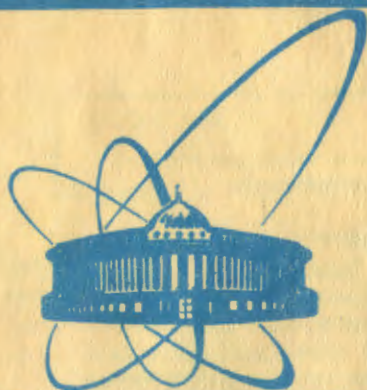


сообщения  
объединенного  
института  
ядерных  
исследований  
дубна



982/2-81

23/11-81

P10-80-743

Г.Балука, Ю.Намсрай, И.М.Саламатин,  
Г.Я.Яновский

МЕТОД НАСТРОЙКИ В ПРИМЕНЕНИИ  
К ПРОГРАММИРОВАНИЮ РАБОТЫ ЭВМ  
НА ЛИНИИ С ЭКСПЕРИМЕНТАЛЬНЫМ  
ОБОРУДОВАНИЕМ

4. Генерация прикладных программ

1980

## ВВЕДЕНИЕ

В <sup>1/</sup> сформулирован подход к программированию работы ЭВМ на линии с экспериментальным оборудованием. Согласно введенной методике, для выполнения типичных прикладных операций с отдельными блоками /или группами блоков/ экспериментального оборудования <sup>2/3/</sup> создаются унифицированные программные модули /УПМ/. Эти модули, предварительно отлаженные, заносятся в библиотеки в виде, не зависящем ни от способа подключения оборудования /например, от номера крейта или номера станции в крейте/, ни от методики конкретного эксперимента.

При необходимости создать для конкретного эксперимента программное обеспечение, работающее с ЭО, на специализированном языке описываются состав оборудования, способ его подключения к ЭВМ, состав и последовательность выполнения прикладных операций. Служебные программы на основании этого описания собирают из библиотек запасенные модули, объединяют их, настраивают и, в конечном счете, создают подсистему с заданными функциями для работы с ЭО.

Для реализации этого метода разработаны структура УПМ <sup>2/</sup>, библиотеки УПМ, язык генерации прикладных систем <sup>3,4/</sup>, компилятор, средства управления библиотеками <sup>5/</sup>, программа-компоновщик и ряд служебных и прикладных программ.

В данной работе описывается программа-компоновщик и продукт компоновки - подсистема для работы с экспериментальным оборудованием.

Цель разработки - сократить сроки создания и стоимость программного обеспечения экспериментов на реакторе ИБР-2 <sup>6/</sup>. Реализация выполнена для мини-ЭВМ, программно совместимых с СМ-3 <sup>7/</sup> в рамках проекта, описанного в работе <sup>8/</sup>.

## ОБЩЕЕ ОПИСАНИЕ ПОДСИСТЕМ ДЛЯ РАБОТЫ С ОБОРУДОВАНИЕМ В СТАНДАРТЕ КАМАК

В рамках описываемого подхода из УПМ создаются подсистемы, обеспечивающие взаимодействие программной системы с экспериментальным оборудованием.

Подсистемы обычно состоят из резидента и набора стандартных программ /СП/.

```

. . . операторы языка САНПО
/ LET COM=22
/ LET DIS=19
/ LET BK=15, AK1=11, KND=9
/ LET AK2=4
//
/ SUBSYSTEM HALF, TYPE=NE/RESIDENT
/ DEVICE 1, AV:170V, VECTOR:1-23=170V
//      THIS ACTIVE CHANNEL HAS HIGHEST PRIORITY
/ HCASE AK2:AK2 DO HISTO(B)
/ HCASE AK1:BK, AK1, KND DO BAD2(A1,A2)
//      B, A2 AND A2 ARE BUFFERS DECLARED ABOVE
/ ATTACH DIS TO TV
//
/ LINK COM TO TRANS
/ ENDDHARDWARE SYSTEM
. . . операторы языка САНПО

```

Рис.1. Фрагмент описания прикладной системы, содержащий описание подсистемы для работы с оборудованием в стандарте КАМАК.

загрузки обязательно фиксируется на рабочем поле памяти, а СП - нет. При необходимости освободить память для других программ СП могут быть автоматически затерты служебными программами. Второе отличие резидента от СП в том, что он может создаваться из произвольного количества УПМ, в то время как СП - только из одного. Очевидно, что резидент должен содержать программные компоненты, к скоростным характеристикам которых предъявляются повышенные требования.

Резидент компонуется из диспетчера и одного или нескольких УПМ. Каждый УПМ имеет 4 управляющих входа с фиксированными названиями / SET , UNSET , RESUME , SUSPEND / и назначением, а также функциональный вход. В результате компоновки одноименные управляющие входы всех компонентов резидента /УПМ и диспетчера/ объединяются в 4 последовательные цепочки, как это показано на рис.2. Такая цепочка программ выполняется при работе прикладной системы за время одной операции управления.

Управляющие части резидента могут исполняться лишь в определенной последовательности и переводят как программные компо-

Способ создания подсистемы, ее состав, структура, тип ЭО и детали его конфигурации описываются на специализированном языке<sup>3/</sup>. На рис.1 приведен пример такого описания.

Наличие или отсутствие резидента или СП в составе подсистемы определяется видом инструкций на языке генерации подсистемы.

Резидент генерируется в случае, если описание содержит инструкции HCASE /для программ обработки прерываний/ или /ATTACH для программ, обслуживающих пассивные каналы/. Инструкция /LINK обеспечивает создание СП. И резидент, и СП формируются в перемещаемом формате загрузки REL<sup>11/</sup> и используются в режиме динамического распределения памяти /ДРП/<sup>10/</sup>. Различие их в том, что резидент после

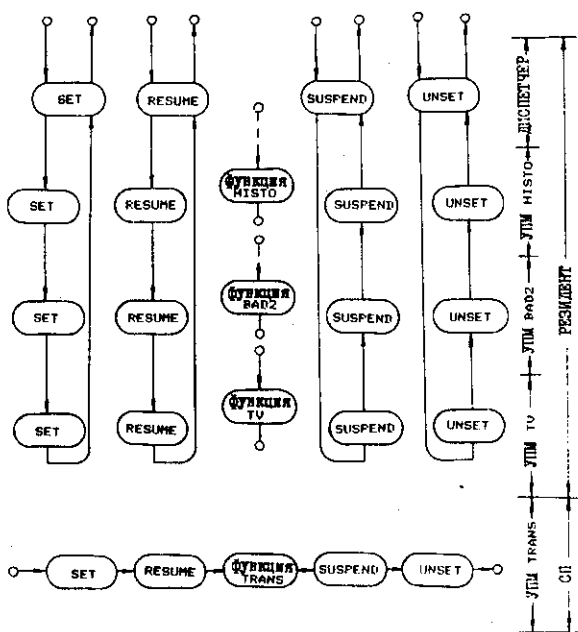


Рис.2. Реализованный способ организации управления подсистемой. Кружочками обозначены точки входа и выхода. Пунктирные линии обозначают передачу управления по прерыванию.

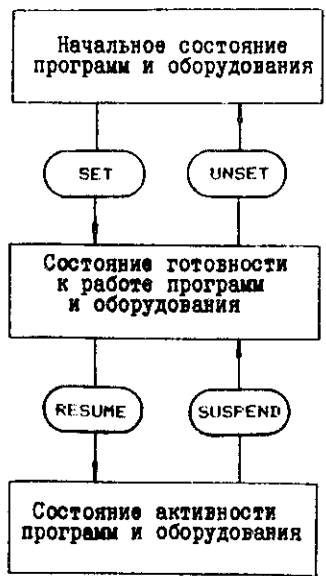


Рис.3. Три возможных состояния подсистемы для работы с экспериментальным оборудованием, операции, изменяющие ее состояния, и возможные последовательности исполнения этих операций.

ненты системы, так и оборудование из одного состояния в другое. Мы различаем три таких состояния: начальное, состояние готовности и состояние активности. На рис.3 приведена диаграмма состояний, поясняющая назначение четырех управляющих входов и допустимые операции управления в каждом из состояний.

В отличие от управляющих частей УПМ, их функциональные части исполня-

|   | 1       | 2          | 3                           |     | 4      |   |        |         |         |       |
|---|---------|------------|-----------------------------|-----|--------|---|--------|---------|---------|-------|
|   |         |            | Текущее состояние резидента | SET | RESUME | Совокупность выполняемых операций при инициации функции УПМ |        |         |         |       |
|   |         |            |                             |     |        | SET   | RESUME | ФУНКЦИЯ | SUSPEND | UNSET |
| 1 | /HCASE  | прерывание | +                           | +   |        |   | +      |         |         |       |
| 2 | /ATTACH | CALL (JSR) | +                           | +   |        |   | +      |         |         |       |
| 3 | /ATTACH | CALL (JSR) | +                           |     |        | +   | +      | +       |         |       |
| 4 | /LINK   | CALL (JSR) |                             |     | +      | +   | +      | +       | +       |       |

Рис. 4. Способы исполнения прикладных операций.

ются независимо друг от друга. Передача управления функциональной части возможна лишь после соответствующих операций управления. На рис. 4 приведена таблица, описывающая способы исполнения функциональных частей УПМ и необходимые операции управления, которые должны предварять исполнение функциональных частей. Во всех случаях перед исполнением функциональной части УПМ должны быть выполнены операции SET и RESUME программными сегментами, содержащимися в этом УПМ. Таблица иллюстрирует также возможность выполнения операции управления различными способами - одновременно для всех УПМ или индивидуально. Одновременное исполнение операции управления над всеми УПМ подсистемы переводит ее в другое состояние. Соблюдение правильной последовательности операций управления контролируется программами.

Объединение отдельных УПМ в подсистему позволило сократить число отдельных операций управления и сформулировать одинаковые управляющие приказы для подсистем, применяемых в различных экспериментах.

#### ДИСПЕТЧЕР РЕЗИДЕНТА ПОДСИСТЕМЫ

В состав диспетчера входят следующие элементы: четыре управляющие части, соответствующие объявленным входам SET, RESUME, SUSPEND и UNSET; программа обработки прерываний, возникающих при отсутствии сигнала "X" во время исполнения КАМАК-функции; служебные программы, обслуживающие окончание работы УПМ резидента; слово состояния подсистемы.

При работе каждой управляющей части диспетчера организовано последовательное обращение к одноименным управляющим частям всех УПМ, входящих в состав данного резидента. Таким образом оборудование, элементы данных и т.д. приводятся в состояние, определяемое назначением каждого управляющего входа.

Способ исполнения управляющих частей резидента имеет следующую особенность /см. рис.2/: по входам SET и RESUME соответствующие части всех УПМ получают управление в том порядке, в котором эти программы появляются в описании подсистемы на языке САНПО<sup>3/</sup>; для входов UNSET и SUSPEND организован обратный порядок исполнения.

Опишем функции каждой управляющей части диспетчера в строгом соответствии с его алгоритмом работы.

#### Инициализация /вход SET /:

1. Разрешение прерывания при отсутствии сигнала "X".
2. Запоминание состояния векторов прерывания, используемых в данном резиденте для обработки прерываний от оборудования.
3. Заполнение вектора, обслуживающего прерывание при отсутствии сигнала "X", адресом входа в соответствующую программу диспетчера.
4. Последовательное исполнение управляющих частей всех УПМ данного резидента, входы в которые имеют имя SET.

#### Отключение /вход UNSET /:

1. Последовательное исполнение для всех УПМ резидента управляющих частей с названиями входов UNSET.
2. Восстановление начального состояния векторов прерывания, использовавшихся в данном резиденте для обработки внешних прерываний.

#### Возобновление /вход RESUME /:

1. Заполнение векторов прерываний, обслуживающих обработку запросов от активных каналов оборудования для данного резидента, адресами входов в функциональные части соответствующих УПМ.
2. Последовательное исполнение управляющих частей всех УПМ резидента с входами RESUME.
3. Разрешение прерываний от контроллеров используемого оборудования.

#### Приостановка /вход SUSPEND /:

1. Запрещение прерываний от контроллеров используемого оборудования.

2. Последовательное исполнение управляющих частей всех УПМ резидента с входами SUSPEND.

Программа обработки прерываний в случае отсутствия сигнала "X" во время исполнения КАМАК-функции обеспечивает формирование обращения к подсистеме обработки аварийных ситуаций /ПОАС/, т.к. такого рода ситуации являются аварийными. В этом случае ПОАС печатает диагностическое сообщение о нефатальной ошибке. Если такая аварийная ситуация повторяется неоднократно, то после 8-го раза программа резидента объявляет фатальную ошибку, ПОАС печатает соответствующее сообщение и приостанавливает исполнение программы эксперимента.

Три служебные программы диспетчера: CONTRT, EXITRT и FAILRT предназначены для обслуживания окончания работы УПМ по командам JMP CONTRT, JMP EXITRT, JMP FAILRT, соответственно /2/.

Программа CONTRT выполняет следующие действия:

- 1/ разрешает прерывания от контроллеров используемого оборудования;
- 2/ исполняет выход из прерывания (RTI).

Программа EXITRT выполняет следующие действия:

- 1/ приостанавливает работу оборудования, используемого данным резидентом, посредством исполнения управляющей операции SUSPEND над резидентом;
- 2/ передает монитору САНПО информацию, обеспечивающую возможность возобновить работу оборудования после завершения обработанного события, объявленного УПМ;
- 3/ исполняет выход из прерывания.

Программа FAILRT анализирует информацию об ошибке, сформированную в УПМ, а именно:

- 1/ если идентифицирована фатальная ошибка, то программа формирует обращение к ПОАС, сообщая тип ошибки и сопутствующую информацию;
- 2/ если обнаружена нефатальная ошибка и счетчик числа повторений не указан /2/, то программа формирует обращение к ПОАС соответствующего вида. Обработка ошибки завершается возвратом управления в программу FAILRT с последующим выходом из резидента или СП;
- 3/ если обнаружена нефатальная ошибка с указанным счетчиком числа повторений, то вид обращения к ПОАС определяется текущим значением счетчика. При значении счетчика, равном 1 /данная ошибка обнаружена впервые/, выполняются действия, аналогичные пункту 2. Если значение счетчика равно верхнему граничному, указанному в УПМ, то ошибка рассматривается как фатальная, и выполняется пункт 1. При промежуточном значении счетчика программа выполняет выход из резидента или СП, не объявляя ошибки.

Слово состояния подсистемы предназначено для контроля диспетчером последовательности исполнения управляющих частей резидента. Код в этом слове хранит описание текущего состояния резидента.

### СТАНДАРТНЫЕ ПРОГРАММЫ ДЛЯ РАБОТЫ С ОБОРУДОВАНИЕМ В СТАНДАРТЕ КАМАК

СП для работы с оборудованием в стандарте КАМАК компонуются из специальной управляющей программы и одного УПМ. Они могут быть фиксированы на рабочем поле памяти с помощью оператора /GET языка САНПО<sup>4</sup>, но обычно такие СП хранятся в библиотеке прикладной системы. По мере надобности СП для работы с оборудованием загружаются на динамически распределяемый участок рабочего поля и исполняются. При работе прикладной системы эти СП, равно как и СП, выполняющие прикладные операции обработки данных, могут быть затерты на динамическом участке рабочего поля. Поэтому необходимо было обеспечить исполнение управляющих и функциональной частей УПМ за время одной операции. Это достигается тем, что специальная управляющая программа, включенная в состав СП, организует последовательную работу всех входов УПМ в фиксированном порядке: SET, RESUME, функция; SUSPEND, UNSET. Это позволяет использовать СП для работы с КАМАК-оборудованием в режиме ДРП.

### РЕКОМЕНДАЦИИ ПО ПРОГРАММИРОВАНИЮ УПРАВЛЯЮЩИХ ЧАСТЕЙ УПМ

Алгоритм работы каждой управляющей части УПМ должен быть построен с учетом правил работы управляющих частей резидента и функций одноименных частей диспетчера.

Поэтому при программировании УПМ мы рекомендуем в общем случае возлагать на их управляющие части следующие операции /если, конечно, они требуются/:

| Наименование входа | Назначение                             | Примеры возможных функций  |
|--------------------|--|--|
| 1                  | 2                                      | 3  |
| SET                | 1. Приведение УПМ в исходное состояние | Обнуление элементов данных /ЭД/ /например, для формирования спектров/, занесение определенных начальных значений в ЭД. |



| 1       | 2  | 3  |
|---------|--|--|
|         | 2. Подготовка оборудования канала.   | Выбор режима работы блоков, проверка работоспособности регистров блоков.                             |
| RESUME  | 1. Проверка и выбор условий /режима/ работы.                               | Анализ текущего состояния ЭД и рабочих ячеек, присвоение значений параметрам для последующей работы. |
|         | 2. Запуск оборудования канала.   | Разрешение работы блоков, разрешение выдачи сигнала запроса прерывания от блоков.                    |
| SUSPEND | 1. Приведение в состояние, обеспечивающее возможность продолжения работы.  | Запоминание текущего состояния ЭД и рабочих ячеек, обеспечение сохранности структуры данных.         |
|         | 2. Временная остановка /запрет активности/ оборудования канала.            | Запрещение работы блоков, запрещение выдачи сигнала запроса прерывания от блоков.                    |
| UNSET   | 1. Обеспечение сохранности и/или последующей обработки текущей информации. | Объявление событий, запись признаков конца данных.   |
|         | 2. Отключение оборудования канала.   | Спасение текущих значений регистров оборудования канала, переключение в режим off-line.              |

Как видно из перечня рекомендуемых операций по каждому управляющему входу УПМ, эти операции являются типичными при программировании работы с оборудованием в стандарте КАМАК и обычно выполняются в том же порядке, который мы предлагаем. Разработанная структура УПМ, а также последовательность работы управляющих входов 1/ позволяют конкретизировать алгоритм работы каждой управляющей части, тем самым дисциплинируя программиста при создании программного модуля и снижая вероят-

ность отсутствия в алгоритме каких-то обязательных элементов; 2/ облегчают разработку и чтение текста УПМ при проверках и модификациях.

### ОПИСАНИЕ ПРОГРАММЫ-КОМПОНОВЩИКА

Программа-компоновщик предназначена для создания резидентов и СП для работы с оборудованием различных типов из отдельных УПМ.

Под типами оборудования мы понимаем типы контроллеров крейта КАМАК, а также устройств сопряжения других стандартов, различающихся способом программирования работы с ними.

Место программы-компоновщика в общем комплексе средств генерации прикладных систем иллюстрирует рис.5. На основании описания подсистем в некотором промежуточном формате, содержащемся в файле, созданном компилятором САНПО, программа-

компоновщик выбирает из библиотек САНПО те УПМ, которые упомянуты в описании подсистемы и предназначены для работы с указанным типом оборудования. Результатом работы программы являются загрузочные модули резидентов и СП в перемещаемом формате REL, которые могут быть занесены в рабочую библиотеку прикладной системы. Программа-компоновщик написана на языке MACRO-11<sup>12/</sup> и работает под управлением монитора операционной системы RT-11<sup>12/</sup>. Программа может быть инициализирована двумя способами: в автоматическом режиме - компилятором САНПО, в интерактивном - командой RUN монитора RT-11. Во втором случае программа печатает символ \* и ожидает ввода с терминала приказа оператора.

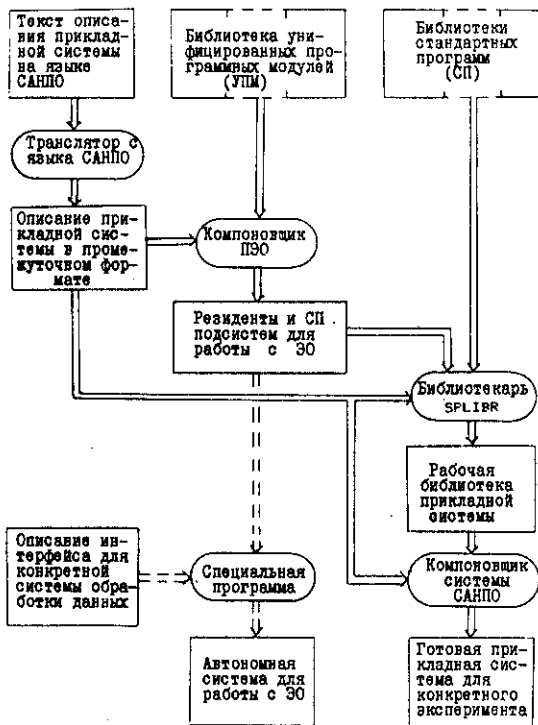


Рис.5. Схема генерации прикладных систем.

Этот приказ имеет следующий формат: FILE1 , FILE2=LBR<sub>1</sub>,...,LBR<sub>N</sub>, где FILE1 , FILE2 , LBR<sub>i</sub> - спецификации файлов в формате, принятом в системе RT-11/12/;

- FILE1 - файл, созданный компилятором САНПО и содержащий описание подсистем в некотором промежуточном формате;
- FILE2 - файл листинга результатов работы программы;
- LBR<sub>i</sub> - файлы библиотек САНПО, содержащих необходимые для компоновки описанных подсистем модули диспетчеров и УПМ.

Результирующие файлы /резиденты и СП/ записываются на устройство, которое указано в спецификации файла FILE1 в строке приказа на компоновку. Помимо этого, в автоматическом режиме работы программы-компоновщика эти файлы программой SPLIBR заносятся в рабочую библиотеку прикладной системы /рис.5/.

Программа-компоновщик имеет оверлейную структуру. В ее состав входят основной сегмент /ядро/ и оверлейные сегменты. Ядром программы выполняются следующие функции: интерпретация описания подсистемы в промежуточном формате, организация работы оверлейных сегментов, загрузка УПМ подсистемы из библиотек, их модификация, обеспечивающая перемещаемый формат результирующих модулей, создание файлов этих модулей на указанном устройстве.

Помимо этого, ядро обеспечивает формирование и печать листинга результатов компоновки, а также диагностических сообщений при обнаружении ошибок в описании подсистемы и/или структуре УПМ.

Каждый оверлейный сегмент соответствует определенному типу оборудования. При интерпретации описания подсистемы ядро программы распознает заданный тип оборудования и передает управление соответствующему оверлею. Оверлей обеспечивает чтение модуля соответствующего диспетчера /или специальной управляющей программы/ в область памяти, где компонуется тело файла подсистемы. Затем эти программы настраиваются в соответствии с конфигурацией и способом подключения данного типа оборудования. Для этой цели организуется таблица, в которую заносится информация о базовых адресах регистров и поля векторов прерывания, номере приоритета линии на общей шине, к которой подключено оборудование, а также формируются код маски запросов прерываний и адреса программ обработки этих запросов.

Помимо этого, в оверлее осуществляется выбор УПМ подсистемы и настройка их на конкретные адреса оборудования. Принцип настройки УПМ на конкретные адреса оборудования подробно описан в работах /1.2/. Он заключается в том, что реальные адреса оборудования записываются в команды обращения к оборудованию, ссылки на которые присутствуют в таблице настройки УПМ.

| SANFO HARDWARE SUBSYSTEM LINK |      |      |        | REL. LOAD MAP |        |        |        |        |
|-------------------------------|------|------|--------|---------------|--------|--------|--------|--------|
| SYSTEM                        | TYPE | SIZE | ENTRY  | ADDR          | ENTRY  | ADDR   | ENTRY  | ADDR   |
| *NALF                         | * NE | 5546 |        |               |        |        |        |        |
|                               |      |      | SET    | 000246        | UNSET  | 000650 | RESUME | 000134 |
|                               |      |      | SUSPND | 000552        | DISPAT | 000134 | HISTD  | 002426 |
|                               |      |      | BAD2   | 003360        | TV     | 005164 |        |        |
| *TRANS                        | * NE | 2706 |        |               |        |        |        |        |
|                               |      |      | SET    | 000616        | UNSET  | 000624 | RESUME | 000632 |
|                               |      |      | SUSPND | 000640        | TRANS  | 000676 |        |        |

Рис.6. Листинг результатов компоновки подсистемы для работы с оборудованием в стандарте KAMAK.

Листинг компоновки содержит описание состава резидентов и СП всех подсистем, созданных программой-компоновщиком. На рис.1 представлено описание некоторой подсистемы на языке SANFO, а на рис.6 приведен листинг ее компоновки.

Первый элемент листинга перечисляет состав резидента данной подсистемы. Имя резидента совпадает с именем, указанным в операторе /SUBSYSTEM описания. Второй элемент листинга относится к СП, которая описана в подсистеме с помощью оператора /LINK. Имена СП и УПМ совпадают. В графе TYPE указывается тип оборудования, в графе SIZE - длина программного модуля /в байтах/. В графах ENTRY и ADDR приведены имена входов и их адреса относительно начала модуля соответственно. Листинг содержит имена / SET , RESUME , SUSPEND , UNSET / управляющих входов диспетчера и специальной управляющей программы для резидента и СП соответственно. Имя DISPAT является именем диспетчера резидента, а имена функциональных входов УПМ подсистемы совпадают с именами этих УПМ.

Тексты диагностических сообщений об ошибках при компоновке подсистем, а также необходимые пояснения к ним приведены в Приложении. При обнаружении ошибок, сообщения о которых приведены в строках 10,14 Приложения, программа-компоновщик обеспечивает создание конечного продукта - резидентов и СП; в случае иных ошибок /из числа приведенных в Приложении/ компоновка на выполняется.

## ЗАКЛЮЧЕНИЕ

Созданы средства генерации прикладных программ для работы с экспериментальным оборудованием из отдельных УПМ, не зависящих от конфигурации ЭО. Этот метод обеспечивает преемствен-

ность программирования и отладки программ, работающих с оборудованием в стандарте КАМАК. Скоростные характеристики прикладных программ отвечают достижимым при программировании на ассемблере.

Систематизировано управление программными компонентами и оборудованием, используемыми в эксперименте, а именно:

- 1/ определена дисциплина работы управляющих входов резидента подсистемы;
- 2/ выработаны рекомендации по программированию управляющих частей УПМ.

Рекомендации содержат перечень типичных операций, обычно требуемых при работе с ЭО, и направлены на то, чтобы, с одной стороны, дисциплинировать программиста при написании УПМ, с другой - облегчить чтение текста УПМ при проверках и редактировании.

Создана программа-компоновщик, обеспечивающая генерацию подсистем для работы с оборудованием в стандарте КАМАК, подключаемым с помощью четырех различных типов контроллеров/9,13-15/.

Структура программы-компоновщика предусматривает возможность легкого расширения ее функций путем добавления оверлейных сегментов, соответствующих новым типам оборудования.

Программа написана на языке MACRO-11 и работает под управлением монитора операционной системы RT-11.

К отличительным особенностям реализованного подхода к программированию работы с ЭО относится следующее:

- 1/ реализован метод настройки программ в соответствии с конкретным размещением экспериментального оборудования. Выполнение настройки на этапе генерации подсистем устранило в командах обращения к оборудованию избыточный уровень косвенной адресации, характерный для двухуровневого подхода/16/;
- 2/ обеспечена возможность без редактирования текста описания подсистемы на основании этого описания генерировать прикладные программы для работы с оборудованием различных типов;
- 3/ реализован способ исполнения СП для работы с оборудованием в стандарте КАМАК, позволяющий использовать их в режиме динамического распределения памяти.

В заключение авторы благодарят коллег за полезные обсуждения и помощь в работе, Т.Б. Журавлеву - за оформление рукописи.

## ПРИЛОЖЕНИЕ

### Тексты сообщений об ошибках при компоновке

| Текст сообщения                                 | Пояснения   |
|---|---|
| 1. ILL MODULE STRUCTURE                         | - неверная структура УПМ.   |
| 2. ILL CRATE NUMBER                             | - номер крейта в описании подсистемы указан неверно.  |
| 3. ILL STATION NUMBER                           | - номер станции в описании подсистемы указан неверно.   |
| 4. ILL PRIORITY LEVEL NUMBER                    | - номер приоритета линии общей шины, к которой подключено оборудование, указан неверно.   |
| 5. DISPATCHER NOT FOUND                         | - в библиотеке отсутствует модуль диспетчера с фиксированными именем и расширением.   |
| 6. MODULE NOT FOUND                             | - в библиотеке отсутствует УПМ.   |
| 7. NOT FOUND ANY UPM-S                          | - в библиотеке не найдены УПМ, требуемые для компоновки некоторой подсистемы.   |
| 8. ILL NUMBER OF ARGUMENTS                      | - число аргументов УПМ, указанное в описании подсистемы и в теле УПМ, не совпадает.   |
| 9. ILL NUMBER OF BLOCKS                         | - число блоков канала оборудования, указанное в описании подсистемы и в теле УПМ, не совпадает.                                 |
| 10. BASE VECTOR ADDRESS IS OUT-<br>OF THE RANGE | базовый адрес поля векторов прерывания задан вне граничных значений, принятых для данного типа оборудования.                    |
| 11. ILL INTERRUPT VECTOR                        | - вектор прерывания в описании подсистемы указан неверно.   |
| 12. NOT FOUND INTERRUPT VECTOR-                 | в таблице диспетчера резидента подсистемы не найден адрес вектора прерывания для указанного в описании номера активной станции. |

13. ILL NO-X INTERRUPT VECTOR - вектор прерывания, обслуживающий обработку запросов при отсутствии сигнала "X", указан неверно.
14. NO-X ERROR HANDLING IS DESTROYED - вектор прерываний, предназначенный для обработки запроса при отсутствии сигнала "X", занимается для обработки запросов от оборудования активного канала.
- 

#### ЛИТЕРАТУРА

1. Балука Г. и др. ОИЯИ, P10-80-424, Дубна, 1980.
2. Намсрай Ю. и др. ОИЯИ, P10-80-456, Дубна, 1980.
3. Намсрай Ю., Островной А.И., Саламатин И.М. ОИЯИ, P10-80-480, Дубна, 1980.
4. Островной А.И., Саламатин И.М. ОИЯИ, P10-80-423, Дубна, 1980.
5. Балука Г., Саламатин И.М., Хрыкин А.С. ОИЯИ, 10-12546, Дубна, 1979.
6. Ананьев В.Д. и др. ОИЯИ, P3-10888, Дубна, 1977.
7. Наумов Б.Н., Боярченков М.А., Кабалецкий А.Н. Приборы и системы управления, 1977, №10, с.12-15.
8. Балука Г. и др. ОИЯИ, P10-12960, Дубна, 1980.
9. Nuclear Enterprises. SAMAC Catalogue. June, 1977, No.108, printed by PSQ Ltd., East Kilbrige, Scotland, 69 p.
10. Балука Г., Островной А.И. ОИЯИ, P10-13004, Дубна, 1980.
11. RT-11 Software Support Manual (DEC-11-ORPGA-B-D), Dec., Maynard, Massachusetts, 1975, 342 p.
12. RT-11 System Reference Manual (DEC-ORIGA-C-D), Dec., Maynard, Massachusetts, 1975, 985 p.
13. Bal F. et al. CC-11. SAMAC CRATE - PDP-11 Interface, Type 116. CERN-NP. SAMAC Note No-43-00, Geneva, 1973, 17p.
14. КАМАК-системы автоматизации в экспериментальной биологии и медицине. Под ред. Ю.Е.Нестерихина. "Наука", Новосибирск, 1978.
15. Хрущев С.Н. и др. ИВК в научных исследованиях. В кн.: Обзор ЦНИИ ТЭИП. "Приборостроение", М., 1979.
16. Halling H., Michelson I. Will Fortran Tolerate with IML? SWG 20/72, 1972.

Рукопись поступила в издательский отдел  
18 ноября 1980 года.