

МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
ИНСТИТУТ ФИЗИКИ ВЫСОКИХ ЭНЕРГИЙ

КРАВЧЕНКО СЕРГЕЙ ВАЛЕРЬЕВИЧ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
НА СТЕПЕНЬ БАКАЛАВРА

Время-цифровой преобразователь.

Факультет Общей и Прикладной Физики
Кафедра Физики Высоких Энергий

Научный руководитель:
ИВАШИН А. В.

МОСКВА
28 июня 2014

Оглавление

1	Введение	3
2	Требуемые параметры ВЦП	6
3	Описание модуля	8
3.1	Блок-схема	8
3.2	Общее описание	9
3.3	Процесс регистрации	9
4	Прошивка ВЦП	11
4.1	Блок-схема	11
4.2	Описание логической схемы	11
4.3	Протоколы обмена данными	14
4.3.1	Протокол считывания буферной памяти	14
4.3.2	Протокол настройки временных задержек	15
4.3.3	Протокол считывания регистра временного штампа конца спилла	16
5	Формат данных	17
5.1	Формат данных события	17
5.2	Формат данных спилла	17
6	Тестирование прошивки	19
7	Заключение	22

Глава 1

Введение

Эксперимент ВЕС в ИФВЭ (Россия г.Протвино) в настоящее время ведет исследования в области спектроскопии легких мезонов на пучке π^- мезонов с импульсом 28 ГэВ (вторичный пучок от ускорителя У-70). Для регистрации различных вспомогательных сигналов от триггерной подсистемы (триггерных признаков), а также сигналов от счётчиков охранной системы, применяется небольшое число стробируемых регистров. Эти модули уже морально устарели, у них фиксированная длительность строба и отсутствует цифровая задержка для входных сигналов. В связи с происходящей в настоящее время модернизацией системы сбора данных эксперимента ВЕС^{[1],[2]}, было принято решение заменить их на многоканальные время-цифровые преобразователи (ВЦП) с регулируемой цифровой задержкой входных сигналов и изменяемой шириной окна регистрации. Это позволит on-line контролировать попадание сигнала в окно, а так же настраивать его ширину для разных подсистем. Поскольку необходимо небольшое число таких модулей, то было принято решение сделать их на базе существующих 48-канальных модулей ВЦП для дрейфовых трубок TDC-48DT, заменив прошивку для ПЛИС. Создание новой прошивки для ПЛИС и является целью данной работы.

Принцип ВЦП может быть различным.

Первое поколение

Традиционный подход время-цифрового преобразования основывается на преобразовании временного интервала в напряжение (см. рисунок 1.1).

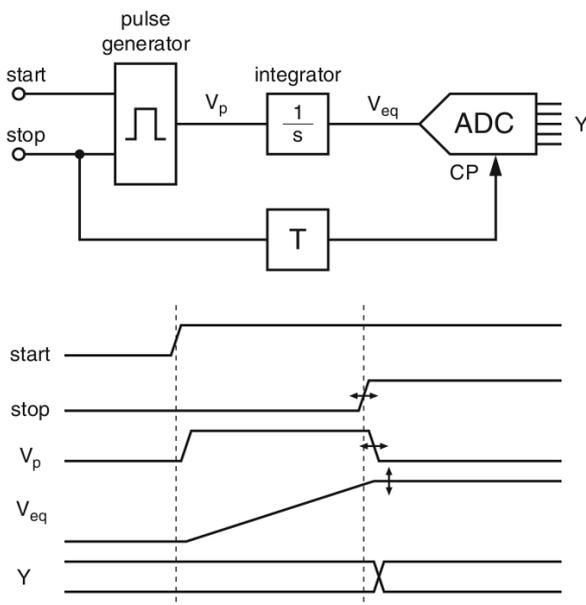


Рис. 1.1: Аналоговая схема ВЦП

Затем это напряжение оцифровывается с помощью обычных АЦП. Сигналы **START** и **STOP** используются для формирования импульса с шириной, соответствующей временному интервалу, который необходимо измерить. Аналоговый интегратор преобразует этот импульс в напряжение, которое затем поступает на АЦП. Динамический диапазон такого преобразования определяется следующим выражением

$$DR = 2^N \cdot T_{LSB},$$

где T_{LSB} - минимальный временной интервал, который может быть измерен. Как следствие возникает компромисс между максимально возможным интервалом измерения и временного разре-

пения. Несмотря на то, что данный подход довольно прост, он не избавлен от серьезных недостатков. Все блоки ВЦП такие как АЦП, интеграторы и т.д. должны удовлетворять требованию линейности. Интегратор представляет себя источник постоянного тока, подключенный к емкости в течении измеряемого интервала. В связи с конечным выходным сопротивлением источника тока линейность слаба. Решением этой проблемы может служить активный RC-интегратор. Однако конечная полоса пропускания операционного усилителя ограничивает скорость, и как следствие T_{LSB} . Для абсолютного измерения временного интервала необходимо точное знание тока и емкости, что невозможно без калибровки. Для решения этой проблемы возможно построение более сложных схем на основе данного аналогового подхода [3].

Второе поколение

На смену первому поколению аналоговых схем ВЦП пришло второе поколение - полностью цифровых схемы время-цифрового преобразования. Самый простой подход к реализации такого ВЦП основывается на подсчете периодов опорного такого сигнала во время измеряемого временного интервала. Он определяется сигналами **START** и **STOP**, полностью асинхронными к опорному тактовому сигналу (см. рисунок 1.2).

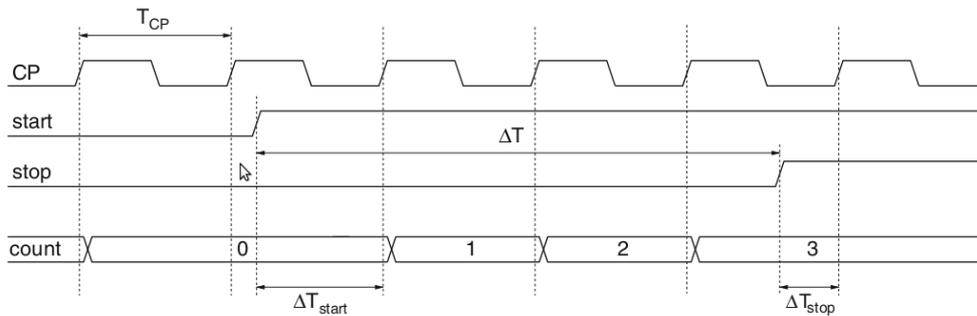


Рис. 1.2: Схема ВЦП, основанная на счетчике тактовых импульсов

Измеренное значение интервала может быть вычислено как

$$\Delta T = N \cdot T_{CP} + (T_{CP} - \Delta T_{stop}) + (T_{CP} - \Delta T_{start}),$$

где N - значение счетчика тактовых сигналов; T_{CP} - период тактового сигнала; ΔT_{stop} и ΔT_{start} временной интервал между передним фронтом сигнала **STOP** и **START** соответственно и следующим растущим фронтом тактового сигнала. Ошибка такого измерения равна $2 \cdot T_{CP}$. Временное разрешение может быть увеличено, если увеличить частоту тактового сигнала [3]. Другой способ увеличения временного разрешения заключается в делении периода тактового сигнала на более мелкие промежутки.

Линии задержки

Чтобы увеличить временное разрешение за пределы максимально возможной частоты тактового сигнала, из опорного тактового сигнала необходимо сформировать несколько вспомогательных сигналов, которые разделят период опорного сигнала на несколько частей. Как правило, это делается с помощью фазового сдвига опорного тактового сигнала. Кольцевой генератор может генерировать, например, k равномерно распределенных вспомогательных тактовых сигналов. Измерение состоит из грубого измерения временного интервала с помощью опорного тактового сигнала и более точного, с помощью k вспомогательных тактовых сигналов, сдвинутых по фазе друг относительно друга (см. рисунок 1.3). При этом точность измерения будет равна $\frac{T_{CP}}{k}$.

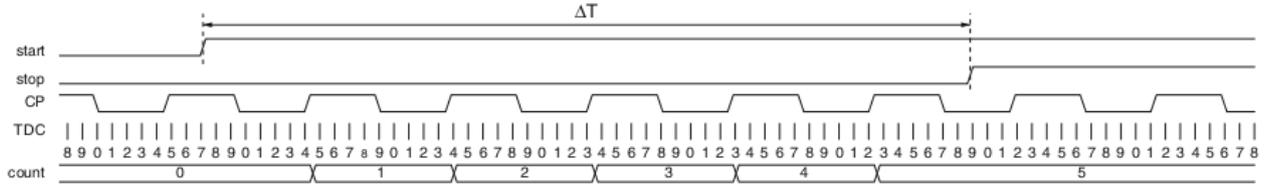


Рис. 1.3: Схема ВЦП с использованием вспомогательных тактовых сигналов, сдвинутых по фазе

Еще большее временное разрешение достигается с помощью цепочки цифровых задерживающих элементов. Принцип работы изображен на рисунке 1.4. Стартовый сигнал задерживается по линии задержки, и в момент прихода сигнала **STOP** состояние линии задержки замораживается. При этом, если на задерживающем элементе уже прошел сигнал **START**, то на данном этапе присутствует высокий уровень, если стартовый сигнал не дошел до задерживающего элемента в цепочке, он имеет низкий уровень. Таким образом можно узнать, как далеко зашёл стартовый сигнал и получается код, соответствующий измеряемому интервалу. Число элементов в цепочке, имеющих высокий уровень соотносится с измеряемым интервалом следующим образом

$$\Delta T = N \cdot T_{LSB} + \varepsilon,$$

где ε - ошибка квантования, которая возникает в задерживающем элементе (был принят стартовый сигнал или нет).

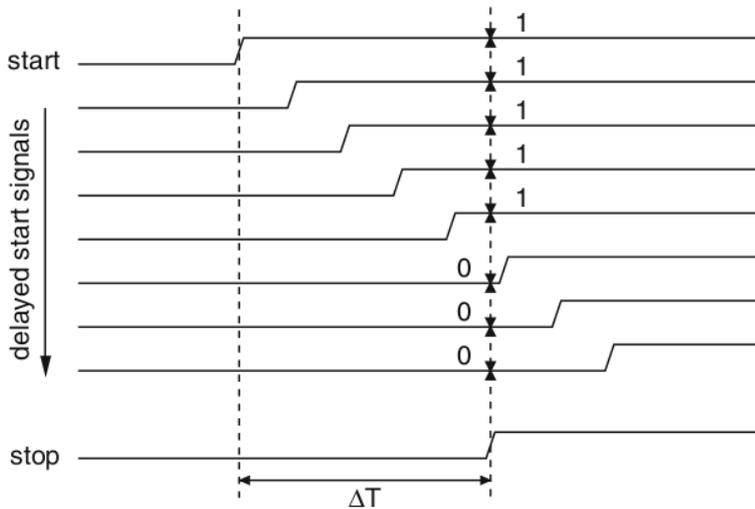


Рис. 1.4: Схема ВЦП с использованием задерживающей линии

В данной работе описывается реализация схемы время-цифрового преобразователя с использованием двух противофазных тактовых сигналов. Она была выбрана в силу того, что полностью соответствует невысоким требованиям, предъявляемым к ВЦП, и того, что имеет относительно несложную реализацию.

Глава 2

Требуемые параметры ВЦП

Сброс частиц на установку ВЕС производится в циклическом режиме. Длительность одного цикла составляет 9,5 секунд, из которых сам сброс составляет 1-2 секунды. Во время сброса необходимо регистрировать сигналы от детекторов экспериментальной установки. Подобная схема работы позволяет, при построении системы сбора данных, во время сброса пучка записывать данные с большой скоростью в локальную память, а в период между сбросами передавать данные используя более медленные линии связи. Модуль TDC-48DT, выбранный в качестве базового для создания ВЦП, построен в соответствии с этим принципом: он имеет локальную буферную память 32 Мбайт с пропускной способностью 50 Мбайт/с и сравнительно медленный интерфейс Fast Ethernet с пропускной способностью 10 Мбайт/с.

Для замены стробируемых регистров в эксперименте ВЕС, ВЦП должен обладать следующими характеристиками:

- Входные сигналы в уровнях NIM.
- Временное разрешение порядка ± 5 нс.
- Возможность регулировки цифровой задержки приблизительно до 1000 нс.
- Возможность регулировки ширины окна регистрации приблизительно от 30 нс до 500 нс

Поскольку модуль TDC-48DT имеет входные сигналы в уровнях LVDS, было решено подключить его плоскими кабелями к преобразователю уровней NIM->LVDS производства ОЭА ИФВЭ.

Ниже приведена временная диаграмма (см. рисунок 2.1), описывающая процесс сброса частиц на установку и регистрацию сигналов, вызванных ими.

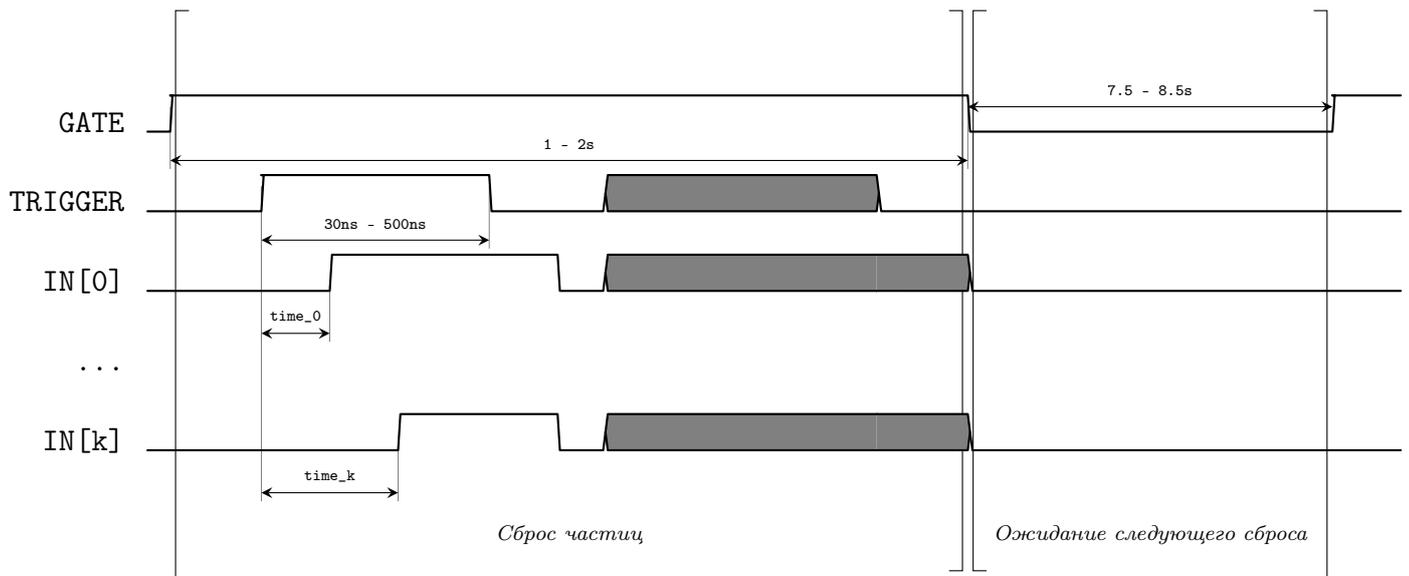


Рис. 2.1: Временная диаграмма процесса сброса частиц на установку

Глава 3

Описание модуля

Ниже, для удобства, приведено краткое описание модуля TDC-48DT, позаимствованное из [4].

3.1 Блок-схема

Модуль предназначен для регистрации и выполнения время-цифрового преобразования 48 дифференциальных цифровых сигналов.

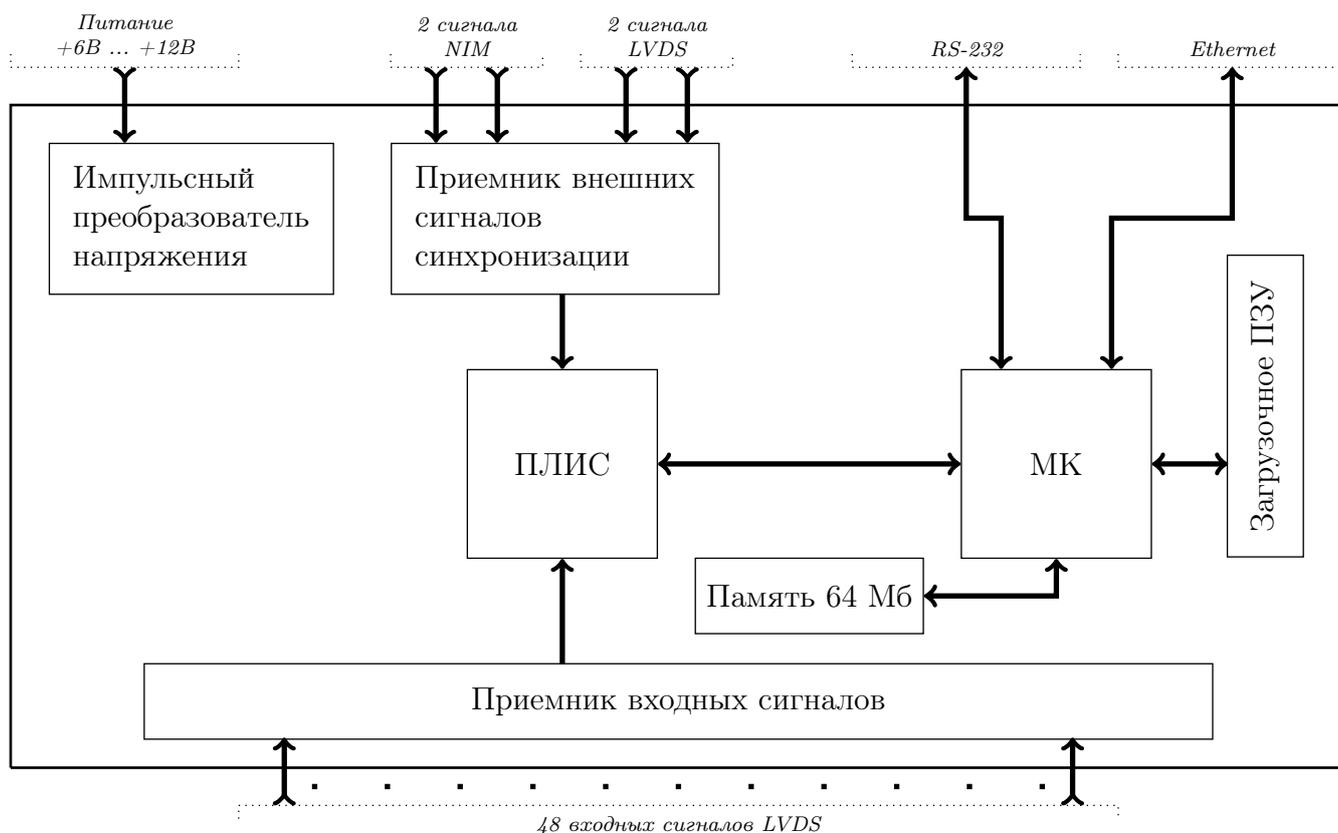


Рис. 3.1: Блок схема модуля TDC-48DT

3.2 Общее описание

Параметры

- 48 дифференциальных входов (LVDS)
- 4 сигнала управления процессом регистрации (2 - NIM, 2 - LVDS)
- 2 интерфейса для связи микроконтроллера и ПК - RS232 и Ethernet 10/100M
- Динамическая память (SDRAM) объемом 64 Мбайт

Для обработки входных сигналов используется ПЛИС Xilinx XC3S500E-5PQG208C. Входные сигналы, подлежащие обработке, подаются на модуль в уровнях LVDS на 48 LVDS-приемников (SN65LVDS348), которые преобразуют дифференциальные сигналы в однофазные в уровнях LVCMOS-3.3, и подают их на входы ПЛИС.

Кроме сигналов, подлежащих регистрации, на ПЛИС также возможно подать до четырёх внешних сигналов для управления процессом регистрации. Два из этих сигналов можно подать в уровнях NIM, а еще два в уровнях LVDS. Для приема как NIM, так и LVDS сигналов используется микросхема SN65LVDS348, которая преобразует их в однофазные сигналы в уровнях LVCMOS-3.3. Преобразованные сигналы поступают непосредственно на входы ПЛИС.

После первичной обработки сигналов в ПЛИС, информация передается в микроконтроллер (в дальнейшем «МК») по интерфейсу ПЛИС \longleftrightarrow МК. В качестве МК используется AT91 RM9200-QU-002. К МК подключена синхронная динамическая память (SDRAM) объемом 64 Мбайт, часть которой используется для накопления информации от ПЛИС, а часть для работы операционной системы Linux с набором служебных программ.

Микроконтроллер осуществляет общее управление модулем, в частности конфигурирует ПЛИС во время инициализации модуля, а также периодически осуществляет передачу накопленной в памяти информации от ПЛИС во внешний мир (систему сбора данных) по интерфейсу Ethernet. Микроконтроллер для связи с внешним миром имеет 2 интерфейса: RS-232 и Ethernet 10/100M. Первый применяется для служебных целей, в частности для конфигурации временных задержек входных сигналов, и во время нормальной работы модуля не используется. Интерфейс Ethernet является основным и единственным для управления модулем и приема данных с него во время нормальной работы.

Описание всех физических интерфейсов, а так же конфигурацию всех входов и выходов ПЛИС и МК можно найти в описании к модулю.^[4]

3.3 Процесс регистрации

Модуль работает в циклическом режиме, переключаясь между режимом регистрации и накопления данных и режимом выдачи накопленных данных через интерфейс Ethernet. Режим, в котором находится модуль, определяется внешним сигналом GATE. Во время высокого логического уровня сигнала GATE модуль находится в режиме регистрации и накопления данных; в противном случае в режиме выдачи данных по интерфейсу Ethernet.

Запуск процесса регистрации события осуществляется нарастающим фронтом внешнего сигнала START. ПЛИС формирует по переднему фронту START временное окно, во время которого происходит регистрация сигналов. Логическая схема в ПЛИС измеряет времена прихода

сигналов IN[47:0] (точнее их передних фронтов) относительно переднего фронта сигнала START, если таковые имеются. Для всех сработавших каналов измеренные значения времен, вместе с заголовком события, записываются во внутреннюю буферную память небольшого объема (16 кбайт). Заголовок события содержит временной штамп события. Процесс регистрации оканчивается после заднего фронта сигнала GATE считыванием микроконтроллером временного штампа конца спилла.

Для записи событий в ПЛИС применяется двойная буферизация с переключением буферов по мере их заполнения на лету за несколько тактов системного тактового сигнала ПЛИС с частотой 100 - 250 МГц (зависит от прошивки). После завершения операции переключения буферов, заполненный буфер готов к вычитыванию микроконтроллером. Логическая схема в ПЛИС оповещает МК о готовности буфера сигналом RQ. Микроконтроллер обязан следить за этим сигналом и успевать вычитывать буфер до заполнения второго буфера. Логическая схема в ПЛИС обязана войти в состояние ошибки, если МК не успел считать буфер до момента следующей операции переключения буферов. Вход в состояние ошибки сопровождается немедленной установкой сигнала ERR. Если установлен сигнал ERR, то соблюдение протокола считывания буферной памяти считается необязательным со стороны ПЛИС. Сигнал ERR логическая схема в ПЛИС имеет право сбросить (и обязана это сделать) только по переднему фронту сигнала GATE.

Сигнал GATE вводит МК в режим быстрого прерывания FIQ, в котором блокируется любая активность, не связанная с обслуживанием протокола считывания буферной памяти ПЛИС, на всем протяжении сигнала GATE. Во время высокого уровня GATE микроконтроллер выполняет протокол считывания буферной памяти ПЛИС и записывает данные от ПЛИС в динамическую память большого объема. По окончании GATE ПЛИС имеет право выдать сигнал RQ не позднее 5 мкс от заднего фронта GATE для обеспечения вычитывания частично заполненного буфера с несколькими последними событиями. По истечении этого времени обработчик прерывания МК выполняет протокол считывания регистра временного штампа конца спилла, сохраняет считанное значение в заголовке спилла, сохраняет в заголовке спилла состояние сигнала ERR и завершается, переключая МК в обычный режим.

В обычном режиме МК работает под управлением операционной системы Linux, в которой запущен сервер выдачи данных по интерфейсу Ethernet. Ниже приведена временная диаграмма (см. рисунок 3.2), которая отражает работу модуля в целом:

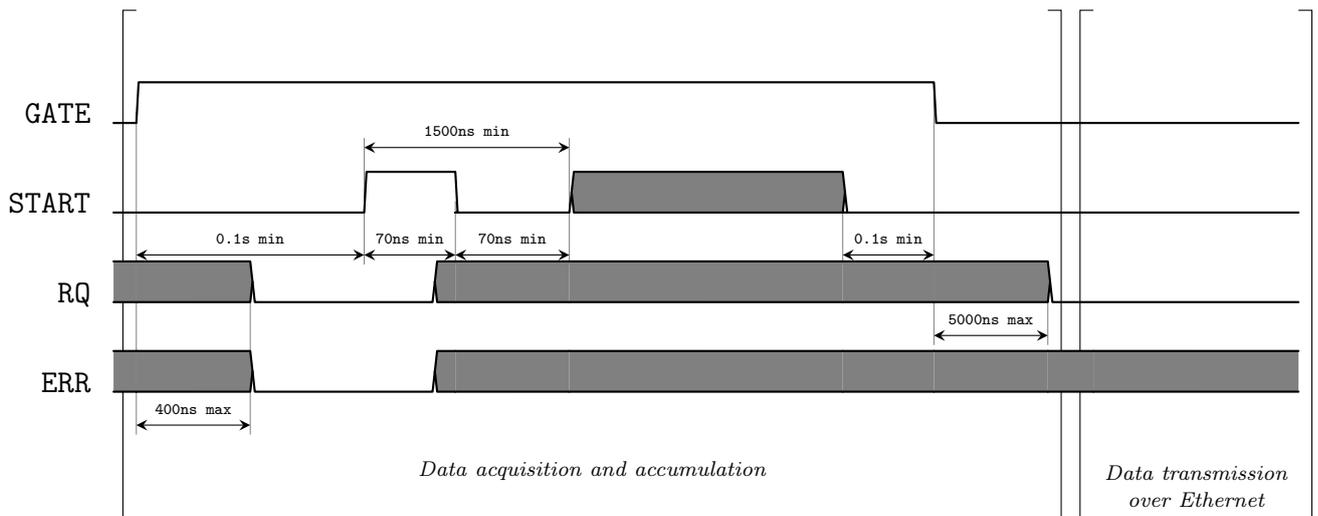


Рис. 3.2: Временная диаграмма работы ВЦП

Глава 4

Прошивка ВЦП

Прошивка написана на языке Verilog. Для синтеза схемы и последующей генерации конфигурационного файла ПЛИС использовался инструментарий Xilinx ISE 14.6. Исходный код прошивки доступен по адресу [5].

4.1 Блок-схема

Блок схема ВЦП представлена на рисунке 4.1

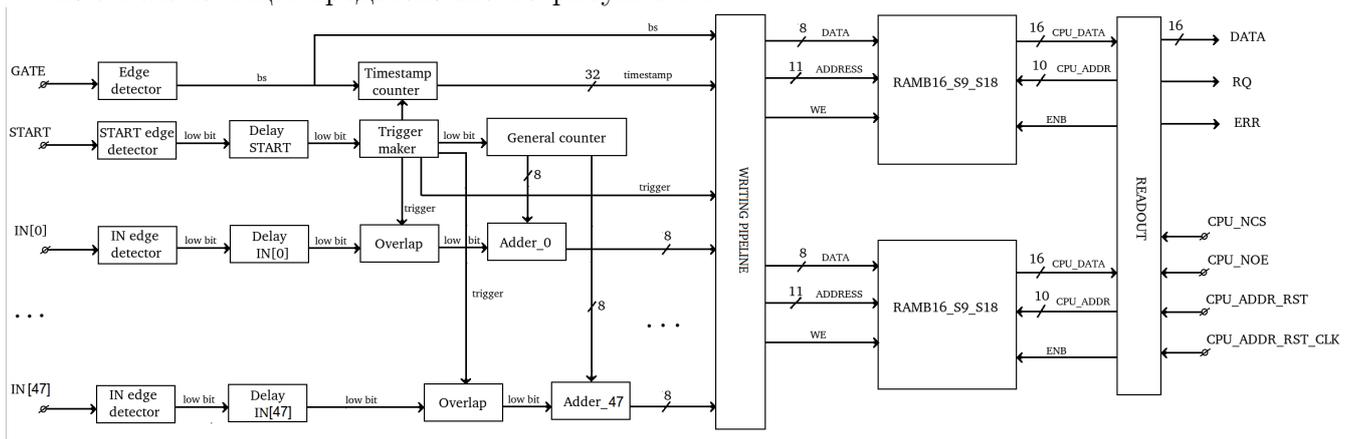


Рис. 4.1: Блок-схема ВЦП

В основе работы ВЦП лежит счетчик тактовых импульсов. По переднему фронту сигнала TRIGGER запускается основной счетчик General_Counter. В случае срабатывания одного, или нескольких входов IN значение счетчика копируется в соответствующий выходной регистр, где ожидает дальнейшей записи в буферную память.

4.2 Описание логической схемы

Регистрация сигнала "edge_detector"

Для повышения временного разрешения используются два противофазных тактовых сигнала. Сигнал, подаваемый на вход IN или START поступает в модуль edge_detector, где определяется, в каком полупериоде тактового сигнала находится передний фронт (см. рисунок 4.2), выставляется младший бит и поднимается флаг срабатывания.

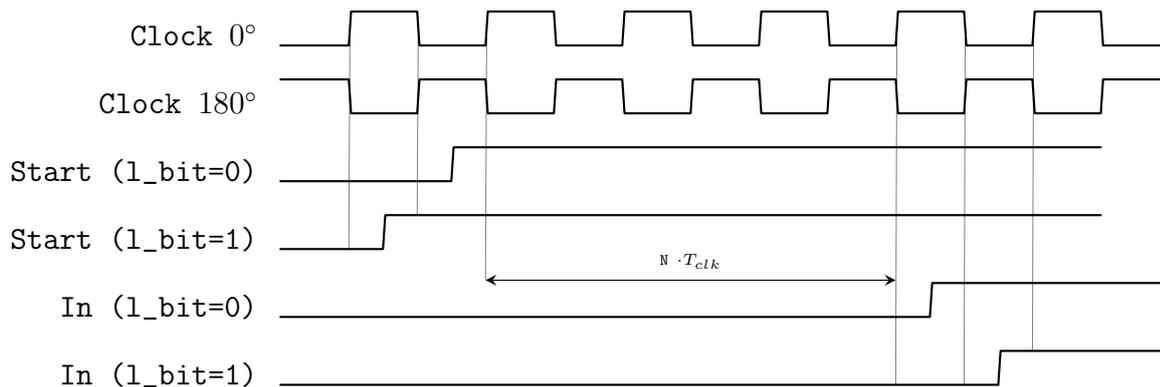


Рис. 4.2: Временная диаграмма определения младшего бита

Младший бит запоминается, а флаг поступает в модуль DELAY.

Временная задержка "Delay"

Временная задержка входящего сигнала основана на регистре сдвига SRL16^[6]. Модуль DELAY_IN и DELAY_START представляют из себя два последовательно соединенных массива из четырех SRL16 регистров сдвига (см. рисунок 4.3). Для инициализации одного массива требуется 16 бит (Delay [15:0]), которые устанавливают величину задержки. Временная задержка второго массива регистров сдвига в модуле Delay выставляется автоматически равной величине в первом массиве. Таким образом, задержка для входных сигналов варьируется в пределах [60 : 970] нс с шагом $2 \cdot T_{clk}$ нс.

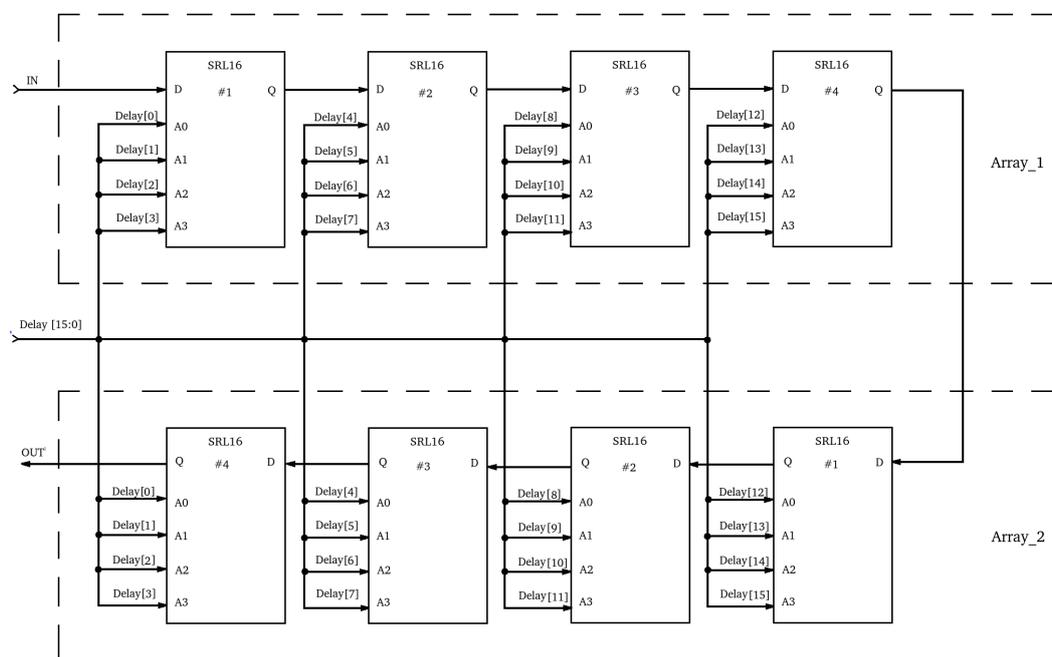


Рис. 4.3: Блок-схема модуля "Delay"

Формирование окна регистрации

После задержки в Delay_START, флаг срабатывания из START_edge_detector попадает в Trigger_maker, где формируется временное окно, в котором будет происходить регистрация

входных сигналов IN. Формирование окна проходит следующим образом: флаг срабатывания переводит сигнал TRIGGER в активное (лог. единица) состояние, и поступает на один массив из четырех сдвиговых регистров SRL16 (см. рисунок 4.3). После задержки флаг срабатывания переводит сигнал TRIGGER в пассивное состояние. Величина задержки в сдвиговом регистре конфигурируется 16 битами в пределах [30 : 480]нс и соответствует ширине окна регистрации.

Счетчики

General_counter имеет разрядность 8 бит. Внутренний счетчик имеет разрядность 7 бит и записывается в [7:1] биты. В 0-ой бит одновременно с переводом сигнала TRIGGER в активное состояние заносится младший бит, полученный в **START_edge_detector**, и запускает внутренний счетчик. Сброс происходит по заднему фронту сигнала TRIGGER.

Timestamp_counter представляет 32-битный счетчик, который обнуляется и запускается по переднему фронту сигнала GATE. По каждому переднему фронту сигнала TRIGGER значение временного штампа события копируется для записи в буферную память. По заднему фронту GATE текущее состояние счетчика записывается в регистр временного штампа конца спилла, который считывается микроконтроллером после заднего фронта сигнала GATE. Все счетчики считают с частотой системного тактового сигнала ПЛИС.

Регистрация сигналов IN

Со входов IN[k] сигналы проходят те же модули **IN_edge_detector** и **DELAY_IN[k]** и попадают в **Overlap**, где проверяется попадание сигнала одного из каналов в окно регистрации. В случае попадания значение основного счетчика копируется, в модуле **ADDER_[k]** к нему прибавляется младший бит k-го канала и результат записывается в буферную память. В противном случае наличие сигнала на входе игнорируется. При регистрации одного входящего сигнала, попавшего в окно регистрации, последующие сигналы того же канала так же игнорируются, пока TRIGGER вновь не примет пассивный уровень.

Младший бит

Суммирование младшего бита сигнала **START** и сигнала IN[k] происходит в модуле **Adder_[k]** по следующему правилу:

$$START_{lbit} + IN_{lbit} + 1 = SUM$$

Таким образом, измеренное значение времени имеет разрядность 8 бит, где младший бит соответствует половине периода тактового сигнала, а оставшиеся 7 бит - числу полных тактов. Например

$$01010111_2 = 87 \cdot \frac{T_{clk}}{2}$$

Запись в буферную память

Для записи событий в ПЛИС применяется двойная буферизация с переключением буферов по мере их заполнения на лету за несколько тактов системного тактового сигнала ПЛИС. После завершения операции переключения буферов заполненный буфер готов к вычитыванию микроконтроллером.

Модуль **Writing_pipeline** формирует строку, содержащую всю информацию о событии, и по заднему фронту сигнала TRIGGER выставляет разрешение **WE** к записи в один из буферов и производит запись 8-битными словами. Для адресации текущего записываемого слова в буферную память ПЛИС содержит внутренний счетчик адреса, который автоматически

наращивается после выполнения записи одного 8-битного слова. В случае переключения буфера, этот счетчик обнуляется.

4.3 Протоколы обмена данными

4.3.1 Протокол считывания буферной памяти

Созданная прошивка реализует протокол считывания буферной памяти (см. рисунок 4.4) в соответствии с описанием к модулю *TDC48DT*^[4]. Ниже приведено краткое описание.

Логическая схема в ПЛИС оповещает МК о готовности буфера сигналом RQ. Микроконтроллер обязан следить за этим сигналом и успевать вычитывать буфер до заполнения второго буфера.

Буферная память ПЛИС не может произвольным образом адресоваться микроконтроллером, возможно только последовательное считывание 16-битными словами. При этом ПЛИС содержит внутренний счетчик адреса для адресации текущего передаваемого слова в буферной памяти, который автоматически наращивается после выполнения микроконтроллером цикла чтения одного 16-битного слова. Перед началом вычитывания памяти микроконтроллер обязан сбросить этот внутренний счетчик с помощью сигналов CPU_ADDR_RST и CPU_ADDR_RST_CLK. Первое считываемое слово является служебным и содержит число байт, которые необходимо считать (буферная память может быть заполнена не полностью), без учета первого служебного слова. После окончания считывания, микроконтроллер обязан оповестить об этом ПЛИС сигналом ACK, после получения сигнала ACK ПЛИС снимает сигнал RQ. Сигналы RQ, DATA[15:0] вырабатываются ПЛИС, все остальные сигналы вырабатываются микроконтроллером.

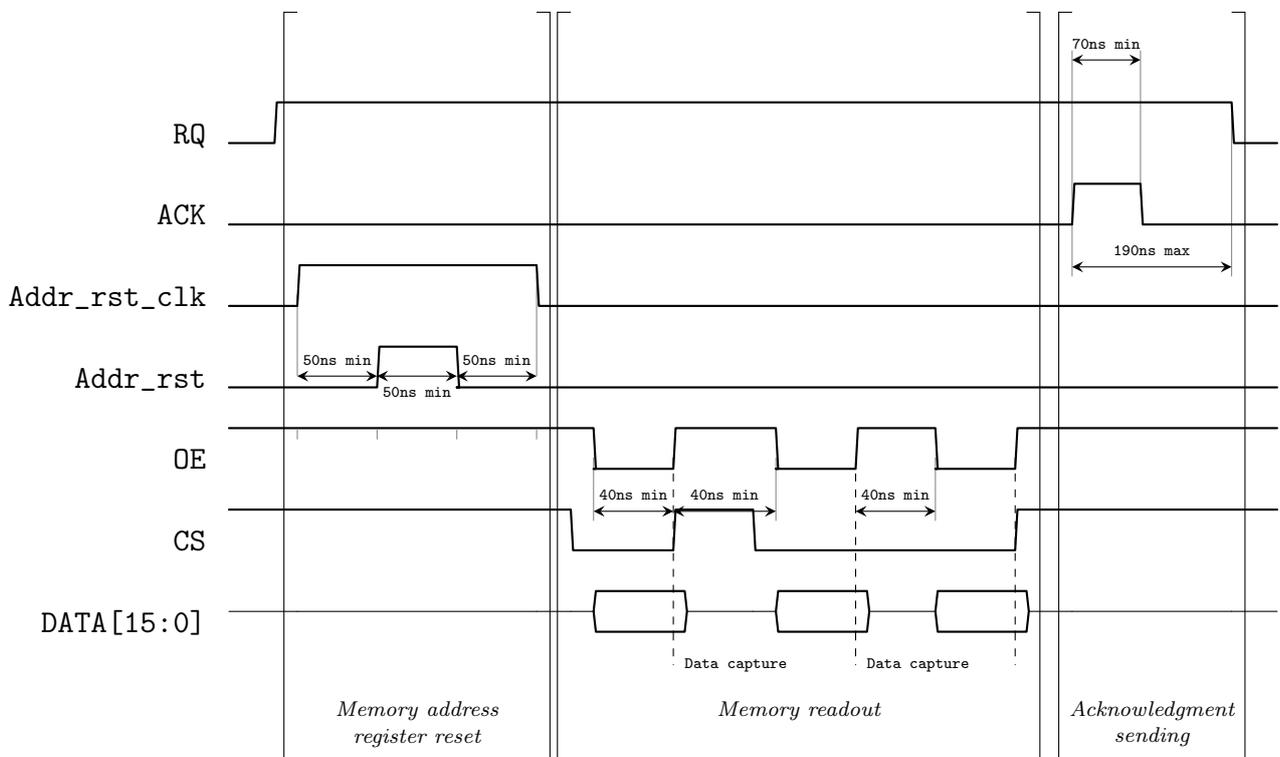


Рис. 4.4: Протокол считывания буферной памяти

4.3.2 Протокол настройки временных задержек

Настройки временных задержек происходит с использованием трёхсигнального SPI-подобного протокола по специально выделенным линиям GPIO[2] (DELAY_DATA), GPIO[3] (DELAY_CLK), GPIO[4] (DELAY_NLD) между МК и ПЛИС.

DELAY_CLK представляет собой тактовый сигнал, генерируемый микроконтроллером, который вызывает сдвиг передаваемого слова вправо;

DELAY_DATA - младший бит записываемого/считываемого слова;

DELAY_NLD - флаг процесса приема/передачи.

Поскольку во время GATE МК входит в режим быстрого прерывания и в нем блокируется любая деятельность, не связанная с процессом вычитывания буферной памяти, на время приема/передачи величины задержки сигнал GATE блокируется в ПЛИС. Поведение же ПЛИС определяется первым битом (RW) принятого слова. В случае, если этот бит равен 0, после окончания процесса приема, ПЛИС записывает величину задержки DELAY по принятому адресу ADDRESS.

Ниже приведена временная диаграмма (см. рисунок 4.5), описывающая взаимодействие ПЛИС и МК в случае записи

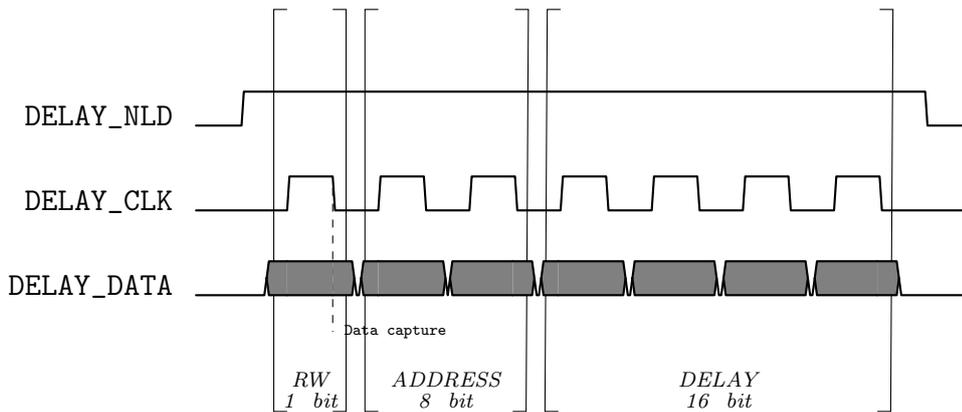


Рис. 4.5: Запись временной задержки в ПЛИС

После окончания приема ПЛИС записывает величину задержки DELAY по принятому адресу ADDRESS.

При считывании величины временной задержки ($RW = 1$) МК сначала выполняет процесс передачи бита чтения/записи, адреса считываемого модуля задержки и 16 нулевых битов, затем конфигурирует DELAY_DATA как вход. ПЛИС после окончания процесса приема вновь ожидает активного уровня DELAY_NLD и передает побитово 16 битное слово. МК по каждому заднему фронту DELAY_CLK производит считывание бита. Ниже приведена временная диаграмма (см. рисунок 4.6), полностью описывающая процесс считывания временной задержки микроконтроллером:

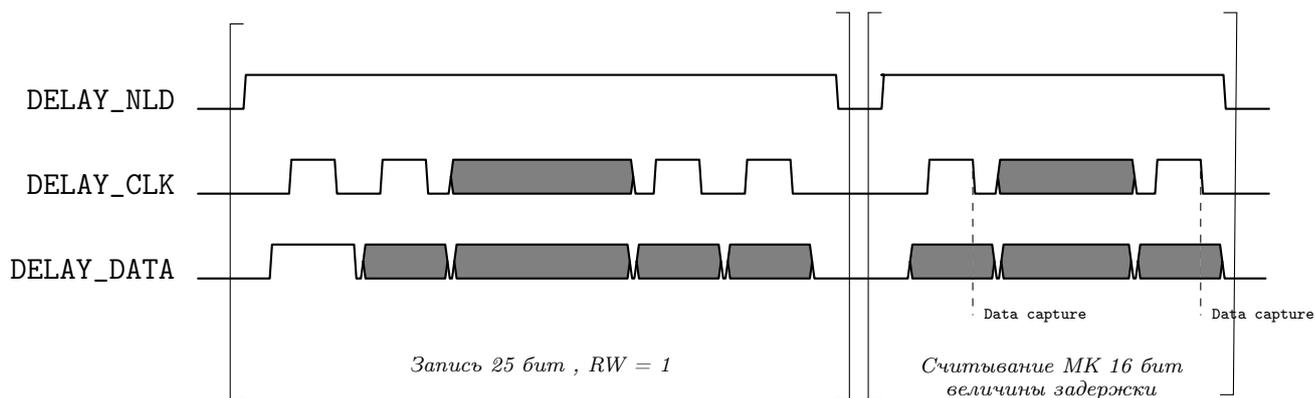


Рис. 4.6: Считывание временной задержки микроконтроллером

ADDRESS представляет 8 битное слово, которое равно адресу модуля “Delay” , в который произойдет запись.

DELAY представляет 16 битное слово, каждая тетрада которого инициализирует 4 бита задержки одного регистра сдвига SRL16 в массиве из четырех таких сдвиговых регистров (см. “Временная задержка Delay” из раздела 4.2).

Для конфигурации временных задержек используется программа *delay.c*, исходный код которой лежит в [5]. Программу следует выполнять на микроконтроллере модуля TDC-48DT, формат командной строки:

`./delay RW ADDRESS [DELAY],`

где

RW - флаг считывания(1)/записи(0) задержки;

ADDRESS - десятичное число в интервале [0 : 49], обозначающее адрес записываемой задержки; Адреса [0:47] отвечают за задержку сигналов IN, 49 - задержка сигнала START, 48- ширина окна TRIGGER.

DELAY- десятичное число, равное величине задержки,измеряемой в периодах тактового сигнала.

4.3.3 Протокол считывания регистра временного штампа конца спилла

Регистр временного штампа конца спилла считывается с использованием простого двух-сигнального последовательного протокола по специально выделенным для этой цели линиям GPIO[5] и GPIO[7] (см. рисунок 4.7). Процесс считывания реализован в соответствии с протоколом, приведенным в описании к модулю *TDC48DT*^[4]

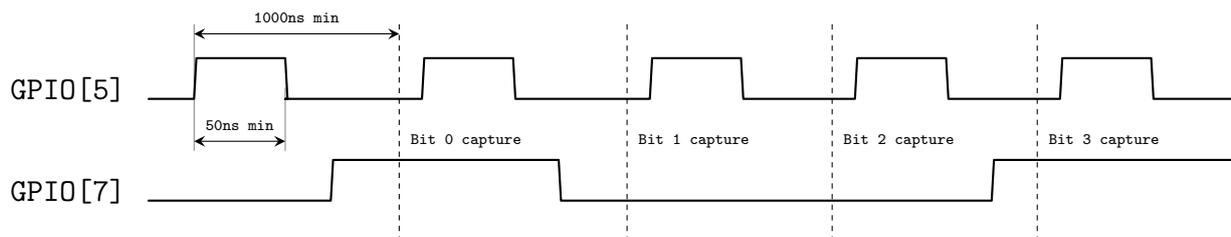


Рис. 4.7: Временная диаграмма считывания временного штампа конца спилла

Глава 5

Формат данных

Поскольку процесс считывания буферной памяти ПЛИС и формирование заголовка спилла производит микроконтроллер, формат данных события, формируемый ПЛИС, был выбран согласно описанию к модулю^[4]. Это позволяет избежать необходимости внесения изменений в протокол считывания данных.

5.1 Формат данных события

Событие полностью формируется в ПЛИС и состоит из заголовка (см. таблицу 5.1) длиной 16 байт и следующего за ним массива хитов. Каждый хит представляет собой 1 байт, значение которого равно измеренному значению времени в соответствующем канале. Хиты упорядочены по возрастанию номера канала. Номер канала, к которому относится хит, определяется с использованием маски сработавших каналов, которая представляет собой 48-битное число, в котором каждый бит соответствует одному каналу модуля. Младший бит маски соответствует каналу 0. Для всех сработавших каналов соответствующие биты в маске устанавливаются в 1. Число единичных битов в маске должно быть равно длине события в байтах (без учета заголовка). И если размер события не кратен 2, то событие дополняется нулевым байтом в конце, при это длина события никак не изменяется.

	31	16 15	0
0	Сигнатура (константа 0xF1FCF7FD)		
1	Временной штамп		
2	Маска (биты 15:0)	Длина события (в байтах)	
3	Маска (биты 47:16)		

Таблица 5.1: Формат данных заголовка события

5.2 Формат данных спилла

Спилл состоит из заголовка (см. таблицу 5.2) длиной 64 байта и следующего за ним массива событий. Формирование заголовка спилла выполняет МК.

	31	16 15	0
0	Зарезервировано		
1	Флаг устаревших данных		
2	Зарезервировано		
3	Зарезервировано		
4	Размер спилла (в байтах)		
5	Состояние сигнала ERR		
6	Флаг переполнения буфера в SDRAM		
7	Временной штамп конца спилла		
8	Зарезервировано		
9	Зарезервировано		
10	Зарезервировано		
11	Зарезервировано		
12	Зарезервировано		
13	Зарезервировано		
14	Зарезервировано		
15	Зарезервировано		

Таблица 5.2: Формат данных заголовка события

Глава 6

Тестирование прошивки

Для тестирования прошивки время-цифрового преобразователя, с генератора импульсов Tektronix AFG3102 было подано два сигнала: сигнал триггера **START** и сигнал **IN**. При чем сигнал, подаваемый на вход, имел изменяющуюся в диапазоне от 0 до 600 нс с шагом 1 нс временную задержку относительно сигнала триггера. Полученные значения кода ВЦП изображены на рисунке 6.1.

Ширина окна регистрации была выбрана максимальной (480 нс). А так же установлена ненулевая временная задержка сигнала триггера для того, что бы можно было изучить процесс регистрации сигнала во всем диапазоне окна регистрации. Для процесса регистрации характерен случайный элемент, то есть значение время-цифрового преобразования зависит не только от временного интервала между передними фронтами сигнала триггера и входящего сигнала, но и от фазы тактового сигнала. А именно от того, на какой именно момент периода тактового сигнала приходится передний фронт внешнего сигнала. Поэтому код одного и того же временного интервала может отличаться на 1 младший значащий разряд (МЗР).

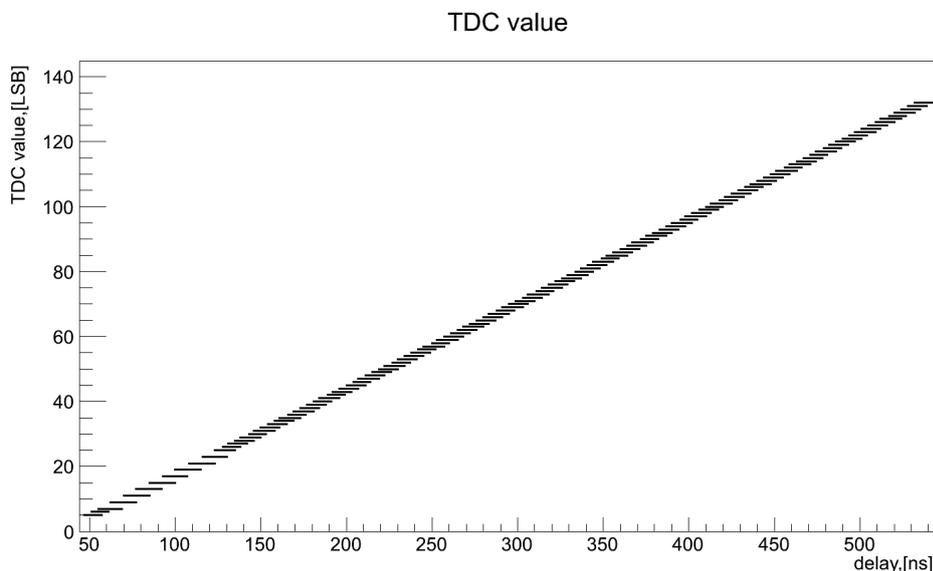


Рис. 6.1: Зависимость кода ВЦП от измеряемого временного интервала

Поскольку для каждого подаваемого временного интервала имеется несколько значений ВЦП, они были усреднены (см. рисунок 6.2).

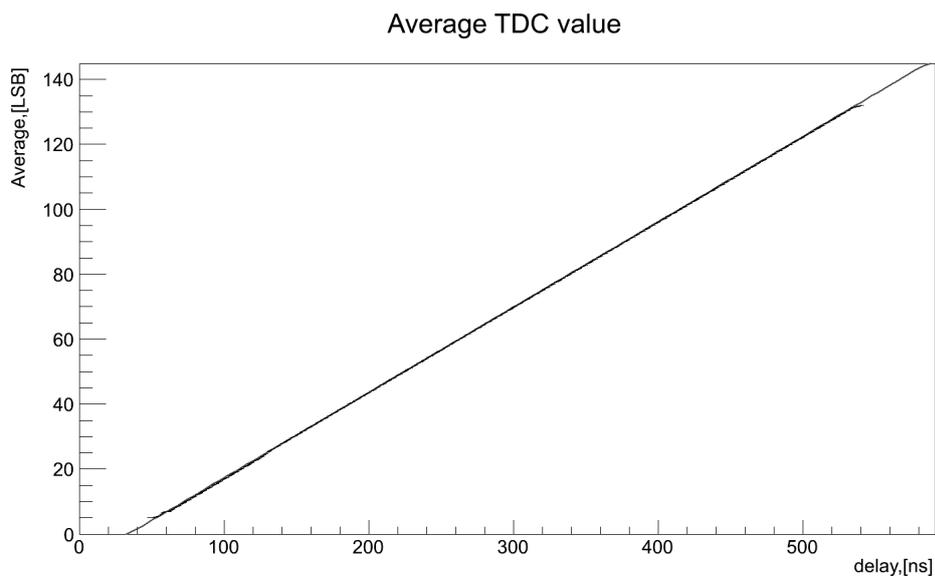


Рис. 6.2: Усредненные показания ВЦП

Под интегральной нелинейностью (INL) подразумевается максимальное отклонение реальной характеристики преобразования от наилучшего фита. Для чего график усредненных значений ВЦП был отфитирован прямой, и построена разность показаний ВЦП и значения фита (см. рисунок 6.3). По вертикали откладывается время в значениях младшего значащего разряда (LSB), по горизонтали - значение подаваемого временного интервала.

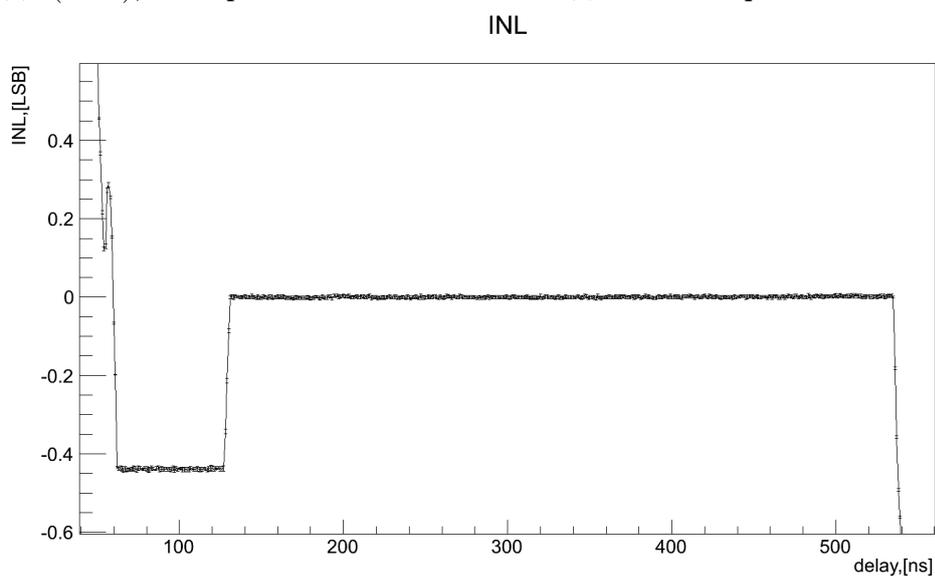


Рис. 6.3: Интегральная нелинейность

В течении 70 нс после переднего фронта сигнала триггера нелинейность составляет $-0.5LSB$, затем следует скачок, происхождение которого пока не известно. На краях нелинейность стремительно растет. Это связано с тем, что из-за конечного временного разрешения, на границах окна регистрации внешний сигнал регистрируется не с максимальной эффективностью (см. рисунок 6.4). Где под эффективностью регистрации сигналов понимается отношение числа хитов к общему числу триггеров за один спилл. Она спадает на краях окна регистрации в

течении 16 нс ($2 \cdot T_{clk}$), а на всем остальном интервале она максимальна и равна 1.

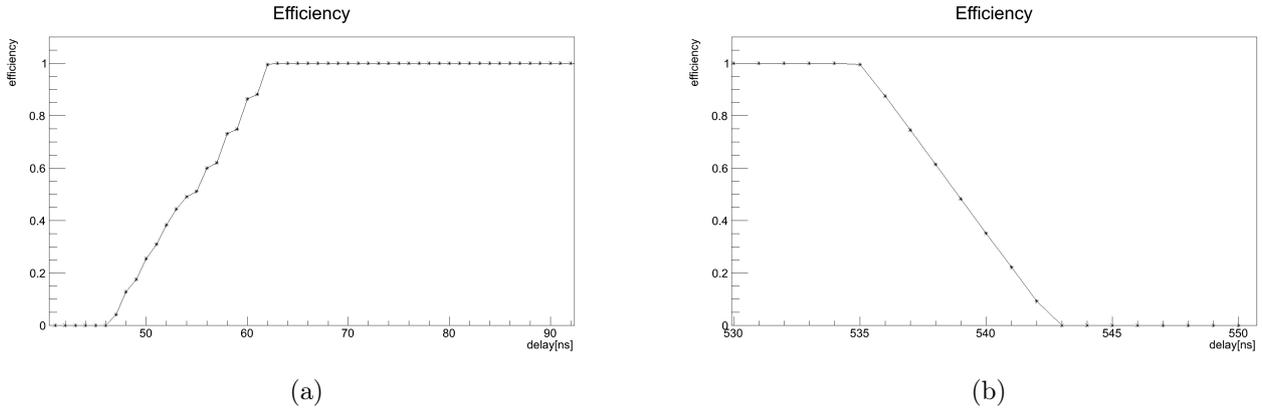


Рис. 6.4: Эффективность регистрации

Так же для каждого значения подаваемого временного интервала можно построить среднеквадратичное отклонение (см. рисунок 6.5).

$$RMS = \sqrt{\frac{\sum_i^N (\bar{y} - y_i)^2}{N}},$$

где \bar{y} - среднее значение кода ВЦП для определенного временного интервала; N - число измерений одного временного интервала.

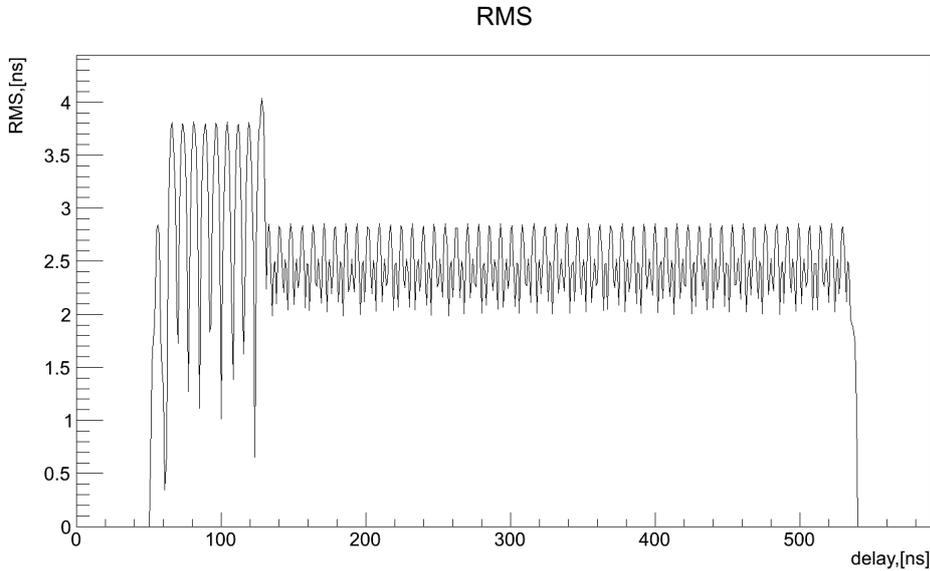


Рис. 6.5: Среднеквадратичное отклонение

В интервале $[0 : 70]$ нс после переднего фронта триггера, RMS довольно большое и изменяется в диапазоне от $[3.8 : 1]$ нс - там же, где и велика интегральная нелинейность. Далее RMS уменьшается до 2.4 ± 1 нс. На краях окна регистрации значение RMS изменяется в широких пределах в силу неэффективности регистрации. Теоретическое значение $T_{RMS} = 2.2$ нс.

Глава 7

Заключение

Выполненная работа позволяет провести модернизацию системы сбора данных установки ВЕС.

Написанная и отлаженная прошивка для ПЛИС модуля TDC-48DT позволяет регистрировать одновременно до 48 внешних сигналов в уровнях LVDS (или NIM при применении дополнительного модуля-конвертора NIM->LVDS) и обладает следующими характеристиками:

- Временное разрешение $T_{LSB} = 3.8$ нс; $T_{RMS} = 2.4 \pm 1$ нс.
- Интегральная нелинейность $T_{INL} = [-0.4 : 0.01] T_{LSB}$.
- Максимально возможное число регистрируемых триггеров $\approx 850000 \frac{1}{c}$
- Минимальное время между триггерами 1.2 мкс
- Объем памяти для хранения данных от одного сброса 32 МВ
- Регулировка временной задержки входящих сигналов в диапазоне [60 : 970] нс
- Регулировка ширины окна регистрации в диапазоне [30 : 480] нс

Было успешно проведено моделирование прошивки, испытание на сигналах от генератора и тестовый запуск в системе сбора данных эксперимента ВЕС.

Литература

- [1] А.В. Ивашин, В.Д. Матвеев, Ю.А. Хохлов,
“Модернизированная система сбора данных установки ВЕС. Технические аспекты.”
Препринт ИФВЭ 2010-10
- [2] Yu. Khokhlov, et al. “Recent results with upgraded VES setup: $\pi^- 3\pi^0$ and other systems”
EPJ Web of Conferences 37, 01029 (2012)
- [3] Henzler. S. “Time to digital converter”
<http://www.springer.com/978-90-481-8627-3>
- [4] А.В. Ивашин, “Модуль TDC-48DT,”
<http://pcbech.ihep.su/~ivashin/tdc-48dt-doc.pdf>
- [5] Исходный код прошивки ВЦП и программы регулировки задержек.
<http://pcbech.ihep.su/~kravchenko/TDC-48DT-Delay.tar.gz>
- [6] “Xilinx ISE 9.2i Software Manuals”
<http://www.xilinx.com>