

Министерство науки и высшего образования Российской Федерации  
Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий

Работа допущена к защите  
директор ВШИСиСТ  
\_\_\_\_\_ В.М. Ицыксон  
« \_\_\_\_\_ » \_\_\_\_\_ 2022 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА**  
**РАЗРАБОТКА УПРАВЛЯЮЩЕГО УСТРОЙСТВА КАЛИБРАТОРА**  
**СУБ-НАНОСЕКУНДНОЙ СИНХРОНИЗАЦИИ**

по направлению 03.09.01 «Информатика и вычислительная техника»  
по образовательной программе  
03.09.01\_01 «Вычислительные машины, комплексы, системы и сети»

Выполнил

студент гр. 3530101/80101 <подпись>

Д.В. Пешков

Руководитель

к.т.н.,

доцент,

<подпись>

А.А. Лавров

Консультант

ассистент

<подпись>

А.А. Антонов

Консультант

по нормоконтролю

<подпись>

А.Г. Новопашенный

Санкт-Петербург  
2022

**САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ПЕТРА ВЕЛИКОГО**

**Институт компьютерных наук и технологий**

УТВЕРЖДАЮ

директор ВШИСиСТ

\_\_\_\_\_ В.М. Ицыксон

« \_\_\_\_\_ » \_\_\_\_\_ 2022г.

**ЗАДАНИЕ**

**на выполнение выпускной квалификационной работы**

студенту Пешкову Даниилу Валерьевичу гр. 3530101/80101

1. Тема работы: Разработка управляющего устройства калибратора суб-наносекундной синхронизации.
2. Срок сдачи студентом законченной работы<sup>1</sup>: дд.мм.202X.
3. Исходные данные по работе<sup>2</sup>: статистические данные с сайта [gosstat], а также из репозитория [uci]; основным источником литературы является монография [Book] и статья [Article].
4. Содержание работы (перечень подлежащих разработке вопросов):
  - 4.1. Обзор литературы по теме ВКР.
  - 4.2. Исследование программных продуктов.
  - 4.3. Разработка метода/алгоритма/программы.
  - 4.4. Апробация разработанного метода/алгоритма/программы.
5. Перечень графического материала (с указанием обязательных чертежей):
  - 5.1. Схема работы метода/алгоритма.
  - 5.2. Архитектура разработанной программы/библиотеки.
6. Консультанты по работе<sup>3</sup>:
  - 6.1. Ассистент, А.А. Антонов.

<sup>1</sup>Определяется руководителем ОП, но не позднее последнего числа преддипломной практики и/или не позднее, чем за 20 дней до защиты в силу п. 6.1. «Порядка обеспечения самостоятельности выполнения письменных работ и проверки письменных работ на объем заимствований».

<sup>2</sup>Текст, который подчеркнут и/или выделен в отдельные элементы нумерационного списка, приведён в качестве примера.

<sup>3</sup>Подпись консультанта по нормоконтролю пока не требуется. Назначается всем по умолчанию.

6.2. Должность, степень, А.Г. Новопашенный (нормоконтроль).

7. Дата выдачи задания<sup>4</sup>: дд.мм.202Х.

Руководитель ВКР \_\_\_\_\_ А.А. Лавров

Консультант<sup>5</sup> \_\_\_\_\_ А.А. Антонов

Задание принял к исполнению дд.мм.202Х

Студент \_\_\_\_\_ Д.В. Пешков

---

<sup>4</sup>Не позднее 3 месяцев до защиты (утверждение тем ВКР по университету) или первого числа преддипломной практики или по решению руководителя ОП или подразделения (открытый вопрос).

<sup>5</sup>В случае, если есть консультант, отличный от консультанта по нормоконтролю.

## РЕФЕРАТ

На ?? с., ?? рисунков, ?? таблиц, ?? приложений

КЛЮЧЕВЫЕ СЛОВА: СТИЛЕВОЕ ОФОРМЛЕНИЕ САЙТА, УПРАВЛЕНИЕ КОНТЕНТОМ, PHP, MYSQL, АРХИТЕКТУРА СИСТЕМЫ.<sup>6</sup>

Тема выпускной квалификационной работы: «Разработка управляющего устройства калибратора суб-наносекундной синхронизации»<sup>7</sup>.

В данной работе изложена сущность подхода к созданию динамического информационного портала на основе использования открытых технологий Apache, MySQL и PHP. Даны общие понятия и классификация IT-систем такого класса. Проведен анализ систем-прототипов. Изучена технология создания указанного класса информационных систем. Разработана конкретная программная реализация динамического информационного портала на примере портала выбранной тематики...<sup>8</sup>

В данной работе изложена сущность подхода к созданию динамического информационного портала на основе использования открытых технологий Apache, MySQL и PHP. Даны общие понятия и классификация IT-систем такого класса. Проведен анализ систем-прототипов. Изучена технология создания указанного класса информационных систем. Разработана конкретная программная реализация динамического информационного портала на примере портала выбранной тематики...

## ABSTRACT

?? pages, ?? figures, ?? tables, ?? appendices

KEYWORDS: STYLE REGISTRATION, CONTENT MANAGEMENT, PHP, MYSQL, SYSTEM ARCHITECTURE.

---

<sup>6</sup>Всего **слов**: от 3 до 15. Всего **слов и словосочетаний**: от 3 до 5. Оформляются в именительном падеже множественного числа (или в единственном числе, если нет другой формы), оформленных по правилам русского языка. *Внимание! Размещение сноски после точки является примером как запрещено оформлять сноски.*

<sup>7</sup>Реферат **должен содержать**: предмет, тему, цель ВКР; метод или методологию проведения ВКР; результаты ВКР: область применения результатов ВКР; выводы.

<sup>8</sup>ОТ 1000 ДО 1500 печатных знаков (ГОСТ Р 7.0.99-2018 СИБИД) на русский или английский текст. Текст реферата повторён дважды на русском и английском языке для демонстрации подхода к нумерации страниц.

The subject of the graduate qualification work is «Title of the thesis».

In the given work the essence of the approach to creation of a dynamic information portal on the basis of use of open technologies Apache, MySQL and PHP is stated. The general concepts and classification of IT-systems of such class are given. The analysis of systems-prototypes is lead. The technology of creation of the specified class of information systems is investigated. Concrete program realization of a dynamic information portal on an example of a portal of the chosen subjects is developed...

In the given work the essence of the approach to creation of a dynamic information portal on the basis of use of open technologies Apache, MySQL and PHP is stated. The general concepts and classification of IT-systems of such class are given. The analysis of systems-prototypes is lead. The technology of creation of the specified class of information systems is investigated. Concrete program realization of a dynamic information portal on an example of a portal of the chosen subjects is developed...

## СОДЕРЖАНИЕ

Введение .....	7
Глава 1. Анализ предметной области .....	9
1.1. White Rabbit .....	9
1.2. Калибровка .....	9
1.3. Актуальность.....	12
1.4. Стробоскопический осциллограф .....	12
Глава 2. Аппаратная платформа и алгоритм проведения измерений.....	14
2.1. Измерительный канал .....	15
2.2. Алгоритм проведения измерений.....	16
2.3. Пороговое напряжение сравнения.....	17
2.4. Линия задержки .....	17
2.5. Компаратор .....	18
Глава 3. Разработка аппаратного описания управляющего устройства .....	20
3.1. Описание верхнего уровня .....	20
3.2. Измерительный модуль .....	21
3.2.1. Модуль stb_gen .....	22

## ВВЕДЕНИЕ

Для надёжного функционирования распределённых систем потоковой обработки данных, работающих в режиме реального времени, на исполняемые в них операции накладываются строгие временные ограничения. Многочисленные узлы таких систем могут находиться друг от друга на значительных расстояниях, что приводит к длительным (и не всегда одинаковым) задержкам при передаче сигналов между ними. Возникающие задержки приводят к рассинхронизации работы устройств систем, что влечёт за собой возникновение потенциально некорректных результатов выполнения операций.

С целью согласования всех устройств и обеспечения общего представления времени во всей сети применяются системы синхронизации часов. Такие системы гарантируют, что часы всех устройств сети отсчитывают время с одинаковой скоростью (выдают одинаковые показания в каждый момент времени).

Для передачи информации при синхронизации могут использоваться различные протоколы. Наибольшее распространение получили следующие два: NTP и PTP (Precision Time Protocol). Протокол NTP способен обеспечивать точность синхронизации времени до одной миллисекунды, а протокол PTP – до десяти.

Протоколы NTP и PTP не подходят для случая, когда необходима синхронизация с субнаносекундной точностью. Например, такая высокая точность требуется в распределённых системах, применяемых в экспериментах по физике высоких энергий. Там они используются для потоковой обработки информации, поступающей с детекторов и ускорителей частиц.

Системы субнаносекундной синхронизации применяются в Большом Адронном Коллайдере, расположенном в ЦЕРН, в Швейцарии. Они также планируются к применению в строящемся комплексе «NICA» Объединённого института ядерных исследований (ОИЯИ) в Дубне. Другим приложением субнаносекундной синхронизации являются системы радиочастотного позиционирования, использующие технологию сверхширокополосной связи (UWB) и алгоритм позиционирования TDoA (Time Difference of Arrival).

**Цель работы:** разработка управляющего устройства для устройства, выполняющего калибровку систем суб-наносекундной синхронизации.

**Решаемые в данной работе задачи:** портирование и интеграция существующих открытых модулей в систему на кристалле, реализация модулей для проведения измерений по принципу стробоскопического осциллографа, написание

драйверов для используемой периферии, написания управляющей программы для процессорного ядра, реализация протокола для управления проведением измерений с персонального компьютера, тестирование и отладка, калибровка полученного устройства.



## ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

### 1.1. White Rabbit

White Rabbit – система синхронизации часов. Разработана при сотрудничестве множества институтов и компаний. Изначально проект был начат для улучшения текущей системы синхронизации в Церне. Предполагалось использование для физических экспериментов, однако в процессе было создано обобщенное решение, которое нашло своё применение в различных сферах.

Характеристики:

- Суб-наносекундная точность
- Большое количество синхронизируемых узлов
- Расстояния в десятки километров
- Канал передачи между двумя узлами — 1 Gbps
- Открытый исходный код

Достоинствами White Rabbit являются полностью открытый исходный код и аппаратура, а также использование существующих стандартов (Ethernet, PTP и т. д.).

### 1.2. Калибровка

Синхронизация в сети White Rabbit выполняется по протоколу WR PTP — модифицированному протоколу PTP. (IEEE 1588). Однако для достижения суб-наносекундной точности необходима дополнительная калибровка.

Обмен данными между двумя устройствами происходит по одной линии оптоволокну, работающей в полнодуплексном режиме. Для передачи в одну и другую сторону используется свет с разной длиной волны, поэтому возникает асимметричность в задержках распространения сигнала. Из-за этого устройство не может само определить задержки, отправив эхо запрос другому устройству — время распространения сигнала в одну и другую сторону не равны.

Определение коэффициента асимметричности оптоволокну позволит протоколу White Rabbit PTP обеспечить требуемую точность синхронизации устройств сети.

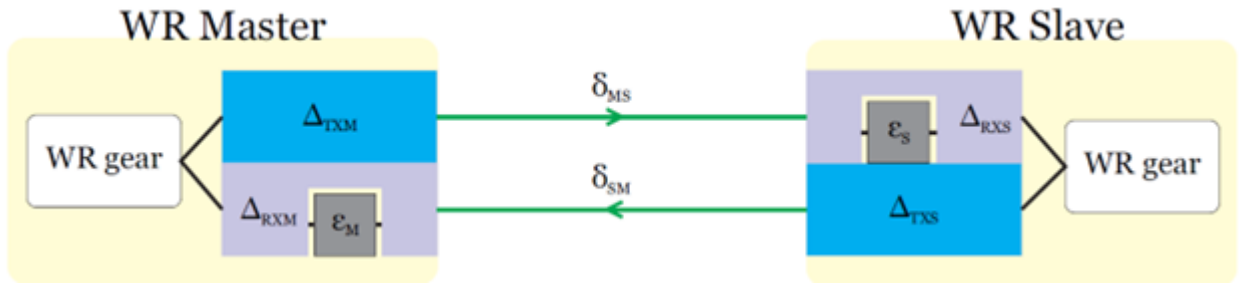


Рис.1.1. Модель соединения между двумя устройствами

На рис.1.1 изображены возникающие задержки, требующие калибровки. Внутри каждого устройства возникают задержки на приёме и отправке ( $\Delta_{TXM}, \Delta_{RXM}, \Delta_{TXS}, \Delta_{RXS}, \epsilon_M, \epsilon_S$ ), которые являются результатом задержек в SFP (Small Form-factor Pluggable) модуле, в электрических цепях и электронных компонентах. Эти задержки калибруются отдельно на каждом устройстве и не являются предметом рассмотрения в данной работе.

Суммарная задержка распространения сигнала от ведущего устройства к ведомому устройству и обратно считается по формуле:

$$delay_{MM} = \Delta_{TXM} + \Delta_{RXS} + \epsilon_S + \Delta_{TXS} + \Delta_{RXM} + \epsilon_M + \delta_{MS} + \delta_{SM} \quad (1.1)$$

В данной работе рассматривается калибровка задержек  $\delta_{MS}$  и  $\delta_{SM}$  – задержек распространения сигнала по оптоволокну.

Когда оптоволокну ещё не установлено, измерить задержки и вычислить коэффициент асимметричности можно без особых усилий. Коэффициент асимметричности определяется, как:

$$\alpha = \frac{\delta_{MS} - \delta_{SM}}{\delta_{SM}} \quad (1.2)$$

Измеряется коэффициент при помощи дополнительного короткого оптоволокну, с известной задержкой.

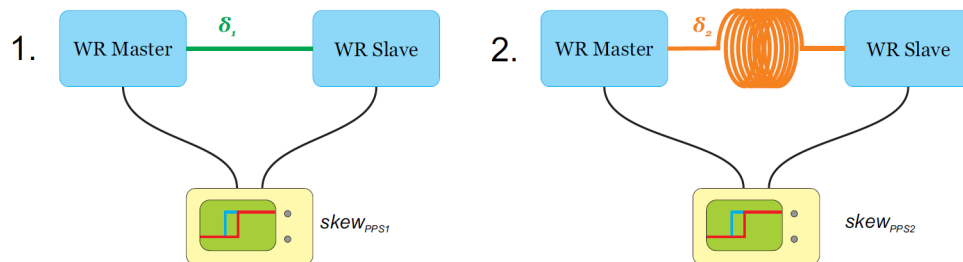


Рис.1.2. Измерение асимметричности про помощи дополнительного оптоволокну

Два устройства подключаются калибруемым оптоволоком ( $\delta_2$ ) и дополнительным ( $\delta_1$ ) отдельно, далее два устройства синхронизируются. После этого измеряется разность фаз синхросигналов 1-PPS (Pulse Per Second), генерируемых ведущим и ведомым устройством.

$$skew_{PPS} = t_{PPS_S} - t_{PPS_M} \quad (1.3)$$

По измеренным значениям можно определить коэффициент асимметричности, как:

$$\alpha = \frac{2 (skew_{PPS2} - skew_{PPS1})}{\frac{1}{2}\delta_2 - (skew_{PPS2} - skew_{PPS1})} \quad (1.4)$$

Однако существуют так же уже установленные линии, нуждающиеся в калибровке. В таких случаях описанный выше метод не подходит.

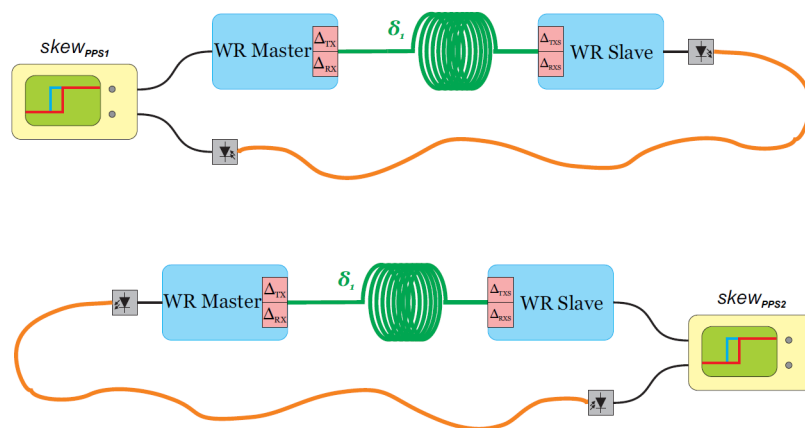


Рис.1.3. Измерение асимметричности про помощи дополнительного оптоволокну

В таких случаях применяется немного изменённый метод с подключением петли от выхода 1-PPS одного устройства к другому (рис.1.3). После синхронизации так же измеряются расхождения фронтов синхросигналов и усредняются.

$$skew_{PPS} = \frac{1}{2} (skew_{PPS1} + skew_{PPS2}) \quad (1.5)$$

Далее это значение может быть использовано для вычисления коэффициента по формуле 1.4.

Разрабатываемое устройство служит для автоматизированного измерения разности фаз и выполнения калибровки.

### 1.3. Актуальность

Для процесса калибровки предполагается использовать устройство, способное измерять отрезки времени меньше, чем 1 нс, чтобы обеспечить суб-наносекундную точность, т.е. иметь частоту дискретизации 1 GSa. Если применить правило «пятикратного превышения частоты дискретизации», то используемый осциллограф должен иметь частоту дискретизации выше 5 GSa.

Цена осциллографов с такими характеристиками крайне высока. Актуальность данной работы заключается в том, что предлагается разработать относительно бюджетное устройство, работающее по принципу стробоскопического осциллографа.

Сигналы PPS от калибруемых устройств являются периодическими. Это позволяет применить для их обнаружения, захвата и анализа устройство, работающее по принципу стробоскопического осциллографа.

### 1.4. Стробоскопический осциллограф

Стробоскопические осциллографы предназначены для обнаружения, захвата и анализа периодических сигналов. Принцип работы стробоскопического осциллографа проиллюстрирован на рис.1.4.

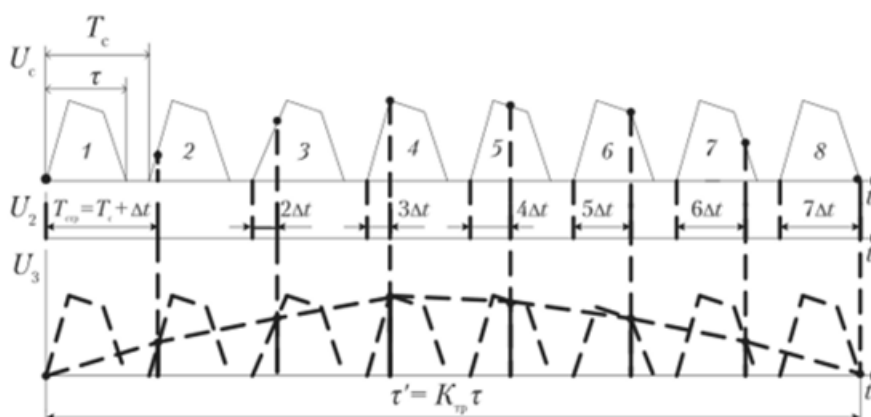


Рис.1.4. Измерение асимметрии про помощи дополнительного оптоволоконна

Условные обозначения:

$U_c$  – исследуемый периодический сигнал

$T_c$  – период исследуемого сигнала

$\tau$  – длительность импульса исследуемого сигнала

$\Delta t$  – шаг считывания исследуемого сигнала

$U_2$  – стробы осциллографа

$U_3$  – снятая осциллограмма сигнала

Таким образом, для снятия очередной точки изменяется смещение  $\Delta t$ . Частота дискретизации определяется минимальным смещением, которое может задать осциллограф.

## ГЛАВА 2. АППАРАТНАЯ ПЛАТФОРМА И АЛГОРИТМ ПРОВЕДЕНИЯ ИЗМЕРЕНИЙ

Управляющее устройство реализовано на базе ПЛИС, так как есть необходимость разрабатывать специфические аппаратные модули.

В качестве аппаратной платформы УУ (управляющее устройство) выступает ПЛИС серии LittleVee китайского производителя Gowin Semiconductor – GW1N-UV9QN88C6/I5.

Данная ПЛИС имеет следующие характеристики:

- количество логических блоков LUT4 – 8640 шт.
- количество триггеров – 6480 шт.
- объём FLASH – 408 Кбит
- количество блоков BSRAM – 26 шт.
- объём блоков SRAM – 468 Кбит
- количество блоков PLL (ФАПЧ) – 2 шт.
- напряжение питания ядра – 1.8–3.3 В
- количество доступных для пользователя выходов I/O – 77
- среди них дифференциальных пар – 19

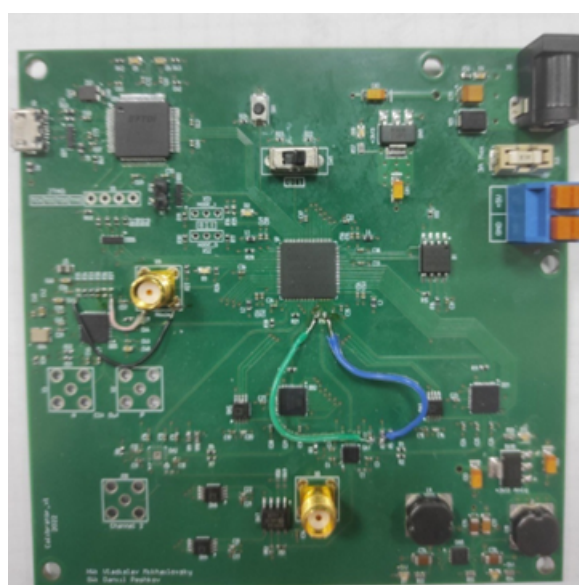


Рис.2.1. Печатная плата устройства

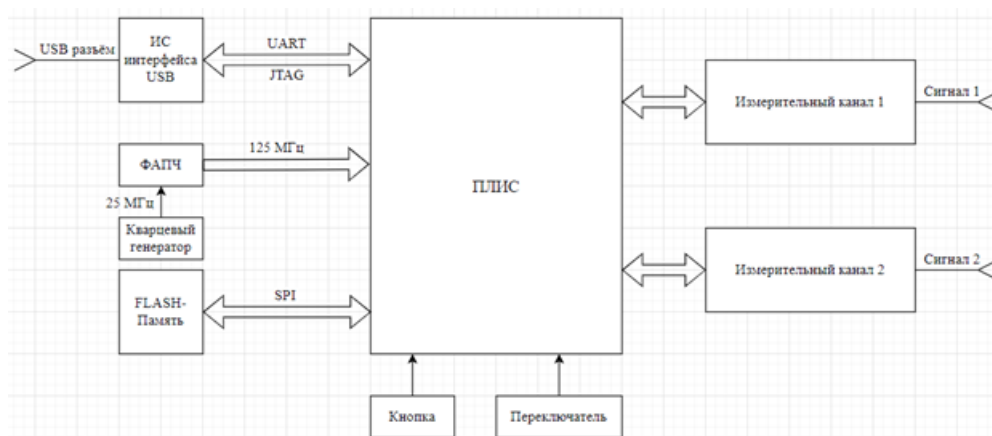


Рис.2.2. Структурная схема устройства

На рис.2.2 изображена структурная схема устройства. К УУ подключаются два измерительных канала, которыми необходимо управлять для проведения измерений. ПЛИС тактируется высокостабильным тактовым сигналом с частотой 125 МГц. Связь с устройством верхнего уровня осуществляется через FTDI чип FT2232H.

Перед разработкой УУ необходимо определить ОУ (объект управления), которым предстоит управлять.

## 2.1. Измерительный канал

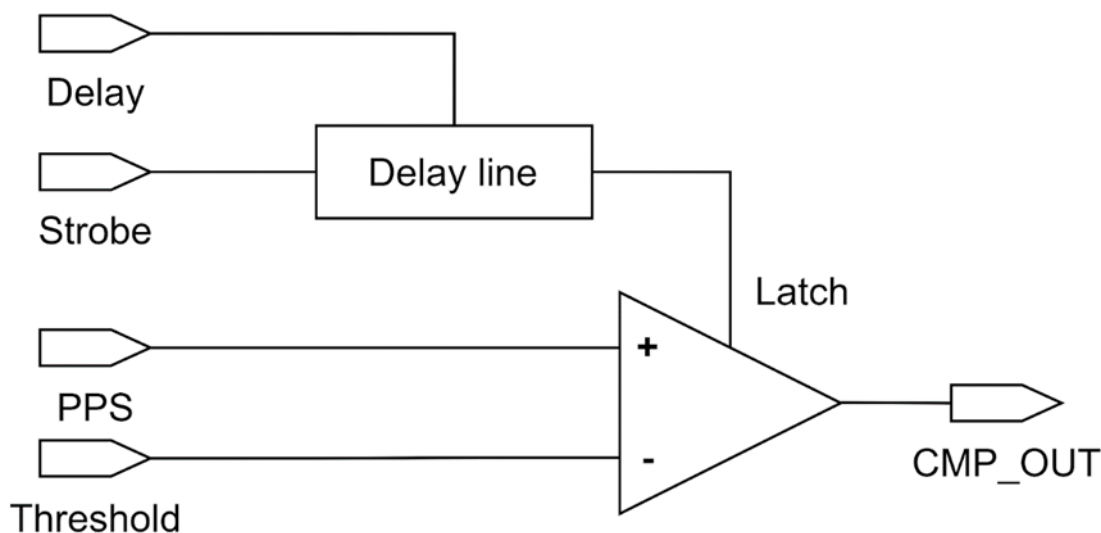


Рис.2.3. Функциональная схема измерительного канала

На рис.2.3 приведена функциональная схема одного из измерительных каналов.

Threshold – пороговое напряжение сравнения на компараторе

Delay – настройка задержки сигнала на линии задержки (Delay line)

Strobe – стробы, выдаваемые УУ и защёлкивающее текущий результат сравнения компаратора.

PPS – измеряемый сигнал 1-PPS (В общем случае, может быть любой частоты, кратной либо 125 МГц, либо частоте внешнего тактового сигнала, подаваемого с SMA разъёма)

## 2.2. Алгоритм проведения измерений

Перед началом проведения измерения необходимо определить частоты измеряемого сигнала и найти его фронт.

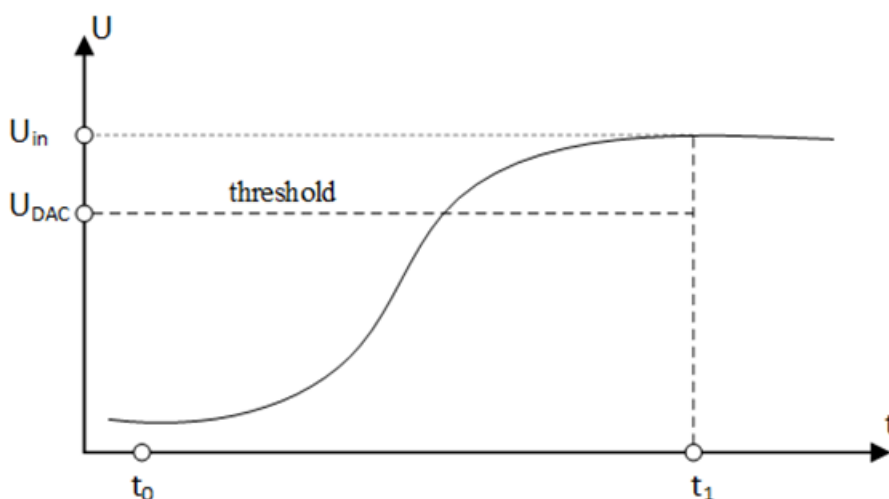


Рис.2.4. Измерение сигнала

Далее необходимо начать генерировать стробы с частотой измеряемого сигнала. Стробы блокируют текущий выход компаратора, и на его выходе остаётся результат сравнения в момент прихода стробы. При помощи изменения задержки можно сдвигать измеряемую точку по оси ОХ. Для поиска значения напряжения в измеряемой точке необходимо изменять пороговое напряжение сравнения на компараторе.

На рис.2.4 приведён пример измерения.  $t_0$  – точка от которой измеряется сигнал (момент прихода стробы),  $t_1$  – измеряемая точка (момент прихода задержанной стробы на компаратор). Временной отрезок  $[t_0, t_1]$  задаётся линией задержки. Если в момент защёлкивания на выходе компаратора логическая единица, значит измеряемое напряжение выше, чем пороговое. Перед приходом следующей стробы пороговое напряжение повышается. Так продолжается до тех



пор, пока на выходе компаратора не будет логического нуля. В таком случае мы считаем, что измеряемая точка имеет значения напряжение равное пороговому. Задержка увеличивается, и измеряется следующая точка. Таким образом снимается осциллограмма входного сигнала.

### 2.3. Пороговое напряжение сравнения

Для задания порогового напряжения используется DAC (Digital to analog converter) AD5662 от Analog Devices. AD5662 – 16-битный DAC с гарантированной точностью 12 бит.

С опорным напряжением 3.3 В, шаг изменения напряжения –  $\frac{3.3}{2^{16}} \approx 50$  мкВ. Для задания напряжения используется интерфейс SPI (рис.2.5).

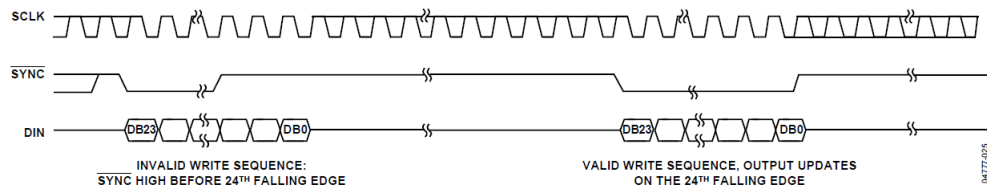
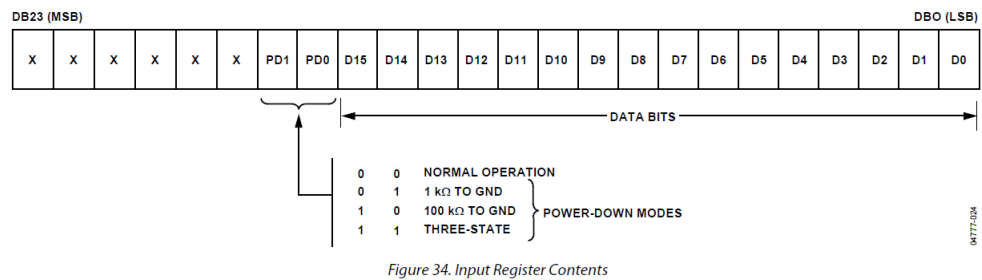


Рис.2.5. Интерфейс взаимодействия с AD5662

При напряжении питания 3.3 В максимальная частота сигнала SCLK – 20 МГц. Максимальное время установки выходного напряжения – 10 мкс (обычно 8 мкс).

### 2.4. Линия задержки

Для сдвига задержки строга используется микросхема SY100EP196V от Microchip. Задержка задаётся 10-битным цифровым кодом с шагом  $\approx 10$  пс (рис.2.6). Так же есть аналоговый вход FTUNE, которым можно точнее подстраивать задержку (рис.2.7) (на данный момент не используется).

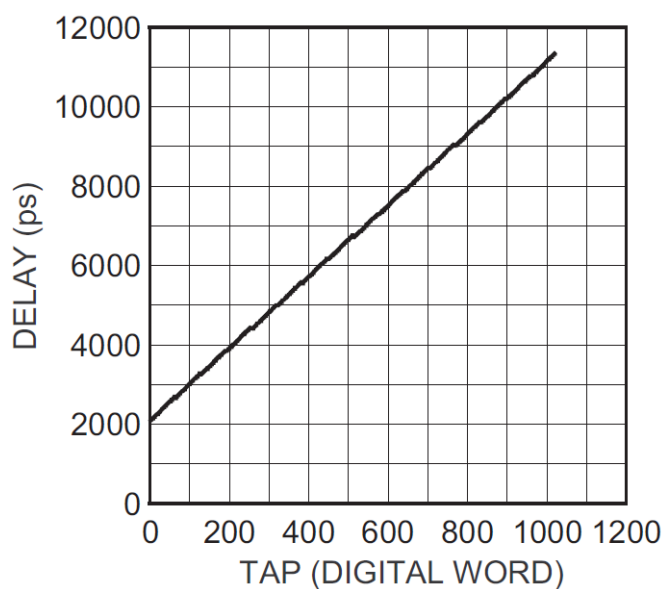


Рис.2.6. Задержка, задаваемая цифровым входом

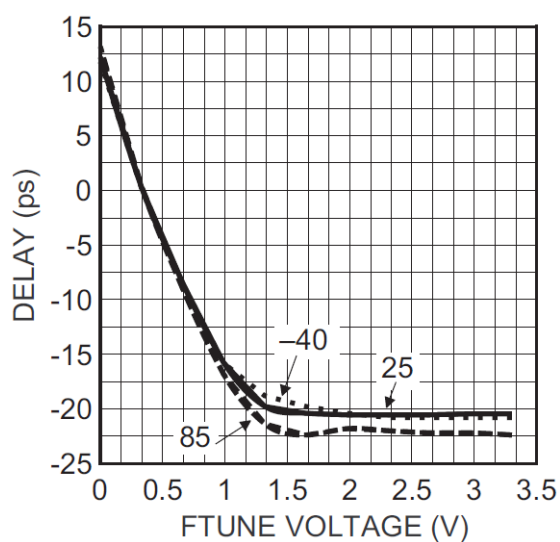


Рис.2.7. Подстройка задержки через аналоговый вход FTUNE

## 2.5. Компаратор

Для сравнения используется компаратор ADCMP582 от Analog Devices. Основным параметром, который необходимо учесть является  $t_{PL}$  (рис.2.8).

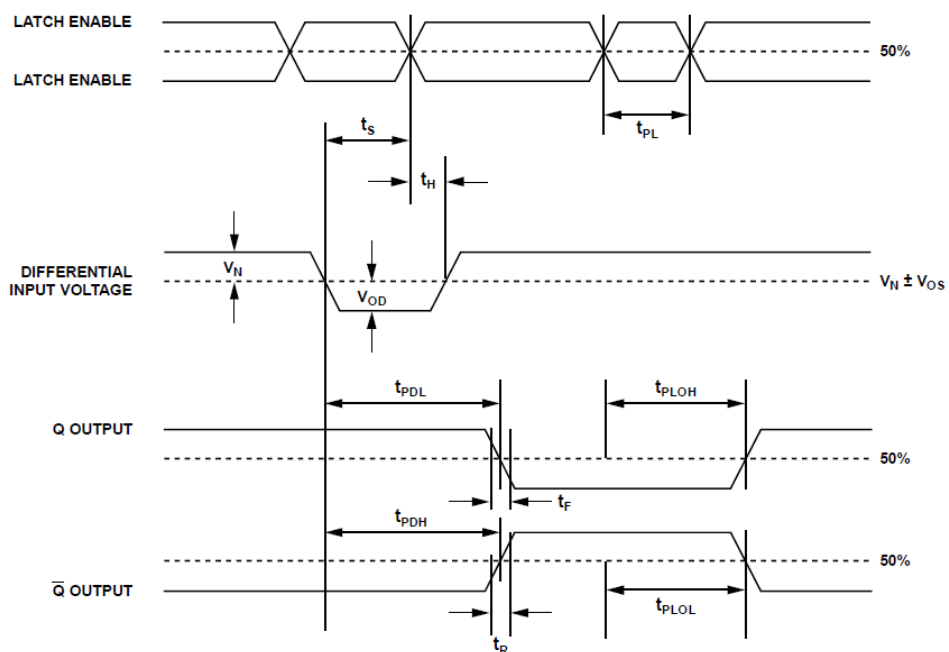


Рис.2.8. Временная диаграмма ADCMP582

$t_{PL}$  – минимальное время, в течение которого сигнал защёлки (Latch Enable) должен быть высоким, чтобы получить результат сравнения попал на выход компаратора.

## ГЛАВА 3. РАЗРАБОТКА АППАРАТНОГО ОПИСАНИЯ УПРАВЛЯЮЩЕГО УСТРОЙСТВА

Все исходные коды аппаратных описаний находятся в директории */rtl/*.

### 3.1. Описание верхнего уровня

Описание верхнего уровня находится в файле */rtl/calsoc\_top.sv*.

Управляющее устройство реализовано в виде СнК (Система на кристалле) (рис.3.1) на базе открытого процессорного ядра *PicoRV32*, основанного на открытой архитектуре RISC-V.

Все периферийные модули подключаются к ядру через шину *Wishbone*. Арбитраж на шине выполняет открытый модуль *wbxbbar*.

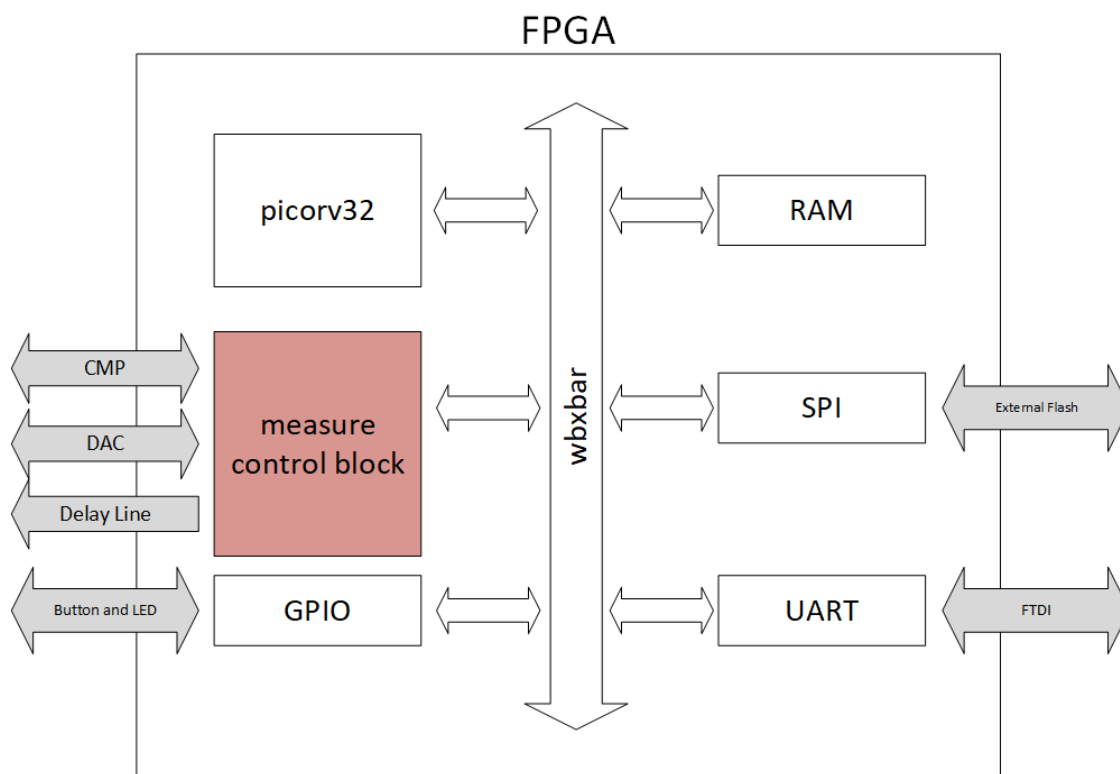


Рис.3.1. Структурная схема СнК

Все периферийные устройства разделяют между собой общее адресное пространство.

0x00000000 – 0x00FFFFFF – RAM

0x01000000 – 0x01FFFFFF – ROM загрузчика

0x02000000 – 0x02FFFFFF – GPIO

0x03000000 – 0x03FFFFFF – UART1

0x04000000 – 0x04FFFFFF – память программы

0x05000000 – 0x05FFFFFF – измерительный модуль

### 3.2. Измерительный модуль

Измерительный модуль выдаёт все необходимые управляющие сигналы для проведения измерений и логически разделён на несколько компонентов (рис.3.2):

- *stb\_gen* – модуль, измеряющий частоту сигнала и генерирующий стробы
- *ch\_measure\_ctl* – модуль, непосредственно управляющий измерением одного канала
- *spi\_master* – модуль, реализующий взаимодействие с AD5662
- *sc\_fifo* – FIFO, накапливающее измеренные значения

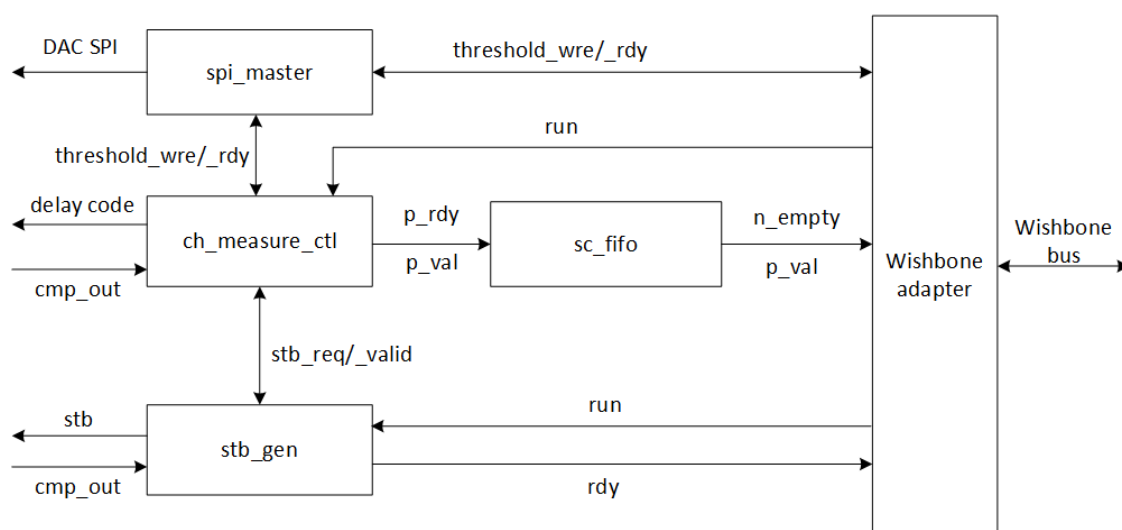


Рис.3.2. Структурная схема измерительного модуля (для одного канала)

Все описанные ранее модули подключены в модуле верхнего уровня *measure\_unit*. В нём же реализована вся логика управления проведением измерений через шину Wishbone.

Далее будут отдельно рассмотрен каждый модуль.

### 3.2.1. Модуль *stb\_gen*

Для определения частоты измеряемого сигнала необходимо измерить время между двумя фронтами. Для измерения используется 32-разрядный счётчик, один отсчёт которого равняется 8 нс (125 МГц).

Реализация «в лоб» не может работать стабильно на целевой ПЛИС из-за задержек на цепочке переносов в 32-разрядном сумматоре. Для корректной работы на такой частоте необходимо конвейеризировать счётчик (рис.3.3).

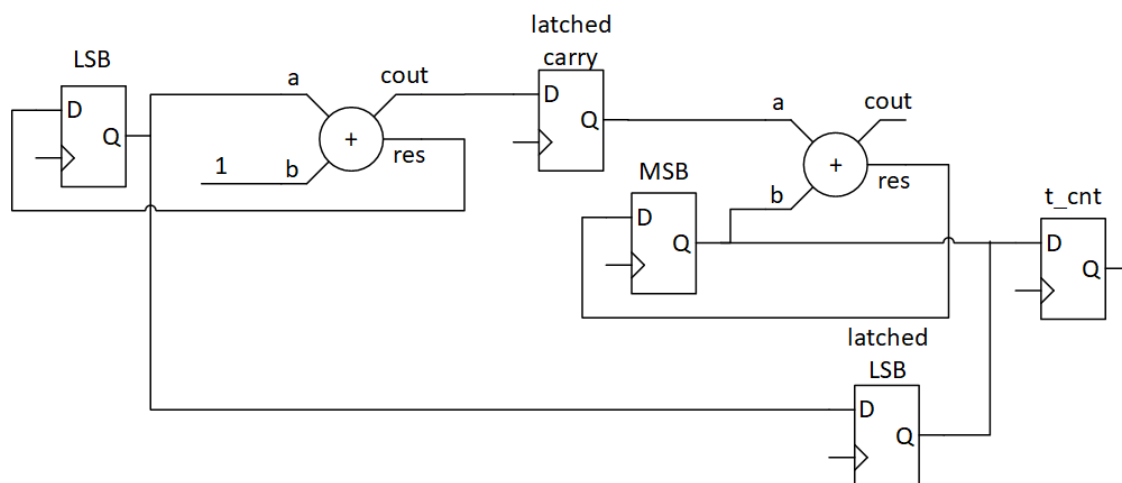


Рис.3.3. Конвейерный 32-х разрядный счётчик

К младшим байтам (регистр *LSB*) каждый такт прибавляется 1, текущее значение младших байт защёлкивается в регистре *latched LSB*. При переполнении, перенос защёлкивается в регистре *latched carry*. К старшим байтам (регистр *MSB*) прибавляется сохранённый перенос из-за младших байт. Выход счётчика – комбинация значений регистров *MSB* и *latched LSB*.

#### Листинг 3.1. Реализация на языке System Verilog

```

2 // pipelined counter
3
4 logic [T_CNT_WIDTH/2-1 : 0] high_bytes = 0;
5 logic [T_CNT_WIDTH/2-1 : 0] latched_low_bytes = 0;
6 logic [T_CNT_WIDTH/2-1 : 0] low_bytes = 0;
7 logic [T_CNT_WIDTH/2-1 : 0] low_bytes_plus_1;
8 logic carry;
9
10 assign {carry, low_bytes_plus_1} = low_bytes + 1;
11
12 //incrementing low bytes

```

```
13 always_ff @(posedge clk_i, negedge arst_i)
14     if (~arst_i) low_bytes = 0;
15     else low_bytes <= low_bytes_plus_1;
16
17 //latching low bytes for 1 cycle
18 always_ff @(posedge clk_i, negedge arst_i)
19     if (~arst_i) latched_low_bytes = 0;
20     else latched_low_bytes <= low_bytes;
21
22 logic latched_carry = 0;
23
24 //latching carry
25 always_ff @(posedge clk_i, negedge arst_i)
26     if (~arst_i) latched_carry = 0;
27     else latched_carry <= carry;
28
29 //adding latched carry to high bytes
30 always_ff @(posedge clk_i, negedge arst_i)
31     if (~arst_i) high_bytes = 0;
32     else high_bytes <= high_bytes + latched_carry;
33
34 //seting t_cnt
35 always_ff @(posedge clk_i, negedge arst_i)
36     if (~arst_i) t_cnt = 0;
37     else t_cnt <= {high_bytes, latched_low_bytes};
```

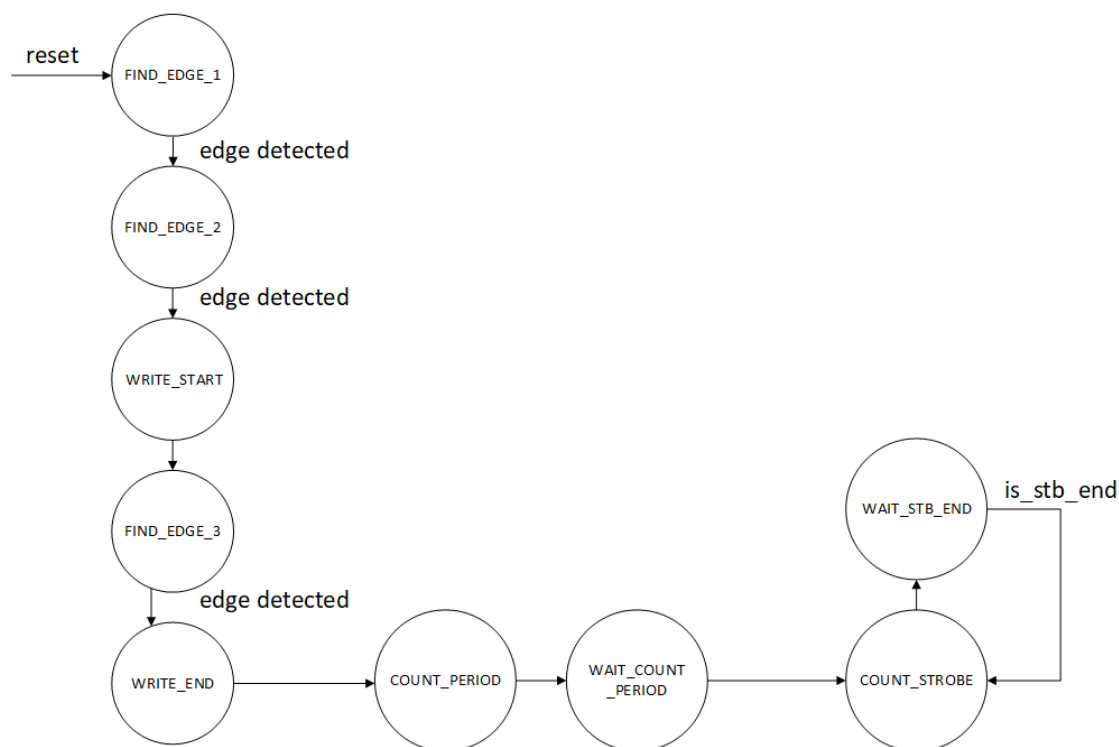


Рис.3.4. Конечный автомат модуля *stb\_gen*

На рис.3.4 представлен конечный автомат модуля *stb\_gen*. Вся логика максимально упрощена, так как, если на одной регистровой передаче будет много комбинаторной логики, при синтезе под целевую ПЛИС не получится выдержать требуемые  $t_{setup}$  и  $t_{hold}$ .

FIND\_EDGE\_1 – состояние после сброса, ожидание первого фронта на входе *sig\_i*

FIND\_EDGE\_2 – ожидание второго фронта на входе *sig\_i*

WRITE\_START – запись текущего значения  $t_{cnt}$  в  $t_{start}$

FIND\_EDGE\_3 – ожидание третьего фронта на входе *sig\_i*

WRITE\_END – запись текущего значения  $t_{cnt}$  в  $t_{end}$

COUNT\_PERIOD – вычисление периода сигнала ( $t_{end} - t_{start}$ )

WAIT\_COUNT\_PERIOD – задержка на 1 такт

COUNT\_STROBE – начало расчёта времени начала и конца отрицательного импульса на выходе стробы

WAIT\_STB\_END – ожидание конца стробы и переход к расчёту новой

Таким образом, модуль определяет частоту входного сигнала и начинает бесконечно выдавать стробы.



Помимо 32-х разрядного счётчика, конвейеризация была необходима для всех операций с 32-х битными числами. В модуле *two\_cycle\_32\_adder* реализован двухтактный сумматор, однако использованный подход идентичен тому, что используется в счётчике.

Отдельного упоминания стоит оптимизация операции проверки на равенство двух чисел. Для этого необходимо сравнить попарно все биты чисел, что выполняется параллельно и не должно увеличивать максимальный путь, однако при синтезе получалась цепочка с последовательным сравнением всех разрядов (рис.??).

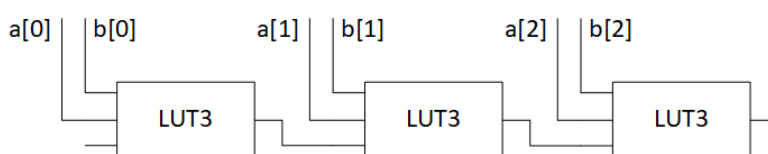


Рис.3.5. Конечный автомат модуля *stb\_gen*

Для исправления этого пришлось явно заменить операцию сравнения на исключающее ИЛИ и свёртку по ИЛИ-НЕ.

### Листинг 3.2. Реализация сравнения

```

2 | always_ff @(posedge clk_i) is_zero_hold_start_lo <= ~(t_cnt
   | [15:0] ^ latched_zero_hold_res[15:0]); //t_cnt ==
   | latched_zero_hold_res;
3 | always_ff @(posedge clk_i) is_zero_hold_start_hi <= ~(
   | t_cnt[31:16] ^ latched_zero_hold_res[31:16]); //t_cnt ==
   | latched_zero_hold_res;
4 | always_ff @(posedge clk_i) is_zero_hold_start <=
   | is_zero_hold_start_lo & is_zero_hold_start_hi;

```

При работе модуль не выдаёт стробы, а держит компаратор в защёлкнутом состоянии. Для запроса стробы используется пара сигналов *stb\_req\_i* и *stb\_valid\_o*. При выдаче стробы, с компаратора снимается защёлка на короткое время, после чего устанавливается обратно.

## **СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ**

WR	–	White Rabbit
PTP	–	Precision Time Protocol
УУ	–	управляющее устройство
FIFO	–	first in, first out