

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ».

Физический факультет

Кафедра автоматизации физико-технических исследований

КУРСОВАЯ РАБОТА ПО СПЕЦИАЛИЗАЦИИ
**Разработка системы автоматизированного тестирования электроники
электромагнитного калориметра детектора Belle II**

Выполнил: студент гр. № 17345

Ильин Иван Евгеньевич

Работа выполнена в НГУ ИАиЭ

Институт Ядерной Физики СО РАН

Научный руководитель:

Ремнев Михаил Анатольевич

«___» мая 2020 г.

Оценка _____

Рекомендации: _____

Разработка системы автоматизированного тестирования электроники электромагнитного калориметра детектора Belle II

Ильин Иван Евгеньевич

Физический факультет. Квантовый практикум. Курсовая работа.

Группа №17345, 6 семестр, 2020 год.

Научный руководитель:

Ремнев Михаил Анатольевич

Аннотация

Данная работа посвящена разработке автоматизированных тестов электроники электромагнитного калориметра детектора Belle II. Основными проблемами при работе с электромагнитным калориметром являются шумы наложения и ложные кластеры. Это приводит к электромагнитным ливням, которые регистрируются калориметром. Так-как в Belle II, по сравнению с Belle, ожидается существенно более высокий уровень фона, то для стабильной и правильной работы электромагнитного калориметра была разработана новая считывающая электроника с более сложной логикой обработки данных. Доступ к электронике осуществляется с помощью специальной библиотеки ECL LIB, разработанной в ИЯФ СО РАН. Поскольку электроника имеет более сложную логику и, кроме того, активно разрабатываются и выходят новые версии прошивки, крайне важно иметь автоматизированные тесты, позволяющие эффективно обнаруживать проблемы с электроникой. Для удаленной работы с сервером ИЯФ использовалось подключение по SSH. Управление удаленным компьютером осуществлялось с помощью программы TightVNC. Нами были выявлены наиболее важные для тестирования параметры, такие как проверка корректности записи и чтения данных из регистров электроники, запись и чтение аттенюаторных коэффициентов, а также проверка корректности загрузки прошивки в ShaperDSP. В результате работы удалось разработать необходимые тесты оборудования. С помощью утилиты Gcov мы выяснили, что доля покрытия тестами библиотеки ECL LIB составляет 63%.

Ключевые слова: электромагнитный калориметр, Belle II, тестирование электроники.

Оглавление

1. Введение	4
2. Постановка задачи.....	5
2.1 Библиотека ECL LIB	5
2.2 Наиболее важные для тестирования компоненты	6
2.3 Критерии тестирования	6
3. Описание эксперимента.....	7
3.1 Подключение по удаленному серверу SSH	7
3.2 Компиляция и исполнение кода	9
3.3 Написание тестов	9
5. Результаты тестирования	11
6. Заключение.....	12
4 Список литературы	13

1. Введение

Belle II – продолжение эксперимента Belle, в котором планируется увеличить светимость до $8 \cdot 10^{35} \text{ см}^{-2}\text{с}^{-1}$ за счет схемы нано пучков. Проектная светимость SuperKEKB в 40 раз больше светимости коллайдера KEKB. Из-за этого уровень фона возрастет в 50 раз.

Данная работа фокусируется на электромагнитном калориметре, очень важной подсистемой детектора частиц. Калориметр нужен для разделения электронов и адронов, формирования сигнала триггера и точной регистрации фотонов. Результат продуктов распада В-мезонов на треть состоит из нейтральных частиц, таких как ρ^0 -мезон. Поэтому электромагнитный калориметр Belle II должен обладать высоким энергетическим разрешением в широком диапазоне энергетического спектра от 20 МэВ до 4 ГэВ. Основными проблемами при работе с электромагнитным калориметром являются шумы наложения и ложные кластеры, вызванные выходом пучка с равновесной орбиты и вследствие чего, попадание в конструкции ускорителя. Это приводит к электромагнитным ливням, которые регистрируются калориметром. Так как в Belle II, по сравнению с Belle, ожидается существенно более высокий уровень фона, то для стабильной и правильной работы электромагнитного калориметра была разработана новая

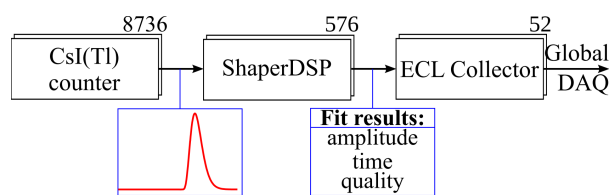


Рис. 1. Упрощенная схема работы электроники калориметра Belle II.

Поступающий из кристалла CsI(Tl), сигнал формируется с помощью модуля Shaper DSP, а затем попадает в ECL коллектор для дальнейшего хранения и обработки.

считывающая электроника с более сложной логикой обработки (Рис. 1). Модернизация электроники в Belle II заключается в уменьшении времени формирования сигнала для устранения шумов, а также применения непрерывной оцифровки сигнала с последующим анализом его формы и

определения времени прихода сигнала. Всё это позволит исключить ложные кластеры и электромагнитные ливни из полученных данных. Доступ к электронике осуществляется с помощью специальной библиотеки ECL LIB,

разработанной в ИЯФ СО РАН. Поскольку электроника имеет более сложную логику и, кроме того, активно разрабатываются и выходят новые версии прошивки, крайне важно иметь автоматизированные тесты, позволяющие эффективно обнаруживать проблемы с электроникой.

2. Постановка задачи

2.1 Библиотека ECL LIB

Для работы с электромагнитным калориметром Belle II используется специальная C++ библиотека ECL LIB, упрощающая работу с коллекторами ECL через Ethernet и b2link (Рис. 2).

ECL LIB разработана в ИЯФ СО РАН. Библиотека ECL LIB предоставляет уровень аппаратной абстракции (автоматически определяет правильные протоколы взаимодействия для использования в зависимости от текущей среды),

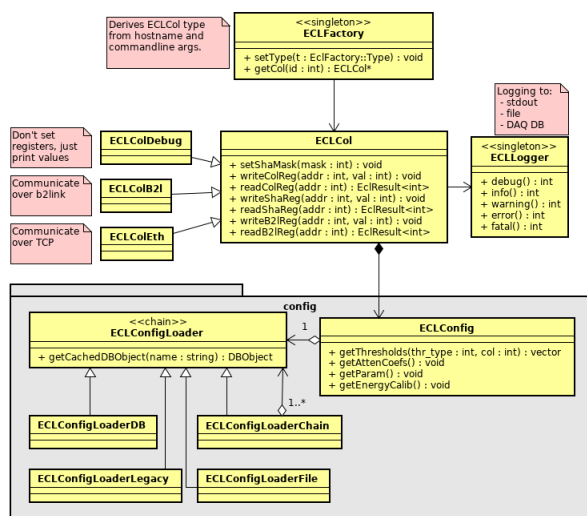


Рис. 2. UML схема библиотеки ECL LIB

настраивает многоцелевое ведение журнала (в основном, на основе класса LogFile daq_slc), предоставляет цепь конфигурации с опциями кэширования и восстановления при сбое базы данных, объединяет две наиболее сложные процедуры ввода-вывода и оптимизирует процедуры записи в регистр (ускорение примерно на 30%).

В ECL LIB есть модуль «EclColDebug»,

который эмулирует некоторые возможности модуля «ECLCollector», позволяя проводить первичное тестирование даже на локальной машине, не имеющей доступа к тестовому стенду со считывающей электроникой калориметра.

С помощью этой библиотеки мы можем управлять всеми стадиями эксперимента, получая или записывая данные нужных регистров. Взаимодействие с параметрами каждого отдельного канала электромагнитного калориметра осуществляются с помощью набора регистров. Основные функции,

которые будут использоваться `ECLCol::writeColReg()`, `ECLCol::writeShaReg()` для ввода и вывода данных в регистры, `writeShaAtten()`, `readShaAtten()` для записи и чтения аттенюаторных коэффициентов, `ECLCol::readShaThresholds()` для чтения значения порогов и `ECLCol::loadShaCoefs()` для загрузки аттенюаторных коэффициентов в регистры. Все указанные высокоуровневые методы чтения-записи данных используют низкоуровневые методы `write*Reg`, `read*Reg`.

2.2 Наиболее важные для тестирования компоненты

Задача состоит в тестировании электроники и проверке корректности параметров. Нами были выделены наиболее важные этапы тестирования.

1. Запись и чтение данных из регистра.
2. Запись и чтение аттенюаторных коэффициентов
3. Проверка загрузки прошивки ShaperDSP
4. Загрузка ECLDSP коэффициентов в ShaperDSP

2.3 Критерии тестирования

1. Первым этапом нужно проверить корректность операции чтения и записи коэффициентов в регистры. Для этого мы сначала записываем в регистры из `ecl_reg_map.h` некоторое значение с помощью `ECLCol::writeColReg()`, `ECLCol::writeShaReg()`, а после этого считываем данные с помощью `ECLCol::readColReg()`, `ECLCol::readShaReg()`. Если прочитанные коэффициенты совпадают с тем, что мы в них записали, то первый этап тестирования можно считать пройденным. Поскольку бывали случаи, при которых чтение-запись в регистры не проходила с маленькой вероятностью, нужно убедиться в повторяемости теста и для каждого регистра повторить вышеизложенные операции 10 раз.
2. После проверки возможности чтения и записи регистров колориметра нужно проверить корректность чтения и записи аттенюаторных коэффициентов. Запись и чтение необходимо провести по всем ECL копперам с номерами `cpr5001-cpr5018`, `cpr6001-cpr6008`. В каждом ECL коппере есть по два индекса калориметра. Для записи и чтения коэффициентов в каждый конкретный

калориметр используем функции `ECLCol::writeShaAtten()` и `ECLCol::readShaAtten()`.

3. На следующем этапе нужно проверить корректность загрузки прошивки `ShaperDSP`. Загрузка прошивки осуществляется с помощью `ECLCol::loadShaFw()`. Проверку корректной загрузки можно сделать, проверив тот факт, что все пороги сбросились в ноль. Чтение порогов следует производить с помощью функции `ECLCol::readShaThresholds()`. К каждому калориметру подключено 12 модулей `shaper DSP`, к каждому `shaper DSP` подключено 16 каналов. В итоге, получаем 192 канала для каждого калориметра. У каждого канала имеется 4 типа порогов: `hit threshold` (самый примитивный вид порогов, позволяющий до фитирования сигнала отбросить большие шумы), `skip amplitude threshold` (отбрасывает шумы, но уже после фитирования сигнала), `low amplitude threshold` (отбрасывает шумы малой амплитуды) и `ADC amplitude threshold` (фильтрует наиболее важные данные высокой амплитуды).
4. На последнем этапе нужно произвести проверку загрузки `ECLDSP` коэффициентов в `ShaperDSP` с помощью `ECLCol::loadShaCoefs()`. При загрузке устанавливается новое значение порогов, которое нужно проверить с помощью `ECLCol::readShaThresholds()`.

3. Описание эксперимента

3.1 Подключение по удаленному серверу SSH

На целевом сервере используется Scientific Linux 6, такая же система, как на `production`-серверах, где используется `ECL LIB`. Кроме того, с этого сервера тривиально настроить развёртку на сервер, подключенный к тестовому стенду и

проверять работу новых версий программной библиотеки. Но в условиях распространения COVID-19 стало невозможно напрямую работать с оборудованием в ИЯФ. Было принято решение запускать и проводить проверку и отладку кода с помощью удаленного сервера по SSH. SSH — сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений. SSH позволяет

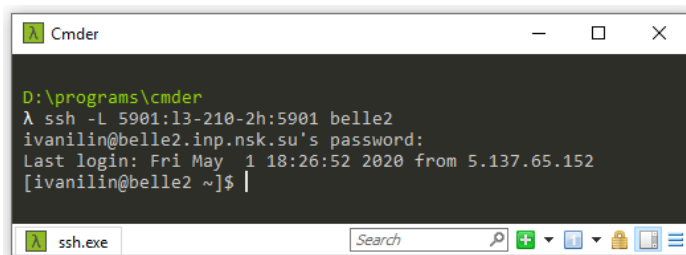


Рис. 3. Эмулятор консоли “Cmder” и подключение к удаленному серверу ИЯФ по SSH.

безопасно передавать в незащищённой среде практически любой сетевой протокол. Таким образом, можно не только удалённо работать на компьютере через командную оболочку, но и передавать по зашифрованному каналу звуковой поток или видео. Также SSH может использовать сжатие передаваемых данных для последующего их шифрования, что удобно, например, для удалённого запуска клиентов X Windows System. Для подключения по SSH использовали эмулятор консоли Cmder (Рис. 3).

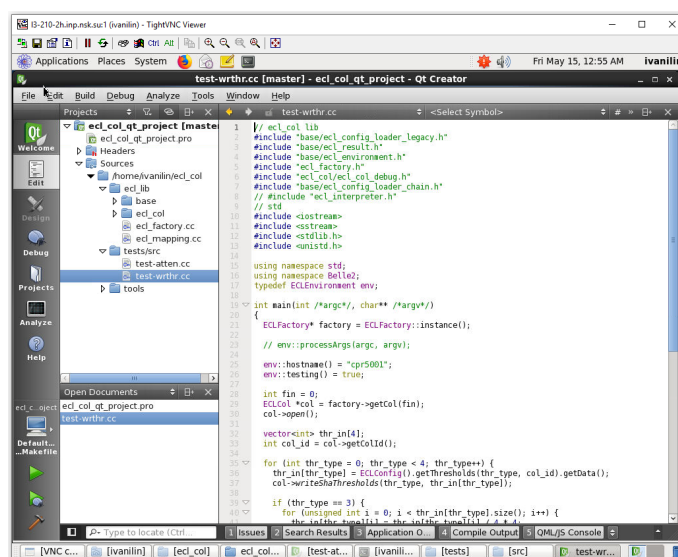


Рис. 4. Работа с удаленным сервером ИЯФ по SSH в среде QtCreator.

Позволяет безопасно обращаться дистанционно к рабочему столу другого компьютера через клиентскую программу или браузер, используя встроенный веб-сервер (Рис. 4).

3.2 Компиляция и исполнение кода

Для программирования использовалась среда QtCreator (Рис. 4), а для исполнения кода – компилятор g++. В качестве системы сборки используется GNU Make.

3.3 Написание тестов

Для проверки корректность операции чтения и записи коэффициентов в регистры мы сначала записываем в регистры некоторое значение с помощью writeColReg(), writeShaReg(), а после этого считываем данные с помощью readColReg(), readShaReg():

```
for (int t = 0; t < 10; t++) {
    vector<int> colValues;
    vector<int> shaValues;
    for (auto k : colRegs)
        colValues.push_back(static_cast<int>(10 * (double) rand() / RAND_MAX));
    for (auto k : shaRegs)
        shaValues.push_back(static_cast<int>(10 * (double) rand() / RAND_MAX));

    for (int i = 0; i < colRegs.size(); i++)
        col->writeShaReg(colRegs[i], colValues[i]);

    for (int i = 0; i < shaRegs.size(); i++)
        col->writeColReg(shaRegs[i], shaValues[i]);

    // Reading data from calorimeter and compare with vector of values
    for (int m = 0; m < colRegs.size(); m++) {
        if (col->readColReg(colRegs[m]) != values[m]) {
            printf("writeColReg() | readColReg() test failed.\n");
            return 1;
        }
    }
    for (int m = 0; m < shaRegs.size(); m++) {
        if (col->readShaReg(shaRegs[m]) != shaValues[m]) {
            printf("writeShaReg() | readShaReg() test failed.\n");
            return 2;
        }
    }
}
```

Рис. 5. Главная часть теста на проверку чтения-записи данных в регистры коллектора.

Поскольку бывали случаи, при которых чтение-запись в регистры не проходила с маленькой вероятностью, нужно убедиться в повторяемости теста и для каждого регистра повторить вышеизложенные операции 10 раз.

После проверки возможности чтения и записи регистров колориметра нужно проверить корректность чтения и записи аттеньюторных коэффициентов.

Запись и чтение необходимо провести по всем ECL копперам с номерами cpr5001-cpr5018, cpr6001-cpr6008. В каждом ECL коппере есть по два индекса калориметра:

```
for(auto host : hosts) {
    env::hostname() = host;
    env::testing() = true;

    for(int fin = 0; fin < 2; fin++) {
        // Open calorimeter with index fin (0 - 1)
        ECLCol *col = factory->getCol(fin);
        col->open();

        // Get the List of Atten Coefs from ECLConfig
        vector<int> atten_1 = ECLConfig().getAttenCoefs(col->getColId()).getData();
        // Write Atten Coefs from ECLConfig to col
        col->writeShaAtten();
        // Read Atten Coefs from col
        vector<int> atten_2 = ((ECLColDebug *) col)->getAtten();
        col->close();

        // Compare Atten Coefs from col with Atten Coefs from ECLConfig
        if (atten_1 != atten_2) {
            printf("writeAtten() test failed.\n");
            return 1;
        }
    }
}
```

Рис. 6. Главная часть теста на проверку чтения-записи данных в регистры коллектора.

Для записи и чтения коэффициентов в каждый конкретный калориметр используем функции `ECLCol::writeShaAtten()` и `ECLCol::readShaAtten()`.

Для проверки корректности загрузки прошивки нужно убедиться в том, что все пороги сбросились в ноль. Чтение порогов следует производить с помощью функции `ECLCol::readShaThresholds()`. К каждому калориметру подключено 12 модулей shaper DSP, к каждому shaper DSP подключено 16 каналов. В итоге, получаем 192 канала для каждого калориметра. У каждого канала имеется 4 типа порогов: hit threshold (самый примитивный вид порогов, позволяющий до фитирования сигнала отбросить большие шумы), skip amplitude threshold (отбрасывает шумы, но уже после фитирования сигнала), low amplitude threshold (отбрасывает шумы малой амплитуды) и ADC amplitude threshold (фильтрует наиболее важные данные высокой амплитуды):

```

for(auto host : hosts) {
    env::hostname() = host;
    env::testing() = true;

    for(int fin = 0; fin < 2; fin++) {
        // Open calorimeter with index fin (0 - 1)
        ECLCol *col = factory->getCol(fin);
        col->open();

        // ShaperDSP firmware downloads
        col->loadShaFw();

        col->close();

        for (int thr_type = 0; thr_type < 4; thr_type++) {
            vector<int> thr_out(192);
            for (int sh = 0; sh < 12; sh++) {
                int mask = 1 << sh;
                if ((col->getShaMask() & mask) == 0) continue;
                vector<int> thr_out_tmp = col->readShaThresholds(sh, thr_type);
                for (int ch = 0; ch < 16; ch++) {
                    thr_out[sh*16 + ch] = thr_out_tmp[ch];
                }
            }
            for(auto thr : thr_out) {
                if (thr != 0) {
                    printf("ECLCol::loadShaFw() test failed.");
                    return 1;
                }
            }
        }
    }
}

```

Рис. 6. Главная часть теста на проверку чтения-записи данных в регистры коллектора.

На последнем этапе нужно произвести проверку загрузки ECLDSP коэффициентов в ShaperDSP с помощью `ECLCol::loadShaCoefs()`. При загрузке устанавливается новое значение порогов, которое нужно проверить с помощью `ECLCol::readShaThresholds()`. Алгоритм данного теста почти полностью повторяет предыдущий шаг, но теперь сравнение нужно сделать не с нулём, а с 20.

5. Результаты тестирования

Для оценки покрытия тестами библиотеки ECL LIB мы использовали утилиту Gcov — свободно распространяемая утилита для исследования покрытия кода. Gcov генерирует точное количество исполнений для каждого оператора в программе. Данная утилита используется совместно с gcc для анализа

программ с целью достижения наибольшей производительности и эффективности кода, и выявления не покрываемых тестами участков кода. В результате исполнения Gcov мы получили долю покрытия в 63%.

6. Заключение

В итоге нам удалось разработать эффективные тесты для проверки работы электроники электромагнитного калориметра Belle-II. С помощью утилиты Gcov мы выяснили, что доля покрытия тестами библиотеки ECL LIB составляет 63%. Для работы с электроникой мы использовали C++ библиотеку ECL LIB, специально разработанную для эксперимента Belle-II в ИЯФ. В условиях распространения COVID-19 стало невозможным проведение экспериментов в ИЯФ. Для доступа к удаленному серверу ИЯФ было решено использовать подключение по SSH. Для управления компьютером ИЯФ использовалась программа TightVNC. Через TightVNC можно было писать и исполнять код на машине ИЯФ в QtCreator. Было решено разбить тестирование на несколько основных этапов: проверка корректности операции чтения и записи коэффициентов в регистры, проверка корректности чтения и записи аттенюаторных коэффициентов, проверка корректности загрузки прошивки ShaperDSP и проверку загрузки ECLDSP коэффициентов в ShaperDSP. Первый этап был выполнен с помощью использования функций `ECLCol::writeColReg()`, `ECLCol::readColReg()`, `ECLCol::writeShaReg()` и `ECLCol::readShaReg()`: нужно просто записать некоторое значение в регистр и после этого проверить, совпадает ли записанное в регистр значение с тем, что было в него записано. Второй этап выполнялся аналогично: с помощью функций `ECLCol::writeShaAtten()` и `ECLCol::readShaAtten()` нужно записать значения аттенюаторных коэффициентов, а затем прочитать их и сравнить с записанными. Третий этап тестирования производился с помощью проверки значений порогов shaper. Чтение порогов производилось с помощью функции `ECLCol::readShaThresholds()`. Последний этап: проверка загрузки ECLDSP коэффициентов в ShaperDSP производилась с помощью отслеживания

изменения значений порогов. Значение порогов считывалось с помощью `ECLCol::readShaThresholds()`. Каждый из этих четырёх этапов тестирования был успешно выполнен. В дальнейшем планируется расширять и дополнять область тестирования, добившись 100% покрытия тестами ECL LIB. Так же планируется доработка модуля `ec1_col_debug`, необходимого для эмуляции работы реальной электроники, совместно с параллельной доработкой библиотеки ECL LIB.

4 Список литературы

1. Belle-ECL^{1,5}, V Aulchenko^{1,5}, A Bobrov^{1,5}, A Bondar^{1,5}, B G Cheon², S Eidelman^{1,5}, D Epifanov⁶, Yu Garmash^{1,5}, Y M Goh², S H Kim², P Krokovny^{1,5}, A Kuzmin^{1,5}, I S Lee², D Matvienko^{1,5}, K Miyabayashi⁴, I Nakamura³, V Shebalin^{1,5}, B Shwartz^{1,5*}, Y Unno², Yu Usov^{1,5}, A Vinokurova^{1,5}, V Vorobjev^{1,5}, V Zhilich^{1,5}, V Zhulanov¹, Electromagnetic calorimeter for Belle II // Journal of Physics: Conference Series 587 (2015) 012045. С. 4 – 5.
2. Institute of Particle and Nuclear Studies 1-1 Oho, Tsukuba-shi, Ibaraki-ken, 305-0801, Japan, Belle II Technical Design Report // Journal of Physics: Conference Series 587 (2015) 012045. С. 4 – 5.