

0!

сообщения
объединенного
института
ядерных
исследований
дубна

5318/2-80

3/11-80

P10-80-456

Ю. Намсрай, А. С. Савватеев, И. М. Саламатин,
Г. Я. Яновский

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

2. Структура унифицированных программных
модулей

1980

Намсрай Ю. и др.

P10-80-456

Метод настройки в применении к программированию работы ЭВМ на линии с экспериментальным оборудованием

2. Структура унифицированных программных модулей.

Данное сообщение входит в цикл работ, посвященных применению метода настройки для программирования функционирования ЭВМ на линии с экспериментальным оборудованием (30). Описывается структура унифицированных программных модулей, настраиваемых на конкретное размещение и способ подключения экспериментального оборудования, а также адреса используемых областей памяти.

Из этих модулей на основании описания на специализированном языке высокого уровня для каждого конкретного эксперимента компонуется программа, которая выполняет инициализацию оборудования, организует начало и остановку эксперимента, обеспечивает обмен данными между ЭВМ и 30 и передает управляющие команды и параметры экспериментальной установке. Программы реализованы на машинах с системой команд СМ-3 для программирования экспериментов на реакторе ИБР-2.

Работа выполнена в Лаборатории нейтронной физики ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1980

Namsraj Yu. et al.

P10-80-456

A Setting Method Applied to Programming Computer Operation On-Line with the Experimental Set-Up.

2. Structure of Unified Program Modules.

The present paper is a part of series of works on the ap

1. ВВЕДЕНИЕ

Данное сообщение является продолжением описания цикла работ, посвященных применению метода настройки для программирования функционирования ЭВМ на линии с экспериментальным оборудованием. В описываемом подходе для обеспечения конкретных экспериментов используется библиотека унифицированных программных модулей /УПМ/, предназначенных для выполнения прикладных операций с оборудованием. Из этих модулей на основании описания на специализированном языке высокого уровня для каждого конкретного эксперимента компонуется программа /или подсистема в терминах системы САНПО^{1/}/, которая выполняет инициализацию оборудования, организует начало и остановку эксперимента, обеспечивает обмен данными и передает управляющие команды и параметры экспериментальной установки.

Ясно, что размещение электронных блоков в каркасах, определяющее адреса регистров этого оборудования, может быть произвольным и определено только для конкретного эксперимента. Но даже во время эксперимента иногда приходится в силу технических причин переставлять блоки на другие места в каркасе. Поэтому настоятельно требуется, чтобы УПМ могли настраиваться на любые позиции блоков в каркасе.

В данном сообщении описывается структура УПМ для работы с экспериментальным оборудованием /в частности, с оборудованием в стандарте КАМАК^{2,3/} и оборудованием фирмы DEC^{4/}/, настраиваемых на конкретное размещение оборудования в каркасе и адреса используемых областей памяти.

Цель данной разработки - сократить сроки создания и стоимость программного обеспечения экспериментов на реакторе ИБР-2^{5/}.

Работа выполнена в рамках проекта проблемно-ориентированной системы САНПО^{1/} для ЭВМ, программно совместимых с СМ-3 и "Электроникой-60"^{6,7/}.

2. ТИПЫ ПРОГРАММНЫХ МОДУЛЕЙ И ИХ СТРУКТУРА

Унифицированным программным модулем /УПМ/ будем называть программу, реализующую прикладную операцию с конкретным оборудованием, составленную для условных адресов оборудования, но

допускающую настройку на конкретные адреса и оформленную по определенным правилам.

Возможность настроить модуль позволяет пользователю не вносить в него никаких изменений, если нужно использовать УПМ с оборудованием, адреса которого /например, В, С, N для стандарта КАМАК/ изменились. Следовательно, УПМ может быть использован без изменений для работы с блоками, размещенными в любых позициях каркаса для оборудования. Помимо этого, для УПМ стандартизован способ обмена данными и управлением с системой обработки данных и управляющими программами.

Программные модули могут быть написаны на языке MACRO-II^{7/}, а перед использованием должны быть транслированы средствами операционной системы общего назначения^{8/} и занесены в двоичном перемещаемом формате REL^{9/} в библиотеку^{10,11/}.

Для пояснения причин, обусловивших появление разных типов УПМ, введем терминологию. Назовем каналом группу блоков оборудования, которые будут обслуживаться одним УПМ. В зависимости от способа использования оборудования мы различаем активные и пассивные каналы. Активными будем считать каналы, оборудование которых может генерировать сигналы запроса прерывания ЭВМ для синхронизации работы ЭО и программ, а пассивными - каналы, работа с которыми осуществляется по инициативе со стороны ЭВМ.

Для обслуживания таких каналов введено два типа модулей.

Модули первого типа введены для обслуживания активных каналов. Исполнение прикладной операции таким модулем инициируется по сигналу запроса прерывания оборудованием канала. Завершается выполнение операции передачей управления служебным программам и, в конечном счете, командой выхода из прерывания.

Модули второго типа, разработанные для обслуживания пассивных каналов, являются подпрограммами типа SUBROUTINE^{12/}.

Модули обоих типов имеют одинаковую структуру. Отдельные элементы структуры модуля помечены цифрами /справа/ в тексте УПМ, приведенном на рис.1 в качестве примера. Это следующие элементы:

1. Операторные скобки на языке MACRO-II;
2. Группа операторов, обеспечивающих кодирование характеристик УПМ в теле загрузочного модуля. Этот элемент занимает 34 байта, его назначение подробно описано в работах^{10,11/};
3. Операторы, определяющие значения констант, используемых при компиляции тела модуля, и макроопределения. Эти операторы не расходуют память;
4. Код дистанции до начала таблицы /пункт 7/, управляющей настройкой модуля;
5. Таблица входов в программное тело модуля. Эта таблица открыта для дальнейшего расширения новыми входами в области меньших адресов;

6. Программное тело;

7. Таблица, управляющая настройкой модуля. В таблице, в свою очередь, содержится несколько элементов, а именно:

а/ код дистанции до первой исполняемой инструкции модуля /а также до таблицы входов/; код дистанции до конца данной таблицы; код количества специальных параметров. /число блоков и число элементов данных/;

б/ и в/ таблица управления настройкой УПМ на конкретные адреса оборудования;

г/ и д/ таблица описания специальных элементов данных;

8. Текстовые и числовые константы, используемые в программном теле.

Ниже в тексте ссылки на эти элементы структуры, помеченные на рис.1, представлены цифрами с буквами в круглых скобках.

В рамках данной структуры доступны: передача фактических параметров, указанных в обращении, настройка модулей при компоновке прикладной системы /на этапе генерации/, настройка на этапе загрузки в оперативную память, либо во время исполнения прикладной задачи /с помощью специальных операций/.

Операции настройки введены для достижения минимального времени реакции системы при сохранении независимости УПМ от конфигурации оборудования и прикладной задачи.

В следующих разделах подробнее остановимся на введенных алгоритмах настройки модуля на конкретные адреса оборудования и настройки на выделенные участки оперативной памяти.

3. НАСТРОЙКА УПМ НА АБСОЛЮТНЫЕ АДРЕСА ОБОРУДОВАНИЯ

Прежде чем описать элементы структуры УПМ, введенные для реализации настройки его на абсолютные адреса оборудования, рассмотрим способы кодирования этой информации в командах ЭВМ для разных типов оборудования. Ограничим это рассмотрение стандартом КАМАК, получившим признание и широкое распространение^{12/}, и стандартом, принятым фирмой DEC^{14/}.

3.1. Работа с оборудованием в стандарте КАМАК выполняется посредством команд, включающих адресную информацию CNA /в случае многоветвевой системы BCNA/ и код функции F. Мы будем пользоваться обозначениями из работы^{13/}: В - номер ветви, С - номер каркаса в ветви, N - номер станции в каркасе, А - суб-адрес и F - код функции.

Формат команды CNAF, т.е. способ представления ее в виде команд ЭВМ, зависит от типа используемого контроллера и от системы команд ЭВМ, к которой подсоединен контроллер. Остановимся отдельно на контроллерах двух типов.

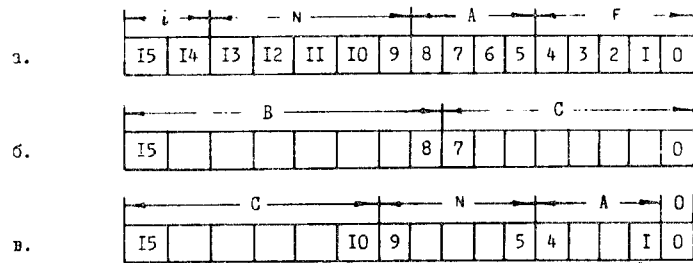


Рис.2. Способ кодирования управляющей информации в регистрах контроллеров двух типов:

- а/ формат данных в командном регистре контроллера, описанного в работе /12/;
- б/ формат данных в регистре номера ветви и крейта в контроллере, описанном в работе /12/;
- в/ способ кодирования адреса для контроллера, описанного в работе /15/.

```
MOV_#const, @#CRTi,
LABEL:MOV_#iNAF, @# CMND,
```

то код iNAF транслятором будет помещен по адресу LABEL+2.

3.1.2. Для контроллеров, описанных в работах /13-15/, адрес любого регистра является адресом на общей шине /4/. Способ кодирования адреса показан на рис.2в. Для того, чтобы выполнить функцию F в блоке CN с регистром A, необходимо:

- 1/ в регистр состояния контроллера (CSR), который имеет адрес CN(0)A (0), занести код функции F;
- 2/ написать команду обращения к блоку в виде одной из следующих инструкций:

```
LABEL:TST_@# CNA;           для функций без передачи /1/
;                           данных (F8÷F15 и F24÷F32).
LABEL:MOV_R,@#CNA;         для F16÷F23. /2/
LABEL:MOV_@# CNA,R;       для F0÷F7. /3/
```

Здесь CNA - адрес на общей шине. Этот адрес получается при упаковке кодов C, N и A /рис.2в/.

При этом код CNA будет помещен транслятором по адресу LABEL+2 или LABEL+4 /в зависимости от способа адресации/ - для инструкции /2/ и по адресу LABEL+2 для инструкций /1,3/.

3.2. Каждое устройство стандарта DEC /4/ имеет группы регистров, которые занимают последовательные адреса на общей шине. Эти адреса можно записать в следующем виде: BASE+0, BASE+2, BASE+4 и т.д., где BASE является базовым /начальным/ адресом поля регистров данного устройства. Если записать команды обращения к i-му регистру устройства в стандарте DEC следующими инструкциями:

```
BASE = 0;
LABEL:MOV_@#BASE+2*i, DATA;           для чтения, /4/
LABEL:MOV_DATA,@#BASE+2*i;           для записи, /5/
```

то относительные адреса регистров транслятор поместит по адресу LABEL+2 /4/ или LABEL+4 /5/.

3.3. Из этого рассмотрения видно, что при использовании определенных способов адресации /непосредственной и абсолютной/ для программирования команд стандарта KAMAK на языке MACRO-II мы получаем возможность настраивать программу так, чтобы она могла работать с одним и тем же блоком, установленным на любой станции (N) любого крейта (C). Для такой настройки в случае контроллера типа, описанного в /12/, требуется изменить значение 9 ÷ 15 разрядов константы, помещенной в слове с адресом LABEL+2, фактическим значением кода jN, а также в j-й регистр CRTj записать значения B и C.

В случае контроллеров, описанных в работах /13-15/, для настройки команды требуется изменить значение разрядов 5 ÷ 15 адресной константы в ячейке LABEL+2 /или LABEL+4/ новыми значениями C и N.

Для устройств стандарта DEC при настройке команды требуется к относительному адресу в ячейке LABEL+2 /или LABEL+4/ прибавить фактический базовый адрес.

3.4. В структуру описываемых УПМ для управления настройкой модулей на конкретные позиции блоков в каркасах оборудования введены специальные элементы.

Таковыми элементами являются специальная таблица и код дистанции до ее начала /этот код указывает также на конец программного тела модуля/. На рис.1 приведен пример текста УПМ для контроллера, рассмотренного в параграфе 3.1.2 данной работы. Указанные элементы помечены на этом рисунке, как /п.7/ и /п.4/ соответственно. В таблице /п.7б,в/ перечислены те команды тела модуля, которые подлежат изменению в соответствии с фактической позицией блока в крейте. Строки таблицы содержат коды дистанций этих строк относительно настраиваемых команд. Нужный код дистанции может быть сформирован в строке таблицы инструкцией на языке MACRO-11

.WORD—LABEL—.,

где LABEL - метка настраиваемой команды.

Заметим, что программный модуль может работать более чем с одним оборудованием. При этом во время настройки мы должны различать команды обращения к первому блоку, второму и т.д., чтобы подставить соответствующие значения С, N. Для этого коды смещений, относящиеся к одному блоку оборудования, образуют в таблице одну группу и отделены нулевым терминатором /например, группы 76 и 7в на рис.1/.

Таблица /п.7/ может быть создана макросом .SETTBL, вызов которого выполняется инструкцией

.SETTBL K,M,ENTRY

Здесь K и M - числа без знака, равные соответственно числу блоков оборудования и числу элементов данных, с которыми работает УПМ; ENTRY - метка первой исполняемой инструкции программного тела модуля. В случае отсутствия в строке фактических параметров идентификатора метки по умолчанию будет принято фиксированное имя START.

Если предполагается использовать макрос .SETTBL, то метки настраиваемых команд должны быть стандартными. Такие метки имеют вид BnLm, где n и m - целые без знака, принимающие значения от 1 до 16; n - порядковый номер блока, m - порядковый номер команды в программном теле, адресованной к данному блоку.

4. ДОСТУП УПМ К ЭЛЕМЕНТАМ ДАННЫХ

Аргументами УПМ могут быть буфера, переменные, константы и так называемые события, являющиеся элементами данных /ЭД/ оперативной базы данных /1/. Доступ к аргументам обеспечивается организующими программами /монитором/ для программ обработки данных /стандартных программных модулей/. Передача параметров, указанных в обращении, обслуживается таблицей, содержащей код числа параметров и список адресов размещения фактических параметров. В момент обращения к модулю адрес такой таблицы находится в пятом общем регистре ЭВМ. Этот механизм является стандартным для обращений к подпрограммам в использованной нами операционной системе RT-II и подробно описан в работах /8,9/. Для того, чтобы сократить мертвое время, в программах регистрации данных и управления экспериментальной установкой удобно иметь прямой доступ к фактическому месту размещения аргументов. Такой доступ обеспечен путем краткого описания

элементов данных в теле УПМ /см. пункт 7г, д. на рис.1/. Для описания каждого элемента отведено 5 слов. Назначение этих слов следующее /см. пункт 7г/:

- 1/ адрес начала ЭД;
- 2/ адрес конца ЭД;
- 3/ адрес слова состояния ЭД в таблице описания базы данных /1/;
- 4/ адрес текущего указателя заполнения буфера;
- 5/ первоначально - адрес начала буфера, в дальнейшем - программа может использовать это слово в качестве рабочей ячейки, например как указатель заполнения буфера.

Адреса в третьем и четвертом словах такого описания используются в теле программы для передачи монитору или системе обработки данных информации о состоянии данного ЭД /о готовности его к дальнейшей обработке /1/. Если используемым ЭД является не буфер, а переменная или константа, то первое и второе слова содержат адреса соответственно старшего и младшего слов значения, третье слово - адрес слова состояния ЭД, а значение четвертого и пятого слов не определено.

В случае использования макроса .SETTBL для формирования таблиц настройки память для описания специальных элементов данных будет зарезервирована автоматически. Группам из 5 слов будут присвоены имена BUF1, BUF2, ...

5. ПРОГРАММНОЕ ТЕЛО УПМ

В программном теле УПМ /см. пункт 6 на рис.1/ можно выделить функциональную /п.6а/ и 4 управляющих /п.6б/ части.

5.1. Функциональная часть реализует алгоритм обмена информацией с каналом. При программировании этого алгоритма автор УПМ должен выполнять следующие требования:

1. Изменять значение текущего указателя границы между свободной и занятой частью ЭД /обычно буфера/ после каждой операции записи /чтения/ в буфер /из буфера/. Очевидно, что изменение этого значения требуется на величину, равную длине записи. Адрес текущего указателя находится в четвертом слове описания /п.7г или д/ соответствующего ЭД;
2. После каждой операции обмена проверять, заполнен ли буфер. Если буфер не заполнен, то модуль должен закончить работу командой JMP CONTRT, где CONTRT = 0. Если буфер заполнен, то работа модуля должна быть завершена командой JMP EXITRT, где EXITRT = 2. Сообщить о готовности буфера /или другого ЭД/ к обработке можно командой

BIS_# 200, @BUFER+4,

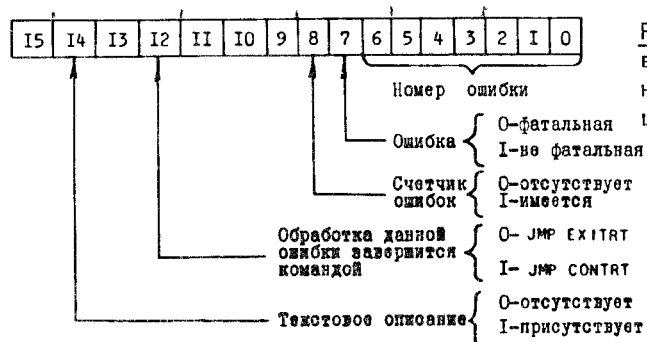


Рис.3. Формат первого слова дополнительной информации об ошибке.

где BUFFER - идентификатор начала краткого описания данного ЭД. Эта команда заносит единицу в 7 разряд слова состояния ЭД.

Аналогично требование продолжить работу канала после обработки буфера записывается командой

BIS $_#400$, @ BUFFER+4,

которая устанавливает единицу в 8 разряд того же слова состояния ЭД.

Если во время работы модуля возникает такая ситуация, о которой необходимо сообщить оператору, либо работа не может быть продолжена вследствие, например, отсутствия свободного буфера, аварии оборудования и т.п., то модуль должен завершить свою работу командой JMP FAILRT, где FAILRT = 4 /передача управления служебной программе/. В этом случае требуется сформировать дополнительную информацию об ошибке. Эта информация должна помещаться в трех последовательных словах и содержать:

- 1/ индивидуальный номер ошибки и описательные признаки, формат этого кода показан на рис.3;
- 2/ адрес текстового описания ошибки;
- 3/ счетчик числа повторений данной ошибки; при возникновении ошибки модуль увеличивает на единицу содержание младшего байта счетчика.

Второе и третье слова нужны только в том случае, когда в первом слове присутствуют соответствующие признаки. Служебной программе адрес сформированной информации передается в стеке.

5.2. Адрес входа в функциональную часть определен содержанием первого слова таблицы настройки /см. рис.1, п.7/. Помимо функционального, модуль имеет еще 4 управляющих входа /п.5/, которые размещаются непосредственно перед функциональным подряд, снизу вверх /в направлении убывания адресов/. Названия и назначение этих входов следующие:

- 1/ SET - подготовка канала к работе;
- 2/ UNSET - прекращение работы канала;
- 3/ RESUME - разрешение продолжить /или начать/ обмен информацией;
- 4/ SUSPEND - приостановка обмена информацией.

Первая управляющая часть программного модуля должна содержать алгоритм инициализации оборудования канала и функциональной части, который выполняется один раз после загрузки УПМ в память перед началом работы /началом эксперимента/. Вторая управляющая часть выполняет обратные действия перед завершением работы.

Третья и четвертая части управления позволяют временно приостановить и продолжить работу канала.

5.3. Проиллюстрируем описание структуры программного модуля конкретным примером. На рис.1 приведен текст на языке MACRO-II программного модуля, который должен регистрировать двухмерный спектр, считывая описание каждого события из регистров двух амплитудных кодировщиков АК-1024^{18/}, размещенных в крейте с контроллером^{13/}. Отметим, что при написании УПМ в качестве адресов регистров контроллера (CSR, DMR и DHR) должны использоваться условные адреса: 164000, 164002 и 164004 соответственно. Данный модуль работает с двумя буферами, причем в первый из них (BUF 1) заносятся описания последовательных событий, а во втором (BUF 2) накапливается интегральный /одномерный/ спектр /гистограмма/ от первого кодировщика. После занесения каждой пары кодов амплитуд в буфер программа запоминает номер /адрес/ текущей свободной строки буфера в его описании /пятая строка описания первого ЭД/. Когда буфер будет заполнен, программа по команде

BIS $_#600$, @ BUF1+4

(BIS # EVENT+DEMAND, @ BUF1+4)

запишет единицы в 7 и 8 разряды слова состояния данного ЭД и закончит работу. Работа канала /и данного модуля/ будет продолжена управляющими программами после освобождения буфера программами обработки.

В случае переполнения 16-разрядного слова памяти при построении гистограммы /спектра/ программа приостанавливает регистрацию данных и выводит сообщение (SPECTRUM $_OVERFLOWED$ $_IN_EXAMPLE(AKAK)$) путем обращения к служебной программе командой JMP FAILRT.

Управляющие части модуля в этом примере выполняют следующие действия:

SET - очистка буфера BUFFERB для интегрального спектра;


```

SET:   MOV   BUF2, R0
1$:   CLR   (R0)+
      CMP   R0, BUF2+2
      BMI   1$ ; CLEANING OF THE "BUF2"

RESUME:
BIL2: MOV   #AOF26, @#CMND
      RTS   PC
      ;

SUSPND:
UNSET:
BIL3: MOV   #AOF24, @#CMND
      RTS   PC ; END OF COMPLETION ROUTINE
      ;

TABLE: .SETTBL 2,2,B2L1 ; n.7
      ;

MSG:   .WORD 40201 ; TEXT & WARNING & EXITRT ; n.8
      MSTXT
      .ASCIZ /SPECTRUM OVERFLOWED IN EXAMPLE(AKAK)/
      ;

.END B2L1 ; n.1

```

ЛИТЕРАТУРА

1. Балука Г. и др. ОИЯИ, 10-12960, Дубна, 1979.
2. CAMAC. A Modular Instrumentation System for Data Handling. Revised Description and Specification. EUR 4100e, 1972.
3. CAMAC. Organisation of Multi-Crate Systems. Specification of the Branch Highway and CAMAC Crate Controller Type A. EUR 4600e, 1972.
4. PDP II. Peripherals Handbook. DEC, Maynard, Massachusetts, 1975.
5. Ананьев В.Д. и др. ОИЯИ, P3-10888, Дубна, 1977.
6. Наумов Б.Н., Боярченков М.А., Кабелевский А.А. Управляющий вычислительный комплекс СМ-3. ПиСУ, 1977, № 10, с.12-15.
7. Борисенко В.Д., Плотников В.В., Талов И.Л. Микро-ЭВМ "Электроника-60". Электронная промышленность, № 10 /70/, 1978, с.20-21.
8. RT-11 System Reference Manual (DEC-11-ORUGA-C-D). DEC, Maynard, Massachusetts, 1975.
9. RT-11 Software Support Manual (DEC-11-ORPGA-B-D). DEC, Maynard, Massachusetts, 1975.
10. Балука Г., Саламатин И.М., Хрыкин А.С. ОИЯИ, 10-12545, Дубна, 1979.
11. Балука Г., Саламатин И.М., Хрыкин А.С. ОИЯИ, 10-12546, Дубна, 1979.
12. Nuclear Enterprises. CAMAC Catalogue. Bulletin No.108. June 1977. Printed by PSG Ltd. East Kilbridge, Scotland.

13. Елизаров О.И., Жуков Г.П., Мячев А.А. ОИЯИ, 11-8396, Дубна, 1974.
14. КАМАК - системы автоматизации в экспериментальной биологии и медицине. Под ред. Ю.Е.Нестерихина. Наука, Новосибирск, 1978.
15. Bal F. et al. CC11 (CAMAC Crate - PDP 11 Interface) Type 116. CERN-NP. CAMAC Note No.43-00, Geneva, 1972.
16. Барабаш И.П. и др. ОИЯИ, 11-8522, Дубна, 1975.

Рукопись поступила в издательский отдел
14 июля 1980 года.

ТЕМАТИЧЕСКИЕ КАТЕГОРИИ ПУБЛИКАЦИЙ
ОБЪЕДИНЕННОГО ИНСТИТУТА ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

Индекс	Тематика
1.	Экспериментальная физика высоких энергий
2.	Теоретическая физика высоких энергий
3.	Экспериментальная нейтронная физика
4.	Теоретическая физика низких энергий
5.	Математика
6.	Ядерная спектроскопия и радиохимия
7.	Физика тяжелых ионов
8.	Криогеника
9.	Ускорители
10.	Автоматизация обработки экспериментальных данных
11.	Вычислительная математика и техника
12.	Химия
13.	Техника физического эксперимента
14.	Исследования твердых тел и жидкостей ядерными методами
15.	Экспериментальная физика ядерных реакций при низких энергиях
16.	Дозиметрия и физика защиты
17.	Теория конденсированного состояния
18.	Использование результатов и методов фундаментальных физических исследований в смежных областях науки и техники