

Виноградов В. Б. и др.
Б1-10-85-837

+



ОБЪЕДИНЕННЫЙ ИНСТИТУТ ЯДЕРНЫХ ИССЛЕДОВАНИЙ

999/86

344.17 + Ц 840

Б 1-10-85-837

ДЕПОНИРОВАННАЯ ПУБЛИКАЦИЯ

Дубна 19 85

B1-10-25-857

ОБЪЕДИНЕННЫЙ ИНСТИТУТ ЯДЕРНЫХ ИССЛЕДОВАНИЙ

Лаборатория ядерных проблем

В.Б.Виноградов, Ю.А.Кульчицкий^ж, А.С.Курилин^ж, В.Г.Одинцов

ПРОГРАММА ГЕОМЕТРИЧЕСКОЙ РЕКОНСТРУКЦИИ СОБЫТИЙ В ЭКСПЕРИМЕНТЕ
ПО ИССЛЕДОВАНИЮ ГИПЕРЗАРЯДОВООБМЕННЫХ ПРОЦЕССОВ НА УСТАНОВКЕ
ГИПЕРОН

Объединенный институт
ядерных исследований
Библиотека

Регистрация документа
в библиотечной системе
.. 21 / 11 1985 г.

Дубна, 1985 г.

^ж Институт физики АН БССР, Минск

/ по крив.

Аннотация

Описана программа геометрической реконструкции событий, зарегистрированных с помощью установки ГИПЕРОН при исследовании гиперзарядовообменных процессов при энергии 12 ГэВ на серпуховском ускорителе. Программа написана на языке ФОРТРАН, занимает объем оперативной памяти ЭВМ З20К и ориентирована на ЭВМ класса ЕС-1060. Алгоритм, заложенный в программу, использует особенности регистрируемых установкой ГИПЕРОН событий и содержит новые элементы по сравнению с ранее описанными алгоритмами. С помощью программы было обработано свыше миллиона событий со средней скоростью 0,2 сек./событие (на ЭВМ ЕС-1060). Применение данной программы при изучении гиперзарядовообменных процессов показало ее адекватность поставленной задаче.

В В Е Д Е Н И Е

Для исследования процессов образования и распадов частиц на Серпуховском ускорителе была создана установка "Гиперон" /1/ и разработано соответствующее математическое обеспечение, описанию которого посвящен ряд работ /2-6/.

В данной работе описывается программа геометрической реконструкции событий *HYPER*, использованная для обработки событий с обменом гиперзарядом /7-8/. Изучение способов реконструкции событий /9-12/ показало, что используемые методы и программы не являются универсальными, а, как правило, создаются и используются для каждой конкретной установки и зависят от многих условий (детектирующая аппаратура, фоновые условия, интенсивность и состав падающего потока частиц и т.п.).

Программа *HYPER* была создана на основе пятилетнего опыта работы на установке "Гиперон" и имеет специфические возможности и особенности. Программа предназначена для поиска траекторий заряженных частиц, зарегистрированных установкой, определения их геометрических и кинематических параметров, отбора по определенным критериям и записи на ЛСР (ленту суммарных результатов) отобранных событий для дальнейшей обработки. Обработка каждого события состоит из следующих последовательных шагов:

1. чтение и декодировка
2. распознавание треков частиц
3. определение геометрических параметров треков
4. проверка восстановленного события по заданным критериям
5. восстановление импульсов заряженных частиц
6. запись информации о событии на ЛСР.

Кроме этого в процессе обработки каждого события накапливается следующая информация:

1. данные о работе трековых детекторов,
2. вычисляются отклонения срабатываний на трековых детекторах от трека и вычисляется среднее отклонение,
3. вычисляются углы поворотов для камер, регистрирующих Y проекцию.

После краткого описания установки и общей схемы работы программы подробно описывается каждый шаг обработки события, далее описаны параметры, определяющие условия работы, и важнейшие подпрограммы, входящие в состав программы *HYPER*.

ОПИСАНИЕ УСТАНОВКИ

На рис. I показано схематическое изображение установки, использованной в гиперзарядовообменном эксперименте. Система координат связана с осью пучка в ускорителе и центром спектрометрического электромагнита МС-12 /13/. Направление оси X - вверх, направление оси Y - вправо, если смотреть по пучку. Центр системы координат привязан к МС-12, начало $Z = 0$ находится в центре электромагнита, плоскость ZOY совпадает с плоскостью орбиты пучка в ускорителе.

Трековые детекторы установки расположены так (рис. I), чтобы регистрировать отдельные прямолинейные участки траекторий заряженных частиц. Моделирование исследуемых процессов с помощью специальной программы /6/ показало, что координатное отклонение от прямой линии из-за многократного рассеяния в условиях установки не превышает 3 мм /5/. Детекторы, регистрирующие отдельные прямолинейные участки траекторий заряженных частиц, объединяются в блоки. На каждом из таких блоков происходит поиск прямолинейных треков. Так, в пучковом спектрометре (рис. I) ПК-1 и ПК-2 объединены в блок, по которому определяется траектория влёта частиц в электромагнит СП-129,

а ПК-3 и ПК-4 объединены в блок, по которому определяется траектория частиц за электромагнитом СП-129. Детекторы спектрометра вторичных частиц (рис. 1) объединены в следующие блоки. Блок за электромагнитом МС-12 состоит из ИПК-4 и ИПК-5. Особо выделен блок детекторов от мишени М до электромагнита МС-12, он состоит из ПК-5, ИПК-1, ИПК-2, ПК-7 и ИПК-3. Между детекторами этого блока располагались два черенковских счетчика C_4 и C_5 , на веществе которых может произойти значительное рассеяние, поэтому блок разбит на два подблока: ПК-5, ИПК-1 и ИПК-2, ПК-7, ИПК-3. Поиск траекторий ведется по полному блоку, а параметры траекторий затем уточняются отдельно для каждого подблока. Это сделано для более точного определения параметров траекторий до черенковских счетчиков и после них.

В состав установки Гиперон входит ЭВМ ЕС-1010, она осуществляет прием, контроль и запись на магнитную ленту информации о регистрируемых событиях. Массивы данных о событиях, зарегистрированных установкой во время сброса пучка на мишень, принимаются ЕС-1010, объединяются и записываются в виде записей на диске ЕС-1010. Затем дисковые записи объединяются в один или несколько массивов, которые записываются на магнитную ленту. Эти массивы образуют записи на магнитной ленте. Кроме записей данных на магнитную ленту записываются *SPILL*-записи, содержащие информацию о каждом сбросе пучка на мишень (мониторные числа, время и дата сброса, номер записи и т.п.). В начале набора событий определенного типа и в конце его на ленту записываются несколько меток *EOF* (конец файла). Такие наборы событий на магнитной ленте, разделенные метками *EOF*, нумеруются и называются *RUN*'ами. Записанные таким образом магнитные ленты называются лентами данных (ЛД), они содержат информацию, обрабатываемую программой геометрической реконструкции *HYPER*.

ОБЩАЯ СХЕМА РАБОТЫ ПРОГРАММЫ

На рисунках 2 и 3 изображена блок-схема программы *HYPER*. Во всех подпрограммах, обведенных на рис. 2 пунктирной линией, задаются константы, определяющие работу программы. Они выполняются в начале работы. Все константы могут считываться или с перфокарт или с файлов на магнитных дисках. После определения условий полная информация о работе программы выдается на печать. Информация об условиях поиска треков на блоках и о составе самих блоков печатается подпрограммами *BLPRIN* и *BLHEAD*. Далее вызывается подпрограмма *ZERRO*, в которой происходит зануление некоторых массивов и занесение в них необходимых для дальнейшей работы величин. Если обработка начинается не с начала *RUN'a*, то вызываются подпрограммы *SKIPDA* и *SKIPDS*, которые пропускают необходимое число уже обработанных или записанных записей на ЛД или ЛСР. После выполнения всех указанных выше подпрограмм вызывается подпрограмма *BLAN*. В результате обращения к подпрограмме *BLAN* происходит последовательная обработка одного *RUN'a*, записанного на ленте данных. В начале работы подпрограмма *BLAN* вызывает подпрограмму *SKIEOF* для пропуска нескольких подряд стоящих меток *EOF* в начале *RUN'a*. Затем организован цикл, в котором происходит обращение к подпрограмме *ES-IOIO*, цикл оканчивается в случае прочтения метки *EOF* на ЛД или в случае получения от оператора ЭВМ приказа о завершении работы программы. При завершении цикла управление передается программе *HYPGEN*, в которой формируется и печатается итоговая информация о работе программы и вызывается подпрограмма *STTPR*, печатающая информацию о работе трековых детекторов.

В результате работы подпрограммы *ES-IOIO* считывается один рекорд с ленты данных и прочитанная информация преобразуется к виду, более удобному для дальнейшей декодировки. Затем, в подпрограмме *BLAN* определяется, какой рекорд считан с ЛД и, если рекорд содержит данные о событиях, то вызывается *DECOD*. *DECOD* представляет собой комплекс подпрограмм, декодирующих всю информацию о событиях, записанных на ЛД. После полной декодировки информации о событии вызывается подпрограмма *STTI*, в которой накапливается информация о работе трековых детекторов и, если требуется, вызывается подпрограмма *EVEPR*, которая печатает всю декодированную информацию о данном событии. Далее, вызывается подпрограмма *EVEN*, в которой последовательно осуществляется обработка одного события. В подпрограмме *EVEN* осуществляется проверка, удовлетворяет ли событие заданным критериям и если удовлетворяет, то вызывается подпрограмма *REDAT*. Подпрограмма *REDAT* управляет поиском треков на блоках (рис.3). При одном вызове подпрограммы *REDAT* проводится поиск треков на заданном блоке в заданной проекции. В начале из подпрограммы вызывается подпрограмма *SPARPA*, в которой происходит замена нескольких подряд сработавших элементов трекового детектора одним, в середине сработавшей группы элементов. В этой же подпрограмме убираются группы, состоящие из большого числа подряд сработавших элементов (обычно ≥ 4). Далее вызывается подпрограмма *COORD*, в которой номера сработавших элементов детекторов переводятся в координаты общей системы координат спектрометра. Затем, в зависимости от заданных условий поиска треков для данного блока, вызывается одна из подпрограмм *I-IO* (рис.2). Если условия поиска треков таковы, что на каждом детекторе можно явно выделить область, по которой могут проходить треки, то вызывается подпрограмма *ZONCHA*, в которой отбрасываются координаты срабатываний, лежащие вне указанной области. Далее выз-

вается подпрограмма *ABCALC*, в которой из массива координат сработавших элементов детекторов выбирается набор, который предположительно принадлежит треку и вычисляются параметры этого кандидата в трек. Затем вызывается подпрограмма *CALXMI*, в которой определяется из скольких срабатываний состоит данный кандидат в треки. Если данный набор координат удовлетворяет заданным критериям, то вызывается подпрограмма *FITAB*, в которой методом наименьших квадратов находятся параметры трека. Далее, если данный блок состоит из подблоков, вызывается подпрограмма *SUBBL*, в которой блок разбивается на подблоки и вызывается подпрограмма *FITAB* для вычисления параметров треков на подблоках. Когда параметры треков на блоках и подблоках найдены, вызывается подпрограмма *AINST*, в которой эти параметры сохраняются для дальнейшего использования. Если это определено условиями работы программы, то из подпрограммы *CALXMI* вызывается подпрограмма *SHIFT*, в которой вычисляются и накапливаются отклонения координаты сработавшего на трек элемента на детекторе от вычисленной координаты трека на детекторе. После нахождения очередного трека вызываются, если это задано в условиях поиска треков на данном блоке, последовательно подпрограммы *DELSP* и *DELCH*. В этих подпрограммах координаты уже вошедших в трек сработавших элементов детекторов убираются из массива координат и детекторы, на которых не осталось сработавших элементов убираются из дальнейшего поиска треков. Затем опять вызывается подпрограмма *ABCALC* и происходит поиск других треков. Информация о числе треков, найденном на данном блоке, запоминается при вызове подпрограммы *STLOST*. Если по условиям работы программы указано, что необходимо вычислять углы поворотов трековых детекторов, то при нахождении только одного трека на данном блоке вызывается подпрограмма *ROTACH*, в которой вычисляются и сохраняются данные, необходимые для определения углов поворотов детекторов /2,3/. Информация

о числе найденных треков передается из *REDAT* в *EVEN*S, и если найдено необходимое число треков на данном блоке, вызывается подпрограмма *REDAT* для поиска треков на других блоках.

После поиска треков на всех блоках вызывается подпрограмма *BLOM* в которой событие проверяется по заданным условиям. Из нее вызываются подпрограммы *TRANS* для проверки стыковки треков на разных блоках и *RENAME* для перестановки треков, затем подпрограммы восстанавливающие импульсы заряженных частиц *PREST*, *TCALC* /4/. В подпрограмме *RBOOK* информация о данном событии заносится в гистограммы.

После полной обработки события, если оно удовлетворяет предъявленным требованиям, вызывается подпрограмма *DSTFOR*, в которой информация об обработанном событии упаковывается и записывается на ЛСР.

Если событие на каком-либо этапе обработки не удовлетворяет предъявленным требованиям, то обработка этого события прекращается и происходит возврат из подпрограммы *EVEN*S к подпрограмме *DECOD* для декодировки следующего события. Обработка прекращается или после обработки заданного числа *RUN*'ов или по команде оператора ЭВМ прекратить работу программы.

ЧТЕНИЕ И ДЕКОДИРОВКА

При обращении к подпрограмме *BLAN* происходит обработка одного *RUN*'а событий, записанных на ленте данных в процессе набора статистики на установке Гиперон.

В начале работы подпрограммы вызывается подпрограмма *SKIEOF*, для пропуска меток "конец файла", записанных в начале данного *RUN*'а. Затем вызывается подпрограмма *ES-IOIO*, в которой специальными подпрограммами (*BUFFER* или *ZONA*) /14,15/ считывается один рекорд с ЛД и информация преобразуется из формата *ES-IOIO* в формат *ES* ЭВМ.

Считанная и преобразованная информация заносится в *COMMON/IDIS/IB* (2048). Далее в подпрограмме *BLAN* определяется, какой рекорд считан с ленты данных. Если прочитан информационный рекорд (*SPILL*) то декодируется записанная в нем информация, которая заносится в следующие *COMMON*-блоки: в *NSPI* - номер *RUN*'а и номер *SPILL*'а, в *REGTIME* дата и время записи на ЛД и в *NEWMON* - мониторные числа. Если же прочитан рекорд, содержащий информацию о зарегистрированных событиях, то вызывается комплекс подпрограмм *DECOD/16/*.

Как уже отмечалось, один рекорд данных на магнитной ленте состоит из одного или нескольких дисковых рекордов. Дисковый рекорд, в свою очередь, может состоять из нескольких массивов информации, каждый из которых содержит данные об одном событии. Массив информации о событии содержит данные о типе данного события, мониторных числах, о работе трековых детекторов и контрольные слова. В контрольных словах должны содержаться определенные значения, если в каком-либо событии записанные контрольные слова не соответствуют заданным, это означает, что при записи события произошел сбой в электронной аппаратуре установки.

В *DECOD* определяется, сколько дисковых рекордов и сколько массивов, описывающих события, содержится в рекорде, прочитанном с ЛД. Из всех данных выбирается первый массив информации о событии и начинается его декодировка. Проверяются контрольные слова, и если они записаны правильно, то декодируется остальная информация. Мониторные числа для данного события заносятся в *COMMON/NEWMON/*. Информация о годоскопах, в *COMMON/NGD/NGD(5,2),NGOD(5,2,16)*, где *NGD(I, NU)* - число сработавших элементов в годоскопе с номером *I*, в проекции *NU*, *NGOD(I, NU, K)* - номер сработавшего элемента, *K* - номер срабатывания. Информация о сработавших каналах пропорциональных и искровых камер заносится в *COMMON/GX/XX(20,80),NSP(80)*, где *XX(Y,I)* -

номер сработавшего канала (J -ое срабатывание) на I -ой камере, $NSA(I)$ число срабатываний на I -ой камере.

На ЛД могут быть записаны события разного типа (триггерное событие, реперное, частица прошла через всю установку без взаимодействия и т.п.). Каждый тип события отмечается в массиве информации о событии своим индексом, этот индекс заносится в $COMMON/JP/INDEX$. После декодировки этой информации вызывается подпрограмма $STTJ$, в которой накапливается информация о работе пропорциональных и искровых камер. Если информация о данном событии должна быть распечатана, вызывается подпрограмма $EVEPR$. Далее вызывается подпрограмма $EVEN5$, в которой проводится дальнейшая обработка. После завершения обработки или в случае обнаружения сбоев электроники (контрольные слова) декодируется следующее событие, записанное в данном рекорде на ЛД. Подпрограмма $DECOD$ заканчивает свою работу после декодировки всех событий, записанных в данном рекорде на ЛД и происходит возврат в подпрограмму $BLAN$. Из подпрограммы $BLAN$ опять вызывается подпрограмма $ES-IOIO$ и весь процесс повторяется до тех пор, пока на ЛД не будет прочитана метка EOF или не поступит команда от оператора ЭВМ закончить работу программы.

РАСПОЗНАВАНИЕ ТРЕКОВ ЧАСТИЦ

Используемый в программе метод распознавания треков частиц заключается в следующем ^{/9,10/}. На блоке камер выбираются две камеры, имеющие Z координаты Z_i и Z_j соответственно, через координаты от сработавших элементов на камере i и камере j проводится прямая линия

$$y = Az + B, \quad A = \frac{y_{kj} - y_{ei}}{z_j - z_i}, \quad B = \frac{y_{ei} z_j - y_{kj} z_i}{z_j - z_i}, \quad (I)$$

где A и B - параметры прямой, y_{kj} - координата k -го сработавшего элемента на камере j и y_{li} - координата l -го сработавшего элемента на камере i . Если параметры A и B находятся в задаваемых для каждого блока пределах, то прямая проводится через все детекторы блока. На каждом детекторе ищутся координаты ближайшего к прямой срабатывания, лежащего в интервале Δy (обычно ± 3 мм). Если число сработавших детекторов больше или равно N мин., то считается, что найденный набор координат срабатываний является треком. Этот набор координат заносится в отдельный массив и используется для вычисления параметров треков. Координаты от срабатываний, вошедшие в трек, вычеркиваются из общего массива координат и в поиске других треков не участвуют. Если число сработавших детекторов в блоке меньше N мин., то проводится прямая через другую пару координат на камерах i и j , затем осуществляется переход к другой паре камер в блоке. На первом шаге в качестве i -ой и j -ой камеры выбираются крайние камеры в блоке, затем происходит последовательное смещение камер от краев.

В программе, после раскодировки каждого события в подпрограмме *DECOD*, вызывается подпрограмма *EVENIS*, в которой обрабатывается одно событие. Из подпрограммы *EVENIS* для поиска треков на блоках вызывается подпрограмма *REDAT*. В подпрограмме *REDAT* осуществляется поиск треков на одном блоке. Вначале вызывается подпрограмма *SPARPA*, в которой происходит замена нескольких подряд сработавших каналов камер на один средний по формуле

$$\bar{n}_{ik} = n_{ik} + \frac{z_{ik} - 1}{2}, \quad (2)$$

где i - номер срабатывания на k -ой камере, n_{ik} - номер первого канала в группе z_{ik} подряд сработавших каналов. Далее вызывается подпрограмма *COORD*, в которой номера сработавших элементов камер преобразуются в координаты

$$y_{ik} = h_k S_k \bar{n}_{ik} + h_{ik} j_k + y_k^0 \quad (3)$$

Здесь в Y_k -координата i -го срабатывания на K -ой камере, h_k -шаг намотки проволок камеры, $S_k = \pm 1$ -направление считывания проволок, y_k^0 -координата первой проволоки, j_k -угол поворота камеры вокруг оси Z . Аналогично, но без учета поворотов j_k , определяются h_{ik} -координаты срабатываний. Затем в подпрограмме *REDAT* или в одной из подпрограмм I-IO (рис. 3) задаются параметры треков для данного блока и вызывается подпрограмма *ABCALC*.

В подпрограмме *ABCALC* вычисляются параметры A и B для каждой пары координат срабатываний и проверяется, удовлетворяют ли параметры A и B заданным критериям и, если удовлетворяют, то вызывается подпрограмма *CALXMI*. В этой подпрограмме определяются координаты ближайших срабатываний к прямой с параметрами A и B . Если число таких срабатываний больше или равно заданной величине, то вычисляются параметры трека.

ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ТРЕКОВ

Как показали расчеты, параметры треков заряженных частиц в условиях спектрометра "Гиперон" могут вычисляться простым способом ^{/5/}. Треки в каждом блоке и для каждой проекции аппроксимируются прямыми линиями с параметрами A и B . Для определения наилучших значений параметров A и B для каждого трека, в каждой проекции минимизируется функционал

$$\chi^2 = \sum_{i=1}^N \frac{(y_i - Az_i - B)^2}{\sigma_i^2}, \quad (4)$$

где y_i - значение координаты трека на i -ом трековом детекторе; z_i - значение Z -координаты i -го детектора; $\sigma_i = h_i/\sqrt{2}$ - пространственное разрешение i -го детектора; A и B - искомые параметры, N - число детекторов в блоке, сработавших на трек.

После отбора в подпрограмме *CALXMI* "трека" вызывается подпрограмма *FITAB*, в которой и вычисляются значения A и B для данного набора координат.

Кроме самих параметров A и B вычисляется матрица ошибок DA, DB, DAB , величина χ^2 и *ZBAS* база, на которой определен трек, т.е. расстояние по Z координате между первым и последним детектором в блоке.

Если по условиям работы программы не требуется двойного фитирования, то найденные параметры считаются параметрами трека. Если требуется двойной фит, то найденные параметры используются для проведения прямой через детекторы в блоке и производится отбор координат срабатываний в детекторах. Координаты должны лежать не дальше, чем $4\sigma_i$ от прямой. После отбора таких координат они используются для повторного фитирования в подпрограмме *FITAB*. То есть практически из фитирования убираются координаты срабатываний, лежащие дальше, чем $4\sigma_i$ от прямой. Найденные параметры треков заносятся в *COMMON/AFIN/NTR(NB, NU), A(NB, NT, NU), B(NB, NT, NU), DA(NB, NT, NU), DB(NB, NT, NU), DAB(NB, NT, NU), SIG(NB, NT, NU), NPOINT(NB, NT, NU)* и *COMMON/ZBAS/ZBAS(NB, NT, NU)* где NB - номер блока, NT - номер трека, NU - номер проекции, NTR - число треков на данном блоке в данной проекции, $SIG = \chi^2$ - для NB - блока, трека NT , проекции NU , $NPOINT$ - число сработавших детекторов, $ZBAS$ - база трека соответственно.

После определения параметров треков, в случае необходимости, вызывается подпрограмма *SHIFT*, в которой накапливаются отклонения координат срабатываний на детекторах от треков. Если данный блок разбит на подблоки, то после выбора кандидатов в треки в подпрограмме *CALXMI* вызывается подпрограмма *SUBBL*, в которой найденные координаты с помощью *FITAB* фитируются отдельно для каждого подблока.

123

ПРОВЕРКА ВОССТАНОВЛЕННОГО СОБЫТИЯ
ПО ЗАДАНЫМ КРИТЕРИЯМ

После каждого обращения к подпрограмме *REDAT* в подпрограмме *EVENIS* проверяется, найден ли хотя бы один трек на блоке камер. Если на блоке не найдено ни одного трека, то дальнейшая обработка события прекращается.

При этом информация об отсутствии трека на блоке заносится в массив потерь. Из подпрограммы *REDAT* вызывается подпрограмма *STLLOST*, в ней в *COMMON/LOST/LOST(NB,I,NU)* при отсутствии *I*-го трека к массиву *LOST* добавляется единица (здесь *NB* - номер блока, *NU* - номер проекции).

Если на блоке найден хотя бы один трек, то обработка продолжается. Когда на всех блоках, во всех проекциях найдены треки, из подпрограммы *EVENIS* вызывается подпрограмма *BLOM*.

Подпрограмма *BLOM*, используя информацию о найденных треках проверяет события по заданным критериям. Эта подпрограмма должна изменяться в зависимости от исследуемой физической реакции, так как критерии отбора событий меняются. При исследовании реакций $\pi^+p \rightarrow K^+X$ подпрограмма *BLOM* отбирала события следующим образом.

Требования к числу треков на блоках.

Только по одному треку должно быть:

а) в *У* проекции на :

- 1) блоке до магнита СП-129,
- 2) после магнита СП-129 на входе в мишень,
- 3) на подблоке при входе в магнит МС-12,
- 4) на блоке за МС-12,

б) в *Х* проекции на подблоке при входе в магнит МС-12.

Если на подблоке за мишенью в *У* проекции не было трека, то выход из мишени определялся по большому блоку от мишени до магнита.

Если на подблоке при входе в магнит МС-12 в X проекции нет трека, то вход в магнит определялся по большому блоку от мишени до магнита МС-12. Далее проверялась сшивка треков в Y проекции для треков: до и после магнита МС-12, до и после магнита СП-129. Требовалось, чтобы величина

$$\Delta y = (A_i - A_j) Z_M + (B_i - B_j) \quad (5)$$

где Z_M — координата центра магнита, A_i и A_j , B_i и B_j — параметры треков до и после магнита соответственно, была меньше, чем заданная величина Δ .

Если треки сшивались, то событие считалось "хорошим". Если событие не удовлетворяло перечисленным выше требованиям, но на каждом блоке был найден хотя бы один трек, то оно считалось "плохим". Информация о всех событиях, прошедших через подпрограмму *BLDM* записывалась на *ЛСР* первого поколения, снабженного меткой "плохое" или "хорошее" событие. Для "хороших" событий перед их записью на *ЛСР* вызывалась подпрограмма *PREST*, в которой восстанавливался импульс заряженной частицы, прошедшей через после магнита МС-12.

ВОССТАНОВЛЕНИЕ ИМПУЛЬСА ЗАРЯЖЕННЫХ ЧАСТИЦ

При восстановлении импульса заряженных частиц, прошедших через магнитное поле магнита МС-12, использовалась методика подробно описанная в работе /4/. В программу геометрической реконструкции вводились заранее подсчитанные коэффициенты при полиномах Чебышева. В начале работы программы вызывалась подпрограмма *CREAD*, в которой с дисков считывались и записывались в *COMMON* блоки значения коэффициентов и границы области, для которых они подсчитаны. Импульс частицы восстанавливался по формуле:

$$\frac{CONST}{P} = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sum_{k=1}^{N_3} \sum_{l=1}^{N_4} \sum_{m=1}^{N_5} C_{ijklm} T_i(\zeta_1) T_j(\zeta_2) T_k(\zeta_3) T_l(\zeta_4) T_m(\zeta_5) \quad (6)$$

где C_{ijklm} - коэффициенты при полиномах Чебышева, $T_i(\zeta)$ - полином Чебышева i -го порядка, от-нормированной в интервале ± 1 переменной ζ .

Нормированным переменным соответствовали: $\zeta_1 \rightarrow X_0$, $\zeta_2 \rightarrow Y_0$ координаты точки траектории частицы, соответствующие середине мишени; $\zeta_3 \rightarrow X$, $\zeta_4 \rightarrow Y$ - координаты, соответствующие точке входа частицы в магнит; $\zeta_5 \rightarrow \varphi$ - проекция угла отклонения частицы, прошедшей через магнит, на плоскость YOZ . Нормировка проводилась по формуле:

$$\zeta_i = \frac{2x_i - X_i^{min} - X_i^{max}}{X_i^{max} - X_i^{min}} \quad (7)$$

где X_i^{min} , X_i^{max} граница изменения переменной x_i .

Для каждой частицы, для которой вычислялся импульс, вызывалась подпрограмма *PREST*, в которой нормировались значения X_0 , Y_0 , X , Y и φ и вычислялся импульс по формуле (6). Значение импульса заносилось в блок *COMMON/РМОН/РМОМЕ* и использовалось далее, для вычисления кинематических характеристик событий.

Импульсы падающих на мишень частиц определялись по формуле

$$P = \frac{1}{C\theta + D} \quad (8)$$

где P - импульс, θ - поправка к среднему углу отклонения частиц пучка в электромагните СП-129, C и D - константы. Для определения констант C и D используется следующая процедура.

1. Обработывались события, в которых частицы пучка прошли без взаимодействия через всю установку.

2. Для частиц, прошедших без взаимодействия через всю установку, используя коэффициенты Чебышева восстанавливался их импульс (по магниту МС-12).

3. Для определения C и D минимизируется функционал

$$\chi^2 = \sum_{i=1}^m \left(\frac{1}{\rho_i} - C \theta_i - D \right)^2 \quad (9)$$

где ρ_i — импульс i -ой частицы имеющей поправку к углу отклонения в электромагните СП-129 θ_i . Определенные таким образом константы C и D используются для восстановления импульсов частиц пучка до взаимодействия в мишени.

ЗАПИСЬ ИНФОРМАЦИИ О СОБЫТИИ НА ЛСР

Информация об обработанном событии записывается на ЛСР в подпрограмме *DSTFOR*. На ЛСР записывается следующая информация. Параметры найденных треков на блоках, их ошибки и χ^2 трека. Базы, по которым определен трек, и число точек, вошедших в трек. Мониторные числа, номер *RUN'a*, номер события в *RUN'e*, номер *SPILL'a* соответствующего данному событию, дата и время записи, индекс данного события. Упаковываются и записываются данные о сработавших элементах годоскопов. Восстановленные импульсы заряженных частиц и их ошибки. Вся эта информация используется при дальнейшей обработке.

ЗАДАНИЕ УСЛОВИЙ РАБОТЫ ПРОГРАММЫ

Задание условий обработки событий происходит в следующих подпрограммах:

- *HYPGEN*
- *CHAMBS* задается информация о трековых детекторах установки.
- *BLOCKS* — задается информация, определяющая блоки камер.
- *DBOOK* — определяет гистограммы.
- *FORDEC* — задает параметры для работы подпрограммы *DECOD*
- *CREAD* — читаются коэффициенты при полиномах Чебышева.

В главной программе задается следующее:

Перфокарта или ее образ на диске или магнитной ленте (далее ПК) № 1:

1. индекс событий, которые не будут обрабатываться,
2. ключ записи на ЛСР, если ≤ 0 , то нет записи на ЛСР,
3. восстанавливается ли импульс вторичных частиц, если ≤ 1 , то нет
4. каждое N -ое событие распечатывается.

ПК № 2:

1. обработка с учетом углов поворота U камер или без, $\neq 0$ без учета
2. подсчитываются углы поворота U камер или нет, $\neq 0$ нет,

ПК № 3:

1. сколько RUN -ов обрабатывается,
2. сколько треков ищется в каждом событии.

ПК № 4:

- 1,2. номера двух камер, при отсутствии срабатываний на которых событие не обрабатывается.

Далее 4-ПК определяют условия начала обработки.

ПК № 5:

- 1,2. сколько событий в данном RUN 'е уже обработано,
3. число событий в данном RUN 'е со сбоями электроники.

ПК № 6:

1. сколько рекордов надо пропустить на ленте данных,
2. сколько EOF пропущено на ленте данных,
3. сколько рекордов прочитано с ошибкой.

ПК № 7:

1. сколько рекордов данных прочитано,
2. сколько рекордов $SPILL$ прочитано,
3. сколько рекордов дисковых прочитано.

ПК № 8:

1. сколько записано в обрабатываемом RUN' уже записано на ЛСР,
2. сколько событий в обрабатываемом RUN' уже записано на ЛСР,
3. сколько EOF в обрабатываемом RUN' уже записано на ЛСР.

ПК № 9:

1. вычисляются ли сдвиги камер, если $\neq 0$, то нет,
2. гистограммируются ли точности камер, если $\neq 0$, то нет,
3. есть ли гистограммы импульсов и параметров треков, если ≤ 0 то нет.

ПК № 10:

1. условия поиска трека на блоке, если ≥ 0 , то ищутся треки с максимальным числом трековых срабатываний, если < 0 , то любой набор срабатываний больший заданного для данного блока, принимается за трек,
2. если на камере сработало подряд больше элементов, чем заданное число, то срабатывание выбрасывается,

ПК № 11:

1. делать или нет двойной фит для выброса "плохих" точек, если $\neq 0$, то не делается,
2. требуется ли трек на подблоках, если $\neq 0$, то не требуется.

ПК № 12:

1. для обработки многотрековых событий, если $\neq 0$, то не учитывается.

ПК № 13:

1. номера блоков X , по которым определяются повороты для соответствующих Y блоков.

ПК № 14:

1. в первых 10-ти позициях задается ширина коридора для поиска трековых срабатываний.

Формат для считывания ПК I-13, 5х, $f_0(15)$, ПК I4 $F(10,0)$.

Другие условия работы программы геометрической реконструкции задаются в подпрограммах.

Подпрограмма CHAMBS

В этой подпрограмме задаются основные параметры трековых детекторов, т.е. искровых и пропорциональных проволочных камер и годоскопов.

Информация считываемая в этой подпрограмме заносится в следующие блоки COMMON:

Имя блока

$|NCHAM|$ - $NCHA$ - число камер на установке, максимальное число 80.

$NCHX$ - число камер в X-проекции

$NCHY$ - число камер в Y-проекции

$|FLAG|$ - $IFLAG(80)$ - принимает следующее значение в зависимости от номера камеры на установке

$IFLAG(N)=0$ - камера с номером N - отключена

$IFLAG(N)=-1$ - камера с номером N - относится к X-проекции

$IFLAG(N)=+1$ - камера с номером N - относится к Y-проекции

$|ISG|$ $ISIG(40,2)$ - задает направление считывания и расстояние между сигнальными проволочками. Например: $ISIG(NC,NU)=2$ означает, что камера с номером NC в проекции NU ($NU=1, X$, $NU=2, Y$) имеет шаг намотки 2 мм и расположена так, что с возрастанием номера проволоки возрастает координата.

$CRX(40,2)$ - координаты первых проволочек для камеры с данным номером в данной проекции. Например:

$CRX(15,1)=-105,1$ означает, что камера с номе-

ром I5 в проекции X имеет координату первой проволоки - 105,1 мм. (формула 3)

IZZXY/ZCHA (40,2) - Z координата камеры данного номера данной проекции.

UPFI/UP (80) - максимальные номера проволок камеры с данным номером, которые будут рассматриваться при обработке.

OL (80) - соответствующие минимальные номера. Эти два массива используются для задания рабочей области камеры. То есть, если сработавшие проволоки имеют номера больше *UP(N)* или меньше *OL(N)* они отбрасываются сразу же после раскодировки и в восстановлении треков участия не принимают.

NUCH/NCN (40,2) - массив соответствия номеров камер в проекции и на установке. Например:

NCN (20,2)=56 означает, что камера, имеющая в проекции Y номер 20, в общей системе установки имеет номер 56.

RCN/ICN (80) - массив обратного соответствия. Например:

ICN (56)=20 означает, что камера с номером 56 в общей системе камер, имеет номер 20 в X или Y проекции.

ICWT/WEIG (80) - при считывании в массив *WEIG(N)* заносится точность для камеры *N*, в подпрограмме считается *B* для соответствующей камеры и в дальнейшем в массиве *WEIG(N)* находится величина $1/B_N$, которая используется при фитировании треков.

ABA/ANGL (40,2) - углы поворота камеры относительно оси Z данной проекции и данного номера.

ROTCH/NROT (2,2) - номера камер в U и V проекции, в каждой проекции может быть до двух камер, если

$NR0T(I,I)=\emptyset$, но камер, повернутых на большие углы относительно оси Z нет.

$NR0T(2,2)$ – соответствующие номера в X и Y проекции

$NR0TA(2,2)$ – число срабатываний на этих камерах

$NR0T(I5,2,2)$ – координаты срабатываний

$IBHOD/SHGD(10,2)$ – координаты крайнего элемента годоскопа данной проекции с данным номером.

$ZG(10,2)$ – соответствующее значение Z координаты

$DG(10,2)$ – соответствующее значение ширины элемента годоскопа

$NGR(10,2)$ – направление считывания на годоскопе. Например:

$SHGD(2,2)=-208,5$

$ZG(2,2)=3578.$

$DG(2,2)=45$

$NGR(2,2)=1.$

Это означает, что годоскоп с номером 2 в проекции Y имеет координату края первого элемента – 208,5 мм., Z' -координату 3578 мм., ширину одного элемента 45 мм. и считывается слева направо, если смотреть на него по пучку (с ростом номера элемента увеличивается значение ее координаты).

Задание всех выше перечисленных величин происходит следующим образом:

Первая ПК считывается форматом 1X, A4, на ней пишется откуда далее идет считывание информации, если на карте записано $\ll DISC$, то дальнейшая информация считывается с файла на диске или ленте с номером I7, в этом файле должны быть записаны образы перфокарт. Если на карте содержится любая другая запись, то информация будет считываться с файла 5, т.е. перфокарт. Следующая карта считывается форматом 35X, I3, на ней должно быть записано $NSHA$ – полное число искровых и пропорциональных камер установки. Далее $NSHA$ карт формата 3(I5),

6 $F(10,0)$ на них содержится: 1 - номер камеры N , 2 - $IFLAG(N)$,
 3 - $ISIG(N, NU)$, 4 - $CRX(N, NU)$, 5 - $ANGL(N, NU)$, 6 - $ZCHA(N, NU)$,
 7 - $OL(N)$, 8 - $UP(N)$, 9 - $WEIG(N)$.

Значение переменных и массивов описано выше.

Далее тем же форматом считывается: 1 - NG - номер годоскопа,
 2 - NU - номер проекции, 3 - $NGR(NG, NU)$, 4 - $ZG(NG, NU)$, 5 - $SHBD(NG, NU)$
 6 - $DG(NG, NU)$.

Считывание происходит до тех пор, пока NG не окажется меньше или равен нулю.

Подпрограмма BLOK(LUN)

В этой подпрограмме происходит разбиение всех камер на условные блоки, на каждом из которых в дальнейшем будут искаться прямолинейные траектории заряженных частиц. Камеры могут объединяться также в подблоки, на которых треки определяются по блоку, в который они входят. Информация считывается с файла с номером LUN . В ней используются следующие COMMON блоки:

Имя

$ABREG/ AMI(10,2)$ и $AMA(10,2)$ - значение минимальных и максимальных тангенсов углов наклона для данного блока в данной проекции.

$BMI(10,2)$ и $BMA(10,2)$ - $ZTA(10,2)$ - значение минимальное и максимальное координаты треков при заданном Z соответственно. Если трек должен сшиваться с треком на другом блоке, то BMI и BMA задает границы сшивки при заданном ZTA . Например:

$$AMI(3.2) = -0,1$$

$$AMA(3.2) = 0,1$$

$$BMI(3.2) = -5.$$

$$BMA(3.2) = 5.$$

$ZTA(3.2) = -8100$ означает, что треки на третьем блоке в проекции Y должны иметь параметры A в интервале $\pm 0,1$ и проходить через область ± 5 мм при $Z = -8100$.

/NBLOC/ $NBL(10,2,2)$ - массив номеров первой ($NBL(Y,1,NU)$) последней ($NBL(Y,2,NU)$) камер в Y -ом блоке, в проекции NU .

$MPR(10,2)$ - массив минимального числа сработавших камер из данного блока, которые дают трек

$NBR(2)$ - реальное число блоков в X и Y проекции

/IBL/ $IBL(10,5,2)$ - массив номеров подблоков, из которых состоит данный блок.

/SPMS/ $L1(5,2)$ - массив номеров соответствия блоков и геометрии установки используется при определении кинематики исследуемых процессов. Должно быть следующее соответствие: $L1(1,NU)$ - номер блока между магнитными линзами и магнитом СП-129, $L1(2,NU)$ - номер блока между магнитом СП-129 и мишенью, $L1(3,NU)$ - номер блока, на котором определяется трек, выходящий из мишени, $L1(4,NU)$ - номер блока, по которому определяется вход частицы в магнит МС-12, $L1(5,NU)$ - номер блока, на котором определяется трек частицы за мишенью.

/TAIC e/ $MTA(10,5,2)$ - в этот массив заносятся: номер блока, с которым сшивается трек на данном блоке и номера трековых детекторов, через которые должен проходить трек.

Основным управляющим массивом при дальнейшем поиске треков на блоках является массив $IR(10,2,2)$ (переменная IRA из *COMMON* блока IRA в данной подпрограмме не используется).

Элементы массива $IR(NB,1,NU)$ определяют блок NB или подблок

$IR(NB,1,NU) < 0$ - блок NB входит в состав другого блока, т.е.

является подблоком.

$= 0$ - является обычным блоком.

$= N$ - блок NB разбивается на N подблоков, номера которых содержатся в $IBL(NB,1,NU)$.

Элементы массива $IR(NB,2,NU)$ определяют условия поиска треков на блоке NB .

$IR(NB,2,NU) = \emptyset$ ищутся любые треки в заданных границах.

$IR(NB,2,NU) = 1$ треки сшиваются с треками на блоке с номером $NTA(NB,5,NU)$

$IR(NB,2,NU) = 2$ треки должны быть продолжениями треков на блоке с номером $NTA(NB,5,NU)$

$IR(NB,2,NU) = 3$ треки сшиваются с треками на блоке с номером $NTA(NB,5,NU)$ и должны иметь срабатывание на камере с номером $NTA(NB,1,NU)$

$IR(NB,2,NU) = 4$ треки сшиваются с треками на блоке с номером $NTA(NB,5,NU)$ и должны проходить через сработавший элемент годоскопа $NTA(NB,1,NU)$

$IR(NB,2,NU) = 5$ трек должен иметь срабатывание на камере с номером $NTA(NB,1,NU)$

$IR(NB,2,NU) = 6$ трек должен проходить через сработавший элемент годоскопа $NTA(NB,1,NU)$

$IR(NB,2,NU) = 7$ трек должен проходить через сработавшие элементы на годоскопах $NTA(NB,1,NU)$ и $NTA(NB,2,NU)$

$IR(NB, 2, NU) = 8$ трек должен иметь срабатывание на камерах с номерами $NTA(NB, 1, NU)$, $NTA(NB, 2, NU)$.

$IR(NB, 2, NU) = 9$ трек должен иметь срабатывание на камере с номером $NTA(NB, 1, NU)$ и проходить через сработавший элемент годоскопа $NTA(NB, 2, NU)$.

$IR(NB, 2, NU) = 10$ поиск треков без вычеркивания сработавших уже вошедших в трек.

В начале программы форматом $1-(5X, 10(I5))$ считывается значение $NBR(1)$ и $NBR(2)$.

Затем в цикле сначала по X и затем по Y считывается значение $L1$, формат 1 , потом в цикле по числу блоков форматом 1

$NB, NBL(NB, 1, NU), NBL(NB, 2, NU), YPR(NB, NU), IR(NB, 1, NU), IR(NB, 2, NU)$,

далее форматом $8(F10, 0)$

$AMI(NB, NU), AMA(NB, NU), BMI(NB, NU), BMA(NB, NU), ZTA(NB, NU)$,

далее форматом 1

$IBL(NB, L, NU)$,

затем форматом 1

$NTA(NB, 5, NU)$

и, наконец, форматом 1

$NTA(NB, 1, NU)$ и $NTA(NB, 2, NU)$.

Значение массивов объяснено выше. Цикл повторяется по числу блоков в X ($NBR(1)$) и Y ($NBR(2)$) проекциях.

На этом подпрограмма заканчивает свою работу.

ПОДПРОГРАММА CREAD

Подпрограмма *CREAD* вызывается только в том случае, если необходимо восстанавливать импульсы заряженных частиц, прошедших через установку. В данной подпрограмме считываются значения коэффициентов при полиномах Чебышева и границы областей, для которых они подсчитаны.

Информация считывается с файла на диске или ленте и заносится в следующие *COMMON* блоки /CII/ C (3,3,5,II,5) – коэффициенты при полиномах Чебышева.

/XMX/ – в этом блоке задаются все граничные условия используемые при восстановлении импульсов и максимальные степени используемых полиномов.

Следует отметить, что все реальные величины задаются в виде *REAL*8*, т.е. имеют двойную длину на ЕС ЭВМ.

З А К Л Ю Ч Е Н И Е

В работе описана программа геометрической реконструкции событий зарегистрированных установкой Гиперон. Программой было обработано *1.5 млн.* триггеров гиперзарядообменных событий. Обработка проводилась на ЭВМ ЕС-1060. Среднее время затрачиваемое ЭВМ на обработку одного события 0,2 сек., объем оперативной памяти занимаемой программой 320 К. Специфические возможности программы позволили точно определить пространственное положение трековых детекторов /2/ и добиться высокого разрешения при определении кинематических параметров заряженных частиц, которое необходимо при выделении канала реакции методом недостающих масс /1,7,8/.

Созданная программа ориентирована на специфику установки Гиперон, однако может быть с успехом использована для определения геомет-

рических констант и обработки событий на установках, подобных спектрометру Гиперон.

В заключение авторы благодарят Ю.А.Будагова и В.Б.Флягина за постановку задачи, постоянное внимание к работе и ряд ценных советов, А.А.Богуша, Л.Г.Мороза, Е.М.Маниева, Н.А.Русаковича, Л.А.Пермякову, В.И.Белоусова, В.С.Румянцева, Л.Шандора за многократные полезные обсуждения.

Литература

1. Антхков В.А. и др. ОИЯИ, Р13-84-562, Дубна, 1984.
2. Виноградов В.Б. и др. ОИЯИ, 13-84-805, Дубна, 1984.
3. Виноградов В.Б. и др. ОИЯИ, Р10-85-77, Дубна, 1985.
4. Виноградов В.Б. и др. ОИЯИ, Р1-83-390, Дубна, 1983.
5. Виноградов В.Б. и др. ОИЯИ, 1-13015, Дубна, 1980.
6. Виноградов В.Б. и др. ОИЯИ, 1-10997, Дубна, 1977.
7. Бицадзе Г.С. и др. ОИЯИ, Р1-84-657, Дубна, 1984.
8. Бицадзе Г.С. и др. ОИЯИ, Р1-84-658, Дубна, 1984.
9. Zanella P. CERN-DD/DP/68-3, 1968.
10. Говорун Н.Н. и др. ОИЯИ, 10-7303, Дубна, 1973.
11. Джелядин Р.И. и др. ИФВЭ, 84-70, Серпухов, 1984.
12. Вестергомби Д. и др. ОИЯИ, Р10-7284, Дубна, 1973.
13. Акименко С.А. и др. ОИЯИ, Р13-80-155, Дубна, 1980.
14. Балашов В.К. и др. ОИЯИ, 11-30-251, Дубна, 1980.
15. Дорогов В.И., Кунике М. ОИЯИ, Р10-11919, Дубна, 1978.
16. Виноградов В.Б. и др. ОИЯИ, 51-10-85-651, Дубна, 1985.
17. Виноградов В.Б. и др. ОИЯИ, Р1-85-761, Дубна, 1985.

Подписи к рисункам

Рис.1 Схема спектрометра Гиперон. $\check{C}_1-\check{C}_5$ - черенковские счетчики, S_1-S_{12} - сцинтилляционные счетчики, ПК-1, ПК-7 - пропорциональные камеры, ИПК-1, ИПК-5 - искровые проводочные камеры. $\Gamma_1-\Gamma_4$ - сцинтилляционные фотоскопы. СП-129 и МС-12 - электромагниты.

Рис.2 Общая блок-схема программы **HYPER** .

Рис.3 Блок-схема поиска и вычисления геометрических параметров треков.

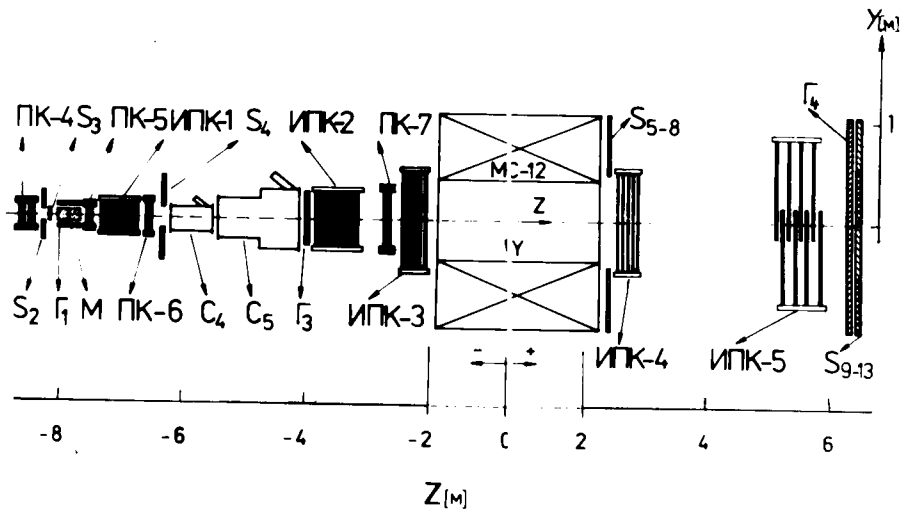
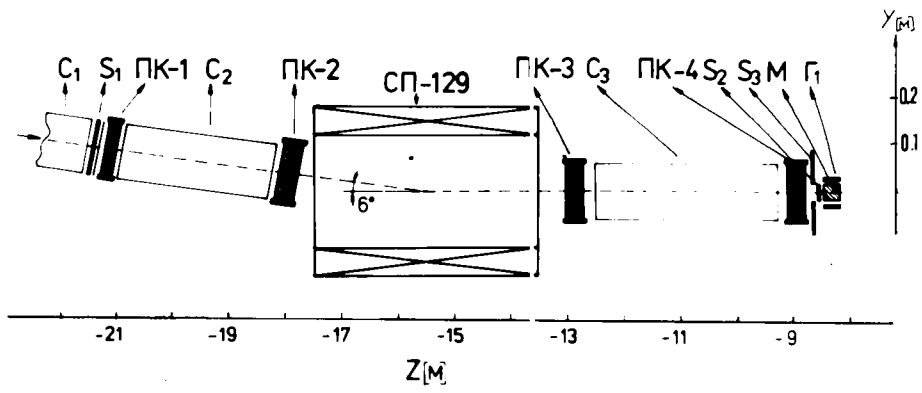


Рис. 1

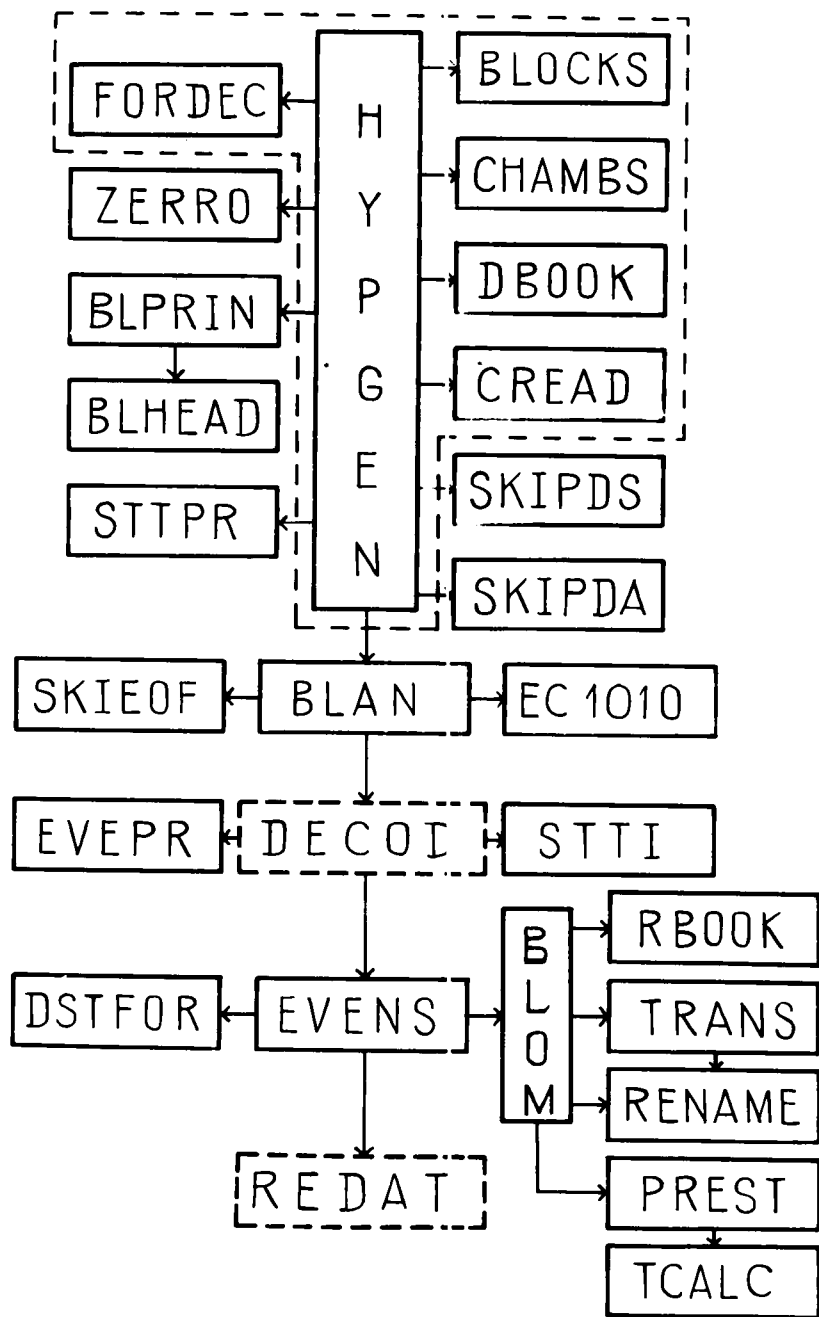


Рис. 2

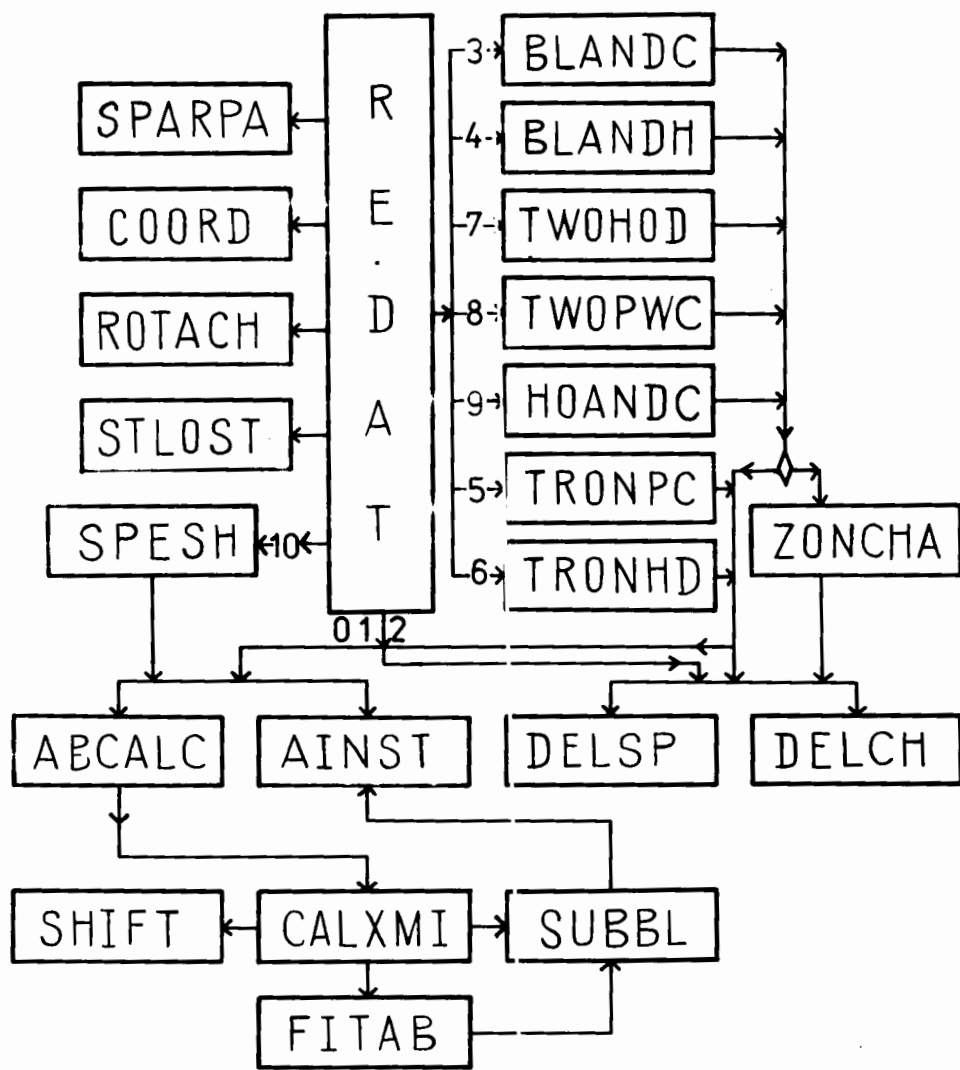


Рис. 3