

Описание FIT readout unit

Финогеев Дмитрий, ИЯИ РАН

17 декабря 2020 г.

Актуальная версия документа доступна для скачивания по ссылке
https://github.com/dfinogee/FIT-readout-manual/raw/master/FIT_readout_manual.pdf

Содержание

1	FIT readout unit	2
2	Описание алгоритма коррекции ВС и отбора событий	3
3	Описание функциональных модулей	4
3.1	модуль Event Selector	4
3.2	модуль RX Data Decoder	4
3.3	модуль CTP emulator	5
3.4	модуль Data converter	6
3.5	модуль TX data gen	6
3.6	модуль GBT	6
3.7	модуль RX data clock sync	6
3.8	модуль CRU packet builder	7
3.9	модуль PM/TCM emulator	7
4	Описание работы модуля FTM	7
5	Список ошибок и процедура их обработки	8
6	Описание клоков	9
7	Описание формата данных детектора FIT	9
8	Используемые компоненты блоков памяти	11
9	Регистры управления и статуса	12
9.1	Описание регистров управления	12
9.2	Описание регистров статуса	13
9.3	Инициализация регистров при запуске электроники	14
10	Описание пакета симуляции	15
11	Ссылки на рабочие документы и программное обеспечение по проекту FIT	16

1 FIT readout unit

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

- Получение и обработка данных с СТП (через CRU): Триггеры, ВС, 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

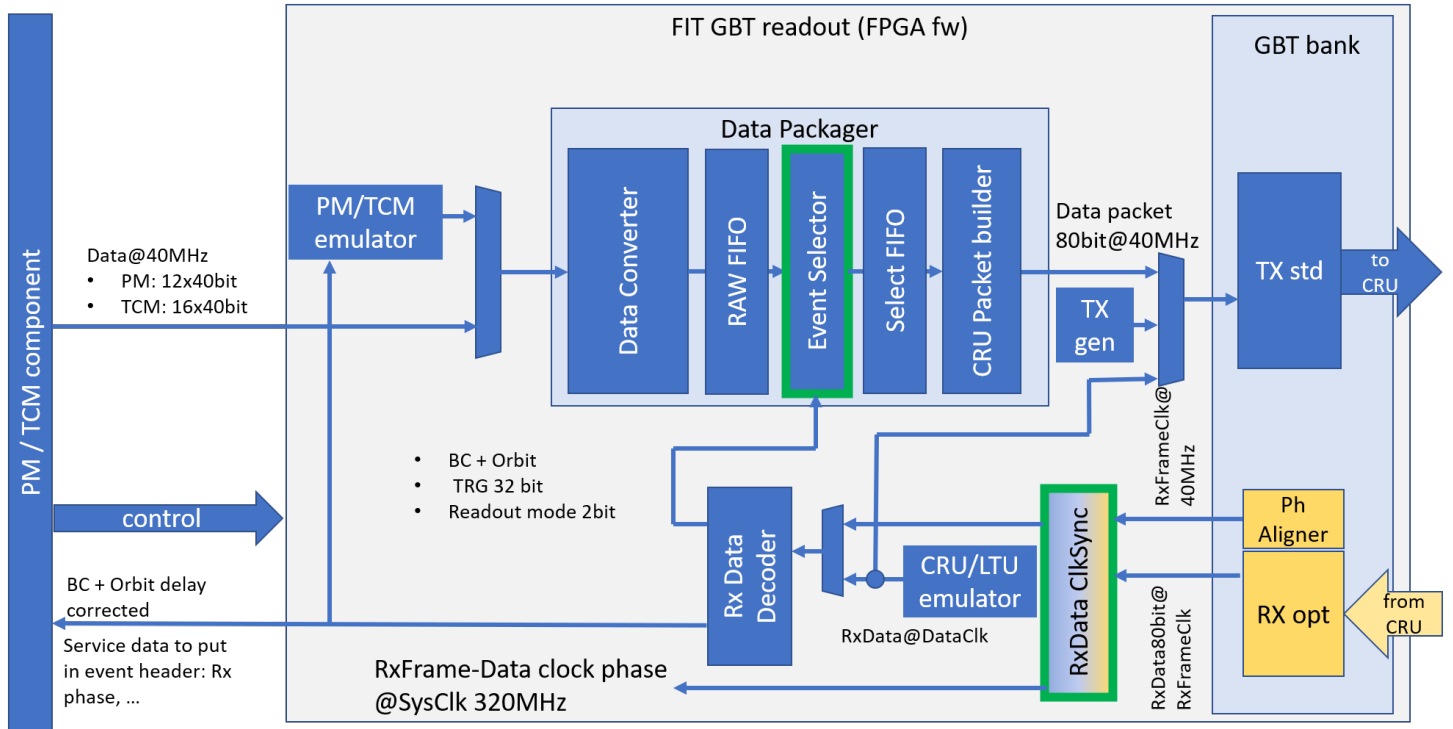


Рис. 1: Функциональная схема FIT readout unit

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 клок 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 синхронизацией с внутренним клок 40МГц, фаза между клокками измеряется по клок 320МГц. После синхронизации данные передаются в модуль 'RX Data Decoder' где принимаются команды запуска и остановки рана и данные о ВС и orbit. Данные ВС 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.

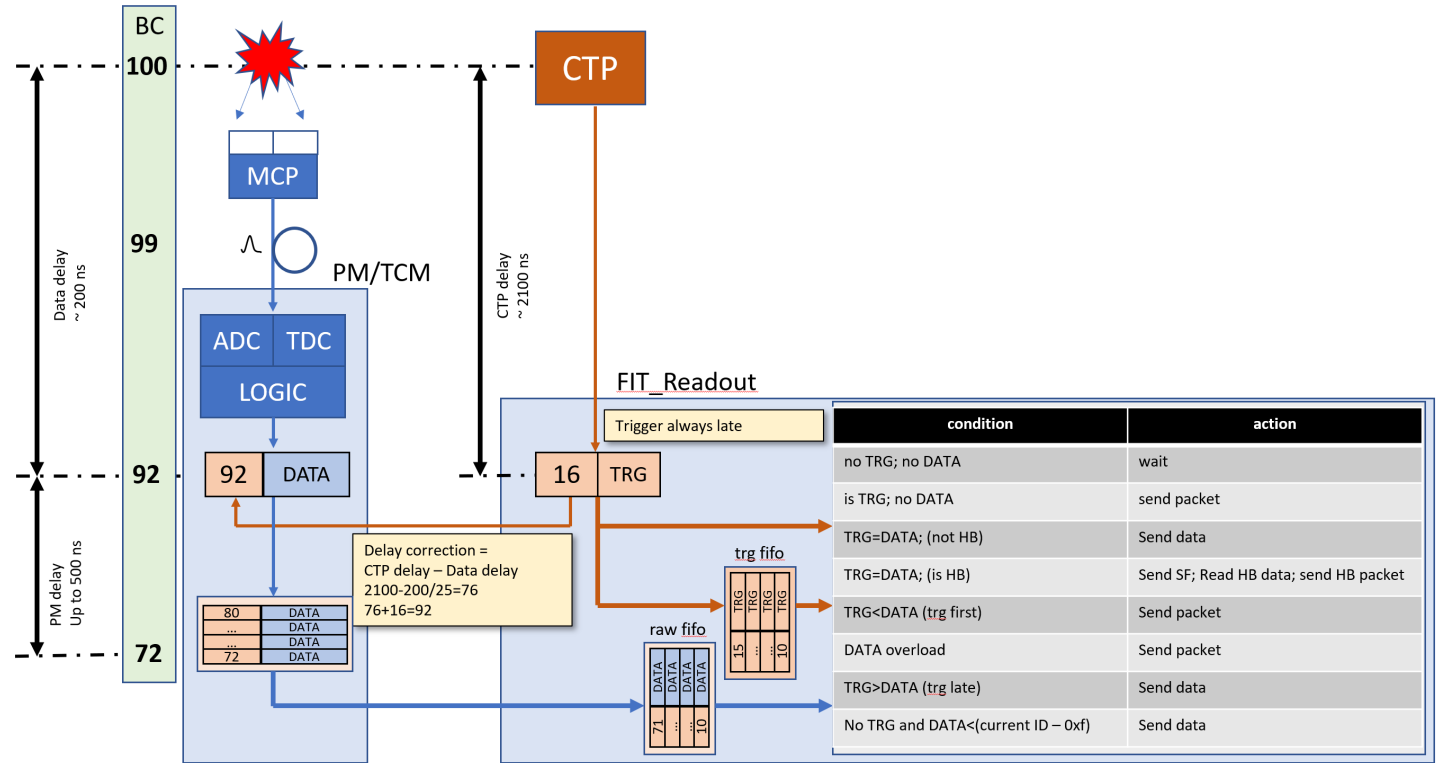


Рис. 2: Схема временной коррекции ВС и отбора событий по триггеру

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

Триггерная информация, поступившая от CTP относится к поступившему вместе с ней номеру 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 CTP, 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 PM может сформировать до 6 слов плюс одно слово заголовки события, модуль TCM 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 соответствует значению счетчика GBT слов при котором закрывается RDH и должно быть равно 0x1F4 (500). $500 + 7(\text{максимальный размер события}) + 5(\text{RDH заголовков, трейлер}) = 512$. 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 kloku 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'. При запущенном режиме вычитывания модуль СТР (и соответственно СТР emulator) отправляет триггера соответствующие режиму вычитывания (idle, continuous, trigger - статус режима вычитывания CRU) каждый орбит вместе с триггером НВ. При получении такого триггера, FIT readout запускает передачу данных так же как и при получении команды запуска вычитывания. Это необходимо для автоматического продолжения режима вычитывания после сбоя или принудительной остановки. Последнее значение статуса режима вычитывания CRU доступно по регистру статуса 'CRU 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'. При переключении регистра 'Trigger generator' в состояние 'TX generator', сформированные данные будут отправлены еще и на выход GBT что позволяет использовать любой модуль как внешний эмулятор СТР. 'СТР emulator' выполняет следующие функции:

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

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

Генерация триггеров запуска и остановки процедуры вычитывания происходит при переводе регистра 'readout command' из состояния 'idle' в состоянии 'continuous' или 'trigger' и наоборот. При этом отправляется соответствующий триггер управления режимом вычитывания. Триггер команды отправляется в момент начала орбита вместе с триггером 'НВ' в соответствии с требованиями ALICE DAQ. После смены режима вычитывания начинают отправляться триггера статуса режима вычитывания CRU.

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/TSM 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 ('НВ' trigger) на величину регистра 'trigger start offset' задающим число отсчетов 'event id'. Если частота 'pattern' равна величине орбита (регистр 'trigger bunch frequency' равен 0xDEC) то сдвиг паттерна относительно 'НВ trigger' будет постоянен.

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

In FTM if Data generator is on, Trigger not used

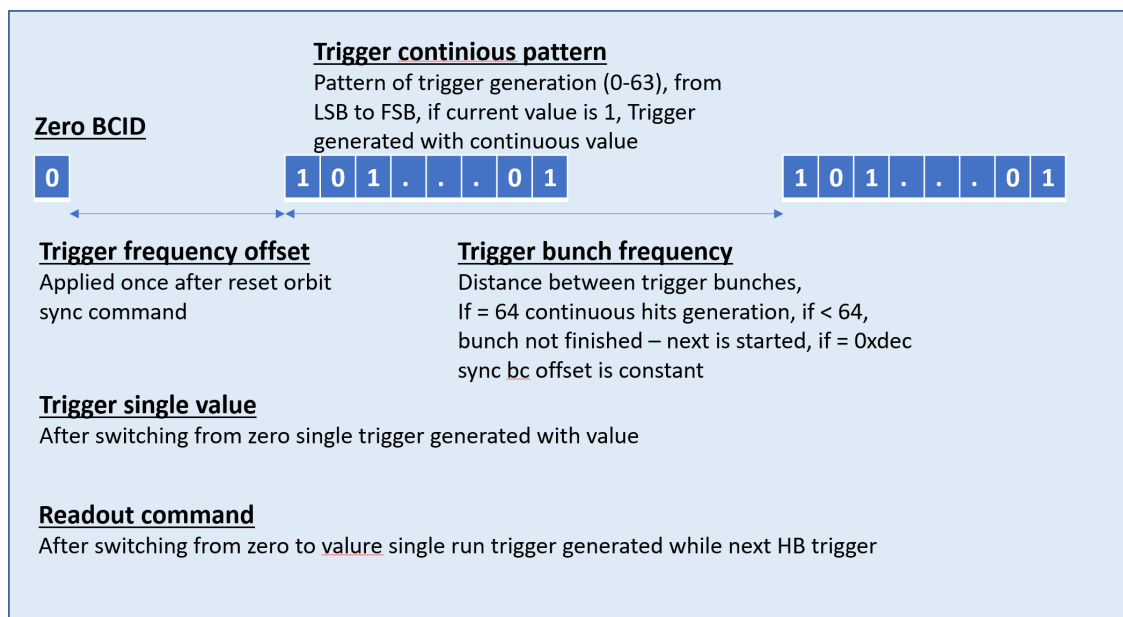


Рис. 3: Схема генерации последовательности триггеров модулем 'CTP emulator'

Так, на пример, если регистр 'trigger bunch frequency' равен 0xDED, регистр 'trigger start offset' равен 0x5 то в момент старта генератора или при переходе бита 'reset start 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 будут 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. Данный модуль различен для PM/TCM проеков, вариант для TCM содержит FIFO 'tcm data 160to80bit fifo' для разделения 160 битных слов на 80 битные.

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

этимися клоками измеряется по клоку 'system clk' 320 МГц и может принимать значения от 0 до 7. При нормальном режиме работы фаза не должна меняться больше чем на ± 1 отсчета от значения при начале синхронизации, если это происходит значение бита 'RX phase error' регистра 'GBT status' принимает значение 0x1 до сброса командой 'reset RX phase error' регистра 'Reset control'

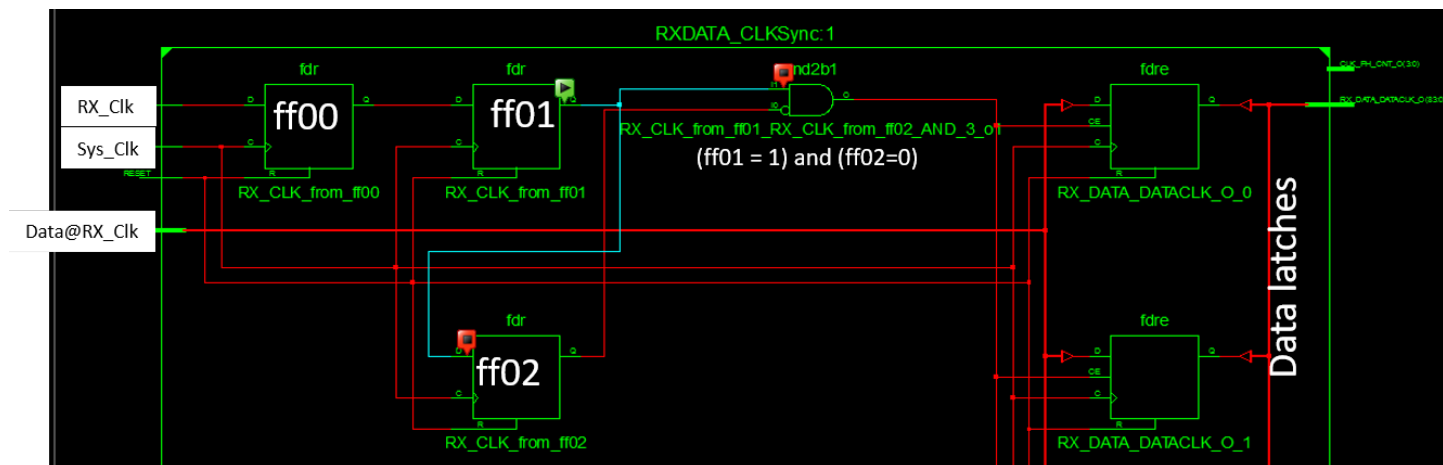


Рис. 4: Схема перехода между асинхронными клоками от RXframeClk 40МГц от CRU и внутренним клоком Data clk 40 МГц

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 start offset' можно задавать последовательность, частоту, и смещение относительно орбита. Алгоритм генерации данных модуля схожа с генерацией непрерывного триггера в модуле модуле 'СТР emulator'. Отличие состоит в том, что при генерации триггера 'pattern' состоит из 64 однобитных значений определяющих наличие триггера на данном шаге. В модуле 'PM/TCM emulator' 'pattern' задается восемью значениями по 4 бита (0-15). Каждое значение определяет количество слов в пакете для текущего события. В случае если значение превышает 0x6, то следующие события не генерируются пока не закончится передача текущего слова. Так на пример если значение регистра 'Data bunch pattern' равно 0x111D21 то последовательность пакетов будет следующая: пакет с одним словом, с двумя, с 13 (на шагах 4 и 5 пакеты генерироваться не будут), с одним словом.

Регистр 'trigger response mask' позволяет определить набор триггеров, при получении которых будет вырабатываться данные генератора с количество слов для первой позиции паттерна (если выполняется непрерывная генерация данных, то количество слов будет соответствовать нулевой позиции 'pattern' а не текущему шагу). Следует отметить что номер события 'event id' пакета будет больше чем у триггера из-за задержки алгоритма.

Бит 'reset start offset' в регистре 'Reset control' перезапускает генерацию данных в модулях 'PM/TCM emulator' и 'СТР emulator' для их синхронизации.

4 Описание работы модуля FTM

Модуль FTM предназначен для тестирования модулей PM без TCM, СТР, LTU, CRU. Модуль FTM выполняет следующие функции:

- Генерация клоков в качестве симуляции модуля TCM
- Конфигурирование модуля PM и получение претриггерных данных по HDMI.
- Генерация импульсов для запуска лазера
- Генерация триггеров в качестве эмуляции модуля СТР/LTU

- Передача данных от модулей PM/TCM полученных по GBT протоколу в PC по ethernet соединению (эмуляция CRU)
- Работа с FIT readout unit (разработка и тестирование в модуля)

Функциональная схема представлена на рис 5. FTM содержит FIT readout unit (FRU - PM version) в котором модуль 'CTP emulator' используется для передачи сгенерированных триггеров по протоколу GBT в модули PM/TCM, для этого необходимо перевести регистр 'Trigger generator' в состояние 'TX generator'. Все данные полученные по GBT RX передаются в модуль FRU для работы с ним и в модуль 'IPbus sender' для отправки в PC по ethernet. Данные полученные по HDMI с PM модуля также передаются в 'IPbus sender' и так же передаются в PC. Для запуска лазера производится сравнение полученного триггера (от эмулятора или от входа GBT) с регистром 'Trigger response mask'. Генерация импульса для лазера может происходить от служебных триггеров (HB, SOT ...), от непрерывной генерации триггера, от единичного триггера. (см. описание модуля CTP emulator)

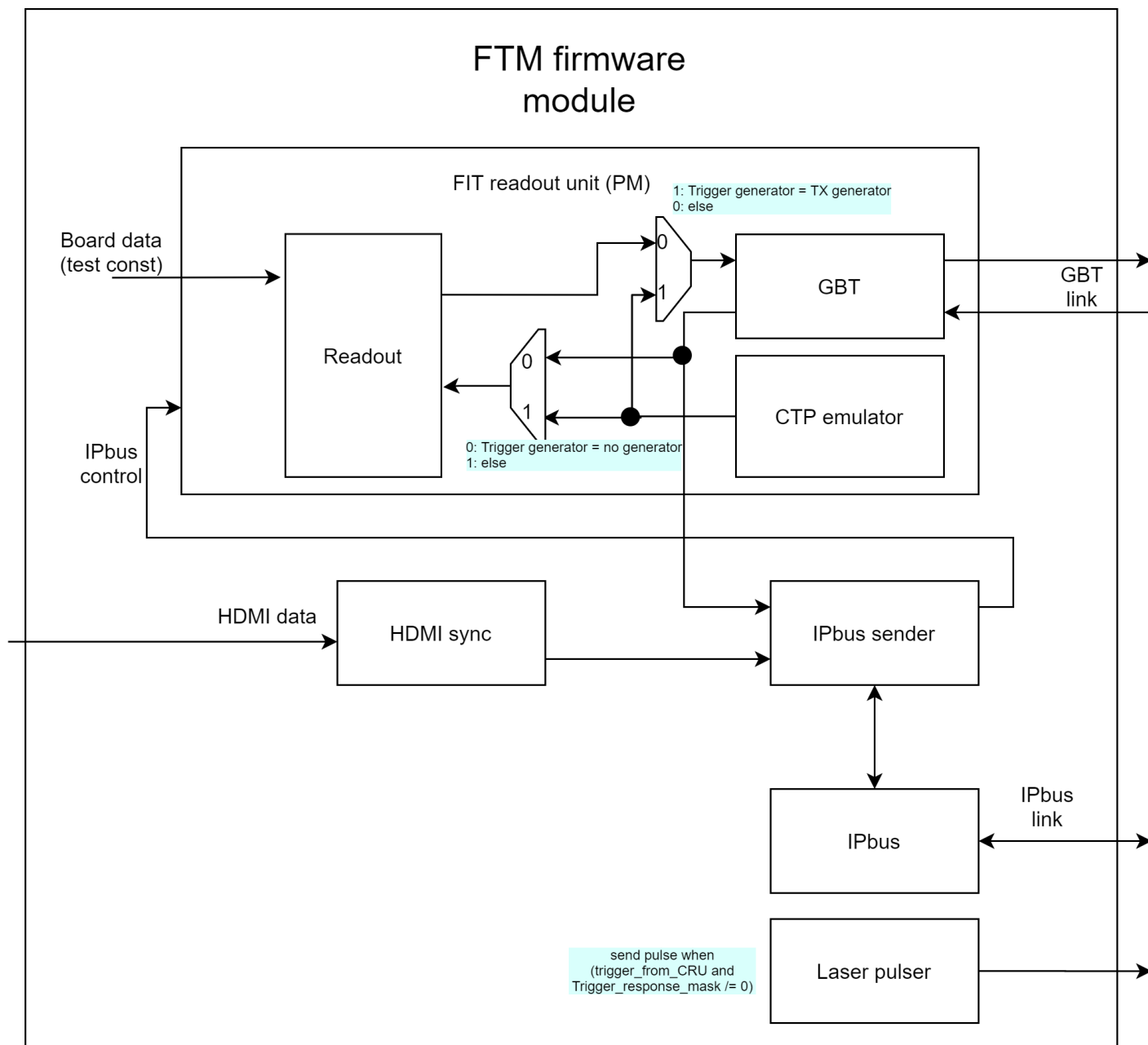


Рис. 5: Функциональная схема модуля FTM

5 Список ошибок и процедура их обработки

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

6 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 РМ/ТСМ. Клок 40 МГц используется для работы с данными GBT, клок 320 МГц используется для обработки данных.

Вторая группа клоков содержит клок '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 клок 'RxFrameClk' не используется.

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

7 Описание формата данных детектора FIT

Структура пакета передаваемых детектором по GBT приведена на рисунке 6. 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>

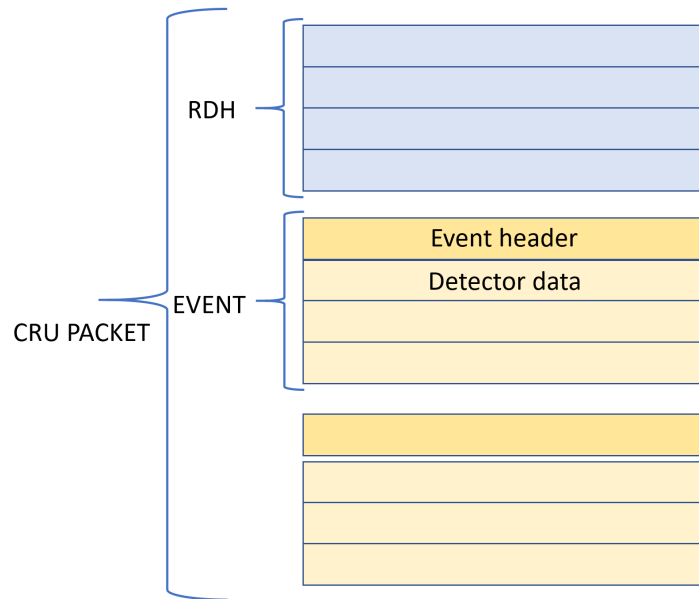


Рис. 6: Схема RDH пакета

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 7. 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 8. Поскольку РМ имеет 12 каналов, размер пакета данных события составляет от двух до 7 слов GBT. Формат данных для ТСМ представлен в таблице 9 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. Актуальная версия формата данных доступна по ссылке.

<https://drive.google.com/file/d/1LxKEuedLqxytCRw072ZqJ2vj7upP0ZgV>

GBT word		80 bit																	
CPU words		1																	
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																	
	data word 0	channel ID (0, 0, 0) flags (8bit) charge (13bit) ± time (12bit) ± channel ID (0, 0, 0) flags (8bit) charge (13bit) ± time (12bit) ±																	
	data word ..	channel data or empty (channel ID = 0)																	
	data word n	channel data																	
PM trigger	header	0xE words: 1.9																	
	word 0	0 TimeC (9bit) ± 0 TimeA (9bit) ± Amplitude (17bit) ± BC ID (12bit, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) NchanC (7bit) 0 NchanA (7bit) 0 0 0 0 V C SC or C or A																	
TCM event	word 1	PM_A2 trigger data (16 LSB)																	
	word 2	PM_A4 trigger data (32bit)																	
	word 3	PM_A7 trigger data (16 LSB)																	
	word 4	PM_A9 trigger data (32bit)																	
	word 5	PM_C2 trigger data (16 LSB)																	
	word 6	PM_C4 trigger data (32bit)																	
	word 7	PM_C7 trigger data (16 LSB)																	
	word 8	PM_C9 trigger data (32bit)																	

Рис. 7: Формат данных детектора

bit	name	bits	min	max	description
0	..	12	-2048	2047	Time with sign bit. (Relative to IP center time value). $\approx 13\text{ps}$ per unit
11	..	13	-100	4095	Charge with sign bit (base line/range corrected), a.u.
24	..	1	0	1	ADC number (0 1)
25	double event	1	0	1	Double event (two CFD pulses during one BC period)
26	time not valid	1	0	1	Time information is not valid
27	ADC in gate	1	0	1	CFD pulse time is inside ADC gate
28	Time information too late	1	0	1	Time information too late (TDC outputs data too late to include in trigger)
29	Amplitude too high	1	0	1	Amplitude too high (CFD saturated, time may be incorrect)
30	Event included in TVDC trigger	1	0	1	Event included in TVDC trigger
31	Time information lost	1	0	1	Time information lost (TDC data did not arrive for any reason)
32		3			
33					
34					
35					
36	Channel ID (1-12)	4	1	12	0x0 means no data
37					
38					
39					

Рис. 8: Формат данных модуля РМ

bit	name	bits	signed	min	max	description
0	ORA	1		0	1	OrA trigger
1	ORC	1		0	1	OrC trigger
2	SC	1		0	1	SemiCentral trigger
3	C	1		0	1	Central trigger
4	V	1		0	1	Vertex trigger
5		3				reserved
6						
7						
8	... NchanA	7		0	127	Number of A-side active channels
14		1				reserved
15						
16	... NchanC	7		0	127	Number of C-side active channels
22		1				reserved
23						
24	... AmplA	18		0	262143	Total charge for A-side, a.u. (1 unit = 8 PM charge units)
41						
42	... AmplC	18		0	262143	Total charge for C-side, a.u. (1 unit = 8 PM charge units)
59						
60	... TimeA	9	\pm	-256	255	Average time for A-side. $\approx 13\text{ps}$ per unit
68		1				reserved
69						
70	... TimeC	9	\pm	-256	255	Average time for C-side. $\approx 13\text{ps}$ per unit
78		1				reserved
79						

Рис. 9: Формат данных модуля ТСМ

8 Используемые компоненты блоков памяти

Название	ширина запи- си, бит	ширина запи- си, бит	размер	клок за- писи	клок чте- ния	использует в модуле	регистр занято- сти	назначение
Raw fifo	80	80	4096	system320	system320	Data packager	Raw FIFO count	Принимает сформированные данные РМ/ТСМ для дальнейшего отбора модулем 'event selector'
Selector fifo	80	80	4096	system320	data40	Data packager	Selector FIFO count	Принимает отобранные данные для формирования RDH
Trg fifo	76	76	512	data40	system320	event selector	нет	Принимает триггеры для отбора событий
Cntpck fifo	160	160	128	system320	data40	event selector	нет	содержит управляющие слова для формирования RDH
data320to40	80	80	4096	system320	data40	TX data generator	нет	принимает данные модуля из raw data fifo для отправки по GBT в режиме 'is readout bypass mode'
hdmi fifo	80	80	4096	system320	RXdata40	IPBUS sender (FTM)	нет	Переход между клоками sys320 и RXdata40 для совмещения данных HDMI с данными принятыми по GBT и дальнейшей отправки в IPbus
ipbus data fifo	96	192	16384	RXdata40	IPbusData	IPBUS sender (FTM)	FTMIPbus fifo count	Буфер на отправку данных от GBT+HDMI по IPbus
tcm data 160to80bit fifo	160	80	4096	system320	system320	Data Converter	нет	Разбиение 160 битных слов от ТСМ на 80 битные для передачи

Таблица 1: Список используемых блоков памяти

9 Регистры управления и статуса

9.1 Описание регистров управления

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 start offset	CTP/LTU emulator	нет
Data start 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

Таблица 2: Список 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 start offset				Data start 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							

Таблица 3: Таблица адресов регистров управления.

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

Таблица 4: Описание битов регистра 'Reset control'

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

Таблица 5: Описание состояний регистра 'Data generator'

Значение регистра	режим работы
0	No generator
1	Continuous generator
2	TX generator

Таблица 6: Описание состояний регистра 'Trigger generator'

Значение регистра	режим работы
0	Idle
1	Continuous
2	Trigger

Таблица 7: Описание состояний регистра 'Readout command'

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

Таблица 8: Описание битов регистра 'Readout mode'

9.2 Описание 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
FTMIPbus FIFO count	FIT TESTMODULE IPBUS sender (FTM)	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

Таблица 9: Список регистров статуса

addr	31 .. 28	27 .. 24	23 .. 20	19 .. 16	15 .. 12	11 .. 8	7 .. 4	3 .. 0
E8	CRU readout mode	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	FTM IPbus fifo count				Readout rate			

Таблица 10: Таблица адресов регистров статуса.

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

Таблица 11: Описание битов регистра 'GBT status'

Значение регистра	режим работы
0	Idle
1	Continuous
2	Triggered

Таблица 12: Описание состояний регистра 'Readout mode', 'CRU readout mode'

Значение регистра	режим работы
0	Start
1	Sync
2	Lost

Таблица 13: Описание состояний регистра 'BCID sync mode'

9.3 Инициализация регистров при запуске электроники

При запуске электроники или после сигнала reset необходимо установить значения по умолчанию в регистры управления что соответствует штатной работе пакета.

Название регистра	значение	комментарий
Readout mode	0x5	ответ НВ включен; редаут выключен
Readout command	0x0	
Reset control	0x0	
Trigger generator	0x0	
Data generator	0x0	
Trigger response mask	0x0	
Data bunch pattern	0x0	
Trigger single value	0x0	
Trigger continuous pattern	0x0	
Trigger continuous value	0x0	
Trigger bunch frequency	0x0	
Data bunch frequency	0x0	
Trigger start offset	0x0	
Data start offset	0x0	
FEE ID		согласно карте
PAR	0x0	
Max RDH payload	0x1F4	см. модуль Event Selector; может быть увеличен на 1 после отключения RDH трей
Detector field	0x0	нет
Trigger compare delay	0xf	будет уточнено после тестов на P2
BCID delay	0x20	будет уточнено после тестов на P2, индивидуально для каждой платы
Data select trigger mask	0x10	отбор по физическому триггеру

Таблица 14: Значения по умолчанию регистров управления для штатной работы

10 Описание пакета симуляции

Пакет симуляции выполнен на языке python и расположен в папке `alice-fit-fpga/software/readout-sim`. Симуляция позволяет выполнить проверку работы модуля FIT-readout-unit при симуляции vhdl кода пакетом Vivado (в настоящее время выполнено только логическая симуляция). Последовательность проверки следующая:

- Запись файла `readout-sim/simulation_inputs/simple_sig_inputs.txt` который содержит конфигурацию всех регистров для каждого цикла @40МГц. Файл записывается в первой части исполняемого файла `testbench.py` и содержит команды `firmware`-генераторов для запуска и остановки записи данных в режимах которые необходимо проверить. На данный момент это `triggered`, `continious`, `continious-load` (несколько пакетов на орбит) раны.
- Запуск симуляции в пакете Vivado. Симуляция использует файл `simple_sig_inputs.txt` для конфигурации пакета редаута. Как результат симуляции записываются файлы `readout-sim/simulation_outputs/readout_gbt_info_output.txt`, `readout_gbt_output.txt`, `readout_status_reg_output.txt`. В файл `readout_status_reg_output.txt` записывается расширенный регистр статуса (добавлены триггер, скорректированный ORBC и размер пакета в `DataGenerator`) для каждого цикла. В файл `readout_gbt_output.txt` записываются все ненулевые слова с GBT, а в `readout_gbt_info_output.txt` номера циклов в которых записаны слова GBT.
- Запуск проверки выходных файлов с выводом ошибок найденных в пакетах данных.

Запуск проверки данных запускается в исполняемом файле `testbehch.py`. В начале создается элемент класса `simulation_data_class`. В данном классе указываются пути для всех выходных файлов и файлы связываются с переменными типа `list` для чтения. При создании класса `simulation_data_class` читается входной файл конфигурации `simple_sig_inputs.txt` и производится поиск команд на запуск и остановки ранов. Для каждого рана создается элемент класса `run_sim_data_class` где читается из файла конфигурации вся необходимая информация о ране (начало и конец рана в файлах, тип рана и его конфигурация) и проверяется корректность регистров конфигурации во время рана (постоянность конфигурации, отсутствие ранов некоторое время до старта и после остановки). Так же при создании класса создается список с данными и триггерами генерированные в тестовых генераторах `firmware`.

После создания класса `simulation_data_class` создается элемент класса `run_testbench_class` для каждого рана, классу передается класс `simulation_data_class` и номер рана который необходимо проверить. При создании класса `run_testbench_class` производится чтение пакетов RDH и пакетов детектора из файла GBT с проверкой ошибок в формате данных. После считывания, пакеты проверяются к корректность (наличие триггеров старта и остановки рана в данных, счетчик пакетов RDH, `stop bit`, `orbit` в заголовке RDH и событиях, наличие всех сгенерированных данных с учетом режима вычитывания `continious/trigger`). Статус выполнения проверки выводится логом в терминал и файл.

11 Ссылки на рабочие документы и программное обеспечение по проекту FIT

Полный список актуальных ссылок на документы находится на странице ALICE FIT Twiki.

- ALICE FIT Twiki <https://twiki.cern.ch/twiki/bin/view/AliceFIT/WebHome>
- Документация по FIT readout unit (этот документ) https://github.com/dfinogee/FIT-readout-manual/raw/master/FIT_readout_manual.pdf