

**Объединённый институт ядерных исследований
Лаборатория физики высоких энергий**

Овчаренко
Егор Владимирович

**РАЗРАБОТКА МЕТОДОВ МОДЕЛИРОВАНИЯ, СБОРА И АНАЛИЗА ДАННЫХ
ФИЗИЧЕСКИХ УСТАНОВОК И ИХ ПРИМЕНЕНИЕ ДЛЯ ДЕТЕКТОРА ЧЕРЕНКОВСКИХ
КОЛЕЦ ЭКСПЕРИМЕНТА СВМ**

Специальность 01.04.01 -
Приборы и методы экспериментальной физики

Диссертация на соискание учёной степени
кандидата физико-математических наук

Научный руководитель: к.ф.-м.н. С.Г. Белогуров

Дубна - 2017

Содержание

Введение	5
0.1 Обработка сигнала с детектора	5
0.1.1 Система считывания и сбора данных эксперимента CBM	8
0.1.2 FLES	8
FLES Interface Board	10
0.2 Обзор существующих детекторов черенковских колец	11
0.2.1 COMPASS RICH-1	11
0.2.2 LHCb	13
LHCb RICH-1	13
LHCb RICH-2	13
0.2.3 HERA-b RICH	13
1 Эксперимент CBM на FAIR	14
1.1 Физическая программа CBM	14
1.2 Экспериментальная установка CBM	14
1.2.1 Вершинный микрорадетектор MVD	14
1.2.2 Кремниевая трекинговая система STS	14
1.2.3 Детектор черенковских колец RICH	15
1.2.4 Мюонная система MUCH	15
1.2.5 Детектор переходного излучения TRD	15
1.2.6 Время-пролётный детектор TOF	15
1.2.7 Электромагнитный калориметр ECAL	18
1.2.8 Детектор PSD	18
1.3 CBM RICH поподробней	18
2 Геометрия	19
2.1 Сравнение представления геометрии в GEANT/ROOT и САПР	21
2.1.1 Представление геометрии в САПР	22
2.1.2 Представление геометрии в GEANT/ROOT	24
2.2 “CATIA-GDML geometry builder”	25
2.2.1 Переменные окружения	27
2.2.2 Примитивы в “Builder”	27
2.2.3 Булевые операции	28
2.2.4 Параметризация	29
2.2.5 Тесселированная геометрия	29
2.2.6 Макропрограммы для CATIA v5 «AddNewPart»	30
	31

«AddShape»	31
«Poly»	31
«Inserter»	31
«ArrayMaker»	32
«Replica»	32
«PointToPointAligner»	32
«Mover»	33
«Measure»	33
«MaterialsManager»	33
«Checker»	34
«CATIA2GDML»	35
«GDML2CATIA»	36
«Duplicator»	36
Визуализация нескольких уровней вложенности объёмов	36
2.2.7 Geometry Description Markup Language (GDML)	37
2.2.8 Избранные подробности реализации “CATIA-GDML geometry builder”	38
Работа с матрицами позиционирования в “CATIA-GDML geometry builder”	38
2.3 Применение “CATIA-GDML geometry builder” к CBM RICH	39
2.3.1 Фокусирующая система — сферические зеркала	40
MC-геометрия фокусирующей системы с возможностью задания инди- видуальных отклонений	41
2.3.2 Магнитный экран вокруг камеры	42
2.3.3 Фоточувствительная камера	42
2.3.4 MC-геометрия механических конструкций RICH	46
2.4 Адаптация CbmRoot для работы с GDML	47
3 Развитие DAQ эксперимента CBM	51
3.1 Фотоэлектронный умножитель	51
3.2 Архитектура системы сбора данных CBM RICH	53
3.2.1 64-канальный модуль считывания	53
3.2.2 Концентрация и ввод данных в ЭВМ	58
3.3 Время-цифровой преобразователь	58
3.4 Экспериментальные установки	61
3.4.1 Экспериментальная установка на пучковых тестах	61
3.4.2 Лабораторный стенд	64
3.5 Программное обеспечение	66
3.5.1 Распаковка	67
3.5.2 Калибровка точного времени	67
3.5.3 Коррекция задержек между каналами	68
3.5.4 Построение хита	68
3.5.5 Построение события	69
3.5.6 Реконструкция	71
4 Результаты анализа	73
4.1 Испытание системы сбора данных с использованием FLIB	73
4.2 LeadingEdgeDiff	73
4.3 Калибровка точного времени (Fine time calibration)	74
4.4 Определение коррекций задержек между каналами	79
4.5 Временное разрешение	80

4.6	Исследование профиля высвечивания сместителя спектра	83
4.6.1	Исследование спектросместителя с помощью флюориметра	85
4.6.2	Прямые измерения временного профиля спектросместителя	86
	Прямые фотоны	86
	Фитирование WLS_diff	87
4.7	Время над порогом	88
4.8	Сравнение одноэлектронных спектров при временном и амплитудном считывании	89
5	Применение разработанных методов к оптимизации СВМ RICH	93
Заключение		94
Список литературы		97

Введение

0.1 Обработка сигнала с детектора

Экспериментальная установка в физике высоких энергий или физике тяжёлых ионов ставит своей задачей восстанавливать события, произошедшие в области первичного взаимодействия. Такой областью будет являться точка взаимодействия пучка с мишенью в случае эксперимента с фиксированной мишенью, либо точка взаимодействия встречных пучков на коллайдерном эксперименте. Для выполнения этой задачи в некоторой области рядом с точкой взаимодействия ставят набор из нескольких детекторов, регистрирующих вторичные частицы и их продукты распада. Детекторы вырабатывают электрические сигналы, которые в случае современного крупного эксперимента необходимо доставить в ЭВМ, чтобы выполнять программную обработку, включающую в себя реконструкцию треков и частиц и анализ многочисленных распределений.

В этой главе обсуждаются решения, выбранные СВМ для организации считывания данных с детекторной установки.

Задача: дать общую картину — детектор, передняя электроника, передающая электроника, ЭВМ. Затем указать, что вот типа электроника с внешним триггером, а вот самотrigгирующаяся электроника. Далее переходим к free-running DAQ.

Рассмотрим общую последовательность обработки сигнала детектора частиц. Источником является некоторое детектирующее устройство (или просто детектор). В зависимости от типа детектируемых частиц и внешних условий — радиационная среда, температура, частота регистрации — это может быть фотодиод, фотоэлектронный умножитель (ФЭУ), полупроводниковый детектор, микроканальная пластина, газовый электронный умножитель, и т.д. Все эти детекторы обладают таким общим свойством, что выходной сигнал, несущий информацию о зарегистрированной частице, это относительно слабый токовый импульс. В любом случае, для того, чтобы выполнять дальнейшую обработку этого сигнала в ЭВМ его необходимо оцифровать.

В связи с этим обрабатывающую электронику условно можно разделить на переднюю и остальную. Основная задача передней электроники — оцифровать выходной аналоговый сигнал детектирующего устройства, при необходимости предварительно его усилив и отфильтровав. Передача аналогового сигнала затруднена, поэтому его обработка и оцифровка обычно выполняется как можно ближе к месту, где этот сигнал вырабатывается, т.е. непосредственно на детекторе. В простом случае последующие слои электроники концентрируют и передают оцифрованные сигналы со многих каналов передней электроники. В более сложном случае возможна какая-то аппаратная обработка оцифрованных сигналов, например архивация или фильтрация с целью уменьшения потока данных.

Разработан целый ряд подходов к оцифровке сигналов с детекторов. Каждый из них лучше всего подходит к определённому детектору, однако возможно применение методов,

разработанных для одного детектора, для оцифровки сигналов с другого детектора. Перечислим лишь некоторые из них. Если не вдаваться в подробности реализации, можно выделить следующие способы: регистрация момента времени прихода фронта, регистрация амплитуды сигнала, непрерывное семплирование (ещё???). Практически всегда требуется регистрация момента времени прихода сигнала, поэтому широко применяются комбинации перечисленных способов — регистрация амплитуды сигнала вместе с моментом времени прихода переднего фронта, регистрация моментов времени переднего и заднего фронтов без захвата амплитуды, регистрация момента времени переднего фронта и формирование ограниченного числа семплов после него.

Если необходимо предварительное усиление, то применяют один из двух типов усилителей — зарядочувствительный усилитель либо ??? . В силу устройства зарядочувствительный усилитель является также формирователем (шайпером). Любая электроника подвержена шумам, поэтому для подавления шумов сигнал обычно предварительно проходит фильтр нижних частот, эффективно отрезающий частоты выше некоторого значения. (Что-то сказать про шейпирование и его неизбежность и влияние на временные характеристики) Ещё один способ борьбы с шумами — фильтрация по амплитуде, т.е. установление некоторого порога по напряжению, ниже которого сигнал игнорируется.

Описанные процедуры обработки сигнала могут быть реализованы на разной аппаратной платформе. Самый примитивный вариант — это когда канал реализуется с применением “сквозного монтажа” (THT) или “поверхностного монтажа” (SMT, SMD) ??? таких-то компонентов. В принципе, это возможно, но в этом случае геометрические размеры определяют заметное время прохождения сигнала, таким образом ограничивая частоту функционирования. Также габариты ограничивают плотность каналов в пространстве и, вообще говоря, стоимость такой электроники выше современных компактных вариантов. В эксперименте СВМ ожидается огромное число каналов при очень высокой плотности в пространстве. Один только детектор RICH будет иметь более 60000 каналов при плотности каналов около $0.5 \text{ см}^2/\text{канал}$. Следовательно, рассматриваются более продвинутые технологии — интегральные схемы, в том числе программируемые.

На данный момент существует множество разновидностей интегральных схем, однако наибольший интерес в СВМ (экспериментальной физике???) проявляется к интегральным схемам специального назначения (ASIC) и программируемым пользователем вентильным матрицам (ППВМ, FPGA). ASIC уже широко применяются в экспериментальной физике и в быту на протяжении десятилетий, в то время как ППВМ доступны сравнительно недавно. Эти два варианта принципиально отличаются тем, что ASIC невозможно изменить после изготовления, а FPGA не имеет какой-либо программы по умолчанию и программируется пользователем. Более того, программа FPGA стирается при отключении питания, поэтому рядом необходимо обеспечить постоянную память, из которой FPGA берёт программу при загрузке. Процесс проектирования FPGA фактически заключается в процессе разработки прошивки на языке описания аппаратуры (HDL, verilog), при этом в любой момент есть возможность применить прошивку к чипу чтобы выполнить отладку. Достоинство ASIC заключается в том, что чипы изготавливаются относительно дёшево при большом размере партии. Высокую стоимость имеет шаблон, на основе которого можно дёшево изготовить большую партию чипов. С другой стороны процесс проектирования ASIC осложнён тем, что если для отладки требуется физический экземпляр, а не программная модель, то требуется изготовление шаблона.

Рассмотрим схему, когда аналоговый импульс с детектора обрабатывается электроникой, регистрирующей момент времени прихода переднего фронта. Любой сигнал переключается за некоторое ненулевое время, поэтому необходимо определить точку, обозначающую фронт. Для этого применяют дискриминатор — прибор,рабатывающий логический “0”,

когда входной сигнал ниже установленного порога, и логическую “1”, когда входной сигнал выше установленного порога. Таким образом точка пересечения сигнала и порога это точка, условно обозначающая момент времени прихода сигнала. На выходе дискриминатора получается логический сигнал, который необходимо преобразовать в цифровой с помощью время-цифрового преобразователя (ВЦП). На выходе ВЦП уже будет цифровой сигнал, а не логический.

Коллаборацией СВМ ведётся разработка нескольких чипов, на базе которых будут построены платы передней электроники. STS-XYTER — ASIC для кремниевой трековой системы, регистрирующий амплитуду сигнала и временную отметку переднего фронта. SPADIC — ASIC для детектора переходного излучения, имеющий богатый функционал, но, самое главное, выполняющий непрерывное семплирование входного сигнала. (**Наверное, нет смысла описывать подробнее — слишком сложный чип**) Группы СВМ RICH, HADES RICH и PANDA DIRC совместно занимаются разработкой программ для FPGA, выполняющих функции дискриминатора и ВЦП.

(Тут нужно обсудить и почитать ещё. Возможен ли такой сценарий, в наше время или в прошлом, когда триггер чисто аналоговый. То есть передняя электроника игнорирует сигнал до тех пор, пока не сработает схема совпадения с триггером. В той модели традиционного триггера, которую я себе сейчас представляю, входной сигнал обрабатывается непрерывно, но выпускается на выход из буфера только при наличии триггера. Вероятно, возможно и то и то, но вопрос в том, что реально используется, что более распространено.) Рассмотрим систему считывания и сбора данных “традиционного” эксперимента, имеющего аппаратный триггер. Каждый канал передней электроники имеет выходной буфер, куда по принципу FIFO складываются оцифрованные входные сигналы. В экспериментальной установке присутствуют детекторы, вырабатывающие триггер — сигнал, который заводится (условно) на каждый канал считывания, и говорит о том, что произошло интересное событие, которое необходимо сохранить для последующей обработки. Данный подход имеет свои причины. Во-первых, до недавнего времени физические эксперименты не требовали высоких частот регистрации — выполнение физической программы при относительно низких частотах первичного взаимодействия было осуществимо в разумные сроки. Во-вторых, многие регистрирующие приборы имеют заметное “мёртвое время” — время после регистрации одного входного сигнала, в течение которого прибор не может обрабатывать последующие входные сигналы. Следовательно, если канал регистрирует ложный входной сигнал, велика вероятность того, что будет пропущен полезный сигнал. С развитием электроники “мёртвое время” уменьшалось. Более того, возможность применения принципиально другой считающей электроники, как например чисто временной канал, реализованный в ППВМ, для обработки сигналов с МА ФЭУ в СВМ RICH, исследуемая в данной работе, позволяет на порядки снизить “мёртвое время” и повысить точность регистрации временной отметки в ущерб полноты информации.

(Сюда подмешивается секция 2.2 ???)

В СВМ планируется использование программного триггера. Это означает, что для того, чтобы принять решение, сохранять принятые данные или нет, необходимо выполнить полную реконструкцию события, включая реконструкцию треков, которая является высокозатратной задачей. Рассматривается также возможность на определённых этапах работы установки использовать для выработки триггера частичную реконструкцию. Например исследуется возможность триггирования по результатам реконструкций треков только в MUCH, когда стоит задача поиска (??? такой-то частицы).

В “традиционном” эксперименте триггер может формироваться в результате логических операций над сигналами с нескольких детекторов, реализованных аппаратно. Такая логика работает за (??? масштаб времени). Реконструкция треков выполняется за гораздо большее

время (??? на столько-то порядков выше). По этой причине необходимо иметь не только буфер в электронике — его будет недостаточно.

(Сказать о том, что нужно хорошо настраивать пороги.)

0.1.1 Система считывания и сбора данных эксперимента СВМ

Блок-схема архитектуры системы считывания и сбора данных эксперимента СВМ приведена на рис. 1.

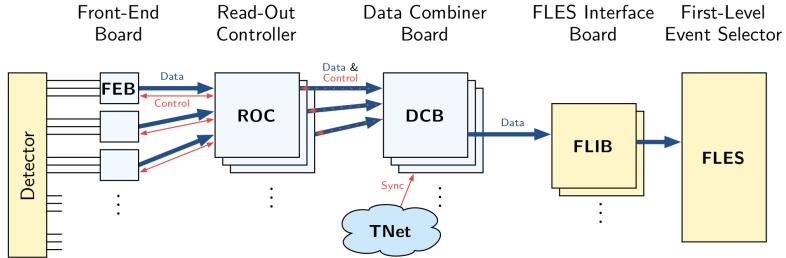


Рис. 1: Система считывания и сбора данных СВМ.

Развёрнутое описание выше делается для понимания этой секции.

Источником электрических сигналов являются детекторы, выполненные по разным технологиям, и требующие оцифровки различными способами. Эта оцифровка выполняется разнообразными платами передней электроники (FEB), имеющими разные принципы и основанными на различных чипах. Затем цифровые данные опрашиваются с плат передней электроники контроллерами считывания (ROC). После этого данные группируются в соответствующих платах (DCB) и поступают во FLES через платы FLIB.

0.1.2 FLES

переработать Всё вместе привело к разработке аппаратно-программного комплекса, называемого системой отбора первого уровня — First Level Event Selector (FLES). По сути DAQ-часть (приём данных) неотделима от FLES, поэтому иногда эту систему называют FLES/DAQ. Отличительные черты FLES — самотриттирующаяся электроника, несколько уровней концентрации данных, многочисленные буферы, формирование срезов времени, построение интервалов и только после этого мы переходим к построению событий.

Функционал FLES/DAQ можно разбить на три части. Первая — приём и объединение данных, поступающих с группы каналов, в срезы времени (timeslice). Такую группу образует некоторое множество каналов одного детектора, количество которых определяется ожидаемым потоком данных, а ограничение диктуется максимальной пропускной способностью входного канала FLES.

Один из наиболее важных и нестандартных этапов работы FLES — это построение интервалов (Interval building, IB). Построение интервалов — это получение контейнеров с данными со всех детекторов за некоторый интервал времени путём перегруппировки данных из срезов времени. Суть построения интервалов показана на рис. 2. Данные от одного “входного узла” (input node, IN) приходят от группы каналов какого-то детектора и представляет собой последовательность срезов времени. Задача заключается в том, чтобы объединить все срезы, соответствующие одному интервалу времени, чтобы дальше передать на “вычислительный узел” (processing node, computing node, CN).

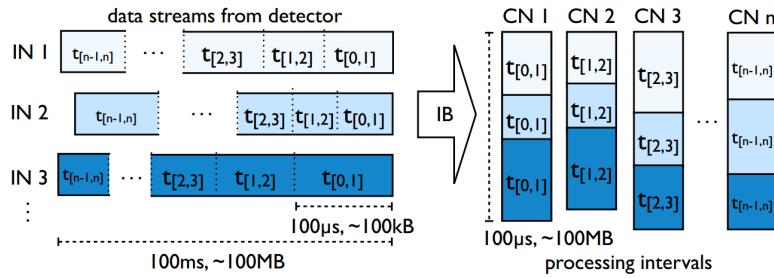


Рис. 2: Построение интервала.

Заключительный этап обработки данных во FLES — восстановление событий в интервале, реконструкция треков и частиц в восстановленных событиях и выработка сигнала о сохранении или отбрасывании интервала.

Для эксперимента СВМ был выполнен оценочный расчёт. Отправная точка — возможно сохранение 1 Гбайт/сек данных. Считается, что одно событие СВМ в среднем имеет объём 40 Кбайт. Отсюда следует, что максимальная частота первичного взаимодействия может быть 25 кГц. В стартовой конфигурации СВМ частота первичного взаимодействия равна 10 МГц, следовательно, необходимо уменьшить поток данных в 400 раз. В полноценном режиме работы СВМ ожидается 25 МГц, т.е. $25 \cdot 10^6 \cdot 40$ Кбайт = 1 Тбайт/сек. Планируется разбить этот поток в 1 Тбайт/сек на 1000 входных каналов FLES, каждый по 1 Гбайт/сек, передающихся по 10-Гбитным оптическим каналам связи. Один входной канал FLES соответствует одному “входному узлу” — ЭВМ с установленной платой FLIB. Все вычисления, необходимые для отбора данных будут осуществляться на так называемых “вычислительных узлах”. Входные и вычислительные узлы объединены в компьютерную сеть посредством InfiniBand QDR, образуя уникальную распределённую вычислительную систему, называемую FLES, см. рис. 3. Вычислительная подсеть будет иметь приблизительно 60000 ядер.

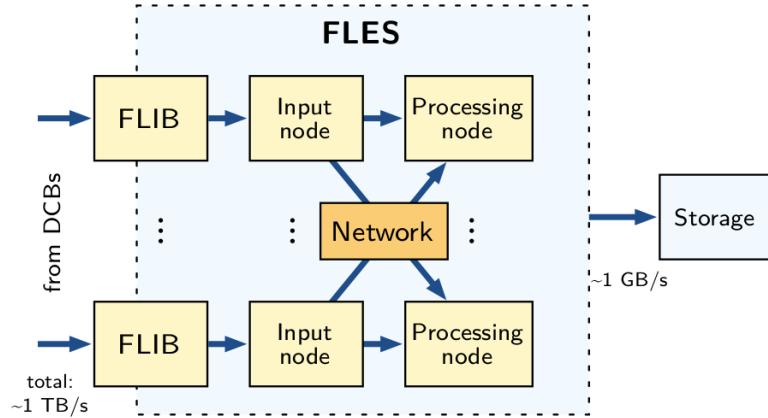


Рис. 3: Общая схема устройства FLES.

Планируется также, что FLES сможет функционировать в особом режиме, когда для выполнения реконструкции с целью отбора данных для сохранения будет использоваться только часть входного потока. На рис. 4 приведена блок-схема функционирования FLES в

таком режиме. Это имеет смысл при работе эксперимента над некоторыми пунктами физической программы, но принципиальная возможность и эффективность реконструкции частиц с целью триггирования по ограниченному набору детекторов является предметом исследований. Например, можно восстанавливать ??? такую-то частицу только по трекам в MUCH. При этом система приёма данных работает в полную силу — идёт приём со всех детекторов и никакие данные не выбрасываются до тех пор, пока не будет выполнена реконструкция по данным с MUCH. Если в результате реконструкции выясняется, что принятая порция данных потенциально интересна, то она извлекается из буферов и записывается. Это позволяет ??? снизить поток сохраняемых данных в ??? раз, что особенно актуально при экстремально высоких частотах взаимодействия.

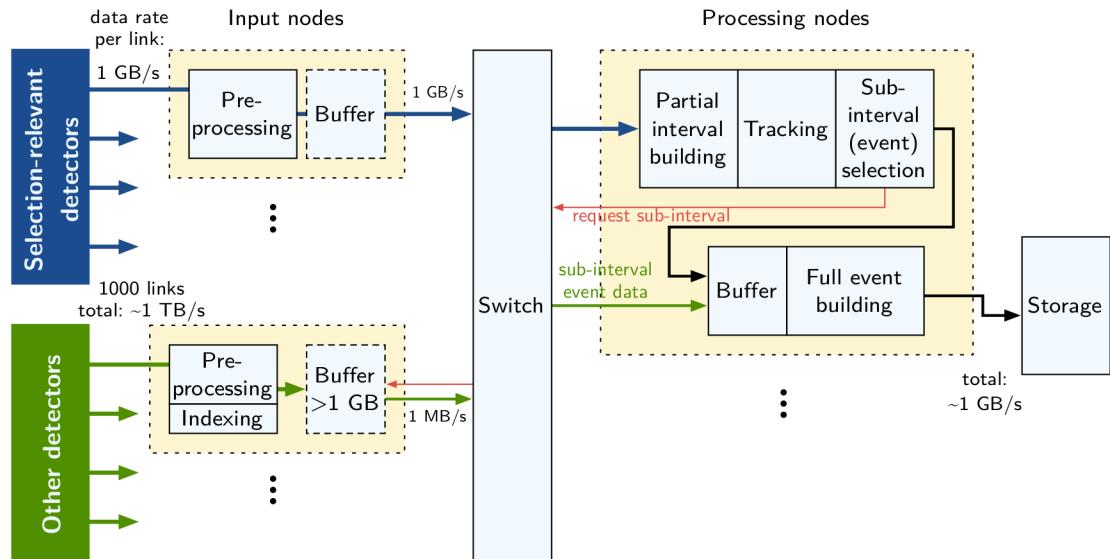


Рис. 4: Блок-схема функционирования FLES в режиме отбора по данным с части детекторов.

??? Подробное описание того, что на картинке

FLES Interface Board

В качестве платы-интерфейса между электроникой, разрабатываемой специально для СВМ, и ЭВМ стандартной архитектуры, будет выступать плата, в общем называемая платой интерфейса FLES – FLES Interface Board, или FLIB. FLIB можно рассматривать как особый сетевой интерфейс, основной задачей которого является предоставление данных, поступающих по входным каналам, центральному процессору ЭВМ. В отличие от сетевого интерфейса Ethernet, данные поступают от разнородных плат передней электроники по разным протоколам. В качестве платформы для реализации FLIB в СВМ рассматривается коммерческая PCI-E плата HTG K-7. Архитектура FLIB показана на рис. 5. Основные компоненты FLIB – драйверы входных оптических портов, программируемая пользователем вентильная матрица (ППВМ, FPGA), оперативное запоминающее устройство (ОЗУ) и драйвер шины PCI-Express.

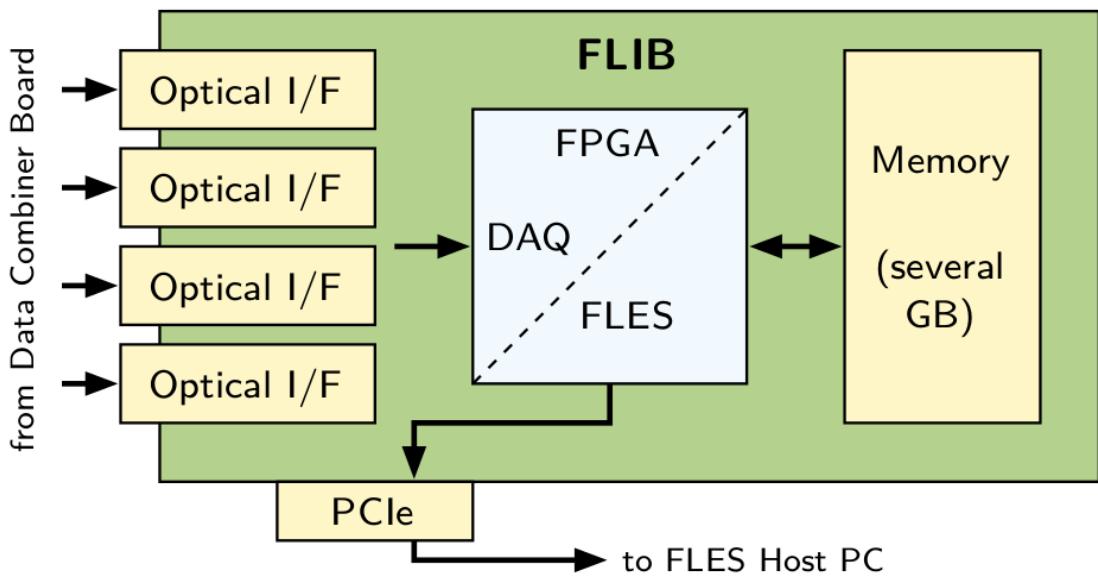


Рис. 5: Архитектура FLIB.

0.2 Обзор существующих детекторов черенковских колец

0.2.1 COMPASS RICH-1

Экспериментальная установка NA58, также известная как COMPASS (“Common Muon and Proton Apparatus for Structure and Spectroscopy”) представляет собой систему из двух спектрометров длиной 60 метров за неподвижной мишенью на отводе пучка M2 ускорителя SPS в CERN. Эксперимент был предложен в 1996 году, в период с 1999 по 2001 года выполнялись работы по установке и в 2001 был выполнен первый (commissioning) запуск. Набор данных разбит на два этапа: COMPASS I (2002–2011) и COMPASS II (2012–2018).

Детектор Черенковских колец RICH-1 был спроектирован в 1996 г. и функционирует с 2002 г. RICH-1 подвергается постоянной оптимизации, а в 2006 было выполнено обновление. Ожидается второе обновление детектора в 2016 г. ??? оно было?

Основная задача детектора Черенковских колец — разделение π , p и K в диапазоне импульсов от 3 до 55 ГэВ/с в условиях высокой интенсивности (что бы это значило??? в статье 2014 г. написано, что beam rate 40 МГц, и частота триггеров 20 кГц) в полном аксептансе спектрометра, составляющем ± 250 мрад по горизонтали и ± 180 мрад по вертикали. Для минимизации отрицательного влияния на эффективность стоящих ниже по пучку электромагнитного и адронного калориметров детектор RICH-1 должен иметь минимум количества материала в аксептансе. Также в процессе проектирования COMPASS RICH-1 необходимо было развивать технологии для реализации возможности регистрировать и справляться с высоким для того времени потоком данных. ??? По большому счёту, задачи те же, что и у СВМ

Габариты корпуса детектора COMPASS RICH-1, выполненного из алюминия, составляют $6.6 \times 5.3 \times 3.3$ м³. Внутри расположен газовый радиатор C_4F_{10} длиной 3 м и объёмом около 83 м³. Пороги по импульсу для Черенковского света: для π — 2.5 ГэВ/с, для K — 8.9 ГэВ/с

и 17 ГэВ/с для p . В центре детектора проходит цилиндрический ионопровод диаметром 100 мм, наполненный гелием. Газовая система закрытого типа поддерживает радиатор под избыточным давлением 100 ± 10 Па.

Система фокусировки состоит из двух сферических зеркал радиусом 6.6 м, составленных из 116 сегментов шести- и пятиугольной формы, общей площадью более 21 м^2 . Для фокусировки на фоточувствительные камеры, расположенные за пределами геометрического аксептанса, центры сфер зеркал смешены по вертикали от пучка на 1.6 м, а образовавшийся в результате этого зазор между двумя зеркалами приводит к потере 4% площади отражающей поверхности. Зеркала были произведены компанией IMMA, Ltd., Kinskeho 703, Turnov, Czech Republic. Коллаборацией COMPASS был разработан метод контроля индивидуальных отклонений сегментов зеркал “на лету” (онлайн), называемый CLAM (“a continuous line alignment and monitoring method”).

??? картинка - схема, оч похоже на CBM

Исходя из необходимости иметь суммарную площадь фоточувствительных камер 5.3 м^2 , изначально для реализации были выбраны многопроволочные пропорциональные камеры (MWPC) с сегментированным фотокатодом из CsI. RICH-1 оборудован восемью идентичными камерами, каждая площадью $576 \times 1152 \text{ мм}^2$. Фотокатоды выполнены из двух двухсторонних печатных плат размером $576 \times 576 \text{ мм}^2$. Окна из silica-quartz состоят из двух одинаковых quartz-plates размером $600 \times 600 \times 5 \text{ мм}^3$. Сегментированный фотокатод обеспечивает размер пикселя $8 \times 8 \text{ мм}^2$ и в общей сложности 82944 канала.

В 2006 г. с целью повышения эффективности детектора было выполнено комплексное обновление центральной области фоточувствительной камеры, составляющей 25% от всей площади. MWPC были заменены на МА ФЭУ с индивидуальными линзами и соответствующей считающей электроникой. В общей сложности было установлено 4 панели по 144 МА ФЭУ Hamamatsu R7600-03-M16, имеющими 16 каналов и входное стекло, прозрачное в ультрафиолетовой области, и специальный делитель напряжения.

Сигнал с МА ФЭУ считывается платами передней электроники, основанными на ASIC “CMAD”, реализующем 8-канальный предусилитель-дискриминатор, разработанный на основе “MAD4” специально для COMPASS RICH-1. CMAD позволяет работать на частоте до 5 МГц на канал.

??? перефразировать Хорошее временное разрешение МА ФЭУ не портится за счёт использования цифровых карт DREISAM, в которых реализован ВЦП F1, имеющий временное разрешение 110 пс и может работать с чатотой до 10 МГц на канал и частоте триггера до 100 кГц.

Считывающая электроника COMPASS RICH-1 монтируется на детектор, образуя очень компактную установку, которая экранирована от внешнего электромагнитного поля медными пластинами, выполняющими также и роль радиаторов, охлаждаемых водой циркулирующей по медным трубкам.

Оцифрованные данные с плат передней электроники передаются по оптике платам считывания CATCH, которые группируют данные и передают дальше также по оптике через S-LINK в систему сбора данных эксперимента.

Заключение такое, что в целом конструкция этого RICH очень схожа с конструкцией CBM RICH.

0.2.2 LHCb

LHCb RICH-1

LHCb RICH-2

0.2.3 HERA-b RICH

На защиту выносятся следующие результаты:

1. Разработка методологии и реализация “CATIA-GDML geometry builder”, средства построения сложной, основанной на инженерном дизайне геометрии детекторов для моделирования прохождения и взаимодействия частиц.
2. Применение “CATIA-GDML geometry builder” для построения беспрецедентно точного параметризованного описания геометрии CBM RICH в среде CbmRoot.
3. Реализация прототипа системы считывания и сбора данных CBM RICH и проведение его тестов на пучке в составе полнофункционального прототипа этого детектора а также дополнительных тестов на лабораторном стенде.
4. Разработка алгоритмов и программного обеспечения для приема, упаковки и передачи бестриггерного потока данных, для калибровки точного времени и относительных задержек каналов и для построения событий из потока данных с детектора CBM RICH в среде CbmRoot.
5. Результаты комплексного исследования временных свойств канала считывания и сбора данных для CBM RICH, реализованного на основе многоанодного ФЭУ H12700 с системой динодов “metal channel”, специально разработанных передней электроники типа предсилитель-дискриминатор и высокоточного ВЦП с последующим прямым вводом данных в единую среду моделирования, сбора и анализа данных CbmRoot.
6. Исследование временных свойств нанесенного на окно МА ФЭУ сместителя спектра при возбуждении черенковскими фотонами.
7. Сравнительный анализ особенностей считывания многоанодного ФЭУ временным и аналоговым трактами.
8. Исследование характеристик детектора CBM RICH с учетом неидеальности геометрии и шумов электроники.

Глава 1

Эксперимент СВМ на FAIR

1.1 Физическая программа СВМ

1.2 Экспериментальная установка СВМ

1.2.1 Вершинный микродетектор MVD

1.2.2 Кремниевая трекинговая система STS

Задача кремниевой трековой системы — измерение траекторий и импульсов заряженных частиц, вылетающих из точки взаимодействия пучка тяжёлых ионов с мишенью. Для выполнения физической программы СВМ — наблюдения редких **явлений** — необходима частота взаимодействий до 10 МГц, при том что в одном взаимодействии будет рождаться до 1000 заряженных частиц. Реконструкция треков должна выполняться с эффективностью порядка 95% и разрешением по импульсу порядка $\Delta p/p = 1\%$. Для удовлетворения перечисленных требований STS должна состоять из 8 слоёв кремниевых микростриповых сенсоров, расположенных внутри поля от дипольного магнита на расстоянии от 30 см до 100 см от точки взаимодействия вниз по пучку с шагом 10 см.

Сенсоры будут монтироваться на легкую механическую опору в виде карбоновых ферм. Считывание будет осуществляться по многоканальным микро-кабелям самотриггирующейся электроникой, расположенной по краям станций вместе с линиями охлаждения и другими инфраструктурными подсистемами. Микро-кабели будут выполнены из sandwiched polyimide-Aluminum layers толщиной несколько десятков мкм. Микростриповые сенсоры будут двухсторонними ???, шаг между стрипами $58\mu\text{m}$, длина стрипов от 20 до 60 мм, а толщина кремния $300\mu\text{m}$. По текущим оценкам максимальная неионизирующая доза в СВМ для сенсоров, расположенных ближе всего к пучку, не будет превышать $10^{14} n_{eq} \text{ см}^{-2}$. STS будет работать в терmostатическом корпусе, обеспечивающем постоянную температуру около -5°C . Тепло, рассеиваемоечитывающей электроникой, отводится с помощью CO_2 системы охлаждения. Механические опоры детектора и соединения спроектированы так, чтобы была возможность заменить отдельный модуль системы не отсоединяя остальные.

??? картинка

1.2.3 Детектор черенковских колец RICH

1.2.4 Мюонная система MuCh

From CBM MuCh TDR:

The MuCh system is designed to identify muon pairs which are produced in high-energy heavy-ion collisions in the beam energy range from 4 to 40 AGeV. The measurement of lepton pairs is a central part of the CBM research program, as they are very sensitive diagnostic probes of the conditions inside the fireball. At low invariant masses, dileptons provide information on the in-medium modification of vector mesons which is a promising observable for the restoration of chiral symmetry. At intermediate invariant masses, the dilepton spectrum is dominated by thermal radiation from the fireball reflecting its temperature. At invariant masses around $3\text{GeV}/c^2$, dileptons are the appropriate tool to study the anomalous charmonium suppression in the deconfined phase. In the CBM experiment both electrons and muons will be measured in order to obtain a consistent and comprehensive picture of the dilepton physics.

The experimental challenge for muon measurements in heavy-ion collisions at FAIR energies is to identify low-momentum muons in an environment of high particle densities. The CBM strategy is to track the particles through a hadron absorber system, and to perform a momentum-dependent muon identification. This concept is realized by an instrumented hadron absorber, consisting of staggered absorber plates and tracking stations. The hadron absorbers vary in material and thickness, and the tracking stations consist of detector triplets based on different technologies. The MuCh system is placed downstream of the dipole magnet hosting the Silicon Tracking System (STS) which determines the particle momentum. In order to reduce the number of muons from pion and kaon weak decays, the absorber/detector system has to be as compact as possible.

The MuCh system will be built in stages which are adapted to the beam energies available. Within the FAIR modularized start version the SIS100 ring will provide heavy ion beams with energies up to 14 AGeV, and proton beams up to 29 GeV. The first two versions of MuCh (SIS100-A and SIS100-B) will comprise of 3 and 4 stations suitable for the measurement of low-mass vector mesons in $A + A$ collisions at 4-6 AGeV and 8-14 AGeV, respectively. The third version of the MuCh system (SIS100-C) will be equipped with an additional iron absorber of 1 m thickness in order to be able to identify charmonium at the highest SIS100 energies. The absorber slices will be built only once so that they could be rearranged properly to obtain required absorber thicknesses. Once SIS300 is operational, we will upgrade the MuCh system further by inserting additional absorbers and detector stations for the measurement of low-mass vector mesons and charmonium at beam energies above 14 AGeV (MuCh versions SIS300-A and SIS300-B).

1.2.5 Детектор переходного излучения TRD

1.2.6 Время-пролётный детектор TOF

From CBM TOF TDR:

The detector system's prime task is to measure the arrival time of charged particles to allow for their identification after having matched the TOF - hit with the corresponding STS track obtained from the CBM silicon tracking system (STS). The system is designed to cope with all the reactions anticipated in the physics program of CBM. This ranges from heavy-ion reactions of different system size from C+C to Au+Au over a large incident energy range from 2A GeV to 45A GeV (35A GeV for the heaviest collision system) to proton - nucleus reactions up to a projectile energy of 90 GeV.

Основная задача время-пролётного детектора — измерять момент времени прихода заряженных частиц, чтобы провести идентификацию после сопоставления хита TOF с соответствующим треком кремниевой трековой системы STS. Детекторная система TOF спроектирована так, чтобы справляться с реакциями, ожидаемыми физической программой CBM. Это подразумевает широкий диапазон взаимодействий от тяжёлых ионов от C+C до Au+Au с энергией от 2A ГэВ до 45A ГэВ до протон-ядерных столкновений с энергией до 90 ГэВ.

The general layout considerations following from the CBM concept are discussed in chapter 2.

Since CBM is operated in fixed target mode the flux of charged particles is strongly varying with the polar angle. In order to fit to the forward acceptance of the CBM experimental setup that is based on an instrumented dipole magnet with particle identification detectors placed downstream of the target, a modular TOF wall is proposed whose elements, called modules in the following, are adapted to the granularity and rate requirements. To allow for sufficient separation power especially for the identification of charged kaons, distances of up to 10 m and a system time resolution of 80 ps are necessary leading to an overall size of the TOF - wall of approx. $12 \times 9 \text{ m}^2$. To achieve the overall time resolution this area needs to get instrumented with timing detectors with an intrinsic timing resolution better than 60 ps and an efficiency better than 95%.

Так как CBM представляет собой эксперимент с фиксированной мишенью, поток заряженных частиц сильно зависит от направления.

Чтобы иметь достаточную степень разделения, особенно для идентификации заряженных каонов, нужно расстояние до 10 м и временное разрешение 80 пс, что приводит к размеру TOF около $12 \times 9 \text{ m}^2$. Чтобы достичь необходимого такого временного разрешения детектор должен быть оборудован электроникой с гарантированным времененным разрешением лучше 60 пс и эффективностью лучше 95%.

We propose to build the CBM - TOF wall on the basis of state-of-the-art Multigap Resistive Plate Chambers (MRPC). The basic element of this robust detector concept is a stack of resistive plates made out of glass or ceramics that are separated by thin gas gaps. At sufficiently high electric field strength avalanches are created in a very uniform manner and can be read out via capacitive coupling. As is demonstrated in chapters 7 this 10 - years old detector technology is well advanced by now, largely due to the effort of the CBM - TOF group, and offers the flexibility to cope with the high demands posed by the CBM physics goals. Most notably, the proposed concept that requires a detector technology with a sustained rate capability in the order of 25 kHz/cm² became possible only by the development of a low - resistivity glass (see section 7.1.1) that is available for CBM - TOF by now through the CBM - TOF group at Tsinghua University, Beijing, China at reasonable costs.

Предлагается построить CBM TOF на основе современных MRPC. Базовый элемент этого robust детектора — пачка резистивных пластин из стекла или керамики, которые разделены тонким зазором с газом. При достаточно высоком электрическом поле создаются очень равномерные ливни, которые можно считывать с помощью capacitive coupling.

The proposed structure of the CBM - TOF wall is based on detector layouts that have already been realized and tested in in-beam experiments and demonstrated MRPC counter timing resolutions below 60 ps with efficiencies above 95% at rates relevant for CBM (see chapter 7 for details). The basic element of the proposed wall are MRPC strip counters where a single avalanche is generating two signals at the two ends of a readout-electrode. The length of the readout - strip can be easily adjusted to the required granularity. The leading design goal of the proposed strip structures is operation stability and signal integrity. This is achieved by differential signal processing and the matching of the readout strip impedance to the input impedance of the newly developed preamplifier discriminator chip (PADI). Thus the number of spurious hits due to

reflections is minimized and an optimal response of the detectors is guaranteed with minimal dead time. This aspect is considered to be especially important for the high rate running conditions of CBM where even independent events will be overlapping in time space and any additional spurious hit will deteriorate the system performance.

In order to minimize the number of components, the CBM - TOF wall is designed to be composed of only 4 different types of MRPC counters, arranged into 6 different types of modules. The proposed CBM -TOF system is described in detail in chapter 3. The modules and counters are realized in 2 different technologies: 1. The modules placed at small polar angles close to the beampipe (M1 - M3) are housing counters (MRPC1,MRPC2) that employ a double stack HV - design. The counters are equipped with low resistivity glass and have electrodes of $32 \times 10 \text{ cm}^2$ and $32 \times 20 \text{ cm}^2$ dimensions, 10 gas gaps and a strip pitch of 4.7 mm. The front-end electronics is mounted outside of the gas volume. A total of 36.608 electronics channels is needed to readout this part of the TOF wall. The proposed design emphasize the full coverage of the solid angle and is rather conservative in terms of data integrity since each hit is registered by 2 readout channels and allows to determine the hit position with an accuracy of a few mm in each direction. More detailed measurements of the response under fully loaded conditions and time based simulations of the system response will show whether this strip solution could be replaced by a cheaper pad - type MRPC solution that is described in detail in the appendix E and requires only 27.792 electronic readout channels. A decision will be taken at latest in Dec. 2015 after careful analysis of test beam data that will be taken at GSI in October 2014 and at CERN SPS in April 2015. With currently available HI - beam, conditions close to the operating conditions of CBM will be realized. 2. For the remaining part of the wall (modules M4 - M6) two different sizes of a MRPC strip counter with active area of $32 \times 27 \text{ cm}^2$ and $32 \times 53 \text{ cm}^2$, and a strip pitch of 1 cm are proposed. These counters are equipped with low - resistivity glass or thin float glass electrodes according to the rate requirements and use a single HV stack with 8 gas gaps. In order to optimize the signal integrity the preamplifier is connected as directly as possible to the readout electrode and put into the gas volume of the modules. As largest benefit this arrangement offers the possibility to operate the detectors at lower discriminator thresholds and the lower pulse height results in turn also in a higher rate capability. The design of independent module units including the analog electronics also offers the necessary robustness to operate a system of 218 modules and 70.000 readout channels. The disadvantage of the proposed single stack architecture is the need of two high voltages ($\pm 11 \text{ kV}$) causing substantial costs. An alternate design with a double stack configuration of the high voltage is available and can be operated at about 5.6 kV. However, this counter configuration needs more glass electrodes (12 instead of 9) and the cost benefit from the HV has to be compared to the additional cost caused by additional electrodes especially when these are made from low resistivity glass. Counters of this type that are described in appendix F.2.4 will be evaluated in comparison to the proposed ones also in the upcoming HI - test beam campaigns addressing the system features of the designs. A decision will be taken at latest in Dec. 2015 after careful analysis of the test beam data. The high performance electronics for the whole CBM - TOF wall is independent of the choice of a specific module or counter type and is based on the PADI and the GET4 - ASIC chips developed by the CBM - TOF group (see sections 3.3.3 and 3.3.5). The combination of both chips in conjunction with a custom - designed clock distribution system (section 3.3.8) offers the possibility to build a large scale system with an electronics contribution to the timing resolution of about 30 ps. This system was tested using a readout controller that implements the full CBMnet - protocol and allows for online inspection and control of the GET4 - data. The details of the readout-chain are described in section 3.3. However, since up to now no long - term stably working GET4 - system could be demonstrated, an alternative backup solution is also included in this report. It is built on FPGA based TDCs that became available recently (see appendix G.2) and provides timing

resolution on the level of 10 ps. A free - streaming readout chain is not yet available, but is planned to be realized in the near future. As a consequence the number of necessary readout controllers would be substantially reduced leading eventually to a more simple and cost effective system. More R'n'D is required in the direction of an FPGA - based digitizer / readout system, specifically considering the radiation environment (see chapter 2.5). Since FPGA - based digitizers are offering a better resolution and more flexibility and would allow e.g. for an improvement of the double hit recognition the decision about the production of the GET4 - system will be taken only after a) the stability of the GET4 readout chain is proven, b) no competitive FPGA based solution is available until start of mass production (Dec.2015). Therefore R'n'D work on FPGA - based TDCs is continuing as part of the project. The proposal of a high performance TOF wall would be incomplete without the discussion of the time - zero (T_0) - reference that is needed for any velocity measurement. Details can be found in chapter 4 demonstrating that for all of the anticipated running scenarios solutions are available. For most of the heavy-ion reaction running at the initial SIS100 accelerator the software based T_0 - extraction is sufficient (see section 4.2). This method can be calibrated by making use of the diamond based start detector system developed for the HADES experiment (section 4.4) that is operational up to interaction rates of 100 kHz. As has been shown by Monte-Carlo simulations using the SHIELD event generator the quality of the T_0 - reference determination can be improved by installing the so called Beam Fragmentation T_0 Counter (BFTC) as the innermost part of the TOF wall (see section 4.3). It should cover the region from about 20 to 60 cm from the beam pipe (overlapping with the PSD acceptance). Using a very simple algorithm without tracking information a T_0 - signal with a precision of better than 50 ps can be achieved. A detector concept, anticipated to be able to operate in this harsh environment with fluxes being as high as 100 kHz/cm², built from very radiation-hard materials and possessing minimum gas aging characteristics, is suggested to be based on the ceramic electrodes (see section F.1). Single cell pads with the size of 2×2 cm² seem to fulfill the double hit, maximum rate and cross-talks requirements. Further R'n'D is necessary in order to prove the validity of the concept. The very large dynamical range in terms of incident beam energies that has to be covered by the CBM experiment reflects itself by the movable wall concept where the whole TOF - system is mounted on rails and can be placed at various distances. The rail system is sketched in section 5.1. It allows e.g. to balance the decay losses for kaons against the purity of an anticipated online kaon multiplicity selection. It also enables the servicing of the components independently from the other subsystems of CBM during shutdown periods.

1.2.7 Электромагнитный калориметр ECAL

1.2.8 Детектор PSD

1.3 CBM RICH поподробней

Глава 2

Геометрия

Процесс проектирования современной экспериментальной установки подразумевает разнообразное компьютерное моделирование этой установки. В первую очередь выполняется компьютерное геометрическое моделирование в трёхмерном пространстве с целью получения конструкторской документации и анализа расположения элементов в пространстве. Геометрическая модель для этих целей обычно строится средствами систем автоматизированного проектирования (САПР), в которых стандартным способом представления геометрической информации является граничное представление (BREP).

Далее представление геометрической информации в САПР мы будем называть просто BREP, хотя строго говоря это не совсем верно.

Также, как и в любой другой прикладной области, необходимо выполнять многочисленные расчёты, которые нередко требуют геометрическую модель в качестве входных данных. Так, например, в инженерно-конструкторской среде широкое распространение получил метод конечных элементов (МКЭ) для решения задач прочности и устойчивости механических конструкций, динамики жидкостей и газов и т.д. МКЭ получил своё название от способа разбиения расчётной области на элементарные блоки — конечные элементы. В простом случае расчётной областью является пространство заполненное материалом, т.е. сама деталь, а конечным элементом — тетраэдр. Такую модель, в которой деталь представлена множеством конечных элементов, называют КЭ-моделью. Существуют алгоритмы, позволяющие эффективно получить разбиение исходной геометрической модели, например представленной с помощью BREP и построенной в САПР, на конечные элементы. Многие алгоритмы основаны на триангуляции Делоне, разработанной в начале 20 века.

Отличительной особенностью сферы физики частиц является то, что в процессе проектирования установки помимо типовых расчётов требуется выполнение моделирования прохождения частиц через материал, которое чаще всего выполняется физиками, формулирующими требования к конструкции установки, но в общей массе не владеющими САПР. Такое моделирование достаточно специфично, но оно также выполняется над геометрической моделью, в идеальной ситуации — максимально подробной, совпадающей с полной детальной моделью, полученной инженерами-конструкторами с помощью САПР. Также стоит отметить, что процесс конструирования, в том числе получения инженерной геометрической модели, и процесс моделирования физики не имеют чётко определённого порядка и тесно между собой переплетены. В результате обоих процессов уточняются геометрические параметры деталей, компоновка узлов, применяемые материалы, и т.д. Это приводит к необходимости постоянного обмена геометрической информацией.

Как было сказано выше, инженеры для получения геометрической модели используют

САПР. Во многих физических лабораториях, включая CERN, GSI и ОИЯИ, применяется САПР CATIA v5. Моделирование взаимодействия частиц с материалом широко применяет метод Монте-Карло (MC) и реализовано в соответствующих программных пакетах, многие из которых основаны на фреймворках GEANT4 или GEANT3, разработанных в CERN. Также часто применяют поход Virtual Monte-Carlo (VMC), в котором все процедуры, связанные с геометрией, поручены системе ROOT. Все перечисленные физические пакеты (GEANT3, GEANT4, ROOT, далее коротко GEANT/ROOT) используют представление геометрии, принципиально отличающееся от BREP. Модели для GEANT/ROOT часто называют MC-моделями. Это отличие состоит из двух пунктов, подробно описанных в 2.1, и приводит к невозможности прямого обмена геометрическими моделями между физиками и инженерами. Существует теоретическая возможность прямой конвертации из представления, принятого в GEANT/ROOT, в BREP, однако в процессе работы не было найдено существующей реализации подобного перехода. Конвертация в обратном направлении до настоящего времени не была математически описана, хотя теоретически также представляется возможной.

Алгоритмы проведения частиц, реализованные в GEANT/ROOT, оптимизированы для соответствующего описания геометрии, применяемого в этих пакетах. Подходы геометрического моделирования, принятые в САПР, обеспечивают максимально эффективную работу как ЭВМ, так и инженеров, в частности за счёт того, что эти подходы интуитивно понятны человеку. Главным фактором против прямой конвертации в том или ином направлении является то, что она имеет малую практическую пользу. Одна и та же геометрическая модель с точки зрения разных задач может быть одновременно оптимальна и, наоборот, избыточна или недостаточна. Это просто понять на следующем примере. С точки зрения инженерного проекта массив болтов, вкрученных в корпус, представляет собой важную информацию. В чертежах и другой конструкторской документации ошибка в точном положении отверстий, их диаметре, типе резьбы болтов и т.д. может привести к невозможности собрать продукт после изготовления отдельных компонентов. В то же время, в САПР принято не хранить, и следовательно не визуализировать, витки резьбы с целью снижения нагрузки на графический адаптер ЭВМ. Это значит, что резьба присутствует только формально, в документации, а геометрическая модель имеет лишь условное обозначение резьбы в соответствующем месте. С точки зрения моделирования прохождения частиц через материал в зависимости от расположения в общей установке подобные подробности могут оказаться как критическими, так и наоборот излишними и вызывающими значительное увеличение времени выполнения моделирования. Так, например, резьба болта, находящегося близко к области, где проходит пучок, может оказать влияние на функционирование всей установки, а та же резьба где-то за пределами геометрического аксептанса не даст ровно никого эффекта может быть упрощена до цилиндра. Более того, без ущерба реалистичности моделирования упрощения могут носить неожиданно масштабный характер. Например, где-то рассматриваемый массив болтов может быть вообще проигнорирован, а пространство в отверстиях заполнено материалом корпуса.

В связи с этим в GEANT/ROOT принято иметь несколько моделей одной и той же установки, имеющих разный уровень подробностей. Чем выше уровень подробностей — тем больше времени занимает выполнение моделирования. Для оценочных расчётов удобно применять грубые модели, для точного определения каких-либо характеристик — подробные модели. В САПР же подобная проблема решается другим образом. Так, например, в САПР CATIA v5 присутствует возможность автоматического огрубления геометрической модели для снижения нагрузки на графический адаптер и повышения частоты кадров при динамической визуализации трёхмерных объектов. Это становится актуально, когда количество треугольников, которые необходимо визуализировать, составляет десятки миллионов.

Принимая во внимание развитие вычислительной техники, в особенности резкое повышение производительности графических карт, их доступность широким массам, и вообще увеличение их значимости в вычислениях общего назначения, представляется возможным разработка новых алгоритмов проведения частиц через материал, учитывающих особенности геометрического представления в САПР. Более того, возможна также некоторая корректировка подходов САПР к геометрическому моделированию с целью повышения совместимости с пакетами проведения частиц. Однако следует учитывать следующие факты, мешающие движению в данном направлении. Во-первых, САПР — это в большинстве своём коммерческое программное обеспечение с закрытым исходным кодом, а геометрическое ядро САПР — базовая составляющая, которую отлаживают десятилетиями. Внесение изменений в столь важную компоненту коммерческого продукта, вероятно, будет проблемным даже при наличии интереса со стороны фирмы-разработчика. Во-вторых, в обеих сферах накоплен огромный массив моделей, применяемых для поддержки изделий на всех этапах жизненного цикла, даже после окончания процесса проектирования. GEANT/ROOT модели могут применяться для выполнения моделирования даже после того, как физическая экспериментальная установка уже собрана.

Таким образом с целью упрощения взаимодействия физиков и инженеров было принятно решение не пытаться разработать конвертеры или какие-либо новые универсальные способы представления геометрии, а сосредоточиться на облегчении существующей процедуры за счёт плавной корректировки привычных методов и предоставления новых инструментов как физикам, так и инженерам. “CATIA-GDML geometry builder” — это как раз набор таких инструментов. Он описан в 2.2 вместе с предлагаемой организацией рабочего процесса и реальным случаем использования для проектирования детектора RICH эксперимента CBM.

2.1 Сравнение представления геометрии в GEANT/ROOT и САПР

Разница между двумя способами описания геометрической информации в САПР и пакетах моделирования прохождения частиц через материал GEANT/ROOT заключается в двух пунктах. Во-первых, отличается способ задания геометрических форм. В САПР применяется граничное представление (BREP), для описания которого используются понятия типа «поверхность», «грань», «ребро», «кривая», и за которыми стоят соответствующие уравнения, описывающие эти объекты в пространстве. В GEANT/ROOT применяется конструктивная твердотельная геометрия (CSG), которая оперирует понятиями «примитив» и «Булева операция». Очевидно, что и за этими объектами также стоят конкретные уравнения, описывающие кривые и поверхности, однако есть существенное различие описанное ниже. Во-вторых, отличается способ задания взаимоотношения форм в пространстве. В САПР, по аналогии с тем, как человек воспринимает окружающий мир, присутствует некоторое бесконечное окружающее пространство без материала, а все предметы находятся в этом пространстве. Невозможна такая ситуация, чтобы один объект находился внутри другого — в таком случае подразумевается, что во втором есть соответствующая полость, освобождающая место под первый объект. При этом получается, что границы двух тел, стоящих рядом, совпадают, т.е происходит дублирование информации. ??? В силу математики это приводит к неприятным эффектам. ??? В GEANT/ROOT для описания взаимоотношения форм используется иерархия объёмов. Это объясняется тем, что такой метод более удобен для описания геометрии, где главной задачей является однозначное задание материала в каждой точке пространства. Вводится понятие объёма — сущности, имеющей форму и мате-

риал. Из всех объёмов выбирается один, называемый объёмом верхнего уровня, а остальные помещены либо в него, либо в какой-то другой, формируя таким образом дерево объёмов. При этом не происходит дублирования границ и нет упомянутых выше эффектов.

2.1.1 Представление геометрии в САПР

В BREP есть два типа понятий — геометрические («точка», «кривая», «поверхность») и топологические («вершина», «ребро», «грань»). «Точка» — это тройка координат в некоторой системе координат. «Кривая» — это уравнение, задающее множество точек, принадлежащих данной кривой. Кривую удобно описать с помощью параметрического уравнения от одной переменной. «Поверхность» — это уравнение, задающее множество точек, принадлежащих данной поверхности. Соответственно, поверхность удобно описать с помощью параметрического уравнения от двух переменных. Топологические сущности задаются на базе геометрических. «Вершина» лежит в некоторой геометрической точке. «Ребро» лежит на некоторой геометрической кривой и ограничено двумя вершинами. Очевидно, что эти вершины должны принадлежать кривой, то есть и соответствующие геометрические точки должны принадлежать кривой. «Грань» лежит на некоторой поверхности и ограничена замкнутым циклом из рёбер. Также очевидно, что эти рёбра должны принадлежать поверхности, как и кривые, на которых они лежат, как и вершины и точки, ограничивающие эти рёбра. Замкнутая оболочка из граней с указанием внешних сторон этих граней ограничивает некоторую область пространства, называемую «телом».

В соответствии с BREP параллелепипед (к которому эквивалентен примитив box в CSG) задаётся следующим образом.

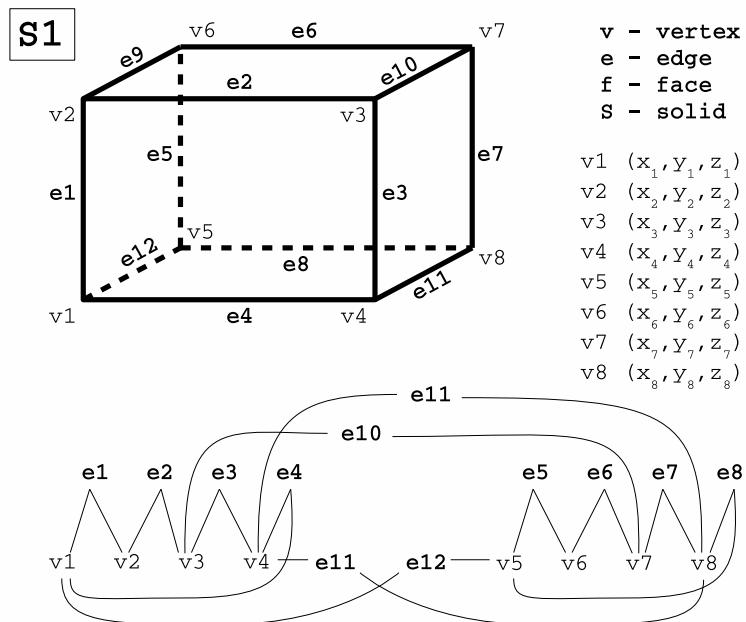


Рис. 2.1: Описание вершин и рёбер примитива box методами BREP.

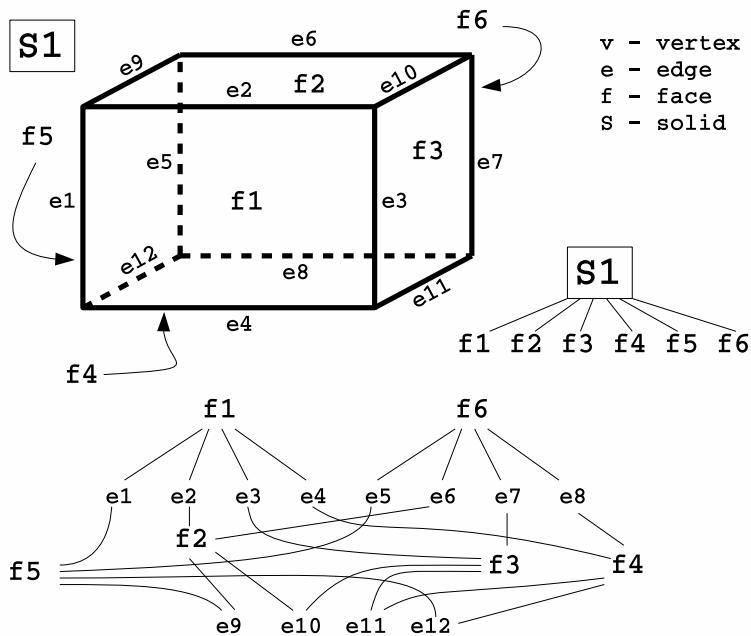


Рис. 2.2: Описание граней и тела примитива box методами BREP.

??? Краткое описание картинки

Стоит однако отметить, что человек, создающий геометрическую модель в САПР, хотя и может выполнять построения в соответствии с базовыми принципами BREP, чаще всего применяет интуитивно понятные формообразования, из которых система точно формирует BREP модель в памяти ЭВМ, которая также необходима для получения триангулированной геометрии для визуализации на дисплее ЭВМ. Есть 4 базовых формообразования и 4 им обратных (с вычитанием) — «выдавливание», «вращение», «протягивание» и «тело по сечениям». Многие другие формообразования, такие как фаски и скругления, разрезы, отверстия, внутри на самом деле являются лишь вариациями перечисленных. Последовательность формообразований, выполненных пользователем для получения итоговой формы, сохраняется в виде дерева построения модели, напоминающего историю построения, но позволяющего навигацию и редактирование. Дерево часто доступно пользователю в основном рабочем окне интерфейса САПР. Однако бывают случаи, когда история построения теряется, например при передаче модели из одной САПР в другую. Таким образом, в результате работы инженера получается модель, описанная с помощью BREP, и во многих случаях имеющая также и дерево построения.

В инженерной практике принято проектировать и соответственно строить 3d-модели, объединяя в сборки детали и другие сборки. Отсюда вытекает, что во многих САПР, в том числе в CATIA v5, существуют стандартные объекты, обозначающие детали и сборки. Например в САПР CATIA v5 существует отдельный тип документа CATPart для детали и отдельный тип документа CATProduct для сборки. Внутри документа типа CATPart есть минимальный набор обязательных элементов — 3 стандартные взаимно перпендикулярные плоскости в начале системы координат детали и главное тело детали, по умолчанию называемое PartBody. В документе типа CATProduct присутствует возможность добавлять в качестве дочерних компонентов либо документы CATPart либо другие документы CATProduct. В 2.2 описывается, как соотносятся перечисленные сущности CATIA v5 с понятиями гео-

метрической подсистемы GEANT/ROOT.

Во многих САПР, в том числе и CATIA v5 присутствует возможность так называемого контекстного редактирования компонентов. Это означает, что пользователь во время работы над сборкой в документе типа CATProduct, имеющей в качестве дочерних компонентов детали в файлах типа CATPart, может также редактировать детали, не переключая активный документ. Эта возможность широко используется в “CATIA-GDM geometry builder” — большая часть работы выполняется в контексте единственного продукта, что с точки зрения пользователя аналогично работе над всей экспериментальной установкой.

2.1.2 Представление геометрии в GEANT/ROOT

Для описания геометрических форм в пакетах GEANT/ROOT применяется CSG. В качестве строительных блоков в CSG используются примитивы из списка реализованных в системе. Список примитивов включает в себя как относительно простые примитивы типа параллелепипеда (box), сегмента цилиндра (tubs), сегмента конуса (cons), так и достаточно сложные, типа эллипсоида, параболоида, скрученных (twisted) примитивов. Принимая во внимание тот факт, что геометрия в GEANT/ROOT нужна для выполнения моделирования взаимодействия частиц с материалом, можно сказать, что примитив — это объект, имеющий геометрическое представление и для которого реализовано решение геометрических задач, возникающих при моделировании. Среди таких геометрических задач можно отметить задачу нахождения расстояния до ближайшей границы примитива от некоторой точки внутри объёма, в одном заданном направлении или в любом возможном направлении. Эту задачу необходимо решать многократно в процессе проведения частицы для того, чтобы определить так называемый максимальный допустимый геометрический шаг. В результате моделирования физических процессов получается максимальный допустимый шаг из соображений физики. Для того чтобы собственно изменить координату частицы из этих двух шагов выбирается минимальный. Каждый примитив, как в GEANT4, так и в ROOT, реализован как отдельный C++ класс, имеющий свои геометрические параметры среди членов данных, и решение описанных выше геометрических задач среди методов. Следует, отметить, что есть только параметры примитива, но нет никакого описания типа BREP. Уравнения границ фигурируют лишь в неявном виде в коде методов для решения геометрических задач. Более подробно примитивы описаны в 2.2.2.

Форма может быть описана как результат Булевой операции над примитивами или другими Булевыми операциями. Есть три Булевые операции — объединение (union), вычитание (subtraction) и пересечение (intersection). Булевые операции позволяют задать практически любую геометрическую форму, имеющую границы из тех, что применяются в примитивах. При этом не требуется дополнительной реализации решения геометрических задач, т.к. удаётся комбинировать то, что реализовано в примитивах. Более подробно применение Булевых операций для задания формы объёма описано в 2.2.3.

Таким образом наблюдается некоторая аналогия между BREP и CSG, заключающаяся в том, что в любом случае сложное тело или базовый примитив имеет некоторые границы, заданные аналитическими выражениями. Корни этой аналогии лежат в фундаментальной математике. Однако решающая разница заключается в том, что для примитива эти границы чётко определены и имеется лишь ограниченное число параметров, позволяющих изменять форму примитива.

Вторая составляющая геометрического представления в GEANT/ROOT это иерархия объёмов. Введём понятия логического и физического объёмов, формы и материала. Логический объём, или просто объём это базовый элемент для построения иерархии объёмов. Объём описывает непозиционированный объект и всё, что находится внутри него. Объём

характеризуется формой и материалом. Форма — это заданные с помощью CSG границы пространства, по методу, описанному выше. Материал включает в себя описание химического состава, плотности, и т.д. При помещении одного логического объёма в другой, например объёма A в объём B , образуется так называемый физический объём, или узел, B_1 , который обозначается взаимоотношение A и B как материнский-дочерний и характеризуется некоторой матрицей позиционирования B внутри A .

Продолжать!???

2.2 “CATIA-GDML geometry builder”

“CATIA-GDML geometry builder” (далее просто “Builder”) представляет собой набор документов-шаблонов и макропрограмм для САПР CATIA v5 вместе с настройками окружения и инструкциями к применению стандартных средств CATIA v5. “Builder” ставит своей задачей упростить процесс создания CSG моделей с иерархией объёмов, напрямую совместимых с GEANT/ROOT.

Центральная идея “Builder” заключается в правилах соответствия сущностей CATIA v5 и сущностей геометрии в GEANT/ROOT. Это соответствие делает возможным конвертацию MC-модели в CATIA v5 в любой внешний файл с целью дальнейшего импорта в ROOT/GEANT. В качестве формата для обмена был выбран XML-подобный формат GDML (geometry description markup language), разработанный в CERN, для которого в GEANT4 и ROOT реализованы методы импорта и экспорта.

Вся геометрия установки создаётся в одном документе типа CATProduct. Объёму соответствует деталь, хранящаяся в файле типа CATPart. Форме соответствует главное тело детали, по умолчанию называемое PartBody. В CATIA не записывается описание материала так, как это принято в GEANT/ROOT, а сохраняется только имя материала в пользовательском параметре Material. Это возможно по той причине, что существует практика хранить описание материалов во внешнем файле или базе данных, считывать его перед выполнением моделирования и приписывать объёмам в соответствии с именами. Для обозначения физического объёма B внутри A в структуре документа, описывающего объём A , создаются тела Body.B.* , где * по умолчанию обозначает номер вхождения, но допускается запись любой идентифицирующей строки.

Может быть частично перенести в описание методов описания геометрии в GEANT/ROOT. Также в “Builder” предусмотрена возможность задания геометрии некоторыми продвинутыми методами, специфичными для GEANT/ROOT. В GEANT/ROOT существует тип объёмов, называемый Assembly, который характеризуется тем, что он не имеет формы и материала. Практически объём типа Assembly является контейнером без границ, который объединяет свои дочерние объёмы, что особенно удобно как минимум в двух случаях. Во-первых, если необходимо многократно позиционировать группу объёмов, которую невозможно охватить простой формой. Во-вторых, если преобразование координат при позиционировании одного или группы объёмов имеет сложную структуру и удобно представить его как суперпозицию двух преобразований. Как частный случай можно упомянуть ситуацию, когда какой-либо параметр преобразования является параметром модели (см. секцию 2.2.4)

Один из плюсов “Builder” заключается в том, что пользователю предоставляется возможность работать с полноценной инженерной моделью и MC-моделью в одной и той же среде, имеющей широкие возможности для анализа и редактирования геометрии. “Builder” не ставит своей задачей перевод модели из одного геометрического представления в другой, но значительно ускоряет процесс создания одной геометрии, на основе другой. Также важно

отметить, что подходы к геометрическому моделированию в САПР подразумевают широкое использование параметров — практически все размеры, значения поворотов и сдвигов, количество вхождений в массивы и прочие числа, определяющие форму и структуру модели, могут подвергаться изменению на любом этапе. Если в процессе построения модели этот принцип параметризированного моделирования нарушается, САПР предупреждает пользователя перед выполнением операции, которая приведёт к разрыву связи с параметром. В CATIA v5 при изменении каких-либо параметров геометрическая модель перестраивается интерактивно — от долей секунды до нескольких секунд в зависимости от сложности модели. Эта стандартная черта САПР очень удобна при работе с МС-моделями и отсутствует, например, в GEANT и ROOT.

“Builder” включает в себя файлы, в которых специальным образом построены примитивы GEANT/ROOT, позволяющие пользователю при построении МС-модели не вникать в подробности реализации, а использовать их практически как и в процессе создания геометрии средствами ROOT или GEANT. “Builder” также включает в себя макропрограммы для CATIA v5, которые также ставят своей задачей сделать процесс построения геометрии в “Builder” максимально похожим на процесс построения геометрии в GEANT или ROOT. Основной макрос — это конвертер «CATIA2GDML», который проектирует дерево построения модели в CATIA в GDML файл. Также разработан обратный конвертер «GDML2CATIA» для импорта GDML файлов.

Целевая аудитория “CATIA-GDML geometry builder” — физики, владеющие CATIA v5 на базовом уровне, и инженеры, продвинутые пользователи САПР, изучившие способ представления геометрии в GEANT/ROOT хотя бы на теоретическом уровне. **Есть опыт, который показывает, что для достижения такого уровня как физикам, так и инженерам, достаточно прохождения двухнедельного курса.**

Предлагается новый алгоритм работы, в котором “Builder” используется как многофункциональный инструмент. Описанный ниже алгоритм сформулирован на успешном опыте разработки СВМ RICH на протяжении 3 (4) лет.

Задача создания и поддержания актуальной МС-модели поручается ответственному человеку, владеющему CATIA, GEANT/ROOT и “CATIA-GDML geometry builder”. В зависимости от того, какая информация и в каких файлах имеется к началу работы, алгоритм немного различается. Если разработка ведётся в нуля и нет никаких данных в ЭВМ, что возможно, например, когда проект находится на таком этапе, когда нужно выполнить грубое моделирование, показывающее принципиальную возможность реализации, то наиболее оптимальный способ — сразу строить МС-модель в CATIA средствами “Builder”. Если, скажем, проект находится на раннем этапе разработки и уже имеется какая-то приблизительная САПР модель, то рекомендуется импортировать её стандартными средствами CATIA, чтобы затем на её основе построить МС-модель в CATIA в автоматизированном режиме с помощью средств “Builder”. Инженерную геометрию можно импортировать практически из любой САПР, например с помощью широко распространённого формата STEP. Третий распространённый случай это когда уже имеется некоторая МС-модель в конечной системе моделирования. Как в GEANT4, так и в ROOT имеется стандартная возможность экспортieren геометрию в GDML файл без потери информации. Этую возможность могут наследовать все дочерние пакеты (как FairRoot и далее CbmRoot), но для этого необходимо явно активировать функциональность GDML. В этом случае можно импортировать модель в CATIA в МС-формате, однако иногда требуются некоторые дополнительные ручные операции после импорта. Они выполняются однократно и лишь делают структуру документа более оптимальной, но не изменяют геометрию.

Во всех этих алгоритмах, независимо от типа и количества исходных данных, получаются файлы CATIA в формате “Builder”, которые в дальнейшем будут являться основны-

ми (первичными) файлами для получения рабочей МС-модели в экспериментальном пакете, которым в случае CBM RICH является CbmRoot. Модель из CATIA экспортируется в GDML файл, который не требует каких-либо последующих изменений в структуре. Для достижения этого условия была проведена огромная работа по мере разработки МС-модели CBM RICH. Допускается и даже рекомендуется текстовое редактирование GDML файла, но только для изменения значений параметров в define секции у параметризованных моделей. Затем, по желанию коллaborации, GDML файл может быть конвертирован в бинарный ROOT-файл, который содержит геометрию, которую невозможно редактировать. Это защищает модель от случайных изменений, что особенно актуально в случае параметризованных моделей. Соответственно, если требуется изменить значения параметров, пользователь может отредактировать GDML файл и экспортить в новый ROOT файл. Практика показывает, что в случае, если требуется множество файлов с МС-геометрией, то обязательно нужно писать комментарии — либо в самом GDML файле, либо в текстовом файле рядом с GDML/ROOT файлом. Обычно в коллаборации вводят правила именования файлов.

2.2.1 Переменные окружения

Для максимального скрытия подробностей реализации от пользователей и минимизации их действий, “CATIA-GDML geometry builder” использует возможности CATIA v5 по настройке окружения. Пакет поставляется с папкой CATSettings, которая должна быть указана как папка с настройками при конфигурировании окружения перед началом использования пакета. Инструкция по настройке поставляется вместе с “Builder”. В настройках окружения помимо прочего определены переменные окружения, значения которых активно используются из кода макропрограмм. В первую очередь необходимо задать переменную BuilderPath, в которой необходимо указать путь к папке с “Builder”, полученной тем или иным способом. Помимо того, что без настроенного окружения не будут работать некоторые макропрограммы, не будет работать и пользовательские панели инструментов “CATIA-GDML geometry builder” в CATIA. проверить ??? Поэтому настоятельно рекомендуется перед использованием пакета выполнить настройку.

2.2.2 Примитивы в “Builder”

Любой примитив можно построить стандартными средствами САПР, используя эскизы и формообразования, но в таком случае конвертер не сможет автоматически определить, является ли построенная форма примитивом и, если да, определить параметры примитива. По этой причине был разработан принцип хранения формы примитива в МС-модели с помощью средства CATIA v5, называемого User-Defined Feature (UDF), и средства для автоматизации создания примитивов — макросы «AddShape» и «Poly». Каждый примитив реализован в своём файле типа CATPart, в котором создаётся описание UDF, превращая этот файл в шаблон. Некоторые объекты модели, в случае примитивов — некоторые стандартные плоскости, и параметры модели объявляются «внешними». Далее в другом документе возможно создать вхождение формы, определённой в файле-шаблоне, вызвав соответствующее формообразование. При этом в текущем документе потребуется лишь выбрать необходимые элементы, с которыми будут совпадать «внешние» объекты шаблона, и задать значения параметрам создаваемого вхождения. Примитивы в “Builder” построены так, чтобы этими совпадающими элементами были стандартные плоскости. За счёт этого при создании вхождения достаточно нажать кнопку use identical names и CATIA автоматически совместит правильные элементы. Более того, использование макроса «AddNewPart» автоматизирует этот процесс.

Из-за того, что полипримитивы polycone и polyhedra имеют переменное число секций, оказалось невозможным реализовать их так же как и остальные примитивы с помощью UDF. Чтобы представить полипримитив в дереве построения модели в CATIA используется последовательность чередующихся UDF, описывающих отдельные секции, и формообразований translate, необходимых для обеспечения сдвига секций относительно друг друга. В случае polycone секция моделируется стандартным примитивом cons (cone section), а для polyhedra аналогичной стандартной формы не существует, поэтому она была введена искусственно. Примитив hedra существует только в “Builder”, он нужен для создания polyhedra и не может быть использован где-либо ещё.

2.2.3 Булевы операции

Форма объёма может быть задана как примитив либо Булева операция над примитивами или результатами Булевых операций, причём работает такое правило, что первый операнд каждой операции не должен подвергаться каким-либо преобразованиям координат. При создании Булевых операций матрица позиционирования второго операнда относительно первого задаётся трёмя поворотами вокруг стандартных фиксированных осей и одним сдвигом — также, как и при позиционировании дочернего объёма в материнском. Разница заключается в том, что порядок поворотов обратный — сначала вокруг X, затем вокруг Y и вокруг Z. Сдвиг выполняется после поворотов.

Определено три Булевые операции (приведены оригинальные названия в GEANT/ROOT и соответствующие названия в CATIA v5):

- Объединение (Union, Add);
- Вычитание (Subtraction, Remove);
- Пересечение (Intersection, Intersect).

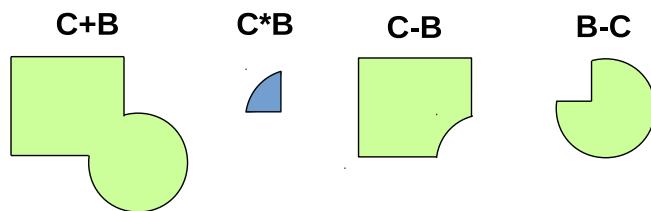


Рис. 2.3: Булевы операции.

Операндом и результатом Булевой операции в МС-модели в CATIA v5 по правилам “Builder” является тело — объект типа Body. Для того, чтобы создать Булеву операцию необходимо к телу, обозначающему результат, присоединить первый операнд с помощью формообразования Assemble, а второй с помощью одного из формообразований Add, Remove, Intersect в зависимости от типа операции. Если операнд это просто примитив, то в теле будет одно формообразование типа UDF. Второй операнд может также иметь преобразования координат.

??? картинка и описание примера с несколькими уровнями вложенности

??? В принципе хочется добавить что-то ещё, но что конкретно?

2.2.4 Параметризация

Одна из наиболее важных возможностей “CATIA-GDML geometry builder” — это возможность создания параметризованных геометрических МС-моделей. У параметризованной модели имеются входные параметры и формулы, задающие зависимость между этими входными параметрами и внутренними переменными, такими как параметры примитивов, значения поворотов и смещений. Данная концепция хорошо ложится на методы работы с геометрией в САПР, особенно CATIA v5. Также параметризация поддерживается форматом GDML и импортёрами GEANT4(???) и ROOT.

В модели CATIA v5 можно вводить пользовательские параметры как в документах типа CATProduct, так и в документах типа CATPart. Причём сборка в CATProduct файле может иметь свои пользовательские параметры и формулы, а дочерние компоненты в CATPart файлах — свои. Обязательное требование “Builder” таково, что все параметры и формулы должны находиться в верхнем продукте. CATIA v5 позволяет задавать зависимости между любыми параметрами, в том числе внутренними, не являющимися пользовательскими, однако для успешного экспорта в GDML файл формула должна в левой части иметь параметр примитива или угол поворота или значение сдвига, а в правой части — формулу только над пользовательскими параметрами. Пользовательский параметр CATIA v5, экспортруемый в переменную в GDML должен обязательно иметь безразмерный тип Real. В связи с этим имеются правила оформления формул и приведения единиц измерения. Также имеется стандартная переменная DEGtoRAD для перевода значения углов из градусов в радианы.

На выходе получается GDML файл, у которого в define секции есть тэги variable, обозначающие входные параметры модели со значениями. При импорте параметризованной геометрии из GDML в ROOT все значения внутренних переменных рассчитываются в соответствии с формулами по значениям входных параметров и параметризация теряется. Следовательно значения входных параметров должны задаваться пользователем непосредственно в GDML файле перед импортом в конечную систему.

2.2.5 Тесселированная геометрия

Помимо BREP и CSG существует ещё один способ представить геометрическую модель в ЭВМ, который не менее распространён, но имеет совершенно другую область применения. Тесселированная геометрия, и как частный случай — триангулированная геометрия, в основном применяется в мультиплексии, где главное внимание уделяется не математически точному описанию границ поверхностей, а реалистичной визуализации модели на дисплее. В общем случае такая модель не требует наличия замкнутой оболочки для задания тела, допускается даже отсутствие части границ, если это не влияет на изображение на экране. Как было отмечено выше, при работе с BREP или CSG, триангулированная геометрия всегда строится графической подсистемой того или иного пакета моделирования, т.к. архитектура графической карты требует на входе именно массив треугольников.

В промышленном геометрическом моделировании триангулированная геометрия применяется не только для визуализации. Например в стереолитографии, и вообще прототипировании, получил распространение формат обмена триангулированной геометрией STL, представляющий собой текстовый файл, в котором перечислены треугольники как группы по три вершины, а вершина задаётся тройкой координат. В GEANT/ROOT реализована поддержка триангулированной геометрии, однако специфика этих систем накладывает строгое ограничение — с помощью треугольников должна быть описана только замкнутая оболочка, ограничивающая область пространства с заданным материалом, т.е. по сути объём. Для реализации поддержки триангулированной геометрии в “CATIA-GDML geometry

“builder” были разработаны правила её хранения в модели и необходимый код в конвертерах «CATIA2GDML» и «GDML2CATIA».

??? описание реализации в билдере

2.2.6 Макропрограммы для CATIA v5

Макропрограммы для CATIA v5 написаны на VBA с применением CATIA API. Все макропрограммы, кроме «AddShape» и «Poly», доступны пользователю в режиме работы над сборкой. В CATIA v5 различают открытый документ (верхний в дереве в текущем окне), активный документ (синий), выделенный объект (оранжевый) и рабочий объект (подчёркнутый). Пользователь может выполнить все операции, необходимые для получения МС-модели, самостоятельно без применения макропрограмм, но в этом случае велика вероятность упустить какой-либо шаг, что приведёт к ошибке, которую сложно диагностировать.

В МС-модели в CATIA есть строгие правила именования. Применение макросов избавляет пользователя от необходимости контролировать имена объектов в документах. Все имена, сгенерированные при использовании “Builder” не конфликтуют между собой и позволяют получить корректный GDML файл на выходе. Практически везде пользователь имеет право изменять суффиксы, не изменения основного названия, несущего информацию о типе объекта — формообразования, тела, и т.д. Однако в редких случаях суффикс имеет решающее значение, как например в именах поворотах (напр, “Rotate.X”) суффикс несёт информацию об оси поворота.

В процессе разработки “Builder” был выработан стандартный алгоритм создания геометрии. Первый шаг — создание нового документа типа CATProduct, который в дальнейшем будет единственным продуктом, и его сохранение на диск. Этот продукт будет представлять модель всей экспериментальной установки. Второй этап — наполнение продукта описанием объёмов без описания взаимосвязей между ними. Для этого используется макрос «AddNewPart», который автоматически открывает в отдельном окне новый документ типа CATPart, сформированный из специального шаблона и соответствующий создаваемому объёму. Система переходит в режим редактирования детали, где доступны только два макрона «AddShape» и «Poly» для создания формы объёма. Здесь же можно и задать имя материала объёма. По окончании редактирования нового объёма в отдельном окне пользователь должен сохранить активный документ и закрыть это окно. CATIA при этом возвращается к редактированию продукта. После того, как созданы объёмы, заданы формы и, возможно, имена материалов, алгоритм подразумевает задание иерархии объёмов, то есть позиционирование одних объёмов в других. Для этого в “Builder” существует целый ряд макропрограмм для создания различных типов взаимосвязей — «Inserter», «ArrayMaker», «Replica». После того, как выполнено размещение дочернего объёма A в материнском объёме B, пользователь может указать поворот и сдвиг, задающие матрицу позиционирования A в B. Для упрощения расчётов в некоторых случаях очень удобно применять макропрограммы «PointToPointAligner» («Pt2PtAligner»), «Mover» и «Measure». Для удобного редактирования материалов всех объёмов был разработан менеджер материалов «MaterialsManager», который обычно имеет смысл вызывать перед экспортом для проверки ранее заданных имён материалов, либо назначения новых. Также перед экспортом рекомендуется проверить модель на наличие ошибок с помощью макроса «Checker». В конце выполняется экспорт макросом «CATIA2GDML». Отдельно стоят макропрограммы «Duplicator» для создания множественных идентичных, но не связанных, параметризованных подсборок и обратный конвертер «GDML2CATIA» для импорта GDML файла.

Для комфортной работы с “Builder” в поставке также имеются файлы для настройки окружения CATIA. Использования окружения в принципе не обязательно, но часть функ-

ционала зависит от путей к файлам, которые прописаны в переменных окружения, поэтому настоятельно рекомендуется перед использованием “Builder” выполнить настройку, следуя инструкции, поставляемой в пакете.

«AddNewPart»

Данная макропрограмма автоматизирует создание нового документа типа CATPart на основе шаблона, содержащего необходимые элементы — публикация главного тела детали, называемая PartBody, пользовательский параметр под названием Material со значением по умолчанию vacsim. Для удобства работы в файле шаблона и во всех генерируемых новых документах типа CATPart погашены стандартные плоскости. Созданный документ автоматически сохраняется на диск в папку, в которой выполняется построение модели, и которая автоматически определяется из открытого документа типа CATProduct в момент вызова макроса. Путь можно изменить в окне графического интерфейса. Также созданная деталь добавляется в качестве компонента в открытую сборку. Новый документ открывается в дочернем окне CATIA и пользователь может как начать работу над объёмом, так и оставить его пустым и отложить редактирование. Достаточно закрыть это окно и система перейдёт обратно к редактированию сборки.

«AddShape»

«AddShape» используется для создания примитивов, в случае необходимости вместе с поворотами и сдвигом. Макропрограмма играет роль интерфейса между пользователем и файлами примитивов. При запуске макроса выводится окно со списком доступных примитивов, по нажатии на кнопку “создать” в рабочее тело детали вставляется выбранный примитив со значениями параметров по умолчанию. Если на форме графического интерфейса выбраны флаги создания поворотов и сдвига, то создаются соответствующие формообразования. Если форма объёма состоит из одного примитива, то повороты и смещение запрещены. Они имеют смысл только при содании второго операнда Булевой операции. Список примитивов формируется ??? проверить! ??? из списка файлов, имеющихся в папке примитивов, определённой с помощью относительного пути в BuilderPath.

«Poly»

В силу ограничений CATIA нет возможности представить полипримитивы (polycone и polyhedra) с помощью тех же средств, что и остальные примитивы, поэтому для них была разработана специальная структура дерева и правила именования. Для автоматизации построения полипримитивов в соответствии с этой структурой предоставляется макрос «Poly». Секции поликонуса представлены стандартными конусами. В случае polyhedra для представления секции используется hedra — специальный примитив, не поддерживаемый GEANT/ROOT.

«Inserter»

Макрос «Inserter» — это инструмент для помещения одного выбранного объёма в другой. Также можно сказать, что «Inserter» создаёт физический объём, задающий связь материнский-дочерний между двумя существующими логическими объёмами. «Inserter» — возможно, самый используемый макрос, в результате работы которого в документе типа CATPart, представляющем материнский объём, создаётся тело с именем “Body.B.*”, где *B* —

имя дочернего объёма. Внутри этого тела имеется ссылка на публикацию PartBody документа типа CATPart, представляющего объём B , и элементы преобразования типа Rotate и Translate — три поворота и сдвиг, задающие матрицу позиционирования B внутри A . Повороты выполняются вокруг фиксированных стандартных осей, а порядок строго определён — сначала вокруг оси Z , затем вокруг оси Y и в конце вокруг оси X . Следует отметить, что такие повороты не являются преобразованиями Эйлера, где система координат вращается вместе с телом и оси меняют своё направление.

«ArrayMaker»

Макрос «ArrayMaker» схож с «Inserter» по идеи и реализации. После выполнения вставки дочернего объёма в материнский, к созданному телу добавляется формообразование pattern вдоль указанной оси и с указанными шагом и количеством вхождений.

«Replica»

Одна из продвинутых возможностей геометрической подсистемы GEANT/ROOT — это деление объёмов. В GEANT4 эта возможность называется *replica*, а в ROOT — *division*. Суть заключается в том, что допускается деление некоторого объёма путём разрезания через равные промежутки вдоль одной из четырёх осей — X , Y , Z и ϕ , где ϕ — круговое направление. В результате деления получается набор одинаковых под-объёмов — долек, которые можно рассматривать как независимые вхождения одного объёма и позиционировать внутри другие объёмы. Отличие долек от обычного объёма заключается в том, что для долек оптимизирована реализация проведения частиц, (??? сформулировать лучше) Деление возможно только для ограниченного числа форм, таких, что все доли имеют одинаковую форму. Box — в любом из трёх линейных направлений X , Y или Z . Tubs — вдоль оси цилиндра, то есть вдоль линейной оси Z , или вдоль круговой оси ϕ . Дальше продолжать???

За счёт того ограничения, что формы долек должны быть одинаковы, для описания разделённых объёмов в CATIA решено использовать тот же принцип, что и для описания массивов с тем отличием, что имя формообразования *pattern* должно соответствовать шаблону “*Replica.Axis*”, где *Axis* — ось реплицирования. За счёт того ограничения, что долки должны заполнять всё пространство материнского объёма возможен автоматический расчёт геометрических размеров долек по двум входным параметрам — направлению деления и количеству.

«PointToPointAligner»

«*PointToPointAligner*» автоматизирует процесс позиционирования дочернего объёма внутри материнского. При вложении одного объёма в другой необходимо задать положение дочернего объёма в материнском. В “*Builder*” для этого используется три формообразования типа Rotate — последовательные повороты вокруг трёх фиксированных стандартных осей Z , Y и X — одно типа Translate — параллельный сдвиг. Пользователь должен каким-то образом рассчитать значения углов и координаты сдвига. Типовая процедура определения этих значений заключается в том, что пользователь использует стандартные средства CATIA для измерения углов и расстояний и затем вручную записывает эти значения в соответствующие параметры. В случае сложного поворота практически невозможно получить углы поворота прямым измерением, требуется активное интеллектуальное участие пользователя. Определение смещения, которое выполняется после всех поворотов, прямолинейно, но требует выполнения достаточно большого количества механических операций, которые автоматизированы в «*PointToPointAligner*».

Этот макрос предоставляет пользователю возможность выбрать две точки, которые совместятся при движении первой ко второй. Первая точка должна быть вершиной, принадлежащей телу, обозначающему дочерний объём — только в этом случае в макропрограмме представляется возможным определить, какому телу принадлежит выбранная вершина, чтобы выбрать формообразования, описывающие матрицу позиционирования, которую необходимо изменить. Вторая точка может быть полученной в результате любой операции — это может быть как вершина тела, так и каркасный элемент. Для неё определяются только координаты, чтобы рассчитать сдвиг как разность координат двух точек.

«Mover»

Нередко возникает такая ситуация, что требуется подвинуть группу дочерних объёмов внутри одного материнского на одинаковое расстояние. Если использовать существующие средства CATIA, то нет никакого способа выполнить этот сдвиг для нескольких объёмов сразу. В ручном режиме пользователю требуется осуществлять сдвиг для каждого объёма отдельно путём изменения значений параметров формообразования Translate, отвечающего за позиционирование дочернего объёма в материнском. Чтобы автоматизировать этот процесс был разработан «Mover». При использовании «Mover» пользователь выбирает какие дочерние объёмы он хочет подвинуть и на какое расстояние в графическом интерфейсе.

«Measure»

Макропрограмма «Measure» автоматизирует процесс измерения и записи результата измерения в какой-либо параметр модели. Необходимость выполнения подобной операции особенно часто возникает при работе одновременно с исходной САПР моделью и разрабатываемой МС-моделью. Если требуется измерить какое-либо расстояние между объектами или размер какого-либо геометрического элемента, то пользователь может воспользоваться стандартными средствами CATIA “Measure between” и “Measure item” соответственно. В результате появляется окно с результатами измерения, обычно включая компоненты по координатам. Чтобы перенести эти результаты измерения куда-либо необходимо выделить и скопировать значение из поля, закрыть окно измерения, найти параметр в МС-модели, открыть его для редактирования и вставить в качестве значения содержимое буфера обмена. При использовании «Measure» список параметров модели отображается в окне графического интерфейса. Пользователь сначала выбирает тип измерения, затем выбирает измеряемые объекты в области геометрии. Результаты измерения выводятся в окне графического интерфейса и для того, чтобы записать выбранный результат в какой-либо параметр МС-модели, достаточно выбрать его в списке параметров.

«MaterialsManager»

Приложение «MaterialsManager» предоставляет пользователю возможность изменять материалы отдельных объёмов, находясь на уровне верхнего продукта. Это избавляет от необходимости часто переключаться между документами либо режимами работы в случае контекстного редактирования. Также заметным преимуществом использования «MaterialsManager» является наглядность — информация о материалах всех объемов представляется в компактном списке, присутствует возможность быстро изменять значения в нескольких элементах списка. Помимо этого, наличие «MaterialsManager» позволяет отложить работу с материалами на последний этап. Использование шаблона файла детали предотвращает от того, что пользователь вообще не укажет материал объема — по умолчанию указан вакуум (vacuum). В графическом интерфейсе «MaterialsManager» представлена таблица из трёх столбцов —

имя документа (объём), текущий материал, новый материал. Все изменения применяются только к этому списку до тех пор, пока пользователь не нажмёт кнопку Apply changes ??? и изменения не внесутся в соответствующие документы типа CATPart.

«Checker»

Чтобы ускорить процесс моделирования необходимо отлавливать ошибки на как можно более раннем этапе. Макрос «Checker» позволяет выполнять проверку правильности построенной пользователем МС-модели в CATIA до экспорта в GDML, таким образом сокращая время отладки, которое ушло бы на экспорт геометрии в GDML, импорт в GEANT/ROOT и запуск проверки в конечной системе. Необходимость проверки возникает в силу того, что в разработанной структуре документов CATIA для МС-модели введено множество правил и ограничений, нетипичных для ??? conventional использования системы. «Checker» выполняет 2 типа проверок. Первый — корректность с точки зрения конвертера, т.е. соблюдение структуры документов, правильность именования, второй — корректность с точки зрения правил построения геометрии в GEANT/ROOT.

Использование разработанных интерактивных приложений ограждает пользователя от ошибок именования в итоговой МС-модели. Есть только одно место, где необходимо вручную указывать имя — формообразование-вращение при позиционировании операнда на уровне формы (??? уточнить). В любом случае, пользователь вправе редактировать модель не применяя макросы — это ускоряет процесс в некоторых случаях, но повышает вероятность нарушения правил именования. Принципиальные ошибки в структуре документов МС-модели с большой долей вероятности приведут к некорректному завершению «Checker», потому что практически невозможно автоматически их проверять.

Корректность геометрии определяется по двум критериям:

1. любые два объёма, находящиеся на одном уровне, не должны пересекаться;
2. любой дочерний объём не должен выходить за пределы материнского объёма.

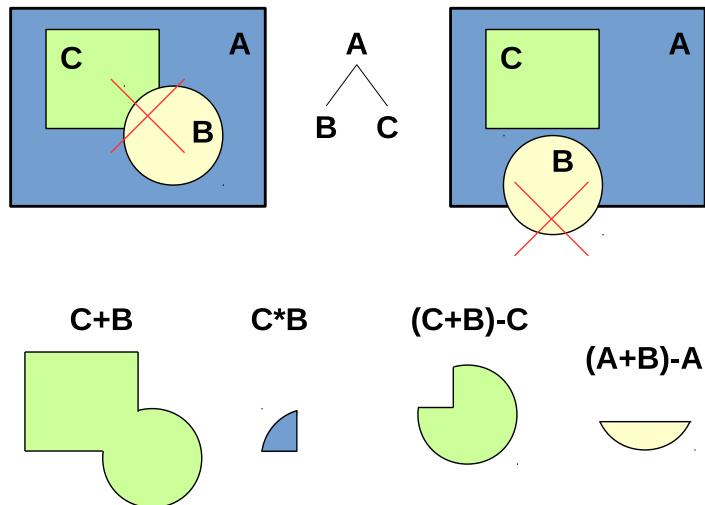


Рис. 2.4: Логические операции для определения правильности позиционирования объёмов.

Чтобы организовать проверку указанных условий, разработан специальный алгоритм и реализован в виде отдельной макропрограммы «Checker» САПР CATIA. В цикле перебираются все пары объёмов, лежащих на одном уровне, и проверяется, не пусто ли множество пересечения текущей пары. В том случае, если не пусто, то возникает событие, оповещающее, что текущая пара объёмов расположена недопустимым образом.

Более детально описанный процесс выглядит следующим образом. В силу того, что физические объёмы описываются телами детали, перебор объёмов, лежащих на одном уровне, сводится к перебору всех тел детали, кроме PartBody. Чтобы исследовать множество пересечения пары тел, создаётся новое пустое тело и копии исходных. Затем применяется Булева операция Intersect над копиями, и результат заносится в ранее созданное пустое тело. На этапе выполнения формообразования булевой операции выдаётся возможность отследить корректность результата. С точки зрения CATIA пустое пересечение является ошибочным результатом и возникает внутренняя ошибка. Именно программный отлов и обработка этой ошибки говорит о корректности расположения объёмов. После выполнения булевой операции результирующее тело удаляется, не оставляя таким образом никаких следов промежуточных преобразований.

Для того чтобы отследить, не выходит ли объём за пределы материнского объёма, применяется схожий подход. Отличие заключается в последовательности булевых операций — вместо пересечения $A * B$ двух объёмов A и B проверяется объём, полученный последовательностью двух операций $(A + B) - A$, где A — материнский объём, а B — дочерний. Присутствие результата операции $(A+B)-A$ говорит о том, что какая-либо часть дочернего объёма расположена за пределами материнского.

«CATIA2GDML»

Конвертер «CATIA2GDML» проецирует дерево потроения МС-модели из CATIA v5 в GDML файл. За счёт того, что правила построения МС-модели в CATIA формулировались с ориентиром на структуру, принятую в GEANT/ROOT, базовый функционал прямого конвертера заключается в том, что он создаёт в выходном GDML файле сущности, соответствующие найденным объектам в дереве построения в CATIA. Важной особенностью является то, что конвертер не обращается к самой геометрии, вся необходимая информацию содержится в дереве построение модели.

Помимо прямолинейного отображения дерева модели в GDML, в «CATIA2GDML» осуществляется преобразование некоторых особенностей, введённых в CATIA для упрощения процесса моделирования. Так, например, для отображения массива (см. ?? ??), которого нет в возможностях GDML, макрос рассчитывает повороты и положения каждого вхождения и создаёт независимые дочерние объёмы в GDML. В случае линейный массивов задача проста. Описание ??? В случае круговых массивов расчёт выполняется с применением матричных преобразований, обсуждаемых в 2.2.8.

В дереве CATIA нет обособленного списка используемых в модели материалов, поэтому для создания такого в GDML выполняется нехитрый алгоритм, который при проходе в цикле по объёмам добавляет в выходной список имя материала, если такого ещё нет.

В “Builder” реализована возможность переноса цвета объёма с помощью вспомогательного тега auxilliary в GDML файле, который позволяет добавлять произвольное поле данных к описанию логического объёма. Цвет тела в CATIA никак не отображается в дереве, поэтому для формирования тега при экспорте модели опрашиваются параметры визуализации главного тела PartBody детали, содержащие в том числе и цвет.

«GDML2CATIA»

«GDML2CATIA» выполняет процедуру, обратную «CATIA2GDML» — проецирует GDML файл на дерево модели CATIA v5. В “Builder” есть возможность задавать линейные и круговые массивы — многократные вхождения дочернего объёма в материнский, позиционированные с некоторым шагом вдоль линейной или круговой оси соответственно. В МС-модели в CATIA для массивов применяется соответствующее стандартное формообразование pattern. Такая возможность отсутствует в GDML, поэтому при экспорте из CATIA в GDML выполняется расчёт поворотов и сдвигов для каждого элемента массива и они представляются как отдельные, независимые дочерние объёмы. Таким образом при конвертации в обратном направлении, из GDML в CATIA, невозможно восстановить массив. Следовательно, одна из немногих (единственная ???) операций, которые необходимо совершать после импорта геометрии из GDML в CATIA — ручной перевод множества дочерних объёмов в массив. Обычно это очень простая процедура, и заключается она в том, что удаляются все вхождения, кроме первого, и в список формообразований первого тела добавляется pattern, которому задаются необходимые параметры и имя.

Аналогично прямому конвертеру, для передачи цвета автоматически выполняется дополнительная операция присвоения параметров визуализации тела если в GDML встречается тэг auxilliary.

«Duplicator»

Создание МС-модели более-менее сложной экспериментальной установки обычно требует создания нескольких вхождений параметризованных подборок с разными значениями параметров. Можно привести следующий пример. Рассмотрим детектор, состоящий из однотипных модулей, содержащих массив чувствительных объёмов (сенсоров), и какие-то другие элементы, например, платы передней электроники. Предположим, что существует несколько типоразмеров модулей, отличающихся количеством и размером сенсоров. Таким детектором может быть, например, калориметр, построенный из нескольких типов модулей, отличающихся гранулярностью — размер чувствительного объёма увеличивается по мере удаления от пучка. Очевидно, что если типы модулей отличаются лишь значениями каких-либо переменных, то представляется возможным построить одну параметризованную модель модуля, чтобы дальше использовать её многократно для построения всего детектора. К сожалению в GEANT/ROOT нет возможности так сделать — нужно для каждого типа иметь отдельное определение геометрии. Также это невозможно и в GDML и в CATIA.

В модели для каждой комбинации параметров, то есть для каждого модуля в нашем примере, должна существовать отдельная параметризованная подборка. При этом имена всех объёмов должны отличаться, а наборы параметров для каждого модуля должны быть независимы. В CATIA нет возможности быстро создать копию подборки вместе со всеми параметрами и зависимостями.

Для того, чтобы автоматизировать процесс создания такой копии используется «Duplicator». Работа с «Duplicator» выполняется в 2 этапа.

???

Визуализация нескольких уровней вложенности объёмов

В соответствии со структурой документов, определённых в “CATIA-GDML geometry builder”, в один момент можно визуализировать только один уровень вложенности.

Для того, чтобы визуализировать несколько уровней вложенности в CATIA предлагается следующий подход. Во всех документах типа CATPart, то есть для всех объёмов, все тела,

обозначающие дочерние объёмы, вычитаются из главного тела PartBody. За счёт того, что в документе, описывающем материнский объём, у тел, обозначающих дочерние объёмы, есть ссылка на PartBody деталей, ???

2.2.7 Geometry Description Markup Language (GDML)

Язык разметки GDML разработан специально для обмена моделями представленными с помощью CSG с иерархией объёмов. GDML — это XML-подобный язык. Файл на GDML имеет следующую структуру. Тэг верхнего уровня <gdml>, в нём 5 разделов:

1. <define>
2. <materials>
3. <solids>
4. <structure>
5. <setup>

В секции <define> объявлены объекты, которые могут многократно использоваться в другой секции.

1. constant
2. quantity
3. variable
4. position
5. rotation
6. scale
7. matrix

Все объекты должны иметь уникальные имена, определённые в значении атрибута name. При работе с “Builder” используются лишь некоторые из перечисленных типов: variable, position, rotation. В “Builder” введено три объекта, которые всегда создаются в <define> секции: нулевой поворот “identity”, нулевой сдвиг “central” и константа DEGtoRAD для перевода из градусов в радианы.

Секция <materials> предназначена для определения материалов, которые будут использоваться в модели. При описании логического объёма в секции <structure> должна быть ссылка на соответствующий тег <material> в секции <materials>. Стандартное применение GDML подразумевает, что в файле имеется полная информация о модели, в том числе и о материалах. Это означает, что при импорте GDML файла в GEANT4 или ROOT описание материалов будет взято из секции <materials>. На практике это оказывается очень неудобным, поэтому в экспериментальных пакетах, как например в CbmRoot, по умолчанию включена опция считывания из GDML только имена материала, игнорируя собственно описание. Однако в любом случае, для того чтобы GDML файл считался корректным с формальной точки зрения, для каждой ссылки должен существовать объект, на который эта ссылка указывает. В “Builder” мы используем минимальное dummy ??? описание, чтобы соблюсти правила XML файла, а фактически используется внешняя база материалов.

В секции `<solids>` приведено описание форм логических объёмов. Как и в случае с материалами, в секции `<structure>`, при описании логического объёма должна быть приведена ссылка на соответствующий тег в секции `<solids>`.

Секция `<structure>` — обычно самая большая секция, в которой описываются логические объёмы и их иерархия. Тег `<volume>` имеет как минимум два дочерних тега — ссылка на материал `<materialref>` с атрибутом `ref`, имеющим в качестве значения имя материала, определённого в секции `<materials>`, и ссылка на форму объёма `<solidref>` с атрибутом `ref`, имеющим в качестве значения имя формы, определённой в секции `<solids>`. Помимо этих двух обязательных дочерних тегов могут присутствовать другие теги, описывающие внутренний состав логического объёма. Самый распространённый случай — тег `<physvol>`, обозначающий дочерний объём и имеющий атрибут `ref`, указывающий на определённое ранее описание другого объёма.

Последняя секция `<setup>` служит для объявления одного логического объёма в качестве объёма верхнего уровня. Также здесь задаётся название установки, которое в `CbmRoot` должно быть `FairGeom`.

2.2.8 Избранные подробности реализации “CATIA-GDML geometry builder”

Каждый макрос “Builder” — это VBA проект, который хранится в отдельном `catvba` файле. Проект состоит из трёх разделов — элементы графического интерфейса (формы), модули и модули классов. Большинство макросов “Builder” написано в соответствии с идеологией структурного программирования, без применения классов, и разделение на модули выполнено из соображений читаемости кода. Обычно в отдельный модуль выносился функционал, объединённый некоторой задачей. Так, например, во многих макросах имеется модуль `???` (имя) для продвинутой работы со строками, модуль `???` (имя) для `???` (задача). В некоторых случаях естественным образом требовалось использовать классы. Так, например, был реализован класс матрицы с методами нахождения углов поворота, который использовался в `???` и более подробно описан в 2.2.8.

Работа с матрицами позиционирования в “CATIA-GDML geometry builder”

При создании иерархии объёмов возникает необходимость задавать положение дочернего объёма в материнском, которое в GEANT/ROOT хранится в виде матрицы 4×4 . Эта матрица легко разбивается на две компоненты — сложный поворот в трёхмерном пространстве и параллельный сдвиг, выполняемый после поворотов. Есть несколько интуитивно понятных человеку способов задать поворот тела в некоторой неподвижной системе координат, которой в нашем случае является система координат материнского объёма. По причине простоты реализации и дружелюбности для пользователя в “CATIA-GDML geometry builder” позиционирование дочернего объёма в материнском выполняется с помощью трёх последовательных поворотов вокруг постоянных осей Z, Y и X с последующим параллельным переносом. Три угла поворота и три координаты сдвига

- α вокруг Z
- β вокруг Y
- γ вокруг X
- a вдоль X
- b вдоль Y
- c вдоль Z

однозначно задают матрицу позиционирования, которая рассчитывается как произведение

$$M = M_t \cdot M_X \cdot M_Y \cdot M_Z$$

$$M_Z = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M_Y = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M_X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) & 0 \\ 0 & \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M_t = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a & b & c & 1 \end{pmatrix}$$

Следует отметить, что α , β и γ не являются углами Эйлера, которые задают повороты вокруг изменяемых осей.

2.3 Применение “CATIA-GDML geometry builder” к СВМ RICH

Значительная часть работы над “Builder” выполнялась при поддержке группы СВМ RICH, поэтому самая сложная МС-модель построенная с помощью “Builder” это СВМ RICH. Построенная за несколько итераций модель имеет достаточно сложную иерархию и характеризуется высокой степенью подробностей.

Для того, чтобы GDML без проблем импортировался в CbmRoot было написано дополнение, описанное в секции 2.4.

Геометрическая МС-модель детектора RICH эксперимента СВМ имеет многоуровневую структуру, в основном обоснованную физической структурой сборки, но иногда и ??? бывает неочевидной и неестественной с целью повышения эффективности проведения частиц.

На момент написания данной работы инженерный проект не был завершён — некоторые узлы были проработаны достаточно подробно и прошли несколько этапов уточнения, в которых модель менялась принципиально. В то же время некоторые узлы существуют лишь на концептуальном уровне. В первую очередь к ним относится форма и конструкция корпуса детектора, проектирование которой не составляет особого труда и может быть отложена на более поздний этап. Большая часть корпуса не оказывает влияния на эффективность

детектора, т.к. лежит за пределами геометрического аксептанса, поэтому допускается моделирование физики детектора с упрощённой моделью корпуса, либо вообще без него.

Аксептанс детектора составляет 25° по вертикали и 37.5° по горизонтали. Длина отвёрдённого под детектор пространства вдоль оси пучка составляет 1900 мм, при этом передняя плоскость находится на расстоянии 1800 мм от точки взаимодействия, а задняя, соответственно, на расстоянии 3700 мм. Ещё 100 мм перед RICH отводится под пространство для стыковки магнита, STS, RICH и пучковой трубы. Исходя из того факта, что зеркала должны полностью покрывать геометрический аксептанс, ширина детектора выбрана 5268 мм, а высота 4420 мм.

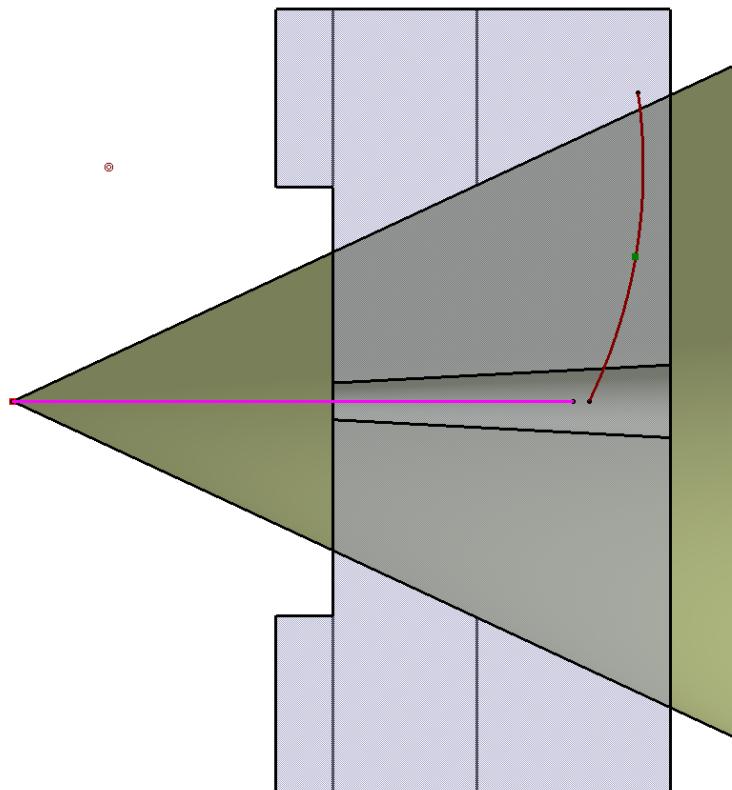


Рис. 2.5:

В детекторе RICH можно выделить несколько подсистем — фоточувствительная камера, магнитный экран вокруг камеры, зеркала, система опор зеркал, часть ионопровода в RICH, корпус детектора. Рассмотрим реализацию каждой подсистемы в МС-модели, построенной с помощью “CATIA-GDML geometry builder”.

2.3.1 Фокусирующая система — сферические зеркала

На раннем этапе проектирования СВМ RICH исходя из доступного пространства в общей экспериментальной установке было рассчитано, что фокусирующая система СВМ RICH должна выполнять одно отражение с помощью двух сферических зеркал радиусом 3 метра, расположенных на расстоянии 3500 мм от точки взаимодействия. Рассматривался также

вариант с двойным отражением, при этом вторая группа зеркал была плоская. Зеркала расположены симметрично относительно горизонтальной плоскости, проходящей через ось пучка, и полностью покрывают аксептанс детектора. Площадь каждого зеркала составляет приблизительно 7.5 m^2 .

Для того чтобы выполнить требование к точности, зеркало такого радиуса технологически возможно изготовить только из сегментов. Рассматривались варианты изготовления различными компаниями, прототипы исследовались на ???, в итоге остановились на ??? . Сегмент зеркала будет иметь размер около $40\text{cm} \times 40\text{cm}$. (точное значение указать невозможно, это же не прямоугольник)

Далее рассматривается только одно, верхнее зеркало. Всё описание распространяется и на симметричное нижнее зеркало.

Основная задача зеркал — отвести черенковские фотоны в область, где они могут быть зарегистрированы фоточувствительной камерой, которую невозможно расположить напрямую на пути этих фотонов, т.е. в геометрической аксептанссе. Это сделает работу последующих детекторов невозможным из-за вторичных частиц.

Вообще есть целое направление, в котором люди занимаются тем, что оценивают и обычно стараются минимизировать material budget.

Для того, чтобы фокусировать фотоны на камеру, расположенную за пределами аксептансса, необходимо, чтобы центр сферической поверхности располагался над осью пучка, а сами зеркала полностью покрывали аксептанс. Есть как минимум два варианта геометрии зеркал. В первом зеркало выполняется симметричным относительно горизонтальной плоскости и поворачивается вокруг оси X. При этом возникает зазор между двумя зеркалами, расширяющийся к краям (см. рис. ??), но все зеркала состоят из сегментов двух типов. Более оптимальный способ — выбрать правильную долю сферы так, чтобы зеркаластыковались без зазора. При этом зеркало получается несимметричным, а следовательно необходимо 4 типа сегментов.

В CBM RICH каждый сегмент зеркал будет крепиться на раме с помощью трёх ??? моторизированных актуаторов с удалённым управлением. Это позволяет корректировать положение отдельных сегментов, корректируя отклонения от правильного положения, связанные с перемещениями точек механической опоры, находящейся в напряжённо-деформированном состоянии под собственным весом и весом зеркал. Также возможны перемещения в связи с термическим расширением рамы при изменении параметров окружающей среды в экспериментальном зале.

Коллаборациями LHCb и COMPASS разработаны методы ???, позволяющие выполнить... Для анализа в моделировании необходимо обеспечить возможность поворота отдельных сегментов зеркал вокруг заданных осей. В связи с этим была построена версия МС-модели, отличающаяся структурой объёмов и обеспечивающая возможность поворота отдельных сегментов зеркал вокруг заданных осей. Эта модель обсуждается в 2.3.1.

МС-геометрия фокусирующей системы с возможностью задания индивидуальных отклонений

Для отладки методов калибровки положения зеркал необходимо выполнять моделирование с геометрией, имеющей отклонения сегментов зеркал, заданные определённым образом. Техника ??? CLAM разработана с расчётом на то что отклонение каждого сегмента может быть получено в результате двух вращений. Если для каждого сегмента зеркала ввести фиксированный центр и локальную систему координат с осями \vec{n} , $\vec{\tau}$ и \vec{b} , то вращение должно осуществляться последовательно вокруг ??? .

Для того, чтобы это было возможно в разработанной МС-модели СВМ RICH потребовалось ввести два промежуточных уровня вложенности — один для каждого типа зеркал и ещё один для каждого сегмента. Первый промежуточный уровень вложенности переносит центр сегмента в начало координат. Второй обеспечивает вращение вокруг оси ??? . При позиционировании в газ вводится отклонение вокруг второй оси ??? .

??? уточнить

Одна из задач, которую приходится решать с описываемой геометрией — выполнять многократно моделирование прохождения частиц с разными значениями отклонения сегментов зеркал. Это означает, что пользователь должен иметь возможность легко модифицировать геометрию. По этой причине значения отклонений каждого сегмента зеркала были вынесены в качестве параметров модели, что выглядит как список из 160 параметров в `<define>` секции GDML файла. Имя каждого параметра построено по правилу “misalign_AXIS_A_B”, где AXIS — ось вращения — либо “x”, либо “y”, $A \in [0, 7]$ — номер сегмента вдоль вертикального направления, а $B \in [0, 9]$ — номер сегмента вдоль горизонтального направления.

??? картинка с нумерации сегментов зеркал

В итоге получается две модели RICH — одна для общего пользования с идеально позиционированными зеркалами и вторая отдельно для отладки методов коррекции положения зеркал. Для того, чтобы не ??? плодить сущности без необходимости оба зеркала смоделированы в одной CATIA сборке, а разделение на два GDML файла выполняется путём комментирования некоторых частей одного GDML файла, экспортруемого из CATIA. Использование геометрии с индивидуальным отклонениями зеркал для моделирования в общей установке СВМ не рационально, т.к. в такой геометрии больше объёмов. Кроме того, в геометрии с отклонениями зеркал по запросу пользователя были выключены некоторые подробности (такие как, например, каркас детектора и опоры зеркал), т.к. они не оказывают никакого влияния на исследование CLAM, но замедляют моделирование.

2.3.2 Магнитный экран вокруг камеры

Из-за близости дипольного магнита и фоточувствительной камеры

На момент написания данной работы проект магнитного экрана не был завершён, однако было выполнено эскизное проектирование и моделирование распределения магнитного поля. Было определено, что магнитный экран должен иметь нижнюю и заднюю (ближнюю к магниту) толщиной 30 мм, а остальные — толщиной 10 мм. Основной проблемой при проектировании магнитного экрана является задача минимизации массы, т.к. экран должен быть выполнен из материала с высоким коэффициентом магнитной проницаемости, а значит лёгкие металлы типа алюминия не подходят. Масса каждого из двух экранов получилась равна 850 кг.

2.3.3 Фоточувствительная камера

Планируется, что фоточувствительная камера СВМ RICH будет составлена из модулей, содержащих 2×3 МА ФЭУ Hamamatsu H12700, см. рис. 2.6. Один такой МА ФЭУ имеет габариты $52 \times 52 \text{ mm}^2$, между МА ФЭУ оставляется зазор 1 мм для запаса по точности, таким образом размер модуля составляет $158\text{mm} \times 105\text{mm}$.

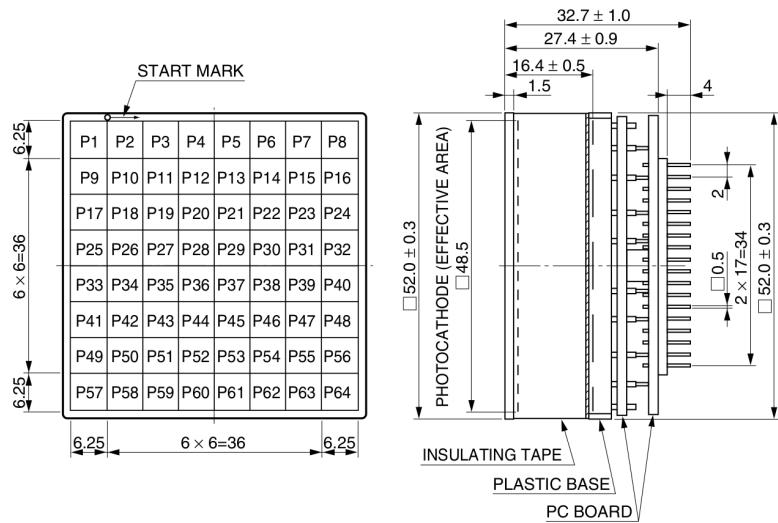


Рис. 2.6: Чертёж МА ФЭУ H12700 из документации.

В момент написания данной работы ведётся работа по проектированию модуля, разработке программ для FPGA, но имеются изготовленный прототип. Помимо МА ФЭУ в модуль входят 12 плат передней электроники DIRICH, одна плата, обеспечивающая питание, и одна плата концентрации данных. В основе модуля лежит плата-адаптер, к которой с одной стороны подсоединяются МА ФЭУ, а с другой — все платы. CAD-модель и МС-модель модуля показаны на рис. 2.7.

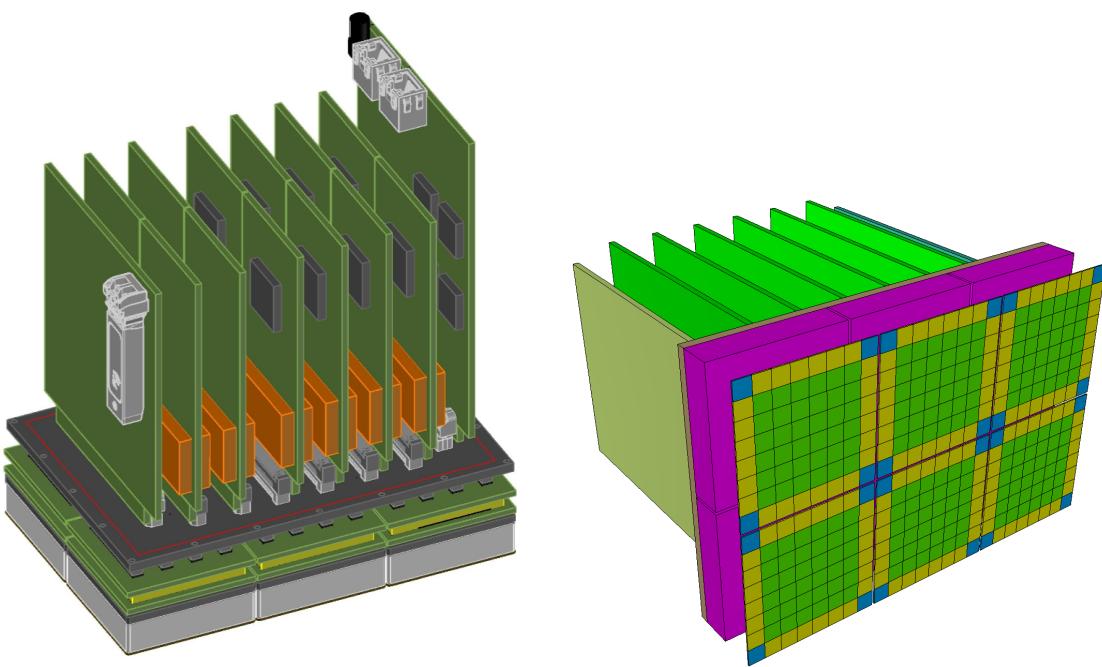


Рис. 2.7: CAD-модель (слева) и MC-модель (справа) модуля фоточувствительной камеры CBM RICH.

Иерархия объёмов, моделирующих модуль фоточувствительной камеры CBM RICH приведена на рис. 2.8.

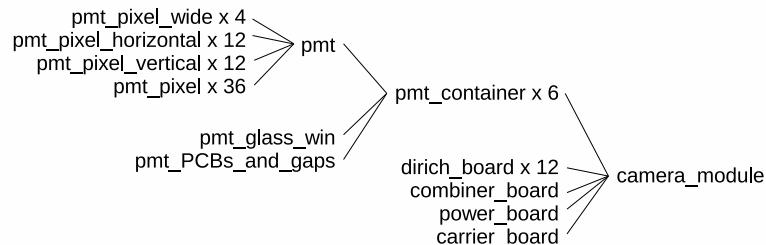


Рис. 2.8: Иерархия объёмов, моделирующих модуль фоточувствительной камеры CBM RICH.

МА ФЭУ моделируется до уровня пикселей. Это позволяет максимально приблизить моделирование прохождения частиц в CbmRoot и обработку реальных данных. В соответствии с документацией, у МА ФЭУ Н12700 пиксели имеют разные размеры (см. рис. 2.6): угловые пиксели (№№ 1, 8, 57, 64) $6.25\text{мм} \times 6.25\text{мм}$, пиксели по краям, кроме угловых, — $6.25\text{мм} \times 6\text{мм}$, остальные, центральные пиксели — $6\text{мм} \times 6\text{мм}$. Для того, чтобы представить три типа пикселей в MC-модели, необходимо три отдельных объема, имеющих разную форму. Чтобы сделать модель максимально понятной и гибкой принято решение отделить пиксели из крайних горизонтальных рядов от пикселей из крайних вертикальных рядов и моделировать их с помощью двух разных объемов размером $6\text{мм} \times 6.25\text{мм}$ и $6.25\text{мм} \times 6\text{мм}$ соответственно. Это позволит позиционировать все пиксели без поворотов.

Таким образом вводится три объёма: *pmt_pixel_wide* для угловых пикселей, *pmt_pixel_horizontal* для пикселей в крайних горизонтальных рядах, *pmt_pixel_vertical* для пикселей в крайних вертикальных рядах и *pmt_pixel* для всех остальных пикселей, расположенных в центральной зоне. Все 4 объёма имеют форму примитива *box* с толщиной вдоль оси Z, равной 0.5мм, материал CsI, который в данный момент используется в моделировании как активный материал для фоточувствительных элементов. Толщина выбрана произвольно, она не имеет значения, т.к. из-за того, что материал объёма активный, т.е. объём объявлен чувствительным, система проведения частиц будет вырабатывать сигнал о пересечении треком границы объёма и передавать управление методу *ProcessHits* класса детектора *CbmRich*. В реализации этого метода вырабатывается *Point*, причём физика не оказывает никакого влияния. Даже если толщина объёма слишком маленькая, чтобы произошло какое-либо взаимодействие, флаг активности обязывает систему вызвать *ProcessHits*.

Объём *pmt* соответствует части МА ФЭУ, включающей в себя фотокатод (пиксели) и динодную систему, и имеет толщину $16.4 - 1.5 = 14.9$ мм. Входное стеклянное окно МА ФЭУ моделируется отдельным объёмом *pmt_glass_win*, имеющим толщину 1.5мм. Пространство за динодной системой, включающее в себя печатные платы и ножки в воздушном пространстве, моделируется объёмом *pmt_PCBs_and_gaps*. Все части МА ФЭУ, моделируемые перечисленными объёмами, вставляются в контейнер *pmt_container*.

Платы передней электроники, питания, концентрации данных и палата-адаптер моделируются объёмами, имеющими форму *box* и одинаковый материал, — *dirih_board*, *power_board*, *combiner_board*, и *carrier_board* соответственно. Объём *camera_module* выполняет роль контейнера, в который помещаются МА ФЭУ и платы. Далее составляется вертикальный массив из ??? 7 модулей, называемый *camera_strip*.

В процессе разработки детектора сначала рассматривался вариант фоточувствительной камеры, состоящей из 4 плоскостей, расположенных симметрично относительно горизонтальной и вертикальной плоскостей, проходящих через ось пучка. Исследовались разные варианты комбинаций размера, поворотов и положения четверти с целью нахождения оптимальных значений с точки зрения эффективность всего детектора. ??? ссылка на несуществующую секцию, где обсуждается эта оптимизация. Одна итерации такой оптимизации заключается в запуске полного моделирования и анализа и является достаточно затратной процедурой.

В настоящее время прорабатывается вариант, в котором верхняя и нижняя половины фоточувствительной камеры составлены из сегментов шириной в один модуль и аппроксимирующих поверхность цилиндра. Радиус 1650 мм, поворот 18° вокруг оси X и положение цилиндра также получены в результате оптимизации.

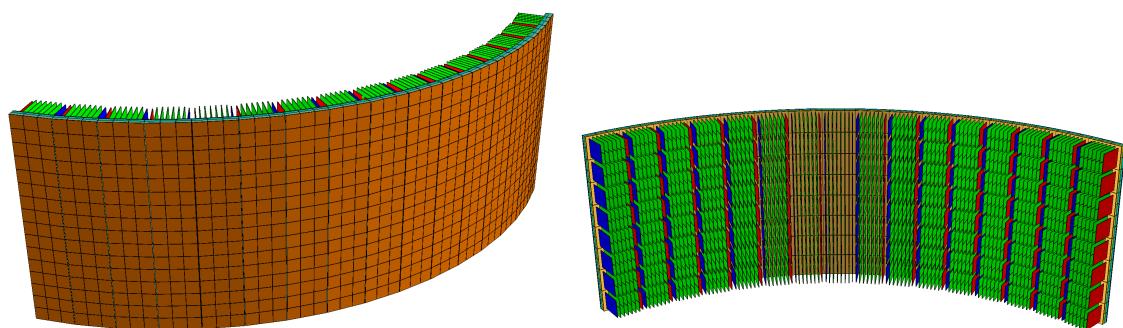


Рис. 2.9: МС-модель фоточувствительной камеры СВМ RICH. На рисунке показаны платы электроники, но не показаны отдельные пиксели МА ФЭУ.

2.3.4 МС-геометрия механических конструкций RICH

Механические конструкции не участвуют в физической части функционирования детектора, а лишь выполняют функцию опоры и, в случае СВМ RICH, создания герметичного контейнера для газового радиатора. Также можно отметить наличие магнитного экрана вокруг фоточувствительной камеры. Однако такие конструкции всё-же оказывают влияние на эффективность детектора, т.к. частицы взаимодействуют с их материалом и в результате могут изменить направление и импульс, поглотиться или произвести вторичные.

Чтобы оценить влияние материала механических конструкций на функционирование детектора необходимо максимально точно смоделировать количество материала в аксептансе. Применение “CATIA-GDML geometry builder” сильно облегчает процесс моделирования пассивного материала, т.к. стандартными средствами CATIA можно измерить объём детали сложной формы, чтобы затем использовать это значение для расчёта упрощённой детали.

Приведём решение типовой задачи. Требуется заменить сложный профиль прямоугольным кольцом с совпадающими внешними размерами. В МС-модели будет box из металла, внутри которого вставлен box из материала окружающей среды (в случае СВМ RICH – газовый радиатор).

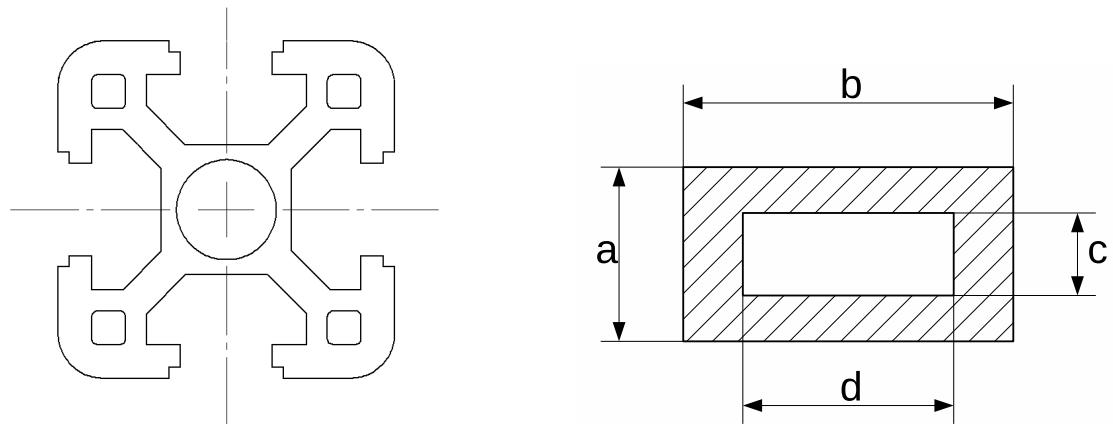


Рис. 2.10: Исходный профиль в CAD-модели (слева) и итоговый профиль в МС-модели.

Площадь исходного профиля S_{CAD} измеряется стандартной функцией CATIA v5. Задача заключается в том, чтобы определить размеры a, b, c, d в МС-модели такие, что площадь обоих профилей совпадает, т.е. $S_{CAD} = S_{MC}$. Пользователь может выбрать внешние размеры a и b , например так, чтобы они совпадали с габаритами исходного профиля. Очевидно, $S_{MC} = ab - cd$. Пусть $c = ka$ и $d = kb$. Тогда $S_{CAD} = S_{MC} = ab - ka \cdot kb = ab(1 - k^2)$. Отсюда $1 - k^2 = \frac{S_{CAD}}{ab}$ и $k = \sqrt{1 - \frac{S_{CAD}}{ab}}$. Отсюда вычисляются c и d .

Механические конструкции RICH были построены с применением описанной методики. На рис. 2.12 приведена модель каркаса детектора в МС-формате в CATIA. Каждая балка моделируется отдельным объёмом, полость внутри балки моделируется с помощью дочернего объёма из материала окружающей среды, в данном случае – газа радиатора. Однаковые балки моделируются одним объёмом, который многократно вставляется в контейнер. Для того чтобы упростить позиционирование каркаса в материнском объёме, вся конструкция была собрана в двух объёмах типа Assembly (см. рис. 2.11).

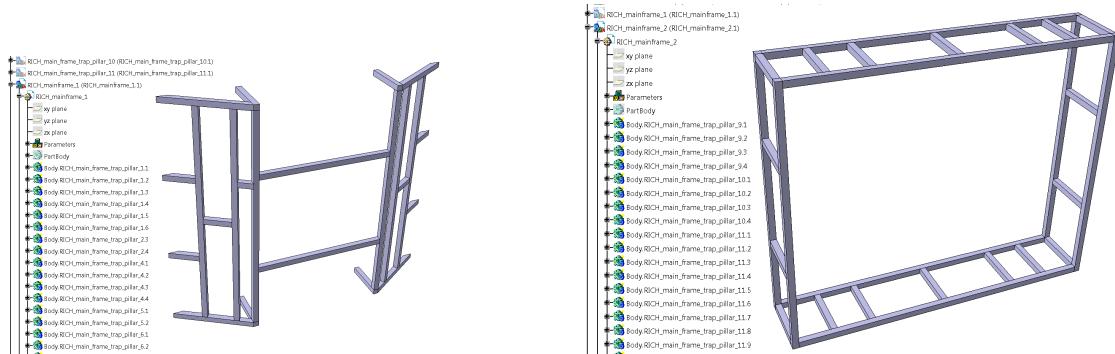


Рис. 2.11: .

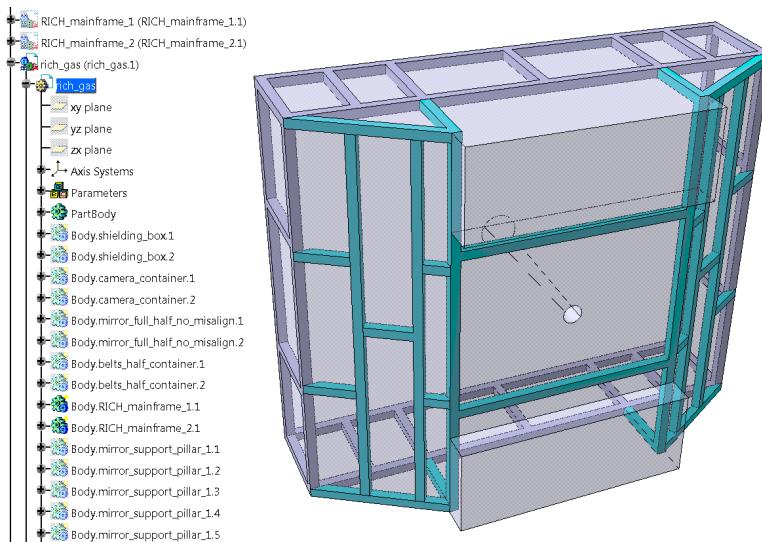


Рис. 2.12: МС-модель каркаса детектора в материнском объёме в CATIA.

2.4 Адаптация CbmRoot для работы с GDML

Я просто перенёс текст из диплома с небольшими изменениями. Боюсь, к настоящему моменту, ситуация заметно изменилась...

Целевым пакетом для большинства МС-моделей, разработанных с помощью “CATIA-GDML geometry builder”, в частности СВМ RICH, является пакет CbmRoot, в котором выполняется моделирование, реконструкция, приём и анализ данных эксперимента СВМ. CbmRoot является фреймворком, написанным на основе FairRoot, который в свою очередь написан на основе ROOT.

Любой элемент экспериментальной установки в системе FairROOT описывается классом, наследованным от базового класса *FairModule*. Для импорта геометрической информации из внешнего файла предусмотрен метод *ConstructGeometry* класса *FairModule*. Данный метод не выполняет непосредственно чтение информации, а лишь определяет тип файла и выбирает какой метод необходимо вызвать.

Поддерживаются различные форматы входных файлов, поэтому существуют отдельные методы, отвечающие за соответствующие форматы:

- *ConstructROOTGeometry*;
- *ConstructASCIIGeometry*;
- *ConstructGEOGeometry*.

Данные методы могут быть имплементированы на различных уровнях — как в базовом классе *FairModule*, так и конечном, описывающем конкретный элемент, или в любом промежуточном классе, если таковые имеются.

В данной секции в качестве конкретного элемента экспериментальной установки будем рассматривать детектор RICH эксперимента СВМ. Соответствующий класс имеет название *CbmRich*. Всё описание с незначительными дополнениями распространяется на любой активный либо пассивный элемент установки (далее детектор).

На рис. 2.13 представлена диаграмма, отражающая иерархию классов и методов, касающихся импорта геометрии до внесения изменений.

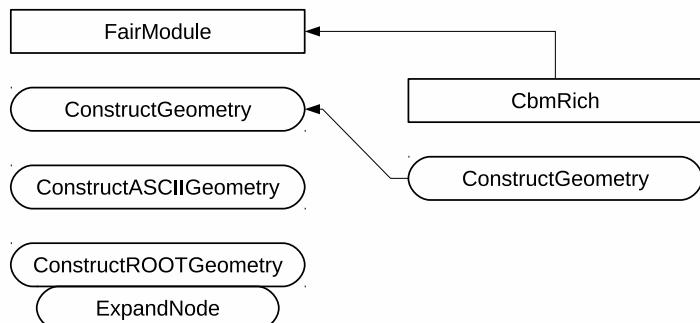


Рис. 2.13: Диаграмма классов CbmROOT до модификации.

Если метод объявлен в материнском классе, то он может быть определён (т.е. собственно его код написан) как в дочернем классе, так и в материнском.

Метод *ConstructGeometry* объявлен в *FairModule*, но определён всегда в классе детектора. Именно этот класс определяет, какой метод вызывать в зависимости от формата входного файла. Метод *ConstructASCIIGeometry* объявлен в *FairModule*, но не реализован в дочернем классе *CbmRich*. Это означает, что для детектора RICH не предусмотрен импорт из текстового файла, но существует какой-то другой элемент установки, который имеет описание импорта из формата ASCII.

Первоначально был введён метод *ConstructGDMLGeometry* на уровне класса *FairModule*. Таким образом для того чтобы обеспечить вызов этого метода из класса детектора, достаточно было лишь добавить условие в методе *ConstructGeometry* в классе детектора. Автоматически вызывался бы метод из FairROOT. Такой подход был реализован и успешно проверен на локальной установке CbmRoot. Однако когда введённые изменения были внесены в промежуточный (trunk) вариант пакета в GSI было обнаружено, что возникают сложности, вызывающие некорректную работу других модулей пакета.

Для работы импорта из GDML файла необходима определённая динамическая библиотека. Эта библиотека не скомпилирована по умолчанию. Если изменить исходный код FairROOT необходимо скомпилировать упомянутую библиотеку и поместить её в соответствующую папку FairROOT. По той причине, что имеется множество наследованных от FairROOT пакетов и ни один из них не скомпилирован с поддержкой GDML, изменения в исходном коде потребуют перекомпиляции всех дочерних программ, что представляет собой трудоёмкую процедуру, требующую прав администратора и значит недопустимую на данном этапе.

Было принято решение временно перенести исходный код импорта из GDML в новый промежуточный класс *CbmModule*. Затем в определённый момент, спустя годы, описываемый функционал был включён в FairRoot, что позволило использовать GDML как формат импортируемой геометрии всем унаследованным от FairRoot пакетам — R3BRoot, PandaRoot, и т.д.

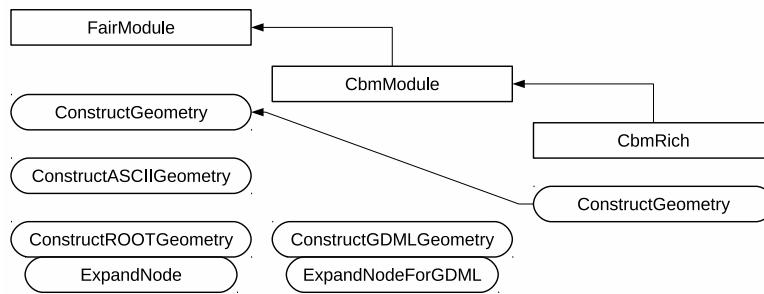


Рис. 2.14: Диаграмма классов CbmROOT после модификации.

Введён промежуточный класс *CbmModule*, имеющий 2 метода — *ConstructGDMLGeometry* и вспомогательный *ExpandNodeForGDML*.

Добавлена ветвь условия, обеспечивающая вызов *ConstructGDMLGeometry* в случае получения входного файла в формате GDML.

```

void CbmRich :: ConstructGeometry ()
{
    TString fileName=GetGeometryFileName ();
    if (fileName.EndsWith (".geo")) {
        ConstructGEOGeometry ();
    } else if (fileName.EndsWith (".root")) {
        ConstructROOTGeometry ();
    } else if (fileName.EndsWith (".gdml")) {
        ConstructGDMLGeometry (fposrot);
    } else {
        std :: cout << "Geometry format not supported." << std :: endl;
    }
}
  
```

Также для геометрии, импортированной из GDML, добавлена опциональная возможность позиционировать детектор непосредственно в макрое путём указания координат смещения и углов поворота. Данные параметры необязательны — в случае их отсутствия детектор позиционируется по умолчанию. Для каждого элемента установки необходимо указать

такие значения по умолчанию в заголовке класса. Данная процедура была выполнена для детектора RICH. Такая возможность позволяет продвинутому пользователю оперативно менять установку, а наличие значений по умолчанию ограждает неопытных пользователей от ошибок.

Все изменения были приняты и внесены в дистрибутив CbmROOT, также как и новая геометрия CBM RICH в форматах GDML и ROOT. Решено хранить как исходный GDML файл, так и ROOT файл, полученный в результате процедуры импорта GDML и экспорта в ROOT.

Таким образом, выполнена разработка модулей FairROOT для гибкой работы с форматом GDML, разработанные модули применены для обновления геометрии детектора RICH эксперимента CBM.

Глава 3

РазвитиеDAQ эксперимента СВМ

3.1 Фотоэлектронный умножитель

Многоанодный фотоэлектронный умножитель (МА ФЭУ) H12700 фирмы Hamamatsu [16], появившийся на рынке в 2013 г., подробно охарактеризован в работах [17, 18]. Он обладает следующими достоинствами: большая доля площади поперечного сечения, приходящаяся на светочувствительные пиксели, квадратная форма, что позволяет перекрывать без потерь значительные площади (плотность упаковки 87%), малое время прохождения однофотоэлектронного сигнала через динодную систему, малый разброс этого времени от события к событию, низкие перекрёстные помехи и низкая скорость счета тепловых электронов. Некоторые свойства данного прибора показаны в табл. 3.1, по большинству параметров он превосходит своего предшественника МА ФЭУ H8500 [19].

Таблица 3.1: Свойства МА ФЭУ H12700B-03.

Темновой счёт на канал, Гц	Темновой счёт на весь МА ФЭУ, кГц	Время нарастания сигнала, нс	Разброс времени развития электронной лавины, нс
≈ 10	<1.0	0.64	0.28

Данный МА ФЭУ имеет двухщелочной фотокатод. Спектральная чувствительность МА ФЭУ в версии H12700B-03, используемой в настоящей работе, соответствует конфигурации с входным стеклом, прозрачным в ультрафиолетовой области.

Коротковолновая граница спектра чувствительности $\lambda_{min}=185$ нм, а максимум квантовой эффективности составляет 33% и достигается при длине волны $\lambda=380$ нм. Такие спектральные характеристики хорошо подходят для регистрации черенковского излучения, лежащего в ультрафиолетовой области. Среднеквадратичное отклонение коэффициентов усиления в каналах от среднего значения не превышает 16% [16]. Разброс квантовой эффективности между пикселями по нашим данным составляет $\pm 10\%$.

Имеются исследования [20, 21, 22], показывающие, что радиационная стойкость прибора достаточна для использования в эксперименте СВМ. Также продемонстрирована работоспособность прибора в магнитном поле до 2.5 мТл [17] без значительного падения характеристик. Использование магнитных экранов и выбор оптимального расположения фотодетектора в пространстве делают этот МА ФЭУ пригодным для использования в эксперименте СВМ. Отметим, что к этому прибору проявляют интерес и другие эксперименты,

например, он рассматривается и для обновления LHCb [17].

Наряду с перечисленными достоинствами, МА ФЭУ Н12700В-03 имеет некоторые особенности, не имеющие аналогов в традиционных ФЭУ и требующие особого внимания при реализации канала считывания. Размножение электронов в динодной системе происходит в одном и том же вакуумном объеме для всех каналов. Помещенная в единый вакуумный объем динодная система типа “Metal Channel”, см. рис. 3.1, отличается тем, что она довольно компактна, едина для всех каналов и позволяет добиться отличных временных свойств. Электронные лавины, соответствующие разным каналам, отличаются местом прохождения через динодную систему. Имеют место такие эффекты как выбивание электронов из динодов фотонами, прошедшими сквозь фотокатод, и отклонение электронов от идеальной траектории за счет разброса энергий. Последняя особенность приводит к попаданию электронов на последующие стадии динодной системы, минуя предыдущие, и перетеканию всей или части электронной лавины в соседний канал. Перетекание части лавины в соседний канал имеет место в более чем 25% случаев при равномерном освещении всего фотокатода. Величина перетекающего заряда составляет от 3% до 7% в зависимости от взаимного расположения пикселей [17]. Вероятность того, что лавина от фотоэлектрона полностью разовьется в соседнем канале зависит от взаимного расположения каналов и составляет при равномерном освещении от 0.1% до 2% [23]. Кроме того, при наличии относительно большого сигнала в одном из каналов, наблюдается биполярная наводка в каналах, расположенных в том же ряду. При интегрировании этой наводки возможно формирование низкоамплитудных импульсов в нескольких каналах. В классическом ФЭУ такие эффекты отсутствуют из-за отсутствия связи с соседними каналами, наличия развитой системы фокусировки и такой конструкции динодной системы, что диноды имеют большую площадь и последующие стадии полностью экранируются предыдущими.

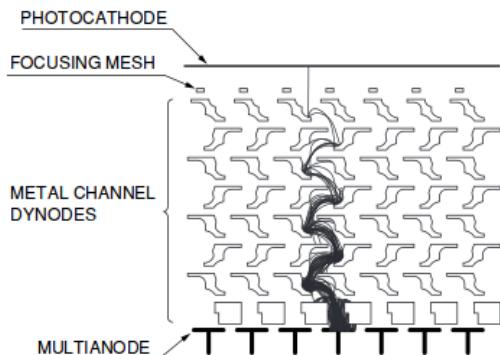


Рис. 3.1: Схема динодной системы типа “Metal Channel”.

Описанные особенности приводят к формированию в одноэлектронном спектре низкоамплитудной части, сливающейся с шумами и отделенной от основного пика довольно глубокой ложбинкой. Проявления этого эффекта в наших измерениях обсуждаются в секции 4.8.

3.2 Архитектура системы сбора данных СВМ RICH

3.2.1 64-канальный модуль считывания

Конструктивно и функционально вся электроника считывания и оцифровки данных СВМ RICH может быть сгруппирована в 64-канальные модули, каждый из которых соответствует одному многоанодному фотоэлектронному умножителю (МА ФЭУ). Схема 64-канального модуля показана на рис. 3.2. Он включает в себя 4 платы PADIWA и одну плату TRB v3.

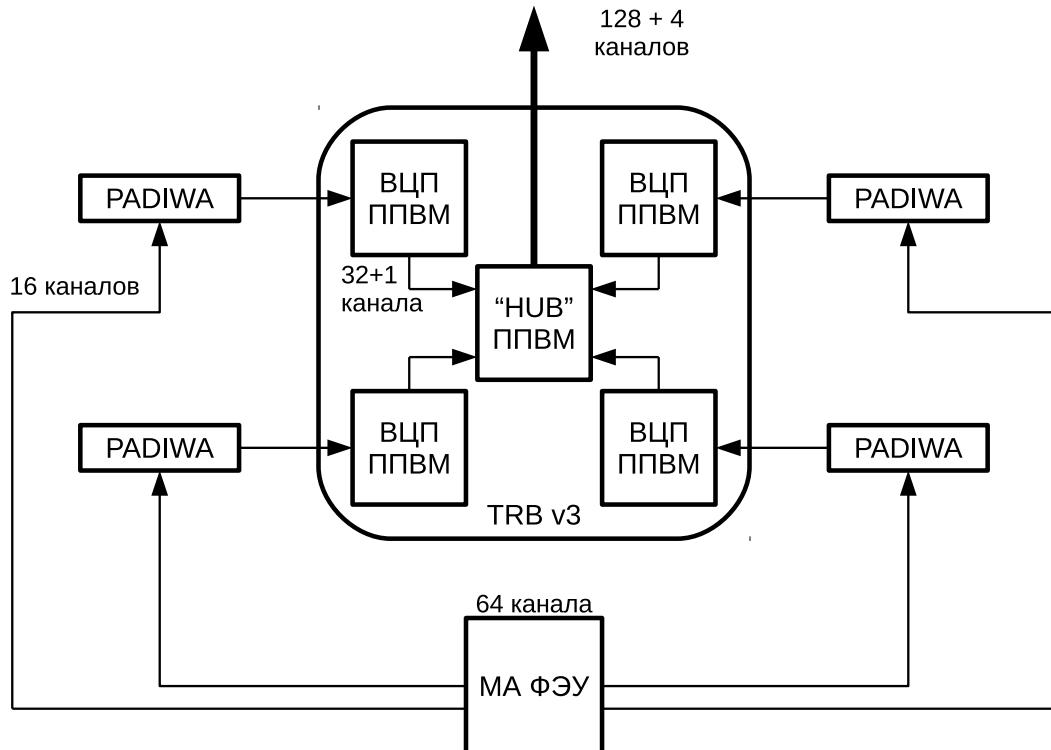


Рис. 3.2: Схема считывания одного МА ФЭУ, состоящая из 4 плат-дискриминаторов PADIWA и одной платы TRB v3.

PADIWA — 16-ти канальная плата передней электроники, разработанная в ГСИ [24]. Общий вид платы PADIWA показан на рис. 3.3. Плата устанавливается на МА ФЭУ через плату-адаптер, единственным назначением которой является соединение анодов МА ФЭУ с соответствующими входами PADIWA. С одной стороны печатной платы PADIWA расположены 16 сигнальных входов с импедансом 100 кОм. На каждый вход приходится два контакта — земля и сигнал. Они чередуются таким образом, чтобы можно было подключить PADIWA к плате-адаптеру любой стороной. Каждый канал PADIWA имеет собственный фильтр низких частот с полосой пропускания около 100 МГц и предусилитель, которые образуют аналоговую часть канала. После усиления сигнал поступает в программируемую пользователем вентильную матрицу (ППВМ). Обычно ППВМ применяются для обработки цифровых (логических) сигналов, однако, в нашем случае на входные цифровые линии подаётся аналоговый сигнал. В ППВМ для каждой входной линии можно задать свой порог,

разделяющий логические уровни входного сигнала. Таким образом, настраиваемые входы ППВМ могут использоваться как дискриминаторы. На выходе каждого канала формируется логический ноль, когда входной сигнал в этом канале ниже установленного порога, и логическая единица, когда входной сигнал выше этого порога, см. рис. 3.4. Далее расположены выходные порты и порты настройки ППВМ, объединённые в разъем, позволяющий подключить 20 LVDS линий. Для управления платой используются 4 LVDS линии, остальные 16 LVDS линий — выходные. Для программирования ППВМ на плате предусмотрен стандартный JTAG порт. Также на плате имеется порт для подключения источника низкого напряжения для питания платы. Помимо этого имеется датчик температуры, подключённый к ППВМ. Сигналы с датчика могут использоваться, например, для того, чтобы обнаружить перегрев, если такая возможность заложена в программе ППВМ.

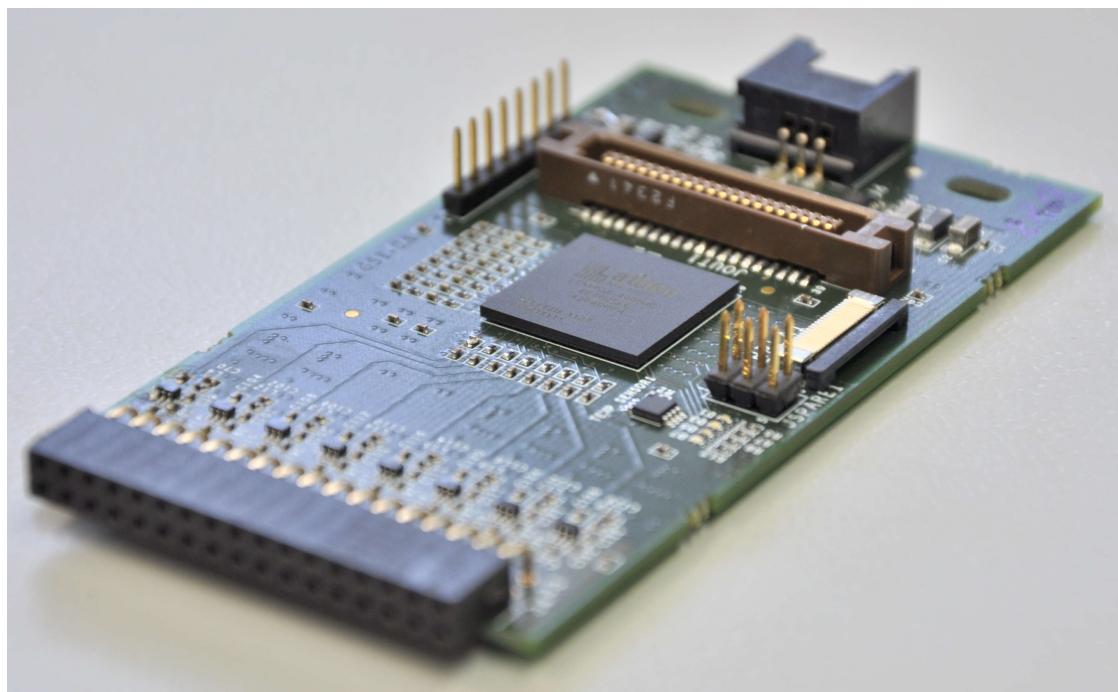


Рис. 3.3: Общий вид платы PADIWA.

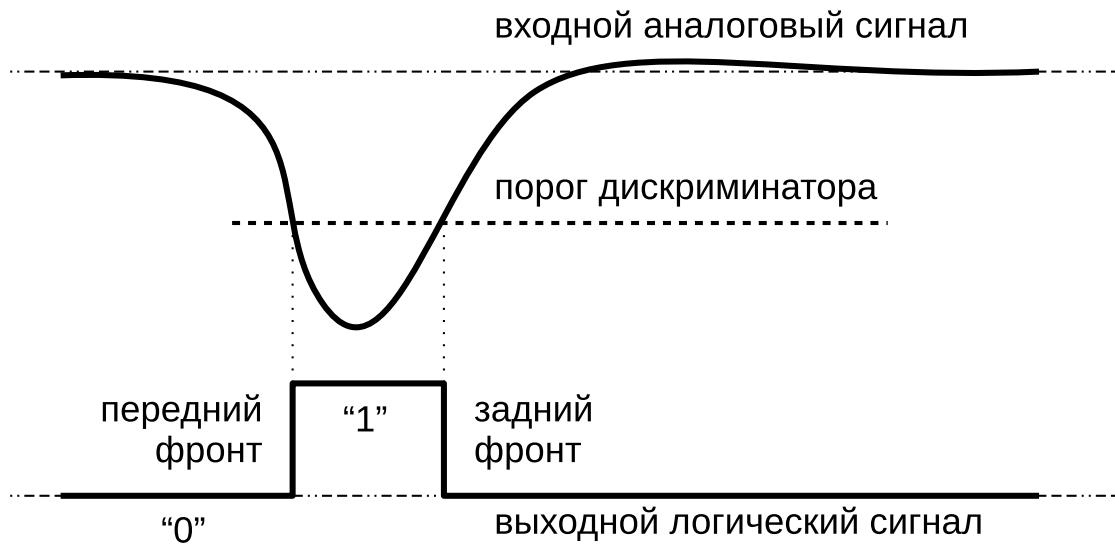


Рис. 3.4: Условная временная диаграмма функционирования дискриминатора.

Многофункциональная плата TRB v3 содержит 5 ППВМ, каждую из которых можно запрограммировать независимо. Различают 1 центральную ППВМ и 4 периферийные. В нашем случае 4 периферийные ППВМ запрограммированы как время-цифровые преобразователи (ВЦП), а центральная ППВМ — как концентратор данных. Такую конфигурацию платы будем называть TRB v3 (конфигурация 1).

Выходные логические LVDS сигналы со всех 16 каналов платы PADIWA поступает в одну из периферийных ППВМ платы TRB v3, где каждый входной канал разветвляется на два канала ВЦП — первый чувствителен к переднему фронту, второй — к заднему. К получившимся 32 каналам в каждой периферийной ППВМ добавляется канал синхронизации. Таким образом, на выходе всей платы TRB v3 имеются 132 канала.

Общий вид платы TRB v3 показан на рис. 3.5. Рядом с каждой периферийной ППВМ имеются специальные порты, к которым можно присоединить платы расширения. В частности, существует специальная плата расширения для подключения шлейфов от плат PADIWA. На плате TRB v3 имеются порты Ethernet, как RG45, так и оптический SFP, которые используются для двусторонней связи с другими платами TRB v3 или с компьютером.

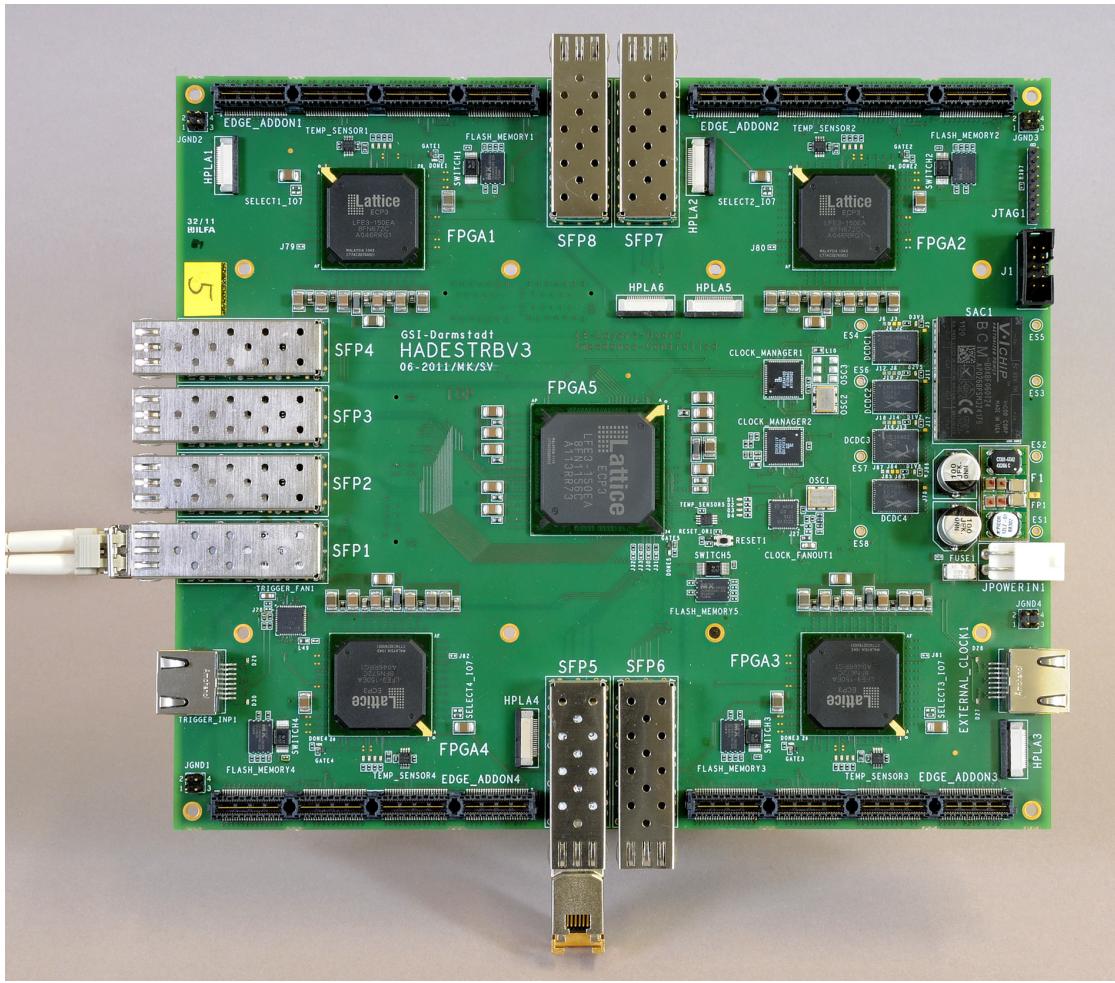


Рис. 3.5: Общий вид платы TRB v3.

Каждая периферийная ППВМ, разбивается на 32 области, в каждой из которых программируется одна и та же схема канала ВЦП. Каналы расположены в разных областях матрицы, поэтому каждый канал ВЦП имеет свою величину пути, проходимого сигналом внутри ППВМ. Нечетные каналы настроены на положительный перепад напряжения, т.е. на передний фронт, а четные каналы — на отрицательный перепад напряжения, т.е. на задний фронт. Обработка импульса из одного входного канала выполняется двумя каналами ВЦП, относительная задержка между которыми должна быть прокалибрована с помощью точного генератора прямоугольных импульсов. Особенности такой калибровки обсуждаются в 4.3. Отметим, что в ППВМ для каждого канала ВЦП имеется специальный счётчик количества зарегистрированных временных отметок, значение которого может быть опрошено независимо от основного потока данных. Этот счётчик может быть использован, например, для получения зависимости скорости счёта от порога дискриминатора с целью определения оптимального порога.

Регистрация момента времени в ВЦП осуществляется в два этапа. Грубое значение регистрируется кольцевым счётчиком, который управляется от тактового генератора с пе-

риодом 5 нс. Старшие 28 разрядов счетчика называются эпохой (epoch), а 11 младших разрядов называются грубым временем (coarse) [25]. При регистрации момента времени входного фронта значение времени кодируется двумя сообщениями — эпохой и собственно так называемой временной отметкой (timestamp). Чтобы уменьшить поток выходных данных значение эпохи, которое увеличивается каждые 10.24 мкс, передаётся однократно для группы временных отметок, принадлежащих данной эпохе.

Для более точного измерения применяется дополнительный 10-битный регистр точного времени (fine). В регистр пишется значение счётчика точного времени, реализованного с помощью технологии Tapped delay line (TDL) на 512-ти элементах. Теоретически, если все элементы задержки идентичны, полный период счётчика грубого времени, равный 5 нс, можно разбить на 512 отсчётов. Тогда точность измеренной временной отметки была бы равна 9.9 пс, а полное время рассчитывалось бы как $T = (\text{epoch} \cdot 2048 + \text{coarse} - (\text{fine}/512)) \cdot 5\text{ нс}$.

Однако, в силу неидеальности компонентов, существует разброс параметров элементов в линии задержки, следовательно, требуется калибровка результатов измерения точного времени относительно диапазона значений регистра. Процедура калибровки и анализ ее качества обсуждаются в секциях 3.5 и 4.3 соответственно.

Находящиеся на TRB v3 ППВМ формируют 4-байтовые сообщения одного из следующих типов: EVENT, SUBEVENT, SUBSUBEVENT HEADER, TDC HEADER, EPOCH COUNTER, TIMESTAMP, DEBUG. Логика формирования сообщений подробно описана в документации [26].

Рассмотрим для примера структуру сообщения типа TIMESTAMP, наиболее информативного для нашего анализа. В зависимости от номера канала это сообщение может нести информацию о фронте синхронизации SYNC, о переднем фронте хита LEAD или о заднем фронте хита TRAIL.

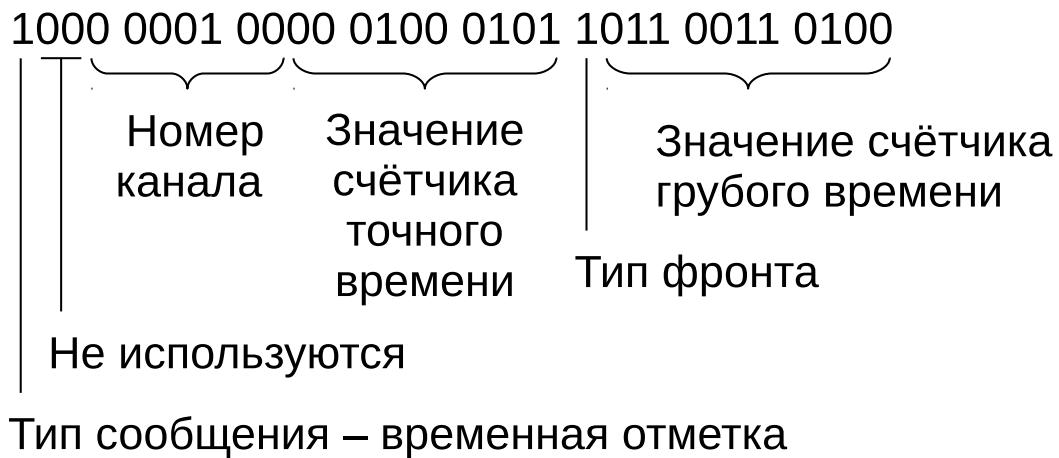


Рис. 3.6: Пример сырого сообщения типа “временная отметка”.

Старший бит (левый) указывает на то, что данное сообщение является временной отметкой. Следующие два бита не используются. Следующие 7 бит указывают номер канала 4. Затем 10 бит указывают значение счётчика точного времени 0x45. Далее вспомогательный бит edge, который на данный момент не используется. Последние 11 бит кодируют значение

ние счётчика грубого времени 0x334. Далее отсюда вычисляется полное значение времени в наносекундах (2681319745539.841309).

Необходимо отметить, что каждый канал считывания характеризуется некоторой индивидуальной задержкой между моментом рождения фотоэлектрона и значением отметки времени переднего фронта. Эта задержка определяется временем развития электронной лавины в динодной системе, временем распространения сигнала по проводникам и временем переключения логических элементов. Процедура коррекции задержек и ее особенности описаны далее в секциях 3.5 и 4.4.

3.2.2 Концентрация и ввод данных в ЭВМ

Это, видимо, нужно удалить в связи с более подробным описанием FLES В концепции системы сбора данных эксперимента СВМ предусмотрено 4 функциональных уровня, каждый из которых реализован соответствующими платами. В общем случае к детектору примыкает плата передней электроники (FEB – front-end board), где осуществляются аналоговые преобразования и оцифровка сигналов. Далее, данные в виде электрических цифровых сигналов поступают в плату считывания (ROB – readout board), где происходит концентрация данных и их пересылка по оптическому каналу. На следующем уровне расположены платы обработки данных (DPB – data processing board). DPB уплотняют данные с различных детекторов за счет удаления избыточной информации специфическим для каждого детектора способом и группируют эти данные в пакеты, называемые срезами времени (time slice). В каждый срез времени попадают сообщения со всех детекторов, имеющие временную отметку в заданном интервале. Далее они передаются по меньшему числу оптических каналов с более высокой пропускной способностью [27]. После этого данные поступают в память, доступную центральному процессору ЭВМ по высокоскоростной шине через платы интерфейса, называемые FLIB. Аббревиатура FLIB обозначает FLES Interface Board, а FLES [28], в свою очередь, обозначает First Level Event Selector, т.е. специализированный аппаратно-программный комплекс для построения событий “на лету” и их отбора по заданным критериям. Плата FLIB может быть реализована, например, путем программирования коммерческой PCI-E платы HTG K-7.

В случае пучковых тестов RICH плата передней электроники реализована как пара PADIWA-TRB v3 (конфигурация 1). В будущем планируется объединение функционала этих плат на одной плате DIRICH [29]. В качестве ROB используется плата TRB v3, сконфигурированная как концентратор. Плата DPB находится в стадии разработки прототипа, а плата FLIB была впервые применена в одном из протестированных вариантов системы сбора данных. При этом значительная часть измерений была выполнена с использованием стабильной системы сбора данных на основе DABC [30] и обычной сетевой карты.

3.3 Время-цифровой преобразователь

На рисунке 3.7 приведена условная схема функционирования одного канала ВЦП, дающая представление о причинах сдвига калибровочной таблицы и объясняющая минус в формуле расчёта полного времени.

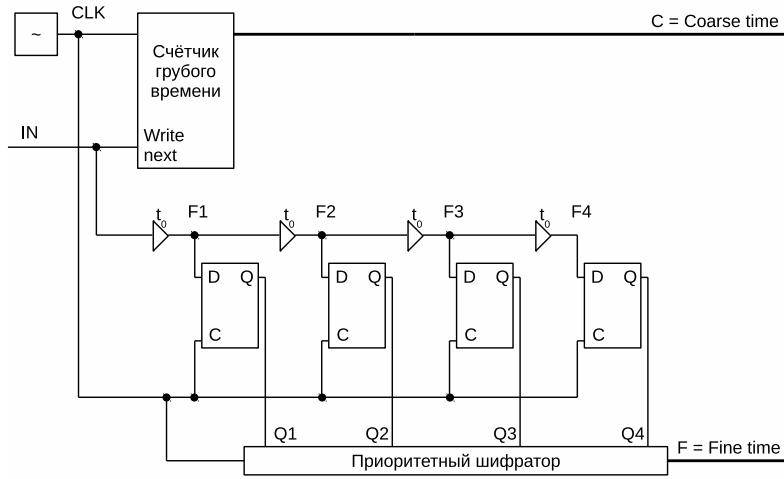


Рис. 3.7: Структурная схема одного канала ВЦП.

Имеется тактовый генератор частотой 200 МГц. Период такого генератора — 5 нс. Он управляет счётчиком грубого времени. Каждые 5 нс значение грубого времени увеличивается, но не выдаётся на выход. Счётчик точного времени выполнен по технологии Tapped delay line (TDL) — цифровая линия задержки (DDL) с промежуточными выходами. Используются элементы задержки t_0 , имеющие одинаковые характеристики в пределах некоторой точности. Количество элементов должно быть таким, чтобы полностью заполнить период между двумя отсчётами грубого времени. Регистрируемый фронт, поступающий на вход IN, проходит линию задержки, состоящую из нескольких элементов задержки. По мере прохождения линии фронтом триггеры переключаются из 0 в 1, каждый следующий через промежуток времени, равный t_0 . При поступлении следующего фронта от тактового генератора происходит считывание грубого времени и перенос значений выходов триггеров в приоритетный шифратор.

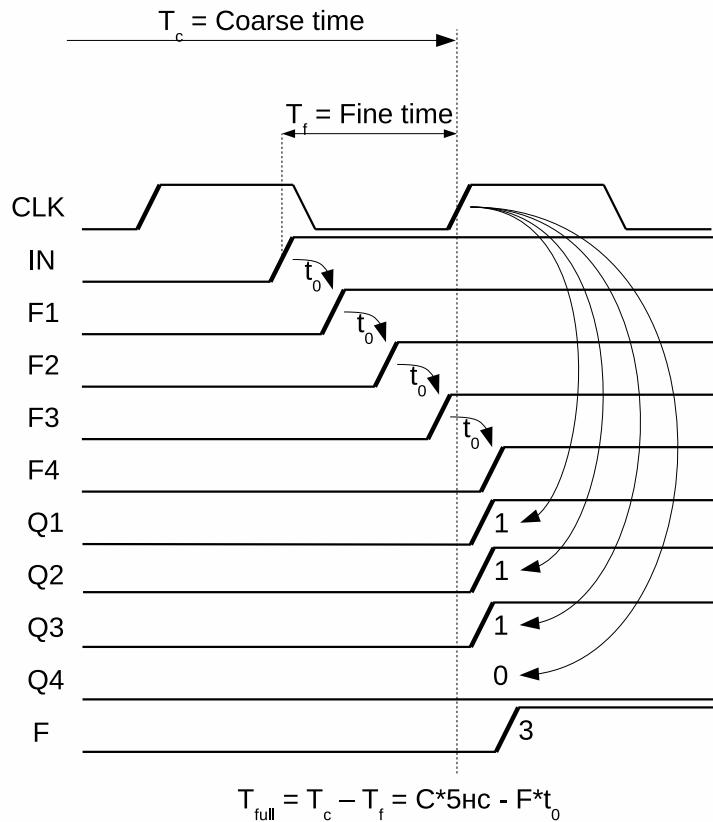


Рис. 3.8: Пример временных диаграмм при регистрации входного фронта.

Отличительная особенность такого шифратора — индифферентность к значению на входах $j < i$ при наличии логической единицы на входе i . Иными словами имеет значение только старший бит, а младшие игнорируются. (У обычного шифратора только один вход должен иметь единицу на входе). Шифратор преобразует номер последнего переключившегося триггера в число, обозначающее значение точного времени. Таким образом точное время должно вычитаться из грубого времени потому что линия задержки измерила время между моментом прихода входного сигнала (start) и моментом прихода следующего отсчёта грубого времени (stop).

Приведена наиболее понятная схема, фактическая же реализация отличается. Например, в качестве элемента задержки может выступать сам триггер. Тогда выход i -го триггера напрямую соединяется со входом $(i+1)$ -го триггера. Используемые нами ВЦП в ППВМ имеют в качестве элемента задержки ячейку матрицы, запрограммированную как полный сумматор.

ВЦП разрабатывался так, чтобы интервал 5 нс между двумя отсчётами грубого счётчика разбивался на 512 элементов. Тогда было бы достаточно 9-битного шифратора для формирования значения точного времени. Из-за того, что существует также и задержка сигналов в проводниках ненулевой длины, реально таблица может быть сдвинута. Например значение

точного времени 0 должно означать, что входной фронт пришёл одновременно с фронтом от тактового генератора (точнее совсем чуть-чуть раньше). Если учитывать задержку в проводниках на приведённой условной схеме, то получится, что команда считывания шифратору идёт дальше, чем до первого триггера. В таком случае никогда не будет принято нулевого значения и 9 бит для хранения точного времени будет недостаточно. По данной причине сообщение, несущее точное время, имеет длину 10 бит, а все калибровочные таблицы имеют правую границу на значении 1024.

Пример таблицы калибровки точного времени представлен в виде графика на рисунке 3.9. По оси абсцисс откладывается значение счётчика точного времени, а по оси ординат — значение точного времени в наносекундах.

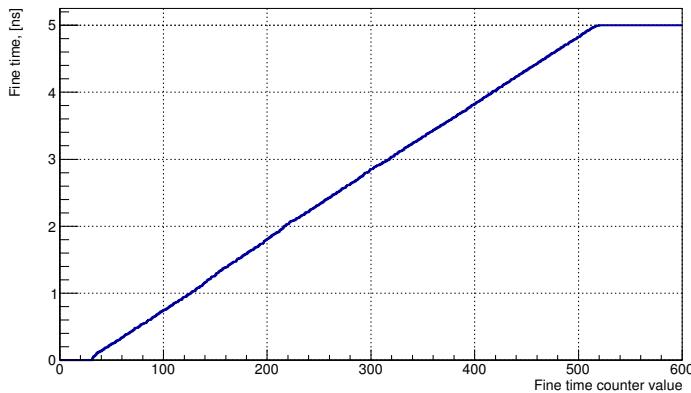


Рис. 3.9: Пример таблицы калибровки точного времени.

3.4 Экспериментальные установки

3.4.1 Экспериментальная установка на пучковых тестах

Исследование системы считывания и сбора данных проводилось в составе полнофункционального прототипа детектора RICH эксперимента СВМ в ходе комплексных пучковых испытаний прототипов нескольких детекторов того же эксперимента [31]. Подробности реализации прототипов детектора переходного излучения и времязадающего детектора содержатся в работах [32] и [33] соответственно. Схема установки представлена на рис. 3.10.

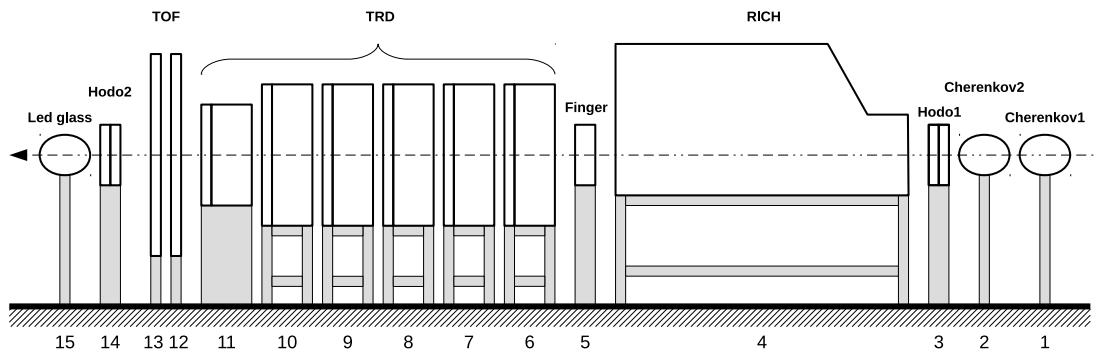


Рис. 3.10: Схема экспериментальной установки на пучковых тестах. 1,2 — пороговые газовые Черенковские счётчики; 3,14 — станции двухкоординатного годоскопа на основе сцинтилляционного оптического волокна; 4 — прототип детектора Черенковских колец; 5 — пластина из органического сцинтиллятора; 6-11 — станции прототипа детектора переходного излучения; 12-13 — станции прототипа время-пролётного детектора; 15 — электромагнитный калориметр из свинцового стекла.

Выход пучка Т9 ускорителя PS [34] в ЦЕРНе представляет собой смешанный вторичный пучок электронов, пионов и мюонов импульсом, настраиваемым в диапазоне 0.5 ГэВ/с — 10 ГэВ/с. В течение пучковых тестов пучок был настроен на импульс от 1 до 3 ГэВ/с. Длительность вывода составляла около 2 секунд, причем за это время регистрировалось в среднем 500 электронов.

Схема прототипа детектора RICH эксперимента СВМ представлена на рис. 3.11.

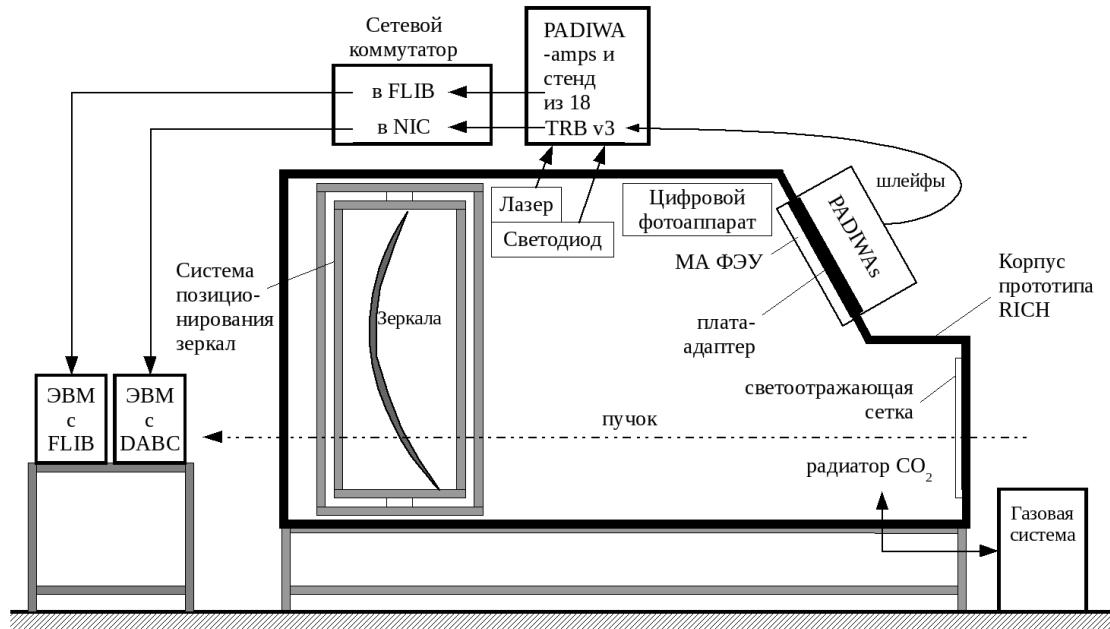


Рис. 3.11: Схема прототипа детектора RICH.

Габариты герметичного алюминиевого корпуса — 1.4 м в ширину, 1.2 м в высоту и 2.4 м вдоль пучка, при этом длина пути частицы в радиаторе до зеркал — 1.7 м. Радиатор детектора — углекислый газ под избыточным давлением 2 мбар при комнатной температуре. Показатель преломления газа для ближнего ультрафиолета составляет при этом $n=1.00045$. Стабилизация избыточного давления газа с точностью 0.1 мбар и его чистота обеспечиваются газовой системой, описанной в [35]. Абсолютное давление газовой смеси и температура мониторируются системой медленного управления. Актуальное значение показателя преломления автоматически вычисляется и сохраняется в данных.

Система позиционирования зеркал представляет собой раму верхнего уровня, вставляющуюся в корпус прототипа; вложенную раму, соединённую с основной рамой через два привода, обеспечивающие вращение вокруг вертикальной оси; внутреннюю раму, соединённую со вложенными рамами через два привода, обеспечивающие вращение вокруг горизонтальной оси. Сферическое зеркало радиусом кривизны 3 м состоит из 4 долей 40 см на 40 см. Каждая из долей крепится к внутренней раме через три моторизированных актуатора. Перечисленные двигатели позволяют удалённо, после установки детектора на пучке, позиционировать зеркала. Более подробно система позиционирования зеркал описана в [36].

Система диагностики положения зеркал [37] состоит из светоотражающей сетки, занимающей всю переднюю стенку корпуса прототипа, светодиода Roithner UVTOP240 [38] с длиной волны 245 нм и фотоаппарата, считываемого удаленно. Сетка сделана из полос ретрорефлектора шириной 10 мм и имеет прямоугольную ячейку шагом 100 мм по горизонтали и 110 мм по вертикали. Эта система позволяет контролировать точность поворота зеркал и, при наличии удалённого управления зеркалами, корректировать его. Также существуют алгоритмы расчёта поправок координат хитов для коррекции ошибок, вызванных неидеальным позиционированием зеркал. Идея метода заключается в следующем. Свет от светодиода, отражаясь от сетки и затем от зеркал, попадает в объектив фотоаппарата. На полученном кадре с помощью алгоритмов распознавания образов находятся линии сетки. При наличии отклонений зеркал от идеального положения, восстановленный образ сетки будет состоять из набора отдельных отрезков. Анализируя параметры отрезков, можно определить значения отклонений отдельных долей зеркала, значения поправок к поворотам отдельных долей зеркала, значения коррекций координат хитов.

Черенковское излучение фокусируется зеркалами на фоточувствительную камеру, содержащую матрицу 4 на 4 МА ФЭУ, шесть из которых — это МА ФЭУ Hamamatsu H12700 и десять — МА ФЭУ Hamamatsu H8500. Данные модели МА ФЭУ имеют сечение 52 мм на 52 мм. Часть фотоумножителей была предварительно покрыта слоем сместителя спектра толщиной 150-200 нм. В качестве сместителя спектра использовался паратерфенил ($\approx 40\%$ по массе) в полимерной матрице Paraloid B72. Сместитель спектра наносился методом погружения в раствор компонентов покрытия в дихлорметане, см. [39]. В определённый момент во время пучковых тестов сместитель спектра был счищен. Это позволило в дальнейшем оценить влияние сместителя спектра на эффективность регистрации одиночных фотонов и на временной разброс хитов, принадлежащих одному кольцу. Для мониторирования системы считывания и калибровки относительных задержек между каналами, наряду со светодиодом, использовался лазер Alphalas Picopower LD405 [40] с длиной волны 405 нм и длительностью импульса по паспорту менее 40 пс. Частота срабатывания лазера, так же как и светодиода, составляла 100 Гц. Интенсивность лазера была подобрана так, чтобы частота срабатывания каждого пикселя была на уровне 10% от частоты запуска лазера.

Считывание с каждого МА ФЭУ осуществлялось модулем, описанным в разделе 3.2.1. Механически все 16 МА ФЭУ монтировались на плату-адаптер, обеспечивающую герметичность корпуса и разводку высокого напряжения. Снаружи к плате-адаптеру монтировались платы предусилителей-дискриминаторов PADIWA, логический сигнал с плат PADIWA пе-

редавался по шлейфам, состоящим из витых пар и имеющих длину 2 м, к платам TRB v3 (конфигурации 1), установленным на корпусе прототипа. Для всей камеры потребовалось всего 64 платы PADIWA и 16 плат TRB v3 (конфигурации 1). Данные с 16 плат TRB v3 поступали на ещё одну, 17-ю плату TRB v3 особой конфигурации, которая также являлась генератором и распределителем триггера считывания для всех плат TRB v3. Импульсы с генераторов, управляющих лазером и светодиодом, а также сигналы от детекторов пучка обрабатывались платами PADIWA-amp (плата, подобная PADIWA, но позволяющая измерять амплитуду сигнала и имеющая в два раза меньшее число каналов [24]) и оцифровывались ВЦП на ещё одной, 18-й плате TRB v3 также нестандартной конфигурации, совмещающей ВЦП и концентратор данных. Параллельно функционировали две системы сбора данных — одна принимала данные через стандартный сетевой интерфейс (сетевой концентратор) с каждой платы TRB v3 по медному носителю, а другая через FLIB с одной (18-й) платы TRB v3. Схема считывания всей камеры и детекторов пучка представлена на рис. 3.12. Отметим, что ЭВМ с установленной в неё платой FLIB, использовалась для приёма данных не только от прототипа RICH, но и от других детекторов.

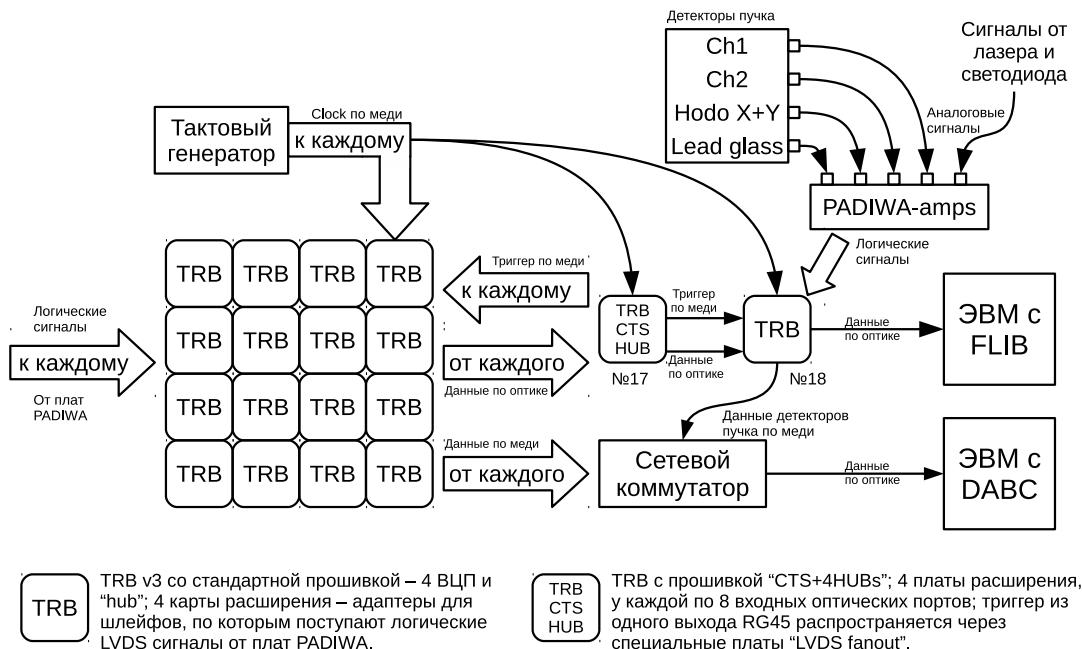


Рис. 3.12: Схема считывания всей камеры и детекторов пучка.

3.4.2 Лабораторный стенд

Система считывания на основе платы PADIWA впервые использовалась на пучковых тестах СВМ в ноябре 2014 г. Простейший анализ набранных данных показал, что некоторые распределения временных отмечок не поддаются очевидному объяснению. В связи с этим потребовалось собрать лабораторный стенд, позволяющий более подробно исследовать особенности работы одного многоканального модуля системы считывания, описанного в разделе 3.2.1. В некоторых измерениях выходной LVDS сигнал с PADIWA не оцифровывался ВПЦ, а считывался осциллографом с помощью активного зонда. Для лучшего понимания

особенностей работы исследуемой системы считывания и сбора данных в том же лабораторном стенде был реализован более информативный, но медленный вариант системы считывания и сбора данных на основе 128-канальной микросхемы n-XYTER, каждый канал которой измеряет момент времени прихода переднего фронта и амплитуду входного сигнала. Эта система состоит из платы передней электроники, подключаемой через печатную плату-адаптер к МА ФЭУ и через контроллер считывания SysCore ROC [41] к ЭВМ. Для считывания одного МА ФЭУ достаточно 64 каналов, то есть половины каналов одной платы передней электроники.

Схема лабораторного стенда приведена на рис. 3.13.

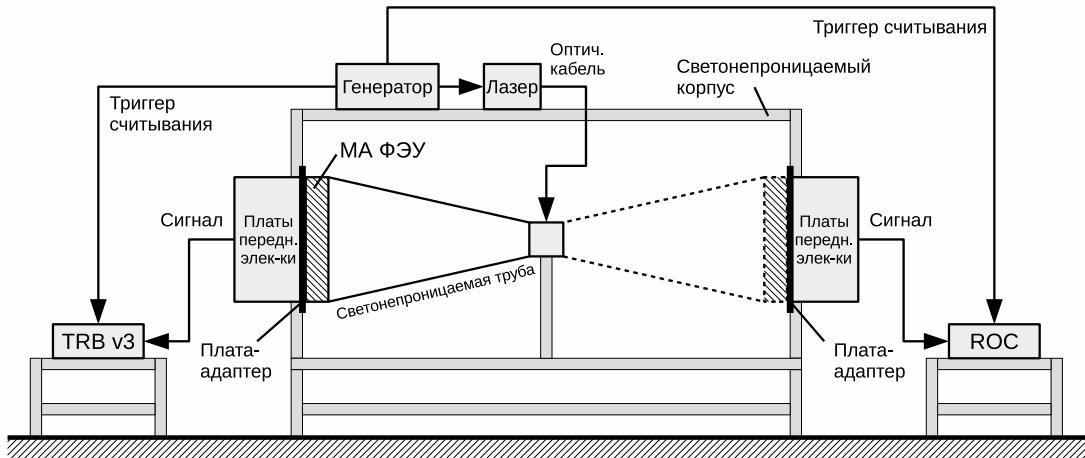


Рис. 3.13: Схема лабораторной установки.

Стенд собран в светонепроницаемом корпусе размером 80 см на 80 см и длиной 2 м. В качестве источника света использовался такой же лазер Alphalas Picopower LD405 [40] с поставляемым с ним генератором Alphalas PLDD-250 [40], как и в пучковых тестах. Свет от лазера поступал внутрь корпуса по оптоволокну. Для того чтобы обеспечить равномерное освещение поверхности МА ФЭУ свет лазера проходил через рассеивающее матовое стекло. Интенсивность лазера подобрана так, чтобы каналы МА ФЭУ работали в одноэлектронном режиме. Частота регистрации фотоэлектронов в каждом канале составляет около 10% от частоты вспышек лазера.

На расстоянии приблизительно 30 см от рассеивающего стекла расположены МА ФЭУ H12700. Для того чтобы обеспечить максимально чистые измерения, выполнена тщательная изоляция МА ФЭУ от внешнего света. Рассеивающее стекло и МА ФЭУ были помещены в черную, специально изготовленную на 3D принтере, пластиковую трубу, которая, в свою очередь, была помещена в светоизолированный корпус.

Известно, что требуется некоторое время, чтобы МА ФЭУ, находившийся на свету, вышел на свет, поэтому перед началом измерений после закрытия корпуса обязательно выдерживался интервал не менее одного часа. В любой момент была возможность удалённо выключить лазер и исследовать темновой шум МА ФЭУ. Для снижения наводок от люминесцентных ламп на время измерений свет в помещении выключался.

Две системы считывания и сбора данных были установлены одновременно, каждая на своей стороне корпуса. Упомянутая выше пластиковая труба, рассеивающее стекло и МА ФЭУ поворачиваются как единое целое, обеспечивая одинаковые условия засветки

МА ФЭУ в положениях, соответствующих работе с обеими системами считывания.

Опорные печатные платы-адаптеры необходимы для того, чтобы на них с одной стороны крепились МА ФЭУ, а с другой — платы передней электроники. Плата-адаптер вмонтирована стенку коробки и выполняет роль каркаса и светоизолятора. Также по ней разведено питание МА ФЭУ. Вся считающая электроника питалась низким напряжением, а МА ФЭУ высоким напряжением от высоковольтного источника.

Обе системы считывания и сбора данных являются самозапускающимися в том смысле, что каждый импульс на входе, при преодолении установленного порога, регистрируется и заносится в выходной буфер. Однако для того, чтобы данные из выходного буфера были отправлены в ЭВМ, необходимо периодически посыпать во вспомогательный вход контроллера считывания специальный импульс, называемый триггером считывания. В нашей установке импульсы генератора, управляющего лазером, одновременно играют роль триггера считывания выходного буфера. В используемых системах считывания и сбора данных триггер считывания автоматически поступает во входной поток данных. Это позволяет анализировать зарегистрированные временные отметки, сопоставляя их с моментом вспышки лазера. Съём данных с обеих систем считывания и сбора данных осуществлялся по стандартному Ethernet кабелю в сетевой интерфейс ЭВМ.

3.5 Программное обеспечение

Программное обеспечение системы считывания и сбора данных прототипа СВМ RICH представляет собой набор модулей приема, первичной обработки и сохранения данных, реализованных в рамках программного каркаса CbmRoot [42]. CbmRoot вместе в FLESnet [43] образуют инфраструктуру, позволяющую выполнять приём данных, моделирование, реконструкцию и анализ данных эксперимента СВМ.

Соответствующим образом сконфигурированное приложение, написанное в рамках CbmRoot, может быть запущено на ЭВМ, как частный случай на распределённой вычислительной системе. Все этапы от считывания до анализа могут быть выполнены “на лету”, без записи промежуточных результатов на диск. В ходе описываемых в данной статье тестов использовалась последовательность обработки данных, изображённая на рис. 3.14. Программная реализация, функционал и взаимодействие отдельных блоков описаны ниже.

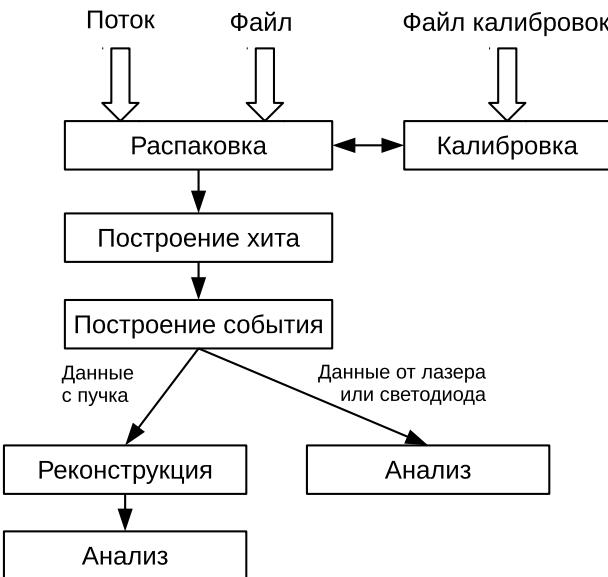


Рис. 3.14: Диаграмма взаимодействия программных модулей.

3.5.1 Распаковка

Распаковка — это первый этап обработки данных, поступающих с электроники. В CbmRoot есть возможность обрабатывать данные как поступающие напрямую с детекторов, так и сохранённые в файле (с помощью DABC в формате HLD — HADES list mode data format [44]). При использовании FLIB распаковке предшествует запуск небольшого интерфейстного модуля, который выделяет из поступающего от FLESnet потока данных в формате временных интервалов (TSA — Time Slice Archive), сообщения, относящиеся к детектору RICH. Распаковка реализована как task-класс *CbmRichTrbUnpack* и в результате выполнения каждой итерации на выходе формируется *TClonesArray* с объектами класса *CbmTrbRawMessage*.

3.5.2 Калибровка точного времени

В процедуре калибровки точного времени воплощена известная техника калибровки счётчика цифровой линии задержки, реализованного с помощью технологии Tapped delay line [45], основанная на том, что распределение времен прихода сигналов должно быть равномерным по временному интервалу, занимаемому всеми элементами задержки. В результате анализа набранной порции данных для каждого канала строится дискретная функция $f_{calib}(Fine)$, называемая таблицей перехода от значения счётчика к значению точного времени в наносекундах. При использовании таблицы калибровки точного времени полное время вычисляется как $T = Epoch \cdot 2048 \cdot 5 + Coarse \cdot 5 - f_{calib}(Fine)$ нс.

Процедура калибровки точного времени реализована в singleton-классе *CbmTrbCalibrator*, который не является частью конвейера обработки данных — обращение к объекту данного класса может производиться из любого места в программе. Присутствует возможность сохранения таблиц калибровки в отдельном файле, что ускоряет многократные расчёты за счёт повторного использования однократно рассчитанных таблиц.

3.5.3 Коррекция задержек между каналами

Для коррекции задержек между каналами в классе *CbmTrbCalibrator* реализована возможность импорта таблицы коррекций, построенной предварительно с помощью CmbRoot-макросов “ExtractDelays” и “BuildDeltaTable” на основе результатов первого прогона анализа. Первый макрос извлекает параметры гистограмм в текстовом виде из многочисленных файлов результатов анализа, полученных с помощью CmbRoot, возможно, с применением параллельных расчётов. Второй макрос стоит таблицу коррекций по данным параметрам.

Алгоритм коррекции задержек состоит в следующем. Введём сплошную нумерацию пикселей по всей фоточувствительной камере, состоящей из множества МА ФЭУ. Полное число пикселей $M = 64 \cdot N$, где N — число МА ФЭУ. Для анализа отбираются все передние фронты, имеющие временную отметку, попадающую в заданное временное окно относительно триггера срабатывания лазера. Ширина и положение окна зависят от экспериментальной установки (например, разницы длин кабелей, точности регистрации триггера) и характеристик лазера и подбираются в соответствии с распределением, построенным по конкретному набору данных. Обычно ширина составляет около 100 нс, а левая граница сдвинута от триггера на 20 нс. По всему массиву отобранных данных строятся гистограммы разности временных отметок i -го и j -го каналов, где i и j пробегают значения от 1 до M . В качестве меры разности задержек между каналами можно взять по выбору пользователя либо среднее значение распределения, либо наиболее вероятное. Полученные значения заполняют кососимметричную матрицу A размерности $M \cdot M$. В дальнейшем пользователь может задать опорный канал, относительно которого будет создана таблица коррекций, являющаяся, по сути, столбцом матрицы A .

3.5.4 Построение хита

Сигнал от каждого зарегистрированного фотона, называемый хитом, состоит из двух сообщений, содержащих временные отметки переднего и заднего фронтов. Т.к. разные каналы имеют разные задержки и вероятность регистрации отдельных фронтов не равна 100%, необходимо было в анализе данных реализовать алгоритм подбора пар фронтов. Данная процедура реализована в task-классе *CbmTrbEdgeMatcher*, который стоит в конвейере после распаковки и фактически выполняется после применения всех калибровок.

Для каждого внешнего канала был реализован буфер сообщений, который наполнялся передними фронтами по мере их поступления. Далее, как только приходил задний фронт, из буфера выбирался наиболее близкий по временной отметке передний фронт внутри заданного допустимого временного окна. На рис. 3.15 приведён пример буфера передних фронтов для заданной пары внутренних каналов в момент прихода одного заднего фронта. Время над порогом (ToT) — параметр хита, говорящий об амплитуде сигнала. Он вычисляется как разница временных отметок заднего и переднего фронтов в подобранной паре. Допускаются как положительные, так и отрицательные значения ToT, однако в обе стороны накладывается ограничение.

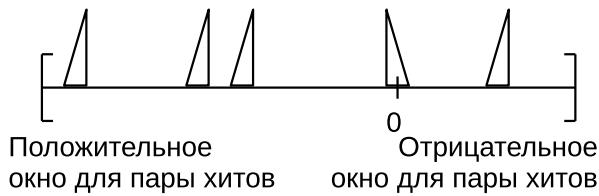


Рис. 3.15: Постановка задачи поиска пар фронтов для одного входного канала.

Т.к. не всегда присутствует соответствующий парный (передний либо задний) фронт, буфер постепенно наполняется и его необходимо очищать, чтобы избежать переполнения. Если для поступившего заднего фронта нет кандидата переднего фронта в буфере это означает, что передний фронт не был зарегистрирован. В таком случае этот задний фронт отбрасывается. Количество ненайденных фронтов сильно зависит от загруженности входного канала ВЦП, которая в свою очередь зависит от порога дискриминатора. При низком пороге регистрируется высокочастотный шум электроники, что приводит к формированию огромного потока выходных сообщений, которые не могут быть переданы из-за ограниченной пропускной способности выходного тракта системы считывания.

Предусмотрена возможность допускать одиночные передние фронты в качестве хитов, однако практика показала, что в этом нет смысла, т.к. в нормальном режиме ненайденные пары в основном обусловлены ошибками ВПЩ и доля таких сообщений пренебрежимо мала — менее $2 \cdot 10^{-4}$.

3.5.5 Построение события

Тот факт, что физическая программа эксперимента СВМ подразумевает исследование очень редких явлений, для которых практически невозможно вырабатывать аппаратный триггер, привёл к решению разработать и использовать бестриггерную систему считывания. В бестриггерной системе считывания каждый канал передней электроники вырабатывает сообщение при преодолении входным аналоговым сигналом установленного порога. Получается, что электроника выдаёт для программного обеспечения непрерывный поток никак не сгруппированных сообщений, содержащих временную отметку. Для того, чтобы выполнять физический анализ, необходимо в этом непрерывном потоке выделять осмысленные группы, которые мы называем событиями. Строго говоря, задача построения событий — это одномерная задача кластеризации на оси времени с последующим отбором кластеров по некоторым критериям.

Задача также усложнена тем фактом, что электроника не может обеспечить непрерывный поток сообщений, упорядоченных по времени регистрации. Происходит группировка сообщений в так называемые DAQ-события, которые необходимы для обеспечения передачи информации, а сообщения внутри DAQ-событий могут быть упорядочены произвольно. Соответственно первый этап построения события — упорядочивание сообщений.

В данных с пучковых тестов 2014 г., для того, чтобы определить, является ли распознанная группа событием, можно использовать сигналы с детекторов пучка — пороговых черенковских счётчиков, гodosкопов и др. В лабораторных данных, где выполнялись измерения с лазером, в качестве триггера можно использовать сигнал от генератора, управляющего лазером. В ситуации, когда нет дополнительной информации, как в случае полного детектора RICH в итоговом эксперименте, необходимо принимать решение о том, является ли распознанная группа событием, или нет, на основе исключительно информации, получен-

ной из этой группы. Распознанный кластер может являться событием, но чаще всего будет состоять из одного сообщения — шумового хита. Следовательно можно использовать кол-во хитов в событии для подавления шумов, что особенно актуально для детектора СВМ RICH, где выполняется реконструкция черенковских колец, требующая некоторого минимального числа хитов в плоскости реконструкции.

(Сначала общая идея, что в любом случае будет некоторое временное окно и все сообщения попадающие в окно формируют событие. Размеры окна — один из параметров построителя событий, которым можно играть с целью повышения эффективности.)

(А ниже описывается предлагаемый алгоритм)

В силу того, что электроника бестриггерная и приём данных осуществляется порциями, называемыми DAQ-событиями, никак не связанными с реальными событиями, для формирования корректной входной информации для реконструкции и дальнейшего анализа данных необходимо выполнять процедуру построения события.

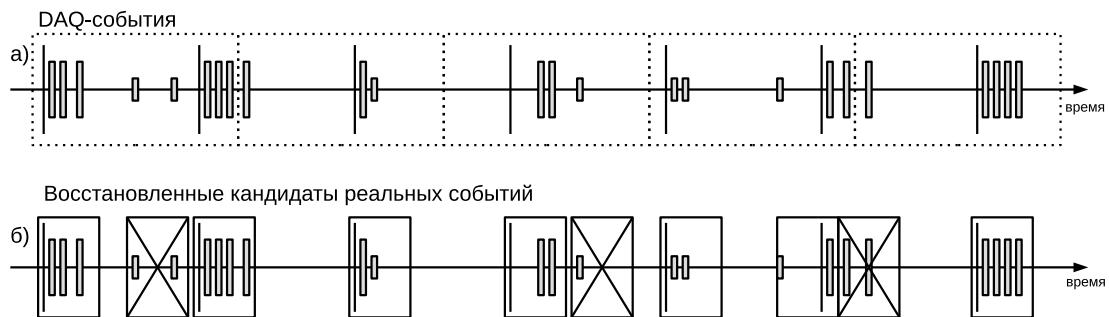


Рис. 3.16: Идея алгоритма построения события: (а) поступающие данные, сгруппированные в DAQ-события; (б) восстановленные кандидаты реальных событий; перечёркнуты отбрасываемые кандидаты, не содержащие триггер. Тонкие длинные линии — триггерные сигналы, прямоугольники средней длины — сигналы, связанные со светом, короткие прямоугольники — шумовые сигналы.

Рассмотрим некоторый интервал времени, приведённый на рис. 3.16, в течение которого поступают:

- Триггерные сигналы — импульсы с генератора, питающего лазер, либо сигналы с детекторов пучка;
- Сигналы, скоррелированные с импульсами с генератора или детекторами пучка, т.е. связанные со светом;
- Шумовые сигналы, распределённые равномерно во времени.

Очевидно, что реальное событие может попасть на границу DAQ-событий, следовательно, необходимо при построении реальных событий смотреть на несколько DAQ-событий. Также в силу особенностей электроники не гарантируется, что входная информация поступает упорядоченной во времени. Поэтому периодически случается, что хиты реального события, пришедшего по большей части в i -м DAQ-событии, обнаруживаются в $i+1$, реже $i+2$, и даже $i+3$ и последующих DAQ-событиях. Следовательно, требуется сначала распознать кластеры хитов в достаточно широком интервале времени — потенциальные события,

а затем по наличию заданного типа триггера выбрать реальные события, содержащие либо черенковские кольца, либо вспышки лазера.

Реализован данный алгоритм с помощью буфера хитов. По мере распаковки входных сообщений, построенные хиты заносятся в буфер. На каждой итерации осуществляется распознавание кандидатов событий в буфере и определяется их количество N . Когда N достигает заданного минимального уровня, на каждой итерации, помимо приёма одного входного DAQ-события, осуществляется выброс выходного кандидата реального события. Так как одно DAQ-событие может содержать несколько кандидатов, буфер будет расти. Для того, чтобы избежать переполнения, устанавливается верхний предел. Когда N достигает этого предела, осуществляется сброс событий на выход по принципу FIFO до заданного минимального уровня. На рис. 3.17 приведён отрывок диаграммы наполненности буфера по мере обработки входного потока. В данном примере были установлены следующие параметры: минимальное кол-во событий в буфере 200, максимальное — 500. Отметим, что кол-во событий в буфере может превышать заданное максимальное значение, если в одном DAQ-событии содержится более одного реального события. По окончании входного потока содержимое буфера обрабатывается полностью и все распознанные события подаются на выход.

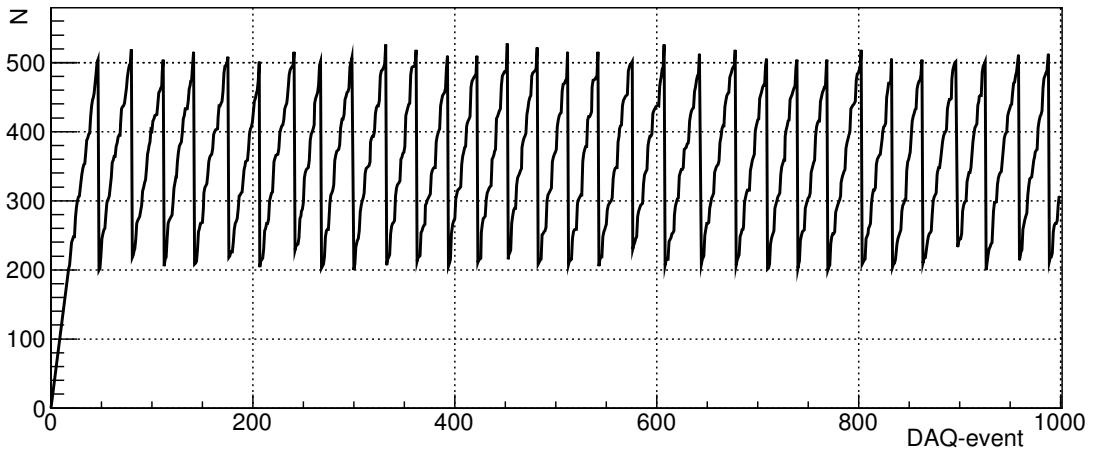


Рис. 3.17: Диаграмма наполненности буфера найденных событий в зависимости от номера обработанного входного DAQ-события.

3.5.6 Реконструкция

Реконструкция в CBM RICH означает поиск колец по хитам в плоскости реконструкции. В контексте реконструкции можно рассматривать хит как загоревшийся пиксель МА ФЭУ. Конус черенковских фотонов, после фокусировки зеркалами, пересекает поверхность фоточувствительной камеры, которая в общем случае может состоять из нескольких плоскостей. Первый этап реконструкции — перевод хитов из плоскостей камеры в плоскость реконструкции. Затем выполняется поиск колец по хитам. В CbmRoot есть реализации нескольких алгоритмов поиска колец. Наибольший практический интерес представляет алгоритм распознавания колец черенковского излучения, основанный на проеобразовании Хаффа, описанный в работах [46, 47]. Реализация данного алгоритма была специально адапти-

рована для данных пучковых тестов, в которых ожидается одно кольцо на событие. Данный алгоритм реализован в классе *CbmRichProtRingFinderHoughImpl*, унаследованном от *CbmRichProtRingFinderHough* и далее от *CbmRichRingFinder*. После этого определяются параметры кольца и далее реконструкция с применением информации с других детекторов.

Глава 4

Результаты анализа

4.1 Испытание системы сбора данных с использованием FLIB

Значительная часть данных была набрана параллельно двумя системами сбора данных. Было проведено побайтное сравнение результатов распаковки обоих потоков. На массиве составляющем примерно 10^7 сообщений расхождений не выявлено. Таким образом, продемонстрирована работоспособность концепции формирования временных интервалов и ввода данных в компьютер с использованием FLIB. Приведённые в следующих разделах результаты получены на основе данных, принятых через стандартный сетевой интерфейс с применением DAQ ПО на основе DABC [30].

4.2 LeadingEdgeDiff

Один из этапов обработки данных — построение событий. В данной работе рассматривается два типа событий — сигналы от лазера и сигналы от черенковского кольца. В любом случае, событие — это структура данных, содержащая информацию о хитах, сгруппированных по времени. Каждый хит содержит, как минимум, временную отметку момента прихода переднего фронта сигнала и номер канала, который в случае СВМ RICH указывает номер пикселя фоточувствительной камеры, т.е. говорит о геометрическом положении зарегистрированного фотона.

Данное исследование посвящено, в первую очередь, временным характеристикам системы считывания, поэтому в основном речь пойдёт о временных отметках.

Очевидно, что для каждого события можно построить несколько распределений, которые на большом массиве данных, т.е. на многих событиях, характеризуют систему считывания и могут быть использованы для калибровки электроники с целью повышения временного разрешения системы. Т.к. событие имеет максимальную ширину, определяемую размерами окна в алгоритме построения событий, распределения могут иметь “обрезанные хвосты”, которые, однако, невозможно избежать.

Пусть событие содержит N хитов. Введём внутри события нумерацию хитов от 0 до N . Пусть внутри события хиты упорядочены по времени, т.е. хит с временной отметкой t_0 был зарегистрирован раньше остальных, а хит с временной отметкой t_N — позже всех. Такой порядок может, например, обеспечиваться естественным образом алгоритмом построения событий. Внутри события все временные отметки зарегистрированы в разных каналах —

множественные хиты в одном канале в одном событии являются признаком того, что порог дискриминатора установлен слишком низко и регистрируются шумы. Введём в рассмотрение распределение ω разностей временных отметок всех хитов, кроме первого, относительно первого, т.е. распределение

$$t_j - t_0, \text{ где } j \in [1..N].$$

Также введём распределение σ_1 всех пар временных отметок одного события, т.е.

$$t_j - t_i, \text{ где } i \in [0..N], j \in [0..N], i \neq j.$$

Очевидно, что в такой формулировке одна и та же пара временных отметок войдёт в распределение дважды с разными знаками — например, $t_1 - t_2$ и $t_2 - t_1 = -(t_1 - t_2)$. Это делает распределение симметричным, среднее значение строго равно 0, а ширина распределения чуть больше, чем в случае, когда нет дублирования информации. Введём непрерывную нумерацию каналов и примем, что в разности $t_j - t_i$ первая временная отметка была зарегистрирована каналом a , а вторая — каналом b . Введём распределение σ_2 , по сути очень похожее на σ_1 , но без дублирования информации, в котором будем учитывать только пары, у которых $b > a$.

В идеальной ситуации, если событие соответствует одной вспышке лазера или одному черенковскому кольцу, и отсутствуют факторы, размывающие время регистрации, все разницы были бы равны нулю. В качестве таких размывающих факторов можно привести, например, следующие: временные характеристики лазера, разброс геометрических путей черенковских фотонов, разброс времени прохождения электронной лавины в динодной системе ФЭУ, дребезг сигналов в передней электронике. Из-за перечисленных явлений распределение ω имеет следующую форму — (описание). Распределение σ_2 — (описание).

Среднее значение либо положение максимума распределения σ_2 можно использовать для того, чтобы определить значение поправки для данной пары каналов. Если выполнить анализ с применением коррекций, то вид всех распределений изменится. ω сгруппируется ближе к нулю, σ_2 передвинется к нулю, а σ_1 сузится к нулю.

Представляется возможность анализировать различные области фоточувствительной камеры. Интересно группировать хиты в соответствии с тем, какой электроникой они обрабатываются. В данном анализе было введено 4 подмножества: 1 пара каналов, 16 каналов одной платы передней электроники, 64 канала одного МА ФЭУ, 256 каналов 4 МА ФЭУ, образующих площадку 2x2 МА ФЭУ в одном углу камеры. При том, что вся фоточувствительная камера на пучковых тестах имела размер 4x4 МА ФЭУ, рассматривать более 4 МА ФЭУ одновременно не имеет смысла, т.к. в прототипе были установлены различные модели МА ФЭУ, некоторые покрыты сместителем спектра, а некоторые нет.

4.3 Калибровка точного времени (Fine time calibration)

Пример таблицы калибровки точного времени, полученной на данных лабораторных тестов, представлен в виде графика на рис. 4.1. По оси абсцисс откладывается значение счётчика точного времени, а по оси ординат — значение точного времени в наносекундах. Вид графика не зависит от того, по каким данным он был построен, так как он определяется архитектурой время-цифрового преобразователя. Обратим внимание, что в показанном примере в диапазоне значений десятибитного счетчика точного времени интервалу равному периоду грубого счетчика, т.е. 5 нс, соответствуют отсчеты от 30 до 520. Точные границы интервала определяются значениями задержек на элементах цифровой линии задержки. Эти величины индивидуальны и зависят от флюктуаций технологического процесса.

С целью понимания особенностей работы счётчиков точного времени, каждая таблица калибровки точного времени была аппроксимирована кусочно-линейной функцией. На

рис. 4.2 показан пример разности значений функции калибровки точного времени и линейной функции. Видно, что отклонения не превышают 60 пс.

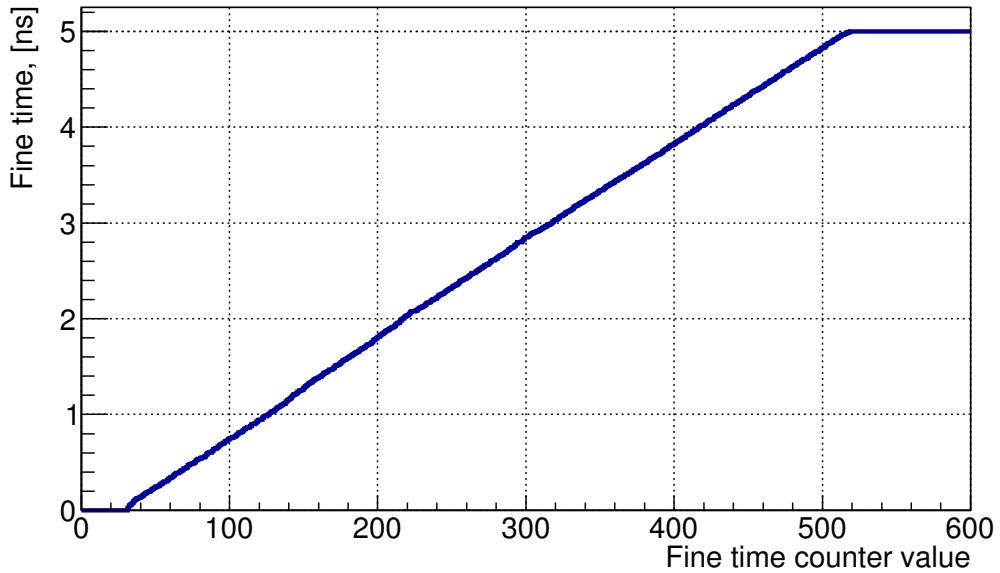


Рис. 4.1: Пример калибровочной кривой.

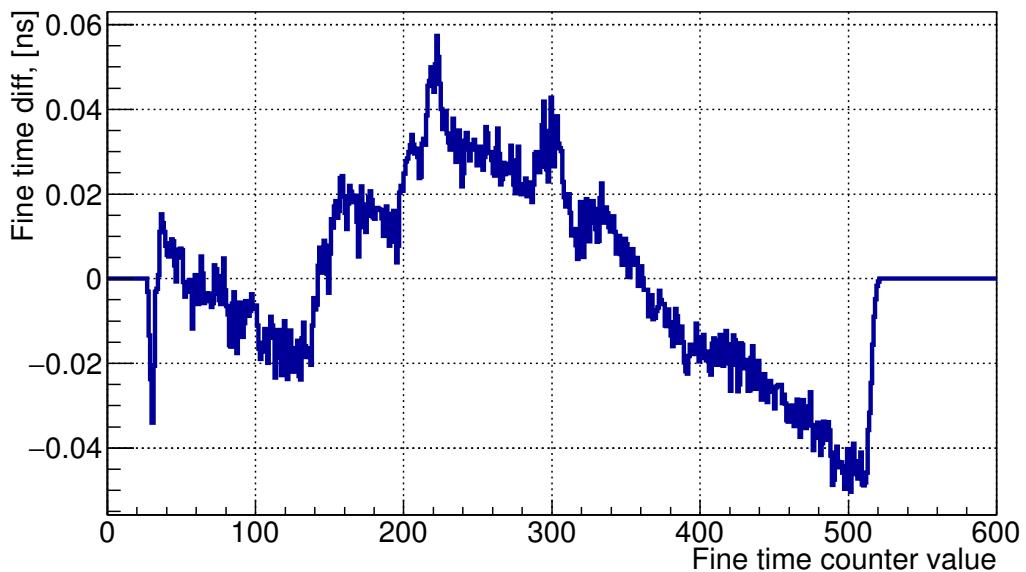


Рис. 4.2: Отклонение калибровочной кривой от линейной функции.

Каждая аппроксимирующая кусочно-линейная функция состоит из трёх отрезков и может быть однозначно описана двумя координатами изломов, которые приблизительно соответствуют двум крайним рабочим значениям счётчика точного времени. Параметры линейных функций для всех каналов отображены на двумерной диаграмме на рис. 4.3. Видно, что хотя параметры и локализованы в двух областях, распределение достаточно компактное.

Для оценки влияния калибровки на точность регистрации временных отметок можно исследовать как одновременные фронты на разных каналах ВЦП, так и длительности прямоугольных импульсов во входных каналах, полученных с помощью высокоточного генератора прямоугольных импульсов. В работе [48] показано, что предельное временное разрешение в обоих случаях одинаково. Ниже мы используем второй подход.

В процедуре калибровки для каждого канала была выполнена замена точной калибровочной таблицы сначала индивидуальной линейной функцией данного канала, а потом общей функцией, усредненной по всем каналам (параметры этой функции показаны на рис. 4.3 сплошным квадратом). Полученные распределения измеренной ширины импульса в исследуемом входном канале показаны на рис. 4.4. Там же показаны результаты без калибровки.

Видно, что точная калибровка точного времени необходима для достижения предельного разрешения ВЦП. Ширина распределения разностей временных отметок в двух независимо флюктуирующих каналах ВЦП составляет 30 пс (FWHM), что соответствует временному разрешению 21 пс. Использование индивидуальной линейной функции приводит к увеличению ширины до 70 пс, а усреднённой — до 90 пс в наиболее неблагоприятных каналах. Отметим, что применение усредненной калибровки устраниет двухпиковую форму, характерную для распределения без калибровки, но в некоторых случаях приводит при этом к увеличению ширины.

Таким образом, при невозможности выполнить калибровку точного времени, например, из-за недостаточного массива данных, предоставленных для анализа, в условиях нашей задачи, когда характерное временное разрешение составляет несколько сотен пикосекунд, возможно применение усреднённой линейной функции без заметного снижения точности.

Использование усреднённой калибровки может быть особенно полезно при измерении разности временных отметок, полученных ВЦП различного типа, поскольку тогда, в отличие от нашего случая, не происходит сокращения начального сдвига кусочно-линейной функции относительно нуля регистра точного времени.

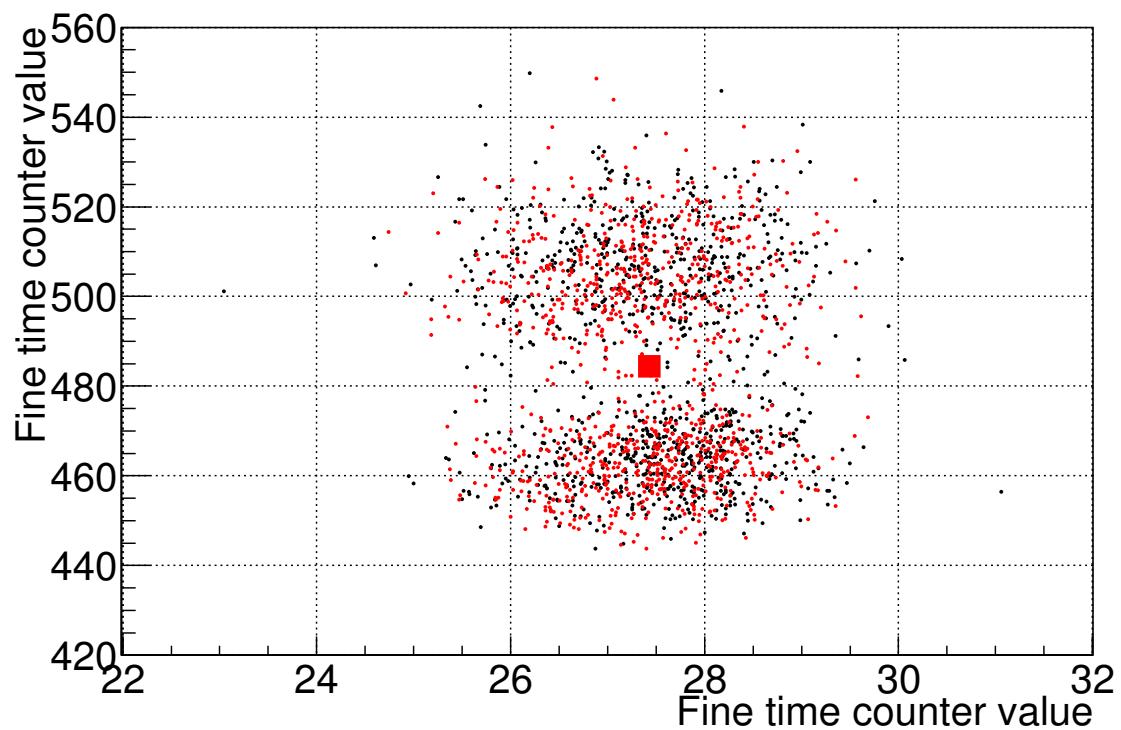


Рис. 4.3: Распределение координат точек излома аппроксимирующих кусочно-линейных функций. Квадратом отмечено среднее значение, используемое для глобальной псевдокалибровки.

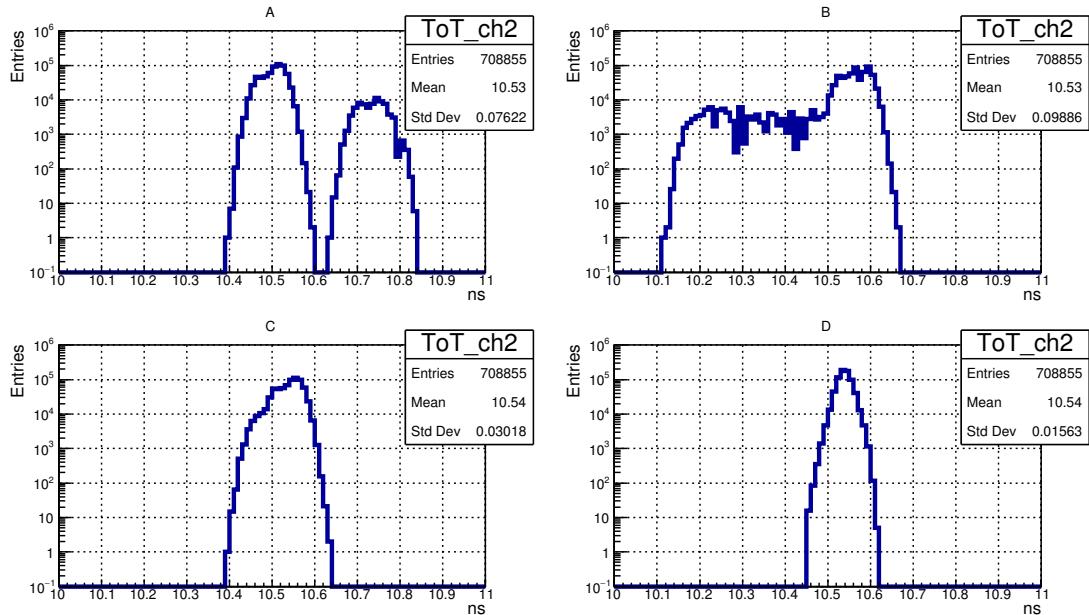


Рис. 4.4: Результаты измерения ширины импульса от генератора в случае: (A) без калибровки точного времени; (B) с применением усреднённой калибровочной функции; (C) с применением индивидуальной линейной калибровочной функции; (D) с применением полноценной калибровочной функции.

Приведённые выше таблицы калибровки были построены по массиву данных, содержащихся в семи файлах. Каждый файл это 2 минуты измерений при частоте генератора 5 кГц, т. е. около 600 тысяч вспышек лазера. Таким образом, всего было 4.2 миллиона вспышек за 14 минут, а один файл составляет приблизительно 15% от полного набора данных. В каждом канале было зарегистрировано от 300 до 400 тысяч временных отметок, которые были использованы для выполнения калибровки. Для иллюстрации стабильности калибровки на рис. 4.5 показана разность функций калибровки, построенных по всему массиву данных и функций, построенных на файлах, составляющих $\approx 15\%$ данных каждый, взятых в начале, середине и конце набора данных. Видно, что отклонения в основном не превышают 10 пс, однако имеются редкие выбросы до 20 пс.

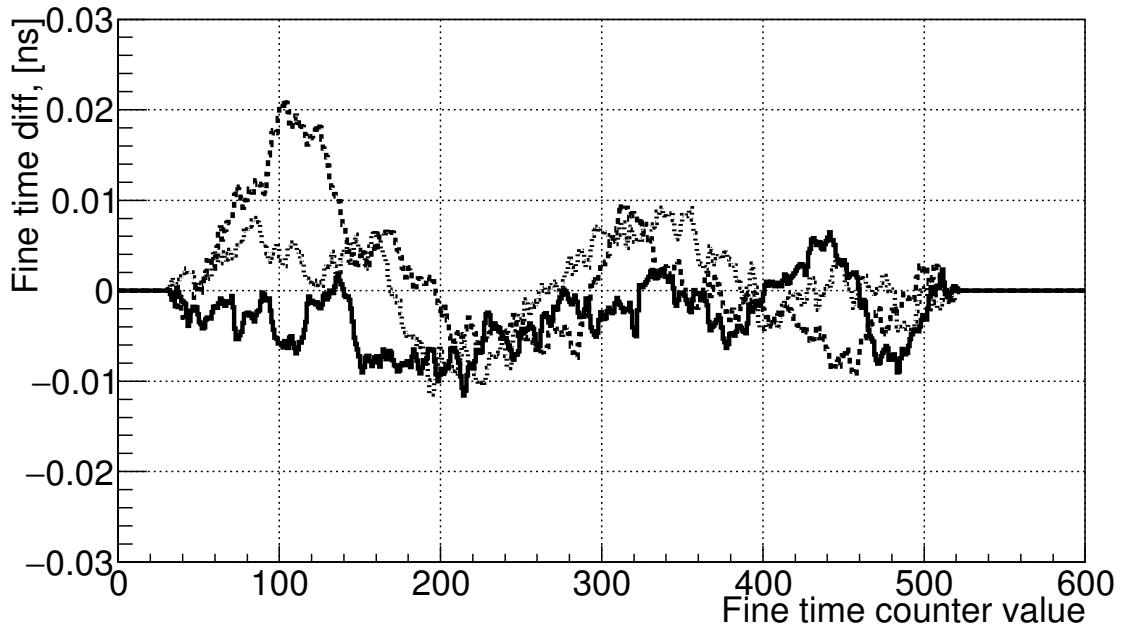


Рис. 4.5: Стабильность калибровок.

4.4 Определение коррекций задержек между каналами

Типичная гистограмма разности временных отметок передних фронтов, соответствующих фотонам из одной вспышки лазера, зарегистрированных в заданной паре каналов, показана на рис. 4.6. Такие гистограммы позволяют определить положение пика и, соответственно, ввести коррекцию задержки. Отметим, что наблюдается дрейф порядка 0.5 нс значений задержек, полученных таким образом, что даёт заметный вклад во временное разрешение системы считывания (см. секцию 4.5).

Наблюдается также аддитивность задержек, т.е. задержка в i -м канале относительно опорного может быть получена с точностью не хуже 400 пс как сумма задержки в j -м канале относительно опорного и задержки в i -м канала относительно j -го. Для некоторых пар каналов вид гистограммы отличается от показанной на рис. 4.6. См., например, рис. 4.7. Подобное распределение можно получить, если один из двух каналов является дефектным в том смысле, что к фронту логического сигнала подмешивается возбужденный или на-введененный колебательный сигнал. Такая гипотеза подтверждается тем фактом, что форма гистограммы зависит от порога дискриминатора на плате PADIWA. При построении аналогичной гистограммы для пары дефектных каналов наблюдается до 5 пиков. Дальнейшее исследование проводилось с исключением дефектных каналов. Доля дефектных каналов составляет около 10% от полного числа каналов. При разработке следующей версии передней электроники для СВМ RICH особое внимание будет уделено электромагнитной чистоте каналов, а гистограммы, подобные обсуждаемым в данном разделе, будут использоваться в качестве диагностического инструмента.

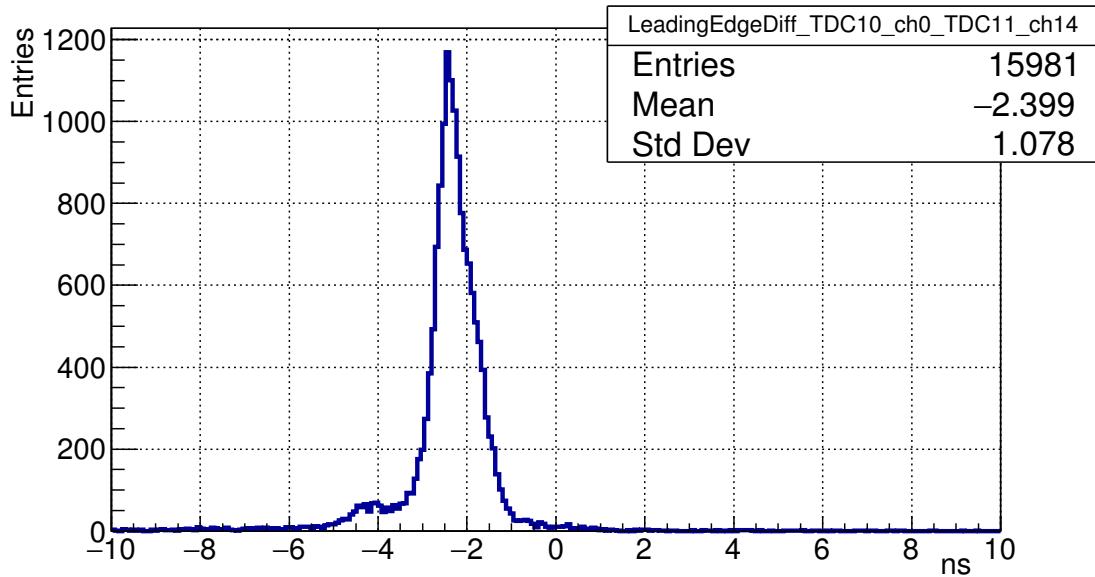


Рис. 4.6: Распределение разности временных отметок передних фронтов, соответствующих фотонам из одной вспышки лазера, зарегистрированных в заданной паре каналов.

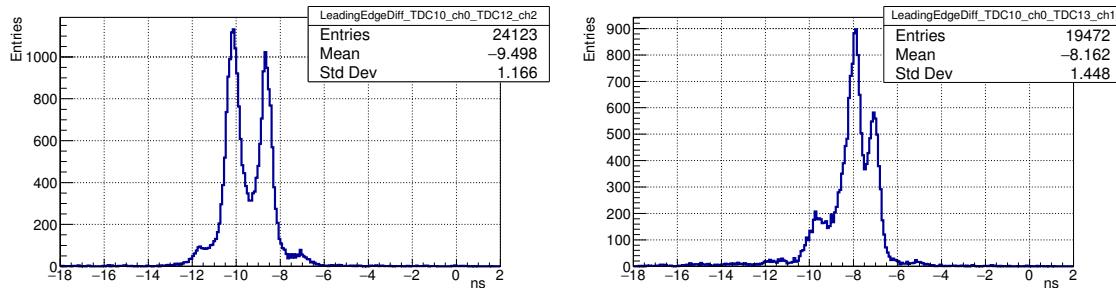


Рис. 4.7: Распределение разности временных отметок передних фронтов, соответствующих фотонам из одной вспышки лазера, зарегистрированных в заданной паре каналов, при условии, что один из каналов — дефектный.

4.5 Временное разрешение

В проведённых пучковых тестах имеют место два типа событий, в которых регистрируются несколько практически одновременно испущенных фотонов. Первый тип — это вспышка лазера, длительность которой ≈ 40 пс, т.е. на порядок меньше разброса времени прохождения сигнала через МА ФЭУ. Второй тип — черенковские кольца. Разброс времени прихода фотонов на МА ФЭУ может достигать 100 пс для колец и 70 пс для вспышек лазера, что определяется в первую очередь наклоном плоскости в которой расположены фотокатоды. Анализ таких событий позволяет охарактеризовать временное разрешение всей системы

считывания, начиная от окна МА ФЭУ и кончая формированием отметок времени. Временное разрешение одного канала определяется разбросом зарегистрированных временных отметок относительно времени прилёта фотона при многократных измерениях. Поскольку точное время прилёта фотона измерить нельзя, нам приходится исследовать разброс разностей временных отметок в паре каналов при регистрации одновременно пришедших фотонов. Временные отметки в каждом из каналов подвержены независимым флуктуациям по одинаковому закону, следовательно, измеренная ширина распределения будет в $\sqrt{2}$ раз больше, чем временное разрешение каждого канала. После применения коррекций задержек и калибровки точного времени в двух каналах, ни один из которых не является дефектным, получается распределение аналогичное показанному на рис. 4.6, отличающееся лишь тем, что положение центра находится в нуле.

Полная ширина на полувысоте (FWHM) этого распределения составляет 750 пс, что соответствует временному разрешению 530 пс. Данное значение превосходит разброс времён прохождения сигнала в МА ФЭУ примерно в 2 раза. Причина расхождения объясняется двумя сравнимыми вкладами: дрейфом задержек в каналах и отсутствием коррекции момента пересечения порога в зависимости от амплитуды сигнала. Для реализации такой коррекции необходимо надёжное измерение времени над порогом, что в нашем случае невозможно, см. секцию 4.7.

Для того чтобы охарактеризовать временное разрешение системы в целом, помимо анализа пар каналов исследовались физически одновременные сигналы на следующих совокупностях каналов: (1) шестнадцать каналов, считываемых одной платой PADIWA, (2) 64 канала, принадлежащих одному МА ФЭУ, (3) 256 каналов, принадлежащих четырём соседним МА ФЭУ. В каждом случае после коррекции задержек и калибровки точного времени, отбирались все хиты, принадлежащие одному событию, и гистограммировались разности временных отметок по всем возможным парам каналов. Результаты для вспышек лазера показаны на рис. 4.8. В таблице 4.1 показано, как эволюционирует среднеквадратичное отклонение и FWHM в зависимости от числа каналов. Отметим, что для событий от лазера среднеквадратичное отклонение меняется слабо, а FWHM возрастает с увеличением числа каналов, одновременно с тем, что распределение последовательно принимает форму, более близкую к распределению Гаусса. Такое поведение можно интерпретировать как размытие индивидуальных особенностей каналов в процессе усреднения. Для хитов, принадлежащих одному черенковскому кольцу (см. рис. 4.9), и FWHM и RMS возрастают с увеличением числа каналов. Вероятно, отличия связаны с тем, что распределения фотонов во времени для черенковского кольца и вспышки лазера отличаются друг от друга.

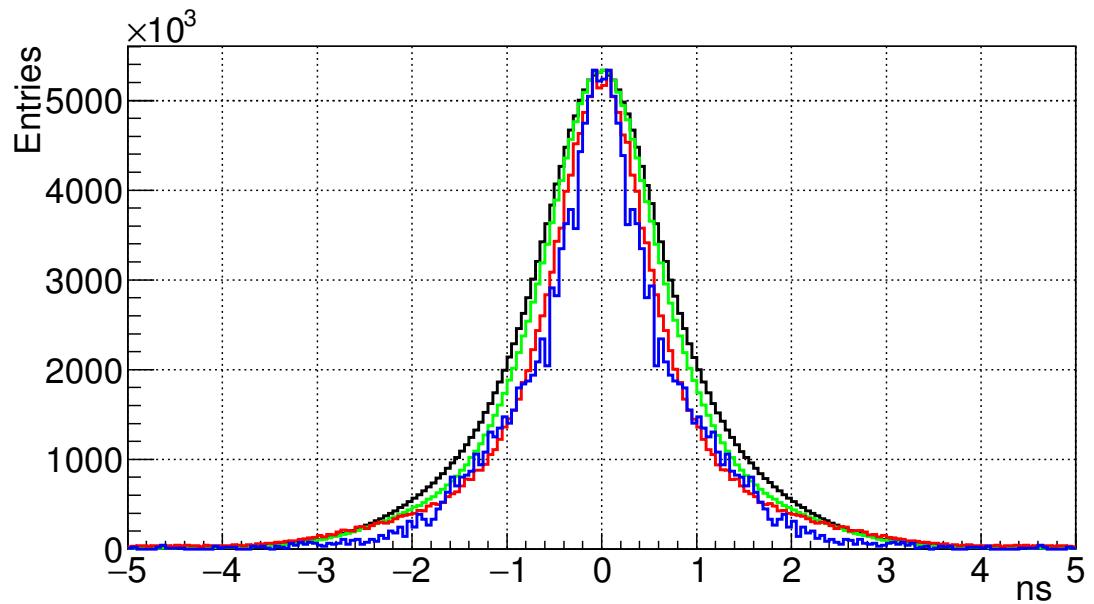


Рис. 4.8: Распределения для четырёх различных наборов каналов для событий от лазера.

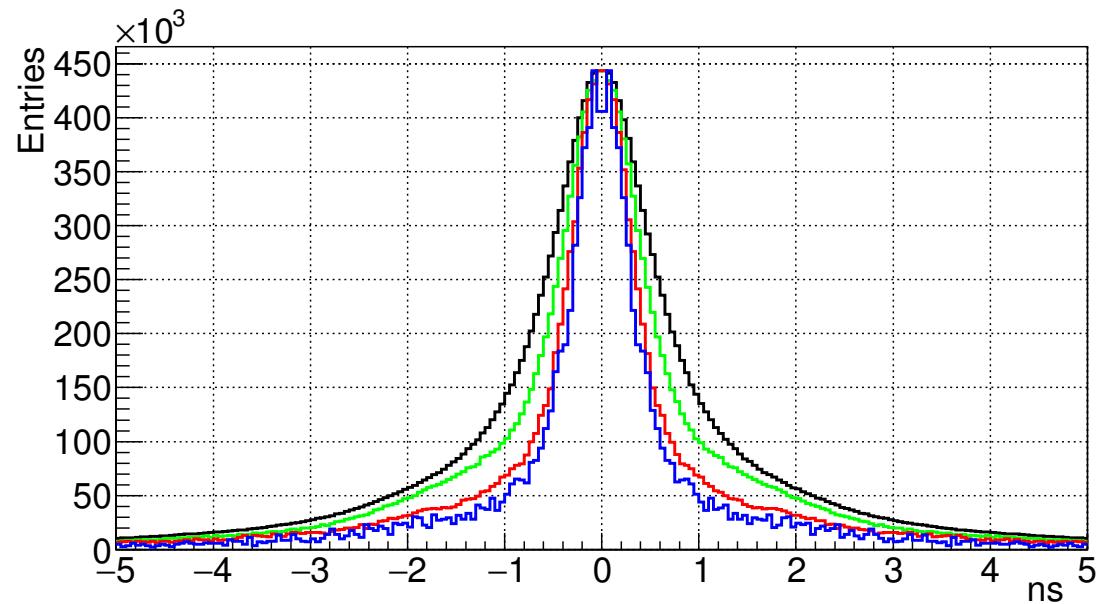


Рис. 4.9: Распределения для четырёх различных наборов каналов для событий от черенковских колец.

Таблица 4.1: FWHM и RMS распределений при различных наборах исследуемых каналов.

Анализируемая область	Пара каналов	Плата PADIWA	Один МА ФЭУ	Четыре МА ФЭУ
Кол-во каналов	2	16	64	256
FWHM, лазер, нс	1.1	1.2	1.5	1.7
FWHM, кольца, нс	0.6	0.8	1.0	1.3
RMS, лазер, нс	0.913	1.093	0.997	1.034
RMS, кольца, нс	1.238	1.379	1.430	1.487

4.6 Исследование профиля высвечивания сместителя спектра

Анализ распределения во времени хитов, принадлежащих одному черенковскому кольцу, позволяет исследовать временные свойства сместителя спектра. Анализу подлежит распределение разностей временных отметок хитов каждого кольца относительно первого по времени хита в данном кольце. В зависимости от длины волны черенковский фотон может с той или иной вероятностью либо поглотиться сместителем спектра и вызвать его свечение, либо пройти сквозь слой сместителя спектра без взаимодействия и попасть фотокатод. В результате, даже при наличии слоя сместителя спектра, часть хитов подчиняется временной зависимости характерной для чистого ФЭУ. Таким образом, для получения кривой высвечивания сместителя спектра необходимо из распределения разностей времен, полученного со сместителем спектра, вычесть должным образом отнормированное в максимуме распределение разностей времён, полученное с чистым ФЭУ.

Нормированные в максимуме кривые высвечивания со сместителем спектра и без него показаны на рис. 4.10, а разность этих распределений — на рис. 4.11. Видно, что за исключением небольшой выпуклости в области 7 нс, связанный с особенностями работы данного семейства МА ФЭУ, кривая выглядит похоже на сумму нескольких экспонент.

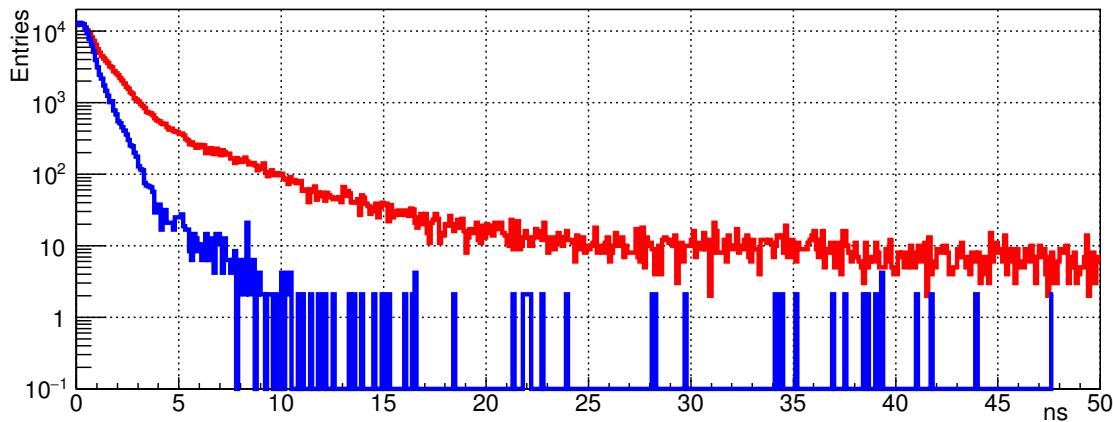


Рис. 4.10: Измеренные распределения, соответствующие кривым высвечивания со сместителем спектра (красный, выше) и без него (синий, ниже).

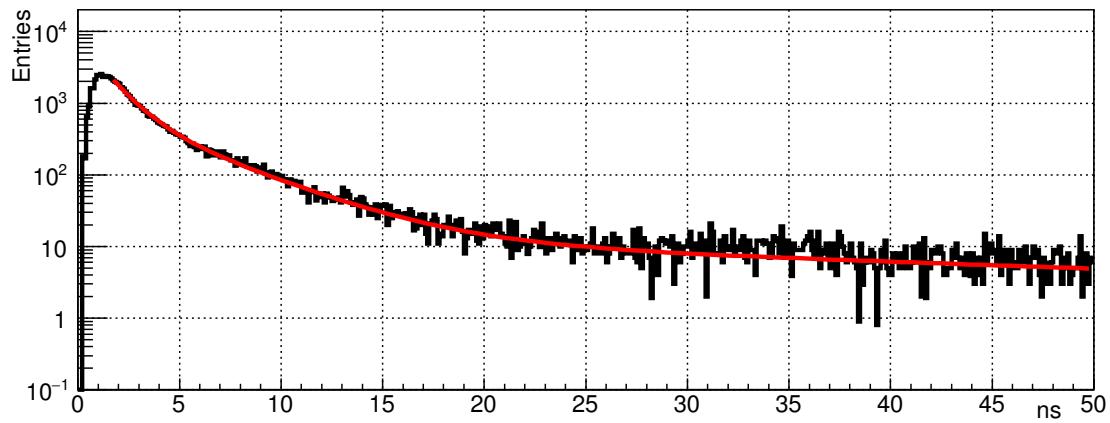


Рис. 4.11: Разница распределений со сместителем спектра и без него и кривая — результат фитирования распределения суммой трёх экспонент.

Указанная выпуклость не позволяет надёжно извлечь характерные времена высвечивания. Интересно, тем не менее, сравнить полученную кривую с результатами флюориметрических исследований. Стеклянная пластина со слоем сместителя спектра, нанесённым точно таким же методом, как и на МА ФЭУ, была исследована с помощью классического метода счёта фотонов при возбуждении светом с длиной волны 280 нм. Были получены ([49]) значения времён высвечивания 1.4 нс, 3.8 нс, и 45 нс и соответствующие относительные интенсивности компонент 1.8996, 1.0000, и 0.8364.

Подгонка кривой с рис. 4.11 суммой трех экспонент с соответствующими временами показывает разумное согласие для времен превышающих 5 нс. Начальный участок лучше подгоняется с характерным временем $\tau_1=1.1$ нс. Сравнение интенсивностей наиболее быстрой компоненты с флюориметрическими измерениями затруднено из-за начального неэкспоненциального участка, а относительный вклад наиболее медленной компоненты в полную интенсивность в нашем случае оказывается в 3.8 раз ниже. Это можно объяснить влиянием способа возбуждения на заселение разных типов центров высвечивания.

В пределе большого числа хитов в кольце использованный нами метод переходит в стандартный метод исследования флюоресценции путем счета единичных фотонов [50]. Однако в нашем случае существует некоторая случайная задержка между моментом попадания чerenковского фотона на поверхность МА ФЭУ и временем прихода первого хита. С целью выявления влияния метода на измеренные времена высвечивания было проведено Монте Карло моделирование.

В модели были заложены разброс времени прохода лавины в МА ФЭУ 300 пс (RMS), три экспоненциальные компоненты с характерными временами 1.4 нс, 3.8 нс, и 45 нс и относительными интенсивностями 2.17, 1.00, 0.22 и средним числом хитов в кольце равным 18. Получившееся распределение времён относительно первого хита в кольце было подогнано трёмя экспонентами со свободными параметрами. Если начать фитирование получившейся зависимости, отступив 4 нс от начала высвечивания, величины постоянных распада экспонент воспроизводятся с точностью лучше 5%, а соответствующие относительные интенсивности несколько искажаются, что естественно, в силу существования начального неэкспоненциального участка кривой. Таким образом, подтверждена корректность применённого метода определения времён высвечивания.

Практическая ценность проведенного исследования состоит в том, что может быть оптимизирована длительность окна, в пределах которого хиты принимаются одновременными и

могут быть приписаны одному событию. Для этого необходимо найти баланс между числом дополнительных хитов, полученных благодаря сместителю спектра и вероятностью наложения сигналов друг на друга или подхвате в кольцо темнового хита. Например, прирост хитов в 19% может быть достигнут при длительности окна 15 нс.

4.6.1 Исследование спектросместителя с помощью флюориметра

Были проведены независимые флюорометрические измерения пара-терфениловой пленки, нанесённой по той же технологии (dip-coating), что применяется для напыления сместителя спектра на поверхность МА ФЭУ в СВМ RICH. Измеренный временной профиль приведён на рисунке 4.12, а результаты фитирования — в таблице 4.2.

Таблица 4.2: Результаты фитирования флюорометрических измерений сместителя спектра.

τ , нс	амплитуда	амплитуда, нормированная по 2-й компоненте
1.4	1387	5.15613
3.8	269	1.00000
45.0	19	0.07063

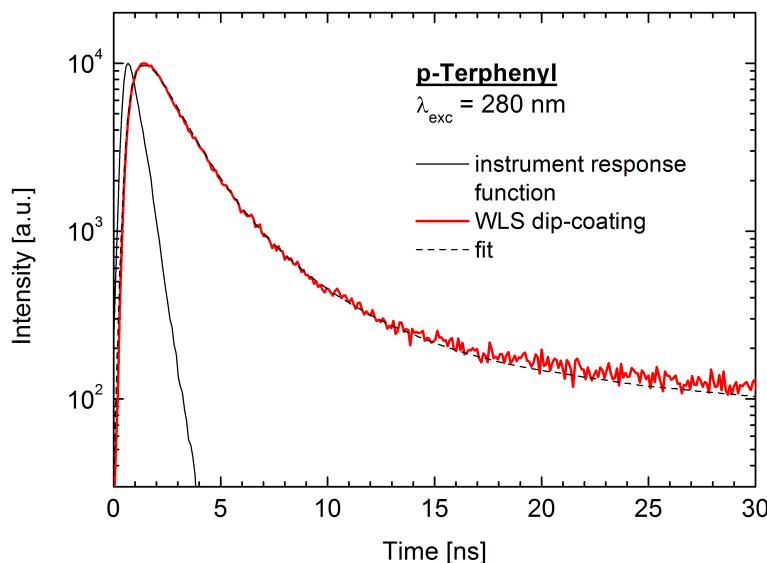


Рис. 4.12:

Данный фит неплохо воспроизводится функцией

$$f(t) = 10 \cdot (A_1 \cdot e^{-(t-1)/\tau_1} + A_2 \cdot e^{-(t-1)/\tau_2} + A_3 \cdot e^{-(t-1)/\tau_3})$$

со значениями амплитуд и времён, приведёнными в таблице 4.2. Здесь единица в скобке возле переменной t означает сдвиг графика по горизонтали на 1 нс и обусловлена разрешением измерительного прибора.

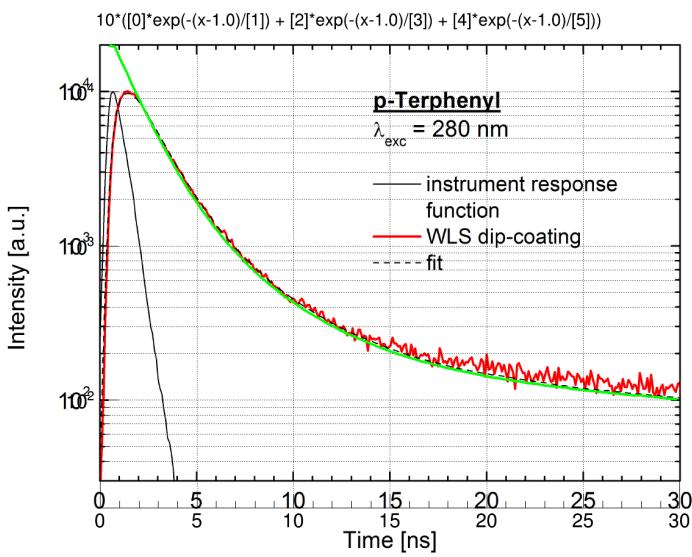


Рис. 4.13:

4.6.2 Прямые измерения временного профиля спектрометриста

Можно фитировать распределение WLS_on или WLS_off 4-мя компонентами, а можно фитировать их разницу WLS_diff трёмя компонентами. Можно фитировать функцией с зафиксированными временами, чтобы определить амплитуды, а можно фитировать функцией, где параметрами являются и времена и амплитуды.

Прямые фотоны

В результате анализа экспериментальных данных были получены две гистограммы — с и без сместителя спектра. Первый этап — фитирование профиля без сместителя спектра. В результате фитирования одной экспонентой в различных диапазонах и с различными начальными условиями было получено значение временной постоянной 0.65 нс.

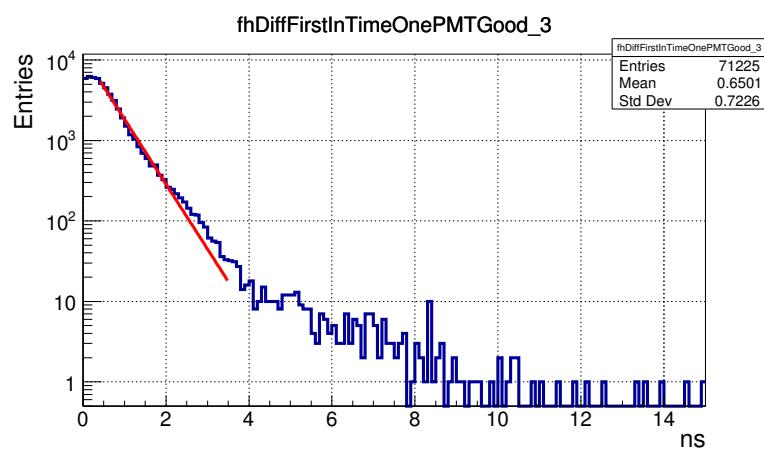


Рис. 4.14:

Фильтрование WLS_diff

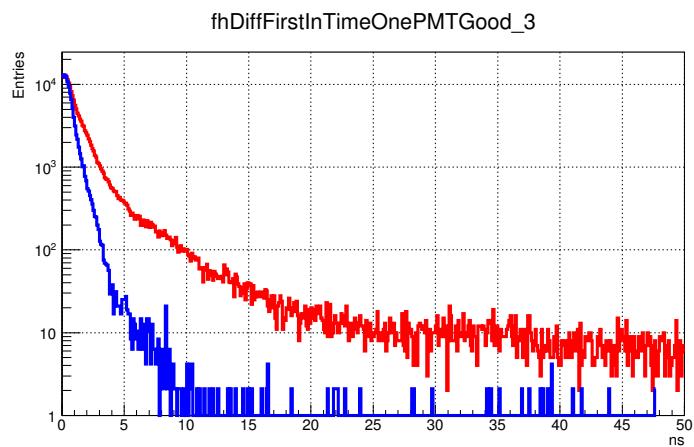


Рис. 4.15:

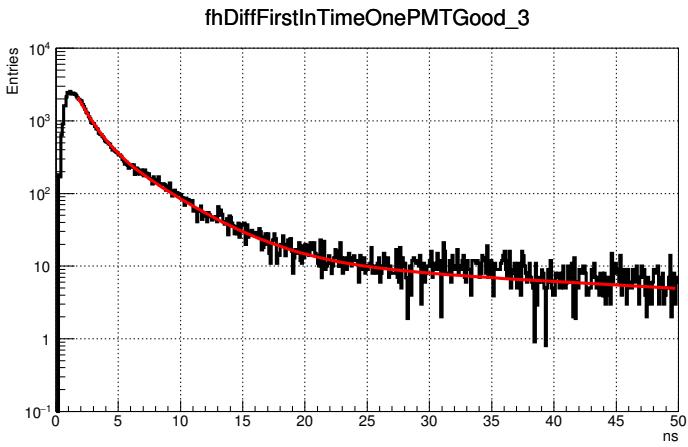


Рис. 4.16:

4.7 Время над порогом

Время над порогом (ToT — time over threshold) — это параметр найденного хита, содержащий в себе, при нормальной работе, информацию об амплитуде зарегистрированного сигнала. В системе считывания и сбора данных CBM RICH ToT может быть использовано для улучшения временного разрешения путём коррекции времени пересечения порога с учетом амплитуды (walk correction), а также для повышения качества отделения однофотоэлектронного сигнала от шума. На рис. 4.18 показано типичное распределение ToT, измеренное с помощью лазера в лабораторных условиях. Вопреки ожиданиям, это распределение имеет несколько пиков. Такая структура, согласно [51], может быть объяснена наличием периодической наводки как на входе дискриминатора, так и между выходом дискриминатора и входом ВЦП. На рис. 4.17 показан экран цифрового осциллографа в режиме накопления сигналов, полученных путем подключения активного зонда к выходу PADIWA. Видно, что сгущение сигналов соответствует наблюдаемым пикам в распределении ToT; имеет место проблема недостаточности амплитуды одноэлектронного сигнала для устойчивой генерации логической единицы; имеется периодическая наводка на выходе дискриминатора, но ее недостаточно для объяснения наблюданной картины; преобладание определенных длительностей логических сигналов позволяет предположить наличие периодической структуры во входном сигнале. Все это говорит о необходимости подстройки аналоговой части для формирования на входе PADIWA более чистого сигнала большей амплитуды и о защите соединения между дискриминатором и ВЦП от наводок. Подобные изменения будут, с учетом результатов данной работы, реализованы в следующем прототипе платы передней электроники, называемом DIRICH [29].

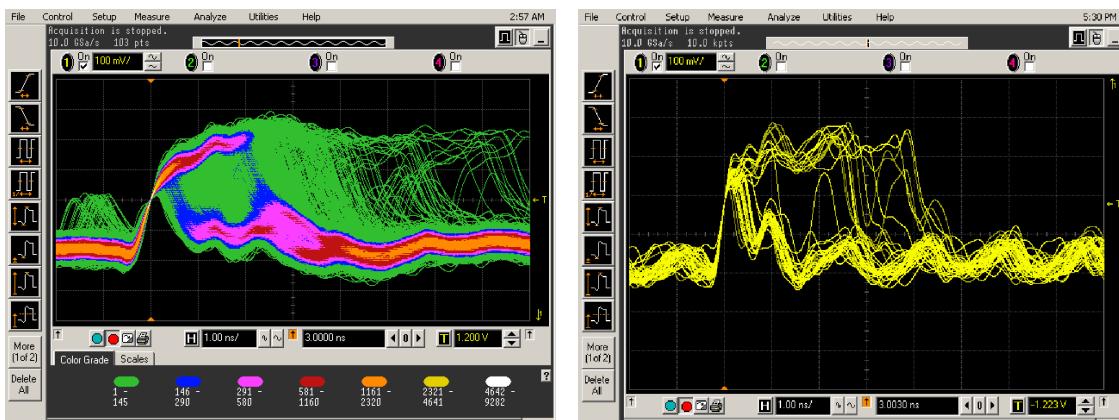


Рис. 4.17: Экран осциллографа, показывающий выходные сигналы PADIWA, регистрируемые по переднему фронту.

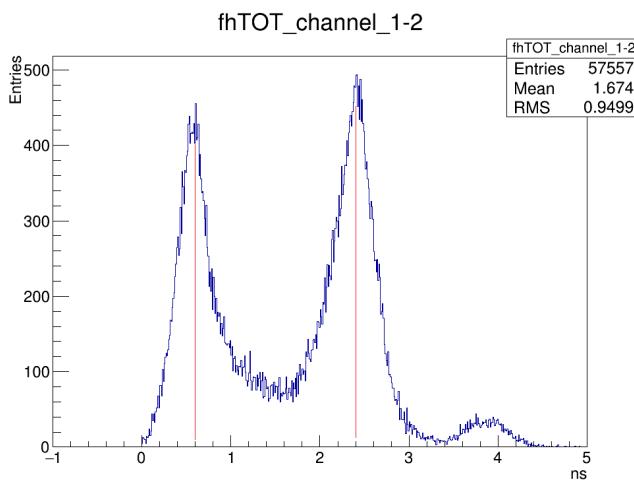


Рис. 4.18: Типичное распределение ТоT.

Отметим, что указанные проблемы не являются критичными в случае CBM RICH, и продемонстрированные в данной работе параметры достаточны для уверенного поиска колец. Тем не менее, улучшение разделения сигналов и шумов и повышение эффективности регистрации поможет создать необходимый запас надежности для долговременной работы детектора в условиях постепенной деградации оптических свойств радиатора, зеркал и фотодетекторов.

4.8 Сравнение одноэлектронных спектров при временном и амплитудном считывании

Как отмечено в секции 3.1, у МА ФЭУ H12700 имеются особенности, которые могут оказать влияние на эффективность регистрации единичных фотоэлектронов и вероятность возник-

новения ложных хитов. Для прояснения этих особенностей были выполнены измерения амплитудных распределений с помощью многоканальной платы на основе микросхемы p-XYTER, см. описание лабораторного стенда в секции 3.4.2. Далее, результаты амплитудных измерений были сопоставлены с данными, полученными с помощью платы PADIWA.

Амплитудные измерения с низким порогом продемонстрировали наличие заметного пика в малых амплитудах в спектре событий, скоррелированных с источником света. Также были выполнены специальные измерения с маской, открывающей только два разнесенных друг от друга на 2.5 см. пикселя. Эти измерения позволили установить, что событие с малой амплитудой в одном из каналов имеет место тогда, когда в другом канале, находящемся в том же ряду динодной системы, был зарегистрирован фотоэлектрон с достаточно большой амплитудой. Таким образом, для каналов с низкими шумами амплитудный спектр одноэлектронных сигналов выглядит как на рис. 4.19.

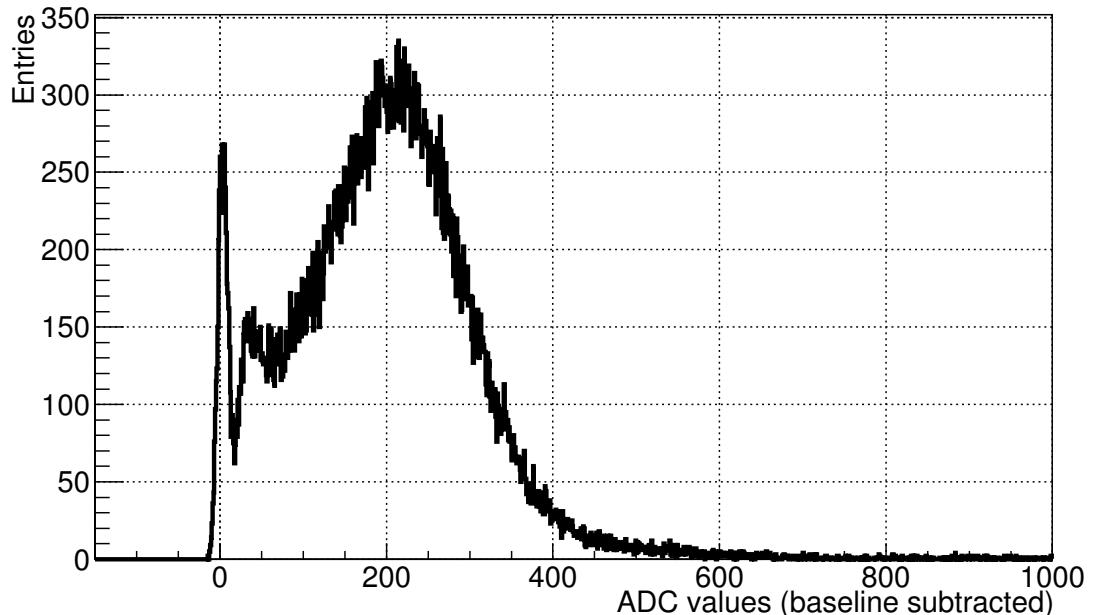


Рис. 4.19: Пример измеренного одноэлектронного спектра, имеющий особую форму, характерную для МА ФЭУ Н12700.

Пик близи нуля соответствует наводке, возникающей в каналах, расположенных в одном ряду с тем, где зарегистрирован одноэлектронный сигнал. Двугорбое распределение справа соответствует настоящим одноэлектронным сигналам. Причем левый пик связан с описанными в секции 3.1 событиями, когда электронная лавина или её часть отклоняется от оптимального пути от динода к диноду. Отметим, что в большинстве каналов уровень шумов оказывается слишком высоким для отделения низкоамплитудного пика, связанного с наводкой, от одноэлектронного сигнала. Таким образом, попытка получить максимальную эффективность регистрации за счет снижения порога приводит к возрастанию паразитных хитов, локализованных не в тех пикселях, где родился фотоэлектрон. Для снижения числа паразитных хитов мы ставили порог регистрации в ложбине между низко- и высоко-амплитудными частями одноэлектронного спектра. Поскольку формы одноэлектронных спектров во всех каналах подобны, анализ формы спектра на рис. 4.19 позволяет

заключить, что выбранный нами порог приводит к потере 12 % одноэлектронных импульсов.

Одно из отличий канала считывания в плате PADIWA — это значительно более быстрая, чем в n-XYTER аналоговая часть. Если в n-XYTER осуществляется формирование со временем интегрирования 190 нс, то в PADIWA происходит лишь подавление частот выше 100 МГц, что соответствует характерному времени нарастания сигнала несколько наносекунд. Такое отличие приводит к возрастанию роли быстрых шумов и наводок при регистрации сигналов с помощью PADIWA.

Информация о форме одноэлектронного спектра при считывании с помощью канала на основе плат PADIWA и TRB v3 может быть получена в виде зависимости от порога регистрации скорости счёта в событиях, построенных вблизи триггера светового импульса. Такие данные могут быть получены из анализа потока данных, набранных при различных значениях порога. Использование счетчика зарегистрированных фронтов, реализованного непосредственно в ВЦП и упомянутого в секции 3.2.1, позволяет получить аналогичную зависимость без отбора вокруг триггера, но позволяет достичь максимальных частот, достаточных для локализации базовой линии. На рис. 4.20 показана зависимость частоты триггеров от порога регистрации. Плечо слева соответствует одноэлектронному спектру, более подробно исследованному ниже, а быстровозрастающие границы вокруг вертикальной штриховой линии ограничивают локализацию базовой линии. Точность локализации базовой линии мы оцениваем как ± 200 отсчетов по шкале, использованной на рис. 4.20 и рис. 4.21B,D.

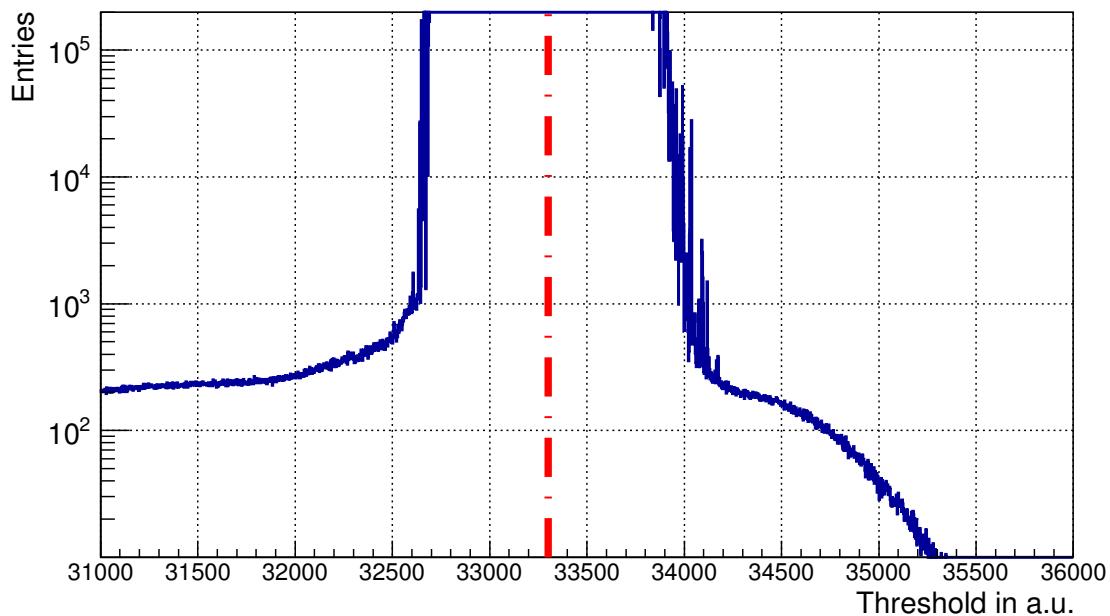


Рис. 4.20: Скан по порогам дискриминатора в диапазоне, включающем базовую линию, изображённую штрихпунктирной линией.

Установлено, что результаты измерения частоты отсчетов, полученные с помощью счетчика и из анализа потока данных, совпадают между собой при условии, что система сбора и передачи данных справляется с передачей потока сообщений с временными отметками.

Интересно сравнить зависимость скорости счёта от порога при использовании двух систем считывания и одинаковых условиях засветки. Результаты такого сравнения для одного из типичных каналов показаны на рис. 4.21. В случае n-XYTER в таком сравнении может быть использован интеграл одноэлектронного спектра, показанный на рис. 4.21С. Соответственно, производная указанной зависимости может быть сопоставлена с одноэлектронным спектром, показанным на рис. 4.21А. Сплошная линия на рис. 4.21В получена дифференцированием кривой, показанной красным цветом на рис. 4.21Д и полученной подгонкой измеренной зависимости полиномом 7-й степени. Отметим, что мы оцениваем равенство световых потоков как $\pm 5\%$. Видно, что скорости счёта в области ложбины и максимума одноэлектронного спектра приблизительно совпадают. Амплитуды, соответствующие максимуму и ложбине соответственно, относятся как 2.6 в обоих случаях. При этом, в случае PADIWA наблюдается, с одной стороны более явно выраженная ложбина, а с другой — избыток счёта в малых амплитудах, что предполагает больший относительный вклад наводок и, следовательно, невозможность отделения от них низкоамплитудной части одноэлектронного спектра и нецелесообразность повышения эффективности за счёт установления порога ниже ложбины.

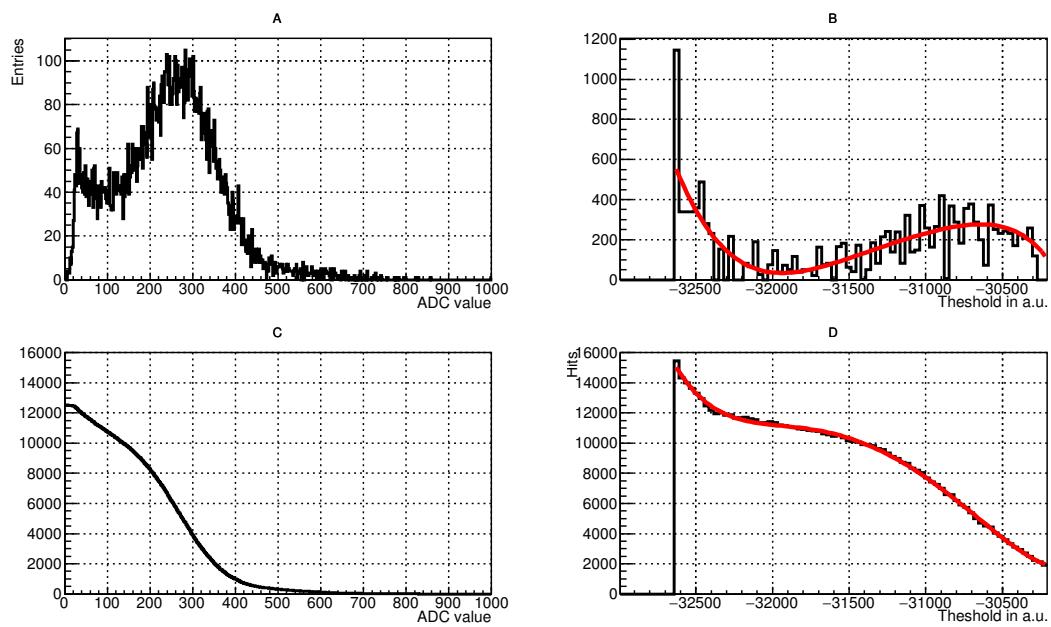


Рис. 4.21: Сравнение (A) одноэлектронного спектра, измеренного напрямую с помощью системы считывания на базе n-XYTER, и (B) производной скана по порогам, полученного с помощью системы считывания на базе PADIWA и TRB v3; сравнение (C) интеграла одноэлектронного спектра и (D) зависимости скорости счёта от порога дискриминатора.

Глава 5

Применение разработанных методов к оптимизации СВМ RICH

Заключение

Основные результаты диссертационной работы

Научная новизна результатов, полученных автором

1. Разработана схема отображения иерархии геометрии, используемой в моделировании транспорта частиц методом Монте Карло (МК), на дерево построений САПР CATIA v5.
2. В среде CATIA создан набор шаблонов для примитивов и сущностей конструктивной твердотельной геометрии, принятой в системах МК моделирования детекторов.
3. Создан набор инструментов для полуавтоматического построения детальной МК геометрии на основе САПР модели и быстрого обмена геометрией между САПР CATIA v5 и пакетами МК моделирования GEANT и ROOT.
4. Выполнены беспрецедентно точные параметризованные описания ряда приборов и детекторов в средах МК моделирования.
5. На основе детального параметризованного описания геометрии СВМ RICH выполнена оптимизация компоновки детектора.
6. Собран прототип системы считывания и сбора данных детектора СВМ RICH.
7. Разработано программное обеспечение для приема, упаковки и передачи бестриггерного потока данных с прототипа системы считывания и сбора данных с частотой до 20 МГц.
8. Разработано программное обеспечение для калибровки точного времени и относительных задержек каналов в потоке данных с детектора СВМ RICH.
9. Разработано программное обеспечение для построения событий из бестриггерного потока данных с детектора СВМ RICH в среде CbmRoot.
10. Проведены пучковые тесты прототипа системы считывания и сбора данных в составе полнофункционального прототипа детектора СВМ RICH и дополнительные тесты на лабораторном стенде.
11. Проведено комплексное исследование свойств канала считывания и сбора данных для СВМ RICH, реализованного на основе многоанодного ФЭУ H12700 с системой динодов “metal channel”, специально разработанных передней электроники типа предусилитель-дискриминатор и высокочастотного ВЦП с последующим прямым вводом данных в единую среду моделирования, сбора и анализа данных CbmRoot.

12. Исследованы временные свойства нанесенного на окно МА ФЭУ сместителя спектра при возбуждении черенковскими фотонами.
13. Изучены возможности работы канала считывания при пониженных порогах.
14. Проведен сравнительный анализ особенностей считывания многоанодного ФЭУ временным и аналоговым трактами.
15. Исследованы характеристики детектора CBM RICH с учетом неидеальности геометрии и шумов электроники.

Представление основных положений и результатов

Основные положения и результаты работы докладывались и обсуждались на научных семинарах ЛИТ ОИЯИ и на различных международных конференциях и совещаниях, в том числе:

1. Семинар НОВФ ЛИТ ОИЯИ, Дубна, Россия, 22.12.2016
Устный доклад “Development of the readout and DAQ system for CBM RICH and EXPERT. ‘CATIA-GDML geometry builder’ and Monte-Carlo geometry of CBM RICH.”
2. Семинар “Contribution of the young Russian scientists into the project FAIR”, ИТЭФ, Москва, Россия, 14-15.12.2016
Устный доклад “Detailed study of the stability and uniformity of the CBM RICH readout and DAQ prototype characteristics. Development and application of the Monte Carlo geometry package”
3. Международная конференция “The 9th International Workshop on Ring Imaging Cherenkov Detectors (RICH 2016)”, Блед, Словения, 05-09.09.2016
Представлен постер “Development of the CBM RICH readout electronics and DAQ”
4. Международная конференция “The 20th IEEE-NPSS Real Time Conference (IEEE-NPSS RT2016)”, Падуя, Италия, 05-10.06.2016
Представлен постер “Development of the CBM RICH readout and DAQ”
5. Международная конференция “The XX International Scientific Conference of Young Scientists and Specialists (AYSS-2016)”, ОИЯИ, Дубна, Россия, 14-18.03.2016
Устный доклад “Development and characterization of CBM RICH readout and DAQ”
6. Семинар “Contribution of the young Russian scientists into the project FAIR”, ИТЭФ, Москва, Россия, 15-17.12.2015
Устный доклад “Development of ‘CATIA-GDML geometry builder’ and CBM RICH software”
7. Международное совещание “26th CBM Collaboration Meeting”, Прага, Чехия, 14-18.09.2015
Устный доклад “PADIWA test measurements, beamtime analysis (TOT, WLS time resolution)”
8. Международное совещание “25th CBM Collaboration Meeting”, ГСИ, Дармштадт, Германия, 20-24.04.2015
Устный доклад “Beamtime analysis: FLIB readout, TOT, timing”
9. Семинар “Contribution of the young Russian scientists into the project FAIR”, ИТЭФ, Москва, Россия, 12-13.11.2013
Устный доклад “Modernization of simulation and data acquisition packages of CBM experiment”

10. Международная конференция “20th International Conference on Computing in High Energy and Nuclear Physics (CHEP)”, Амстердам, Нидерланды, 14-18.10.2013
Представлен постер “Development and application of CATIA-GDML geometry builder”

Публикации

Список литературы

- [1] *H. H. Gutbrod* // FAIR Baseline Technical Report, ISBN: 3-9811298-0-6, 2006.
- [2] *B. Friman, C. Höhne, J. Knoll, S. Leupold, J. Randrup, R. Rapp and P. Senger* // The CBM physics book: Compressed baryonic matter in laboratory experiments, Lect. Notes Phys. **814** (2011) pp. 980.
- [3] *P. Senger and V. Friese* // The CBM Collaboration: Nuclear Matter Physics at SIS-100, GSI, Darmstadt (2012) 18 p.
- [4] Compressed Baryonic Matter Experiment. Technical Status Report. GSI, Darmstadt (2005) 406 p.
- [5] *V. Friese and C. Sturm* // CBM Progress Report 2014, ISBN: 978-3-9815227-2-3, 2015.
- [6] *A. Malakhov and A. Shabunov* // Technical Design Report for the CBM Superconducting Dipole Magnet, GSI, Darmstadt (2013) 80 p.
- [7] *M. Koziel* // MVD Status: Integration, 25th CBM Collaboration Meeting, Darmstadt, 20-24 April 2015. <https://indico.gsi.de/getFile.py/access?contribId=17&sessionId=9&resId=0&materialId=slides&confId=2960>
- [8] *J. Heuser et al.* // Technical Design Report for the CBM Silicon Tracking System (STS), GSI, Darmstadt (2013) 167 p.
- [9] *C. Höhne et al.* // Technical Design Report for the CBM Ring Imagine Cherenkov (RICH), GSI, Darmstadt (2013) 201 p.
- [10] *S. Chattopadhyay et al.* // Technical Design Report for the CBM Muon Chambers (MuCh), GSI, Darmstadt (2014) 192 p.
- [11] *S. Biswas et al.* // Development of a GEM based detector for the CBM Muon Chamber (MUCH), 2013 JINST 8 C12002.
- [12] *M. Petris et al.* // TRD detector development for the CBM experiment, NIM A, Volume 732, 21 December 2013, Pages 375–379.
- [13] *N. Herrmann et al.* // Technical Design Report for the CBM Time-of-Flight System (TOF), GSI, Darmstadt (2014) 182 p.
- [14] *I. Korolko* // CBM Calorimeter for SIS100 (TDR status), 25th CBM Collaboration Meeting, Darmstadt, 20-24 April 2015.
<https://indico.gsi.de/getFile.py/access?contribId=138&sessionId=29&resId=1&materialId=slides&confId=2960>

- [15] *F. Guber et al.* // Technical Design Report for the CBM Projectile spectator detector (PSD), GSI, Darmstadt (2014) 78 p.
- [16] Hamamatsu H12700 manual, <https://www.hamamatsu.com/resources/pdf/etd/H12700 TPMH1348E.pdf>
- [17] *M. Calvi et al.* // Characterization of the Hamamatsu H12700A-03 and R12699-03 multi-anode photomultiplier tubes, 2015 JINST 10 P09021.
- [18] *M. Calvi et al.* // Characterization of the Hamamatsu H12700A-03 multi-anode photomultiplier tube for the LHCb RICH upgrade, LHCb-INT-2015-006.
- [19] https://www.hamamatsu.com/resources/pdf/etd/H8500_H10966 TPMH1327E.pdf
- [20] T. Mahmoud RICH2016 proceedings.
- [21] *C. Pauly et al.* // The CBM RICH project, NIM A 2016, doi: 10.1016/j.nima.2016.05.102.
- [22] *S. Reinecke et al.* // The CBM-RICH detector, JINST 11 (2016) no.05, C05016.
- [23] J. Kopfer PhD thesis, Bergische Universität Wuppertal.
- [24] Official TRB project web site, <http://trb.gsi.de/>
- [25] *C. Ugur, S. Linev, J. Michel, T. Schweitzer, and M. Traxler* // A novel approach for pulse width measurements with a high precision (8 ps RMS) TDC in an FPGA, 2016 JINST 11 C01046.
- [26] TRB v3 documentation, <http://jpsc29.x-matter.uni-frankfurt.de/docu/trb3docu.pdf>
- [27] *W. M. Zabłotny and G. Kasprowicz* // Data processing boards design for CBM experiment, Proc. SPIE 9290, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2014, 929023 (November 25, 2014); doi:10.1117/12.2073377;
- [28] *J de Cuveland et al.* // A First-level Event Selector for the CBM Experiment at FAIR, 2011 J. Phys.: Conf. Ser. 331 022006.
- [29] *J. Michel, M. Faul, J. Friese, C. Höhne, K.-H. Kampert, V. Patel, C. Pauly, D. Pfeifer, P. Skott, M. Traxler, and C. Ugur* // Electronics for the RICH detectors of the HADES and CBM experiments, 2017 JINST 12 C01072.
- [30] *J. Adamczewski-Musch et al.* // Data acquisition and online monitoring software for CBM test beams, 2012 J. Phys.: Conf. Ser. 396 012001.
- [31] *C. Bergmann et al.* // Common CBM beam test of the RICH, TRD and TOF subsystems at the CERN PS T9 beam line in 2014, CBM Progress Report 2014, p.9.
- [32] *C. Bergmann et al.* // Test of Münster CBM-TRD real-size detector and radiator prototypes at the CERN PS/T9 beam line, CBM Progress Report 2014, p.78.
- [33] *M. Petris, D. Batros, G. Caragheorghropol et al.* // Prototype with the basic architecture for the CBM-TOF inner wall tested in close to real conditions, 2016 J. Phys.: Conf. Ser. 724 012037.

- [34] Information about the T9 beam line and experimental facilities, http://home.web.cern.ch/sites/home.web.cern.ch/files/file/spotlight_students/information_about_the_t9_beam_line_and_experimental_facilities.pdf
- [35] *L.M. Kotchenda, P.A. Kravtsov* // CBM RICH PROTOTYPE GAS SYSTEM.
- [36] *J. Adamczewski-Musch et al.* // Determination of tolerances of mirror displacement and radiator gas impurity for the CBM RICH detector, doi: 10.1016/j.nima.2014.04.074
- [37] *J. Bendarouach, C. Höhne, and T. Mahmoud* // Mirror misalignment control system and prototype setup, CBM Progress Report 2014, p.56.
- [38] Roithner UVTOP240 datasheet, http://www.roithner-laser.com/datasheets/led_deeppuv/uvttop240.pdf
- [39] *M. Dürr, J. Kopfer et al.* // Influence of wavelength-shifting films on multianode PMTs with UV-extended windows, NIM A, Volume 783, 21 May 2015, Pages 43–50.
- [40] Alphalas Picopower-LD series datasheet, http://www.alphalas.com/images/stories/products/lasers/Picosecond_Pulse_Diode_Lasers_with_Driver_PICOPOWER-LD_ALPHALAS.pdf
- [41] <http://www.rz.uni-frankfurt.de/39888789/syscore>
- [42] <http://cbmroot.gsi.de/>
- [43] FLESnet development repository, <https://github.com/cbm-fles/flesnet>
- [44] *J. Adamczewski-Musch, S. Linev, E. Ovcharenko, and C. Ugur* // HADES trbnet data formats for DABC and Go4, PHN-SIS18-ACC-41, GSI SCIENTIFIC REPORT 2012, p.297.
- [45] *R. Szplet, J. Kalisz, and R. Pelka* // Nonlinearity correction of the integrated time-to-digital converter with direct coding, IEEE Transactions on Instrumentation and Measurement, 46:449–453, April 1997.
- [46] *С.А. Лебедев, Г.А. Осоков* // Быстрые алгоритмы распознавания колец и идентификации электронов в детекторе RICH эксперимента CBM, Письма в ЭЧАЯ. 2009. Т.6, №2(151). С.260-284.
- [47] *S. Lebedev, C. Höhne, I. Kisiel, G. Ososkov* // Fast Parallel Ring Recognition Algorithm in the RICH Detector of the CBM Experiment at FAIR, ACAT2010 proceedings.
- [48] *E. Ovcharenko, S. Belogurov et al.* // Tests of the CBM RICH readout and DAQ prototype, PEPAN letters.
- [49] M. Dürr, private communication.
- [50] *D. V. O'Connor, D. Phillips* // Time Correlated Single Photon Counting, Academic Press, London 1984.
- [51] *F. Gonnella, V. Kozhuharov, M. Raggi* // Time over threshold in the presence of noise, NIM A, Volume 791, p. 16-21.
- [52] Голованов Н.Н. Геометрическое моделирование 2002 Физматлит ISBN 5-94052-048-0