на применение контроллеров крейта типа 106 [4] и СС-11 [5] на СМ ЭВМ.

**Использование простой базы данных** в программах накопления и обработки обеспечивает унифицированное представление экспериментальной информации и единообразный доступ к ней со стороны большого числа программ. Во время проведения измерений база данных применяется для накопления информации, а по их окончании она позволяет проводить автоматическую обработку накопленных данных, извлекая для каждого файла дополнительные параметры, необходимые при обработке.

База данных может также использоваться для хранения истории работы системы автоматизации, информации о действиях оператора и сбоях оборудования, а в дальнейшем — для анализа качества программно-аппаратной системы, поиска причин случайно возникающих сбоев и т. п.

Для организации базы данных и работы с ней применяется пакет процедур [2]. Проблемная ориентация пакета, его компактность позволяют использовать его в системах автоматизации экспериментов на мини- и микроЭВМ. Одно из применений этой базы данных дано в работе [6].

В заключение следует отметить, что описанная методика использована для создания ряда программных систем на ЭВМ СМ-3, СМ-4 и МЕРА-60 в операционных системах РАФОС и ОС РВ. Отдельные положения описанной методики были опробованы ранее в созданной в нашей лаборатории системе САНПО [7, 8]. В целом получен положительный опыт применения этой методики.

Использование языка высокого уровня Паскаль для программирования систем автоматизации экспериментов обеспечивает простоту модификации и расширения программных систем, является основой для создания качественного и мобильного программного обеспечения.

## ЛИТЕРАТУРА

1. Subroutines for CAMAC. ESONE/SR/01, Sep. 1978.
2. Островной А. И. Методика применения базы данных в системах автоматизации экспериментов и ее реализация. — ОИЯИ, Р10-84-439. Дубна, 1984.
3. Вьюшкин О. В., Храпкин П. Л. Пакет стандартных подпрограмм для работы с аппаратурой КАМАК. — Автометрия, 1982, № 4.

4. Интерфейс СМ-3 — КАМАК типа 106А, 196В. Инструкция по обслуживанию 32 776-00000-002. ZZUJ «POLON», ZPE, 02—673. Warszawa, 1973.
5. Ball F. e. a. CERN-NP, CAMAC Note N—43—00. Geneva, 1973.
6. Балагуров А. М., Миронова Г. М., Островной А. И. Программное обеспечение системы накопления информации дифрактометра ДН-2 на импульсном реакторе ИБР-2. — ОИЯИ, Р10-84-440. Дубна, 1984.
7. Островной А. И. Монитор для специализированных систем автоматизации экспериментов. — В кн.: Структура, технические средства и организация систем автоматизации научных исследований (Материалы XV Всесоюз. школы по автоматизации научных исследований). Л.: ЛИЯФ, 1982.
8. Островной А. И., Саламатин И. М. Дисциплина исполнения параллельных процессов в системе САНПО. — ОИЯИ, Р10-81-342. Дубна, 1981.

Поступило в редакцию 8 июля 1985 г.

УДК 681.3.06

В. Г. СЕЛИВАНОВ

(Томск)

### СПОСОБ ПОСТРОЕНИЯ БИБЛИОТЕКИ ЭЛЕМЕНТОВ ДЛЯ СИСТЕМЫ МОДЕЛИРОВАНИЯ РАБОТЫ ЭЛЕКТРОННЫХ СХЕМ

Реализация алгоритмов моделирования работы электронных схем и решение некоторых других задач автоматизации разработки электронных устройств и производства печатных плат требуют создания библиотеки программ, моделирующих работу элементов схем. В данной статье рассматривается один из возможных способов построения такой библиотеки для микросхем на основе ТТЛ-логики и описан процесс взаимодействия системы моделирования с библиотечными модулями.

Предполагается, что в моделирующем программном комплексе используются два стационарных состояния сигналов («0» и «1»), а также фронты сигналов «0-1», «1-0» и состояния неопределенности [1].

В соответствии с точным справочным описанием параметров и зависимостью выходных сигналов от уровней сигналов на входах [2] для каждого элемента создается программа, реализующая его математическую модель. Исходной информацией для такого программного модуля будет матрица  $Z$  описаний сигналов на соответствующих входах микросхемы или иного электронного прибора

$$Z = \begin{vmatrix} a_1, & \dots, & a_n \\ b_1, & \dots, & b_n \\ s_1, & \dots, & s_n \end{vmatrix}, \quad (1)$$

где  $a_1, \dots, a_n$  — номера входов элемента;  $b_1, \dots, b_n$  — уровни сигналов на соответствующих входах;  $s_1, \dots, s_n$  — длительности фронтов сигналов.

Информация о длительности фронтов сигналов используется при расчете возможности возникновения состояния неопределенности. Кроме того, она необходима для определения задержек, которые могут (дополнительно к стандартным для данной серии микросхем) возникать из-за взаимодействия с нестандартными или принадлежащими к другим сериям элементами [3].

Модель возвращает в вызвавшую программу информацию, которую можно представить в виде матрицы  $Q$ -описания выходной реакции элемента

$$Q = \begin{vmatrix} C_1, & \dots, & C_k \\ U_1, & \dots, & U_k \\ R_1, & \dots, & R_k \\ P_1, & \dots, & P_k \end{vmatrix}, \quad (2)$$

где  $C_1, \dots, C_k$  — номера выходов микросхемы;  $U_1, \dots, U_k$  — уровни сигналов;  $R_1, \dots, R_k$  — времена задержки сигналов;  $P_1, \dots, P_k$  — длительности сигналов на соответствующих выходах элемента.

Модель элемента в предлагаемом способе построения библиотеки не отображает внутренних физических процессов в реальной микросхеме или другом приборе, а сопоставляет входные воздействия с соответствующими выходными реакциями.

Библиотека хранится в виде объектных модулей. Система моделирования, используя эту библиотеку, ориентирована на работу совместно с системой технологического проектирования печатных плат [4], которая формирует информацию о связях в моделируемой схеме и о задействованных в ней элементах. Специальная программа подготовки модели схемы в соответствии с полученной информацией генерирует исходный текст на языке Паскаль программы-библиотекаря, в котором описываются обращения ко всем необходимым для данной схемы библиотечным модулям. Далее в процессе обработки косвенного командного файла подготовки и запуска модели этот текст транслируется, строятся необходимые вспомогательные файлы, задание обрабатывается строителем задач и загружается в память ЭВМ на исполнение.