

Описание FIT readout unit

Финогеев Дмитрий, ИЯИ РАН

23 июня 2020 г.

1 FIT readout unit

FIT readout unit (FRU) является модулем программного обеспечения ПЛИС плат PM/TCM. Основные функции FRU:

- Получение и обработка данных с СТР (через CRU): Триггеры, BC, Orbit
- Формирование пакета данных полученных с PM/TCM для каждого события (event packet, EP)
- Отбор данных по триггеру в зависимости от режима набора данных
- Формирование RDH пакета содержащего набор EP в соответствии требованиями ALICE DAQ
- Отправка RDH пакета в CRU по протоколу GBT

Функциональная схема FRU представлена на рис 1

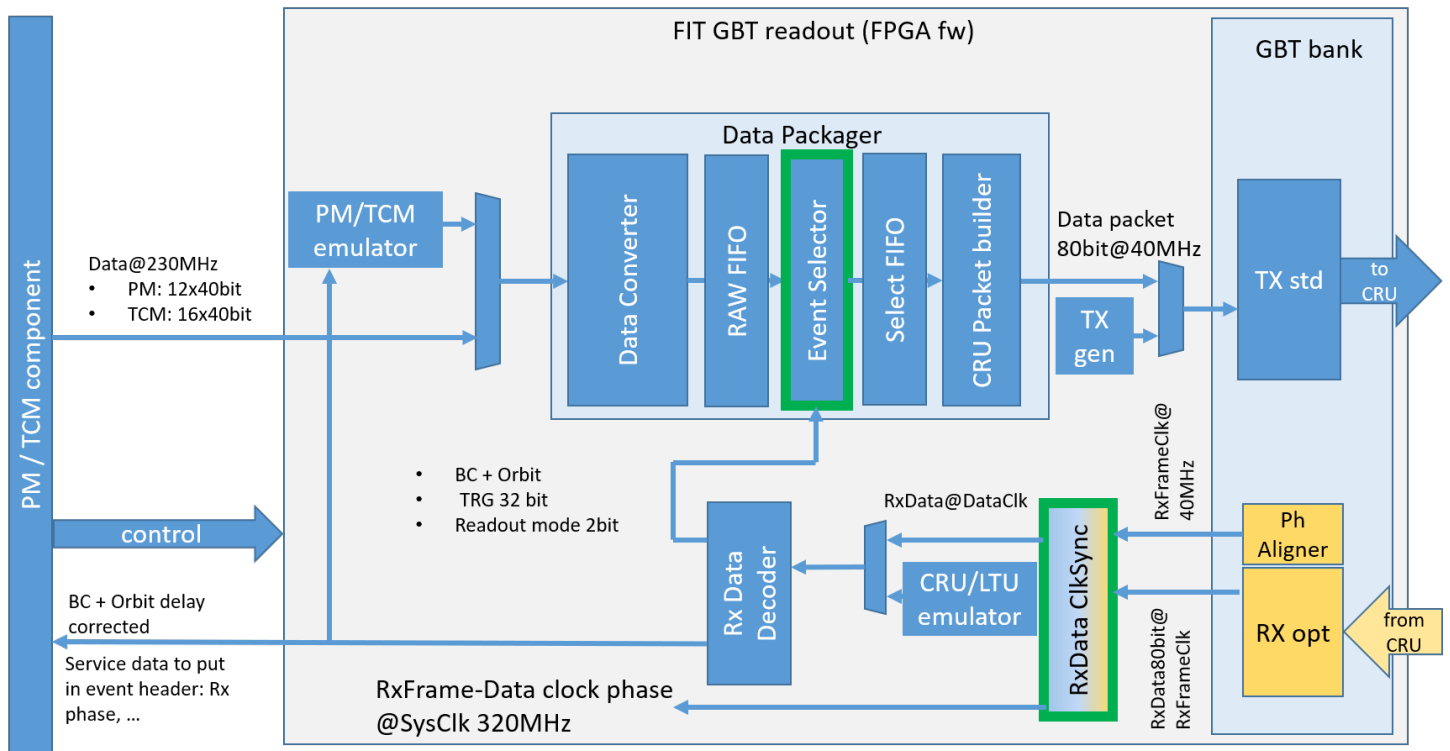


Рис. 1: Функциональная схема FIT readout unit

FRU обменивается данными с PM/TCM с одной стороны и с CRU с другой. Данные в/из CRU передаются по оптической линии по протоколу GBT. Ширина шины составляют 80 бит и тактируются по фрейму 40 МГц синхронным с CRU. Данные содержат информацию orbit и bc 32 и 12 бит соответственно и триггерную информацию - 32 бита каждый из которых отвечает за соответствующие триггера включая триггера запуска и остановки рана, триггер физического события. Данные полученные по GBT передаются в модуль 'RxDataClkSync' для синхронизации с внутренним фреймом 40МГц, фаза между фреймами измеряется по фрейму 320МГц. После синхронизации данные передаются в модуль 'RX Data Decoder' где принимаются команды запуска и остановки рана и данные о BC и orbit. Данные BC и Orbit корректируются в соответствии с задержкой и отправляются в программный модуль PM/TCM. Для тестов без

CRU предусмотрен модуль 'CRU/LTU emulator' в котором генерируются данные содержащие информацию orbit и bc и триггерную информацию включая триггера запуска и остановки рана, триггер физического события.

Данные из PM/TCM передаются по клок 320МГц и обрабатываются модулем 'Data Packager', для тестов на отладочной плате предусмотрен модуль 'PM/TCM emulator' позволяющий генерировать данные с PM/TCM. Модуль 'Data Packager' содержит подмодули 'Data Converter', 'Event selector' и 'CRU Packet builder'. Модуль 'Data Converter' формирует пакеты данных полученных с PM/TCM для каждого события (event packet, EP) и отправляет их в буфер памяти 'RAW FIFO' по клок 320 МГц. Модуль 'Event Selector' принимает EP из памяти 'RAW FIFO' по клок 320МГц, отбирает события в соответствии с триггером в триггерном режиме передачи и группирует данные в RDH пакеты в соответствии с требованиями ALICE DAQ. Отобранные и сгруппированные события с сопутствующей информацией необходимой для заголовка поступают в буфер памяти 'Select FIFO'. Третий модуль 'CRU Packet Builder' принимает данные из блока памяти 'Select FIFO', дополняет сформированные данные RDH заголовком и отправляет результат в модуль GBT для дальнейшей передачи в CRU. Для проверки работы GBT передатчика предусмотрен 'TX generator' передающий постоянный паттерн на вход GBT.

2 Описание алгоритма коррекции ВС и отбора событий

На рис. 2 представлена схема временной коррекции и алгоритма отбора событий.

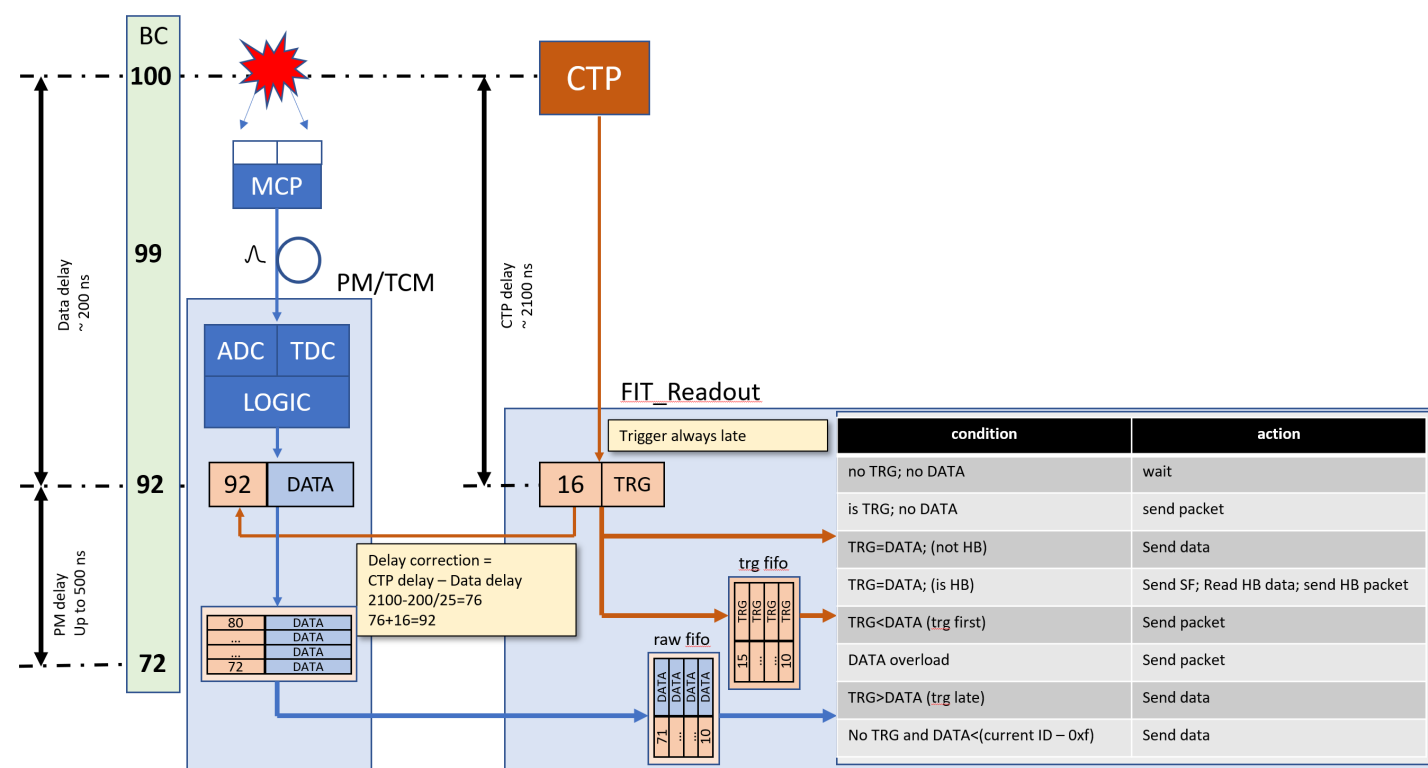


Рис. 2: Схема временной коррекции ВС и отбора событий по триггеру

Каждое столкновение (событие, event) в ALICE@LHC пронумеровано двумя целыми числами: BC - порядковым номером банча в орбите (кольце ускорителя) 0 - 3563 (12 бит) и Orbit - порядковым номером орбита (32 бита).

Каждому событию в эксперименте ALICE соответствует триггерное решение, которое говорит о характере данного события. Так же существуют технические триггеры. Примеры триггеров: команды запуска и остановки набора данных в непрерывном и триггерном режимах (SOC, SOT, EOC, EOT), триггер отброса данных (HBr), триггер начала орбита (Orbit) и другие. Каждому триггеру соответствует один бит в 32 битном слове, каждому событию может соответствовать несколько триггеров. Номер и триггерное решение для каждого события посылаются модулем Central Trigger Processor (CTP) с частотой 40МГц. Данная информация поступает в FIT Readout unit по протоколу GBT через модуль Common readout Unit (CRU) с задержкой порядка нескольких микросекунд.

На схеме рис. 2 эти данные обозначены оранжевой стрелочкой от модуля CRU, для примера, на схеме текущий поступивший в FRU номер обозначен 16 и триггерное решение TRG для данного события. На схеме, для примера, происходит событие под номером 100. Регистрация и оцифровки события детектором происходит с задержкой, для примера на схеме задержка равна 8. Как задержка регистрации сигнала, так и задержка триггерного сигнала постоянны и известны. Задержка регистрации сигнала меньше чем задержка триггерного сигнала и для определения номера регистрируемого события, к номеру полученному от CTP прибавляется разница между задержками регистра-

ции сигнала и распространение триггерного сигнала. На схеме эта разница равна $(2100\text{нс} - 200\text{нс})/25 = 76$ и прибавив 76 к полученному номеру от СТР 16 получаем номер регистрируемого события 92. Данная операция выполняется в модуле 'RxDataDecoder', задержка определяется регистром 'BCID delay'.

Триггерная информация, поступившая от СТР относится к поступившему вместе с ней номером события (на схеме 16-TRG). Она необходима для отбора и сортировки событий по триггеру и сохраняется в буфер памяти 'trg fifo'. Поскольку задержка регистрации событий меньше, чем задержка триггерного сигнала данные о событиях сохраняются в буфер памяти 'raw fifo' для дальнейшей сортировки. Сортировка на схеме обозначена блоком 'Event selection' и на входе принимает данные о текущем принятом номере события от СТР, и данные из буферов памяти 'raw fifo' и 'trg fifo' для сравнения.

Модуль 'Event Selection' выполняет сравнение номеров события для триггерного сообщения и данных детектора из 'raw data fifo' и 'trg fifo'.

- Если номер для триггерного сообщения меньше (соответствует более раннему событию) или данные в 'raw fifo' отсутствуют то это означает что данные для этого триггерного сообщения отсутствуют. В этом случае триггерное сообщение вычитывается из 'raw fifo' и при необходимости формируется RDH пакет (на пример при получении 'HB trigger').
- Если номер события в данных меньше чем номер в триггерном сообщении то это означает что для данных отсутствует триггерное решение. В случае непрерывной передачи 'continuous readout' данные вычитываются из 'raw fifo' и передаются в 'select fifo' для формирования RDH. В случае триггерной передачи 'triggered readout' данные вычитываются из 'raw fifo' и отбрасываются в соответствии с алгоритмом отбора по триггеру.
- Если номер события в данных равен номеру в триггерном сообщении то это означает что триггерное сообщение соответствует данным детектора. В случае непрерывной передачи 'continuous readout' данные вычитываются из 'raw fifo' и передаются в 'select fifo' для формирования RDH. В случае триггерной передачи 'triggered readout' данные вычитываются из 'raw fifo' и передаются в 'select fifo' для формирования RDH только при наличии необходимого триггера для отбора событий. Маска для отбора триггеров содержится в регистре 'Data select trigger mask'. Триггерное сообщение так же вычитывается из 'trg fifo'.
- Если присутствуют данные детектора и отсутствует триггерное сообщение то это означает что триггерное сообщение может прийти позже. В этом случае выполняется сравнение номера события данных и текущего принятого номера события из СТР. Если разница между этими событиями больше фиксированной величины, то это означает что триггерное сообщение отсутствует для данных, данные вычитываются из буфера 'raw fifo' и отбираются в соответствии с режимом вычитывания данных. Если разница меньше или равна фиксированной величине, то данные сохраняются в 'raw fifo'. Величина для сравнения задается регистром 'Trigger compare delay'.

Для формирования RDH пакета данные отобранные из 'raw fifo' помещаются в 'selected fifo'. Для отправления RDH пакета при накоплении максимального объема данных или при иной необходимости (на данный момент согласно требованиям DAQ ALICE необходимо отправлять пакет при получении 'HB trigger' и для отправки 'HB stop frame') формируется управляющее слово. Управляющее слово содержит количество слов в RDH пакете и сопутствующую информацию для RDH заголовка в том числе номер орбита, триггерное решение (актуально для RDH v4). Модуль 'CRU Packet builder' принимает управляющее слово, вычитывает соответствующее количество слов из 'Selected fifo', формирует RDH пакет включая данные из регистров 'FEE ID', 'PAR', 'Detector field' (актуально для RDH v4) и отправляет его в модуль GBT.

3 Описание функциональных модулей

3.1 модуль Event Selector

Назначение модуля описано в разделе "Описание алгоритма коррекции ВС и отбора событий". Модуль 'Event selector' принимает данные и триггерные сообщения для сравнения и отбора событий. Отобранные события помещаются в 'selected fifo' и сопровождаются командными словами для формирования RDH пакетов.

Каждое событие (40 МГц) модуль РМ может сформировать до 6 слов плюс одно слово заголовки события, модуль ТСМ формирует одно слово плюс слово заголовка (в отладочном режиме 9+1 слов, допускается пропуск событий). Пакеты модуля передаются по kloku 320 МГц и максимальная длина пакета для одного события (40 МГц) составляет 8 слов. Модуль 'Event selector' также читает и обрабатывает данные по kloku 320 МГц. 'selected fifo' читается модулем 'CRU packet builder' по kloku 40 МГц, GBT модуль так же передает данные по kloku 40 МГц. В связи с тем что 'selected fifo' читается по kloku 320 МГц а записывается по kloku 40 МГц, возможно его переполнение. В случае когда в 'selected fifo' не остается места для данных, данные отбрасываются. В этом случае запоминаются первый и последний номер орбита и количество пропущенных событий, эта информация доступна в регистрах 'first hit dropped orbit', 'last hit dropped orbit', 'selector hits dropped'. В регистре 'readout rate' отображается количество слов данных (80бит 40МГц) за последний орбит. В регистре 'selector fifo count' доступна занятость 'selector fifo'. Бит 'reset drop counter' в регистре 'Reset control' (при записи в него значения 0x1) позволяет сбросить счетчик пропущенных событий.

Регистр 'is HB response' (при значении 0x1) позволяет отключить отправку RDH пакета по получению триггера HB для снижения потока данных при использовании FTM. Регистр 'Max RDH payload' задает максимальное количество слов в RDH пакете. Максимальный объем RDH пакета составляет 8 Кб, поскольку каждое переданное GBT слово дополняется в CRU до 128 бит, максимум в RDH пакете может содержаться $8192 \cdot 8 / 128 = 512$ слов GBT. Значение регистра означает количество GBT слов и должно быть равно 0x200. Для отладки максимальный размер пакета может быть уменьшен. Регистр 'CRU trigger delay' определяет как долго данные модуля 'ожидают' триггерное сообщение (см. раздел "Описание алгоритма коррекции ВС и отбора событий"). Регистр 'data select trigger mask' определяет маску (набор) триггеров для отбора в режиме триггерного вычитывания ('triggered mode'). При значении регистра 'readout mode' равном 'idle' модуль 'event selector' прекращает работу.

3.2 модуль RX Data Decoder

Модуль предназначен для обработки данных полученных от CRU (СТР) по GBT. Как описано в разделе "Описание алгоритма коррекции ВС и отбора событий" каждому событию соответствует номер и триггерное решение. Допускается что триггерное сообщение, содержащее текущий номер события, может приходить не каждый клок 40 МГц, в случае если триггер отсутствует сообщение может быть пропущено (относится к первым требованиям ALICE DAQ). В связи с чем модуль 'RX data decoder' содержит внутренний счетчик событий, который может находиться в трех состояниях 'Start', 'Sync', 'Lost', состояние счетчика доступно в регистре 'BCID sync mode'. При включении или по команде 'reset orbit sync' (бит 'reset orbit sync' в регистре 'reset control') счетчик переходит в состояние 'start' до получения первого триггера. После получения триггера счетчик переходит в состояние 'sync' и начинает отчет событий с полученного номера от СТР. Для определения номера события используется текущее значение счетчика в связи с чем получение каждого номера события от СТР не обязательно. Текущее значение номера события доступно из регистров 'CRU orbit', 'CRU bc'. При получении триггерного сообщения номер события полученный от СТР сравнивается со значением счетчика. Если эти значения не равны, то это означает рассинхронизацию внутреннего счетчика с нумерацией СТР, счетчик переходит в состояние 'lost' до момента получения сигнала 'reset orbit sync'.

Как было описано в разделе "Описание алгоритма коррекции ВС и отбора событий" номер полученный от СТР, (фактически - номер внутреннего счетчика синхронизированный с СТР) корректируется на величину разности задержек в распространении триггерного сообщения и обработки сигнала детектором. Величина корректировки задается регистром 'bcid delay'.

Модуль 'RX data decoder' обрабатывает триггера запуска и остановки вычитывания (SOT - start of trigger, EOT - end of trigger, SOC - start of continuous, EOC - end of continuous) управляя текущим режимом вычитывания. Состояние режима может принимать значения 'idle', 'continuous', 'triggered', значение доступно по регистру статуса 'readout mode'. Выставив в регистре управления 'readout mode' бит 'force readout idle mode' возможно принудительно перевести режим вычитывания в 'idle' что необходимо при включении или возникновении ошибки. При значении регистра 'readout mode' равном 'idle' модуль 'event selector' прекращает работу.

3.3 модуль СТР emulator

Для работы без СТР/LTU предусмотрен модуль 'СТР emulator' который производит все необходимые данные для отладки работы 'FIT readout unit'. Включение модуля (отключение приема сигналов СТР) происходит переключением регистра 'Trigger generator' в состояние 'Continuous generator'. 'СТР emulator' выполняет следующие функции:

- Генерация триггера 'HB' и номера события event id для эмуляции CRU.
- Генерация триггеров запуска и остановки процедуры вычитывания.
- Генерация одиночного триггера по команде
- Генерация непрерывного триггера с возможностью синхронизацией с тестовым генератором данных

Генерация триггера 'HB' и номера события event id происходит автоматически и не управляется.

Генерация триггеров запуска и остановки процедуры вычитывания происходит по команде. При переводе регистра 'readout command' из состояния 'OFF' в состоянии команды 'SOC', 'SOT', 'EOT', 'EOC' отправляется соответствующий триггер управления режимом вычитывания. Триггер команды отправляется в момент начала орбита вместе с триггером 'HB' в соответствии с требованиями ALICE DAQ.

Для отправки единичного триггера необходимо в регистр 'trigger single value' записать ноль, а потом значение триггера, которое необходимо отправить. Отправка триггера происходит при изменении регистра из значения 0x0 в ненулевое.

Генерация непрерывных триггеров по функциональности схожа с тестовым генератором данных модуля 'PM/TCM emulator'. Значение генерируемого триггера задается регистром 'continuous trigger value'. Регистрами 'trigger continuous pattern' формируется последовательность из 64 значений задающая очередность генерации триггера 'pattern'. Частота запуска 'pattern' определяется регистром 'trigger bunch frequency' задающим число отсчетов 'event id' между 'pattern'. Первая генерация 'pattern' происходит с отступом от начала орбита ('HB' trigger) на величину регистра

'trigger frequency offset' задающим число отсчетов 'event id'. Если частота 'pattern' равна величине орбита (регистр 'trigger bunch frequency' равен 0xDEC) то сдвиг паттерна относительно 'HB trigger' будет постоянен.

На рисунке 2 представлена схема генерации последовательности триггеров модулем 'CTP emulator'.

In FTM if Data generator is on, Trigger not used

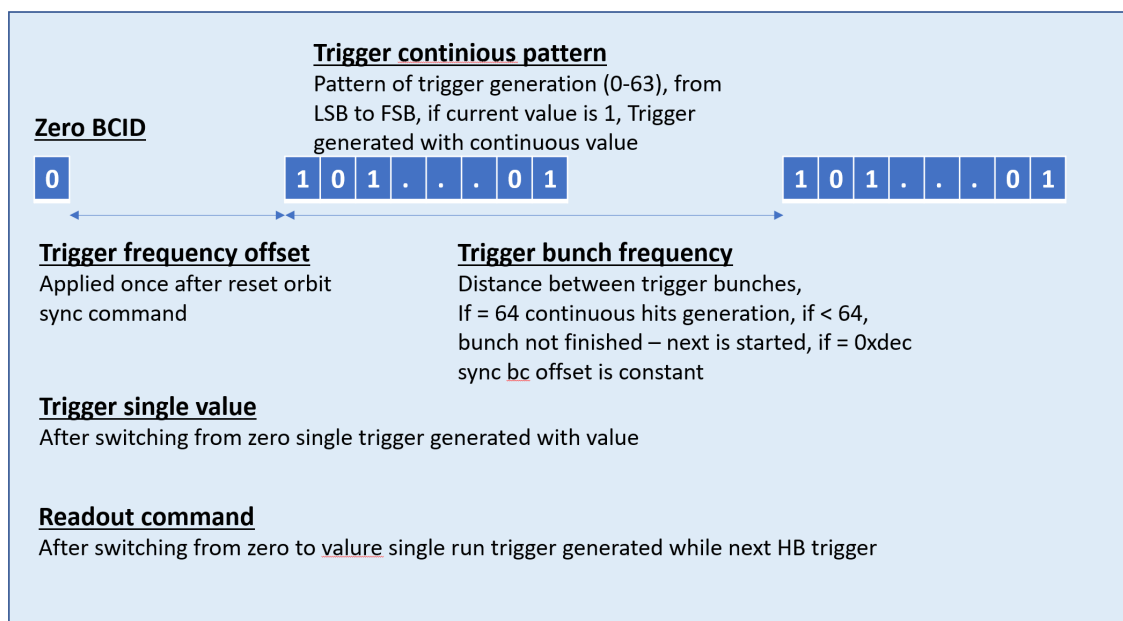


Рис. 3: Схема генерации последовательности триггеров модулем 'CTP emulator'

Так, на пример, если регистр 'trigger bunch frequency' равен 0xDED, регистр 'trigger frequency offset' равен 0x5 то в момент старта генератора или при переходе бита 'reset bunch offset' в регистре 'reset control' из состояния 0x1 в 0x0 будет ожидаться следующий триггер 'HB'. При получении триггера 'HB' будет произведен отсчет в 0x5 событий 'event id'. После этого будет запущена генерация триггеров в соответствии с 'pattern'. Если регистр 'Trigger continuous pattern 31 to 0' равен 0x3 и 'Trigger continuous pattern 63 to 32' равен 0x1 то будет сгенерированы триггера на пятом, шестом и 38 событиях в орбите. После этого будет произведен отсчет в 0xDED событий 'event id' и новый запуск генерации триггеров в соответствии с 'pattern'. Поскольку значение 0xDED на одно больше, чем длительность орбита, в следующем орбите триггера будут в событиях с номерами 6, 7 и 39.

Модуль 'PM/TCM emulator' имеет схожий алгоритм генерации что позволяет обеспечить генерацию данных синхронно с генерацией триггера для отладки работы механизма отбора событий.

3.4 модуль Data converter

Модуль 'data converter' принимает данные от PM/TCM по блоку 320 МГц и дополняет их заголовком 'event header'. Данные, дополненные заголовком опраиваются в 'raw fifo' по блоку 320 МГц. Для контроля занятости 'raw fifo' по регистру 'raw fifo count' доступно количество слов данных в fifo.

3.5 модуль TX data gen

Для отладки GBT передачи возможно подать фиксированный пакет данных в передачи GBT переключив регистр 'Data generator' в режим 'TX generator'. Так же возможно выключить всю логику 'FIT readout unit' выставив в регистре 'Readout mode' бит 'is readout bypass mode'. В этом случае данные, полученные с модуля PM/TCM будут записываться во внутренне fifo по блоку 320 МГц и сразу же, без обработки, передаваться в GBT модуль по блоку 40 МГц.

3.6 модуль GBT

Описание работы модуля GBT доступно по ссылке <https://espace.cern.ch/GBT-Project/GBT-FPGA>. Статус модуля доступен в регистре 'gbt status'. При возникновении ошибки 'GBT RX error detected' бит 'GBT RX error latch' принимает состояние 0x1 до подачи сигнала 'reset GBT errors' (регистр 'Reset control'). Перезагрузка GBT модуля выполняется командой 'GBT reset' (регистр 'Reset control').

3.7 модуль RX data clock sync

На схеме 4 представлена схема регистров (логических защелок) для перехода между асинхронными клоками: внутренним клоком 'Data clk' 40 МГц и клоком от CRU 'RX frame clk' 40 МГц полученным через модуль GBT. Фаза между этими клоками измеряется по клоку 'system clk' 320 МГц и может принимать значения от 0 до 7. При нормальном режиме работы фаза не должна маяться больше чем на два отсчета, если это происходит значение бита 'RX phase error' регистра 'GBT status' принимает значение 0x1 до сброса командой 'reset RX phase error' регистра 'Reset control'

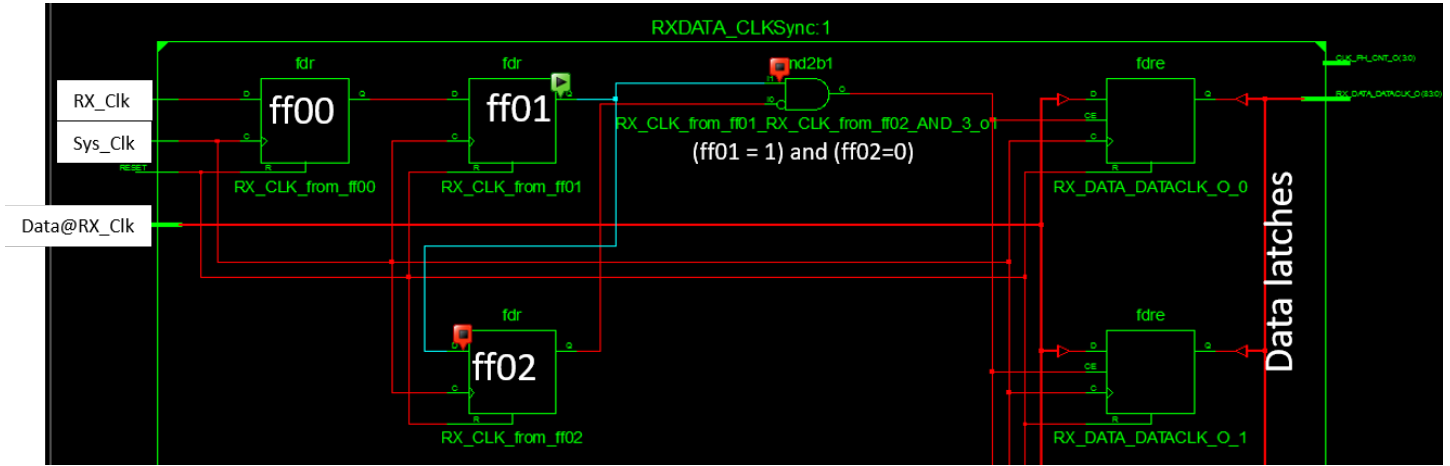


Рис. 4: Схема перехода между асинхронными клоками от RXframeClk 40МГц от CRU и внутренним клоком Data clk 40 МГц

3.8 модуль CRU packet builder

Модуль 'CRU packet builder' принимает данные, сформированные модулем 'event selector' (см разделы "Описание алгоритма коррекции ВС и отбора событий" и "модуль Event Selector") и формирует RDH пакет используя регистры 'FEE ID', 'PAR', 'DET field'.

3.9 модуль PM/TCM emulator

Данный модуль предназначен для отладки 'FIT readout unit' и генерирует данные от модуля PM/TCM, где в качестве значения данных, используется непрерывный счетчик. Для включения модуля в регистр 'Data generator' следует записать значение 'main generator'. Регистрами 'Data bunch pattern', 'data bunch frequency', 'data frequency offset' можно задавать последовательность, частоту, и смещение относительно орбита. Регистр 'trigger response mask' позволяет определить набор триггеров, при получении которых будет вырабатываться данные генератора. Бит 'reset bunch offset' в регистре 'Reset control' перезапускает генерацию данных в модулях 'PM/TCM emulator' и 'CTP emulator' для их синхронизации.

4 Список ошибок и процедура их обработки

При рассинхронизации номера события (регистр BCID sync mode перешел в состояние 'lost' или 'start') отправка событий в модуле 'event selector' останавливается. Требуется процедура перезапуска вычитывания (еще не реализована)

5 Описание клоков

FIT readout unit использует две группы клоков с произвольным фазовым сдвигом: группа платы содержит два синхронных клока 'Data clk' 40 МГц и 'system clk' 320МГц, эти клоки формируются на плате PM/TCM. Клок 40 МГц используется для работы с данными GBT, клон 320 МГц используется для обработки данных.

Вторая группа клоков содержит клон 'RxFrameClk' 40МГц полученный из модуля GBT, синхронный CRU и используемый для передачи данных принятых данных с CRU. Данные принятые из CRU по клоку 'RxFrameClk' синхронизируются с группой клоков платы в модуле 'RxData ClkSync', в последующем клон 'RxFrameClk' не используется.

Физически клоки обеих асинхронных групп имеют один источник что обеспечивает одинаковую частоту.

6 Описание формата данных детектора FIT

Структура пакета передаваемых детектором по GBT приведена на рисунке 5. Заголовком пакета является RDH (Raw Data Header) Актуальную версию формата заголовка можно узнать из комментариев к коду O2.

<https://github.com/AliceO2Group/AliceO2/blob/dev/DataFormats/Headers/include/Headers/RAWDataHeader.h>

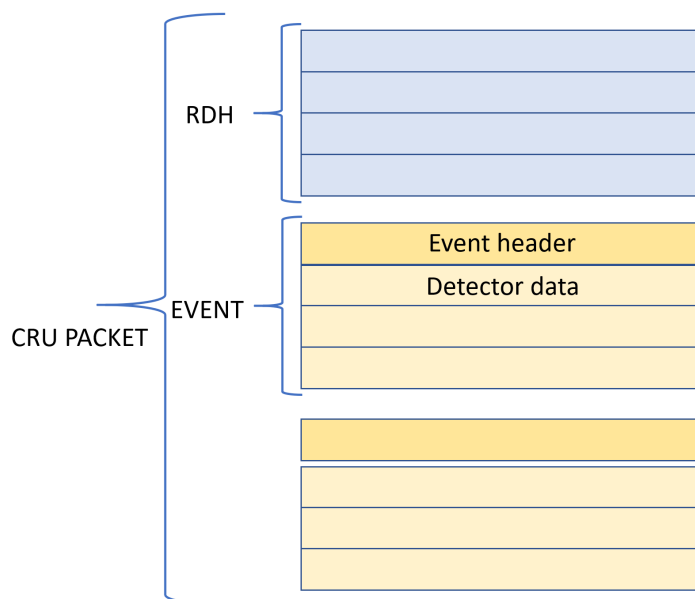


Рис. 5: Схема RDH пакета.

В теле RDH пакета содержится набор пакетов данных каждый из которых относится к отдельному событию и состоит из заголовка 'event header' и блока данных. Пакеты событий всегда относятся к одному орбиту и отсортированы по порядку поступления. Формат заголовка одинаков для модулей РМ/ТСМ, его формат представлен в таблице 6. Формат блока данных различен для РМ/ТСМ, ТСМ имеет два режима передачи: стандартный и расширенный. В РМ каждое GBT слово 80 бит содержит одно или два слова 40 бит для канала: с 0 бита по 39 и с 40 по 79, в случае одного слова, старшие биты с 40 по 79 заполняются нулями. Формат слова для канала представлен в таблице 6. Поскольку РМ имеет 12 каналов, размер пакета данных события составляет от двух до 7 слов GBT. Формат данных для ТСМ в обычном режиме представлен в таблице 6 и имеет размер в 1 GBT слово плюс заголовок. В расширенном режиме пакет данных ТСМ имеет 9 слов плюс заголовок в этом режиме передача данных для двух соседних событий невозможна. Формат данных ТСМ представлен в таблице 6. Актуальная версия формата данных доступна по ссылке.

<https://drive.google.com/file/d/1LxKEuedLqxtvCRw072ZqJ2vj7upP0ZgV>

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------|--------|------|-----------------------------|----|----|----|----|----|----|----|----|----|--|----|----|----|----|----|----|----|----|----|----------------------------|----|----|----|----|----|----|----|----|----|----------------------------|----|----|----|----|----|----|----|----|----|----------------------------|----|----|----|----|----|----|----|----|----|-----------------------------|----|----|----|----|----|----|----|----|----|-----------------------------------|----|----|----|----|----|----|----|----|----|-----------------------------------|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|--|---|--|--|--|--|--|--|--|--|--|--|--|
| | | 80 bit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| GBT word | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CPU words | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 10..15 | 9 | | | | | | | | | | 8 | | | | | | | | | | 7 | | | | | | | | | | 6 | | | | | | | | | | 5 | | | | | | | | | | 4 | | | | | | | | | | 3 | | | | | | | | | | 2 | | | | | | | | | | 1 | | | | | | | | | | 0 | | | | | | | | | | | |
| bits | | 127 | ..80 | 79 | 78 | 77 | 76 | 75 | 74 | 73 | 72 | 71 | 70 | 69 | 68 | 67 | 66 | 65 | 64 | 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | |
| PM event | header | 0xF | | words: 1..6 | | | | | | | | | | 0x000000 [24bit] | | | | | | | | | | err | | | | | | | | | | phase | | | | | | | | | | orbit ID (32bit) | | | | | | | | | | BC ID (12bit, 0x..0xDEB) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | data word 0 | | | | | | | | | | | | | channel data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | channel data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | data word ... | | | channel ID 0 0 0 | | | | | | | | | | flags (8bit) | | | | | | | | | | charge (13bit) ± | | | | | | | | | | time (12bit) ± | | | | | | | | | | channel ID 0 0 0 | | | | | | | | | | flags (8bit) | | | | | | | | | | charge (13bit) ± | | | | | | | | | | time (12bit) ± | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | data word n | | | | | | | | | | | | | channel data or empty (channel ID = 0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | channel data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PM trigger | header | 0x0 | | orbit ID (32bit) | | | | | | | | | | BC ID (12bit, 0x..0xDEB) | | | | | | | | | | amplitude (14bit) ± | | | | | | | | | | bg | | | | | | | | | | time (13bit) ± | | | | | | | | | | code (4bit) | | | | | | | | | | code: 1..12 - number of channels, | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | word 0 | 0xF | | words: 1 9 | | | | | | | | | | 0x000000 [24bit] | | | | | | | | | | err | | | | | | | | | | phase | | | | | | | | | | orbit ID (32bit) | | | | | | | | | | BC ID (12bit, 0x..0xDEB) | | | | | | | | | | 13 - no time information, | | | | | | | | | | bg out-of-gate event (background) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TCM event | Extended data | 0 | | TimeC (9bit) ± | | | | | | | | | | TimeA (9bit) ± | | | | | | | | | | AmplC (17bit) ± | | | | | | | | | | AmplA (17bit) ± | | | | | | | | | | NchanC (7bit) | | | | | | | | | | NchanA (7bit) | | | | | | | | | | 0 0 0 V C | | | | | | | | | | Sc orC orA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | word 1 | | | PM_A2 trigger data (16 LSB) | | | | | | | | | | PM_A1 trigger data (32bit) | | | | | | | | | | | | | | | | | | | | PM_A0 trigger data (32bit) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | word 2 | | | | | | | | | | | | | PM_A4 trigger data (32bit) | | | | | | | | | | | | | | | | | | | | PM_A3 trigger data (32bit) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | word 3 | | | PM_A7 trigger data (16 LSB) | | | | | | | | | | | | | | | | | | | | PM_A6 trigger data (32bit) | | | | | | | | | | | | | | | | | | | | PM_A5 trigger data (32bit) | | | | | | | | | | PM_A2 trigger data (16 MSB) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | word 4 | | | | | | | | | | | | | PM_A9 trigger data (32bit) | | | | | | | | | | | | | | | | | | | | PM_A8 trigger data (32bit) | | | | | | | | | | | | | | | | | | | | PM_A7 trigger data (16 MSB) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | word 5 | | | PM_C2 trigger data (16 LSB) | | | | | | | | | | | | | | | | | | | | PM_C1 trigger data (32bit) | | | | | | | | | | | | | | | | | | | | PM_C0 trigger data (32bit) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | word 6 | | | | | | | | | | | | | PM_C4 trigger data (32bit) | | | | | | | | | | | | | | | | | | | | PM_C3 trigger data (32bit) | | | | | | | | | | | | | | | | | | | | PM_C2 trigger data (16 MSB) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | word 8 | | | PM_C7 trigger data (16 LSB) | | | | | | | | | | | | | | | | | | | | PM_C6 trigger data (32bit) | | | | | | | | | | | | | | | | | | | | PM_C5 trigger data (32bit) | | | | | | | | | | PM_C1 trigger data (16 MSB) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Рис. 6: Формат данных детектора

7 Регистры управления и статуса

Таблица регистров управления.

| addr | 31 .. 28 | 27 .. 24 | 23 .. 20 | 19 .. 16 | 15 .. 12 | 11 .. 8 | 7 .. 4 | 3 .. 0 |
|------|-------------------------------------|----------------------|--------------|-----------------|----------------------|-------------------|-------------------|----------------|
| D8 | | | Readout mode | Readout command | Reset control | | Trigger generator | Data generator |
| D9 | Trigger respond mask | | | | | | | |
| DA | Data bunch pattern | | | | | | | |
| DB | Trigger single value | | | | | | | |
| DC | Trigger continuous pattern 63 to 32 | | | | | | | |
| DD | Trigger continuous pattern 31 to 0 | | | | | | | |
| DE | Trigger continuous value | | | | | | | |
| DF | Trigger bunch frequency | | | | Data bunch frequency | | | |
| E0 | | Trigger bunch offset | | | | Data bunch offset | | |
| E1 | FEE ID (RDH) | | | | PAR (RDH) | | | |
| E2 | Max RDH payload | | | | Detector field (RDH) | | | |
| E3 | Trigger compare delay | | | | BCID delay | | | |
| E4 | Data select trigger mask | | | | | | | |

Reset control.

| No | bit | значение |
|----|-----|------------------------|
| 0 | 8 | Orbit sync |
| 1 | 9 | Drop hit counter |
| 2 | 10 | Generator bunch offset |
| 3 | 11 | GBT errors |
| 4 | 12 | Reset GBT |
| 5 | 13 | RX phase error |

Data generator

| Значение регистра | режим работы |
|-------------------|----------------|
| 0 | No generator |
| 1 | Main generator |
| 2 | TX generator |

Trigger generator

| Значение регистра | режим работы |
|-------------------|----------------------|
| 0 | No generator |
| 1 | Continuous generator |

Readout command

| Значение регистра | режим работы |
|-------------------|--------------|
| 0 | None |
| 1 | SOC |
| 2 | SOT |
| 3 | EOC |
| 4 | EOT |

Readout mode

| No | bit | значение |
|----|-----|-------------------------|
| 0 | 20 | Is HB response |
| 1 | 21 | Is readout bypass mode |
| 2 | 22 | Force readout idle mode |

Таблица регистров статуса.

| addr | 31 .. 28 | 27 .. 24 | 23 .. 20 | 19 .. 16 | 15 .. 12 | 11 .. 8 | 7 .. 4 | 3 .. 0 |
|------|----------------------------------|----------|----------------|--------------|----------------|---------|--------|--------|
| E8 | | RX phase | BCID sync mode | Readout mode | GBT status | | | |
| E9 | CRU Orbit | | | | | | | |
| EA | | | | | CRU BC | | | |
| EB | Selector FIFO count | | | | Raw FIFO count | | | |
| EC | Selector first hit dropped orbit | | | | | | | |
| ED | Selector last hit dropped orbit | | | | | | | |
| EE | Selector total hits dropped | | | | | | | |
| EF | | | | | Readout rate | | | |

GBT status bit

| Бит | значение |
|-----|-------------------------|
| 0 | Phase aligner CPLL lock |
| 1 | RX word clock ready |
| 2 | RX frame clock ready |
| 3 | MGT link ready |
| 4 | TX reset done |
| 5 | TX FSM reset done |
| 6 | GBT RX ready |
| 7 | GBT RX error detected |
| 8 | GBT RX error latch |
| 9 | RX phase error |

Readout mode

| Значение регистра | режим работы |
|-------------------|--------------|
| 0 | Idle |
| 1 | Continuous |
| 2 | Triggered |

BCID sync mode

| Значение регистра | режим работы |
|-------------------|--------------|
| 0 | Start |
| 1 | Sync |
| 2 | Lost |

Список управляющих регистров

| Название регистра | используется в модуле | является побитовым |
|----------------------------|--|--------------------|
| Readout mode | Event selector, TX generator, RX data decoder | да |
| Readout command | CTP/LTU emulator | нет |
| Reset control | GBT, RX data decoder, Event selector, RX data clock sync | да |
| Trigger generator | CTP/LTU emulator | нет |
| Data generator | PM/TCM emulator | нет |
| Trigger response mask | CTP/LTU emulator | нет |
| Data bunch pattern | PM/TCM emulator | нет |
| Trigger single value | CTP/LTU emulator | нет |
| Trigger continuous pattern | CTP/LTU emulator | нет |
| Trigger continuous value | CTP/LTU emulator | нет |
| Trigger bunch frequency | CTP/LTU emulator | нет |
| Data bunch frequency | PM/TCM emulator | нет |
| Trigger bunch offset | CTP/LTU emulator | нет |
| Data bunch offset | PM/TCM emulator | нет |
| FEE ID | CRU packet builder | нет |
| PAR | CRU packet builder | нет |
| Max RDH payload | Event selector | нет |
| Detector field | CRU packet builder | нет |
| Trigger compare delay | Event selector | нет |
| BCID delay | RX data decoder | нет |
| Data select trigger mask | Event selector | нет |

Список регистров статуса

| Название регистра | используется в модуле | является побитовым |
|----------------------------------|-------------------------|--------------------|
| RX phase | RX data clock sync | нет |
| BCID sync mode | RX data decoder | нет |
| Readout mode | RX data decoder | нет |
| GBT status | GBT, RX data clock sync | да |
| CRU orbit | RX data decoder | нет |
| CRU bc | RX data decoder | нет |
| Selector FIFO count | Event selector | нет |
| Raw FIFO count | Data converter | нет |
| Selector first hit dropped orbit | Event selector | нет |
| Selector last hit dropped orbit | Event selector | нет |
| Selector total hits dropped | Event selector | нет |
| Readout rate | Event selector | нет |