

## АННОТАЦИЯ

Дипломная работа: 22 с., 0 рис., 0 табл...

Это пример каркаса расчётно-пояснительной записки, желательный к использованию в РПЗ проекта по курсу РСОИ.

Данный опус, как и более новые версии этого документа, можно взять по адресу (<https://github.com/rominf/latex-g7-32>).

Текст в документе носит совершенно абстрактный характер.

# СОДЕРЖАНИЕ

ПЕРЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ . . . . .	6
ВВЕДЕНИЕ . . . . .	7
1 ПРОБЛЕМАТИКА РАЗРАБОТКИ	
МУЛЬТИСИНХРОННЫХ УСТРОЙСТВ . . . . .	8
1.1 Домен синхрочастоты . . . . .	8
1.2 Основные принципы . . . . .	9
1.3 Решение проблемы метастабильности . . . . .	10
1.4 Асинхронная очередь . . . . .	11
2 РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ И	
СТРУКТУРНОЙ СХЕМ УЗЛА . . . . .	13
2.1 Организация взаимодействия с устройством . . . . .	13
2.2 Организация памяти . . . . .	13
2.3 Тактирование . . . . .	14
2.4 Адресация . . . . .	14
2.5 функциональная схема узла . . . . .	17
ЗАКЛЮЧЕНИЕ . . . . .	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ . . . . .	21

## **ПЕРЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ**

**CDC** — Clock domain crossing. Пересечение доменов синхрочастоты

**FIFO** — .....

**FPGA** —

**MSB** — Most significant bit. Старший бит

## **ВВЕДЕНИЕ**

Целью работы является построение цифрового устройства ресинхронизации данных. Данное устройство представляет собой FIFO с восьмиразрядным входом и тридцатидвух разрядным выходом. Данные на выход подаются по мере образования во входном буфере не менее 4 8-разрядных слов. Тактовые сигналы входного и выходного каналов независимы, однако тактовая частота для выхода FIFO составляет как минимум четверть от тактовой частоты входного интерфейса FIFO.

# 1 ПРОБЛЕМАТИКА РАЗРАБОТКИ МУЛЬТИСИНХРОННЫХ УСТРОЙСТВ

В данном разделе будут рассмотрены проблемы и особенности построение устройств

В этом разделе будут описаны проблемы, возникающие в процессе разработки мультисинхронного проекта, то есть устройства, в котором имеют место пересечения клоковых доменов или доменов синхрочастоты (CDC).

## 1.1 Домен синхрочастоты

Домен синхрочастоты представляет собой ту часть проекта, которая тактируется одной или несколькими синхрочастотами, причем все эти синхрочастоты должны иметь постоянные сдвиги фазы. Если в какой-либо части проекта имеется синхрочастота или инвертированная синхрочастота, или синхрочастота, полученная из исходной путем деления на 2, то такая часть проекта считается клоковым доменом с одной синхрочастотой. Если же домены имеют синхрочастоты с переменной фазой и соотношениями времени, то такие домены считают доменами с различными синхрочастотами.

На Рисунке 1.1 показано, что проект имеет единственный домен синхрочастоты, потому что синхрочастота `divClk` — есть деленная на два частота генератора синхронизации `Clk`.

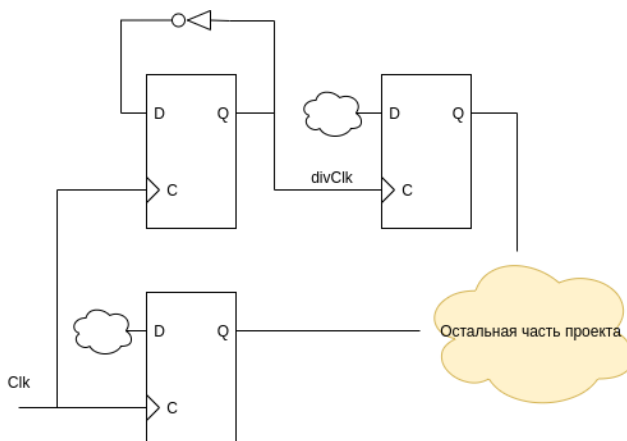
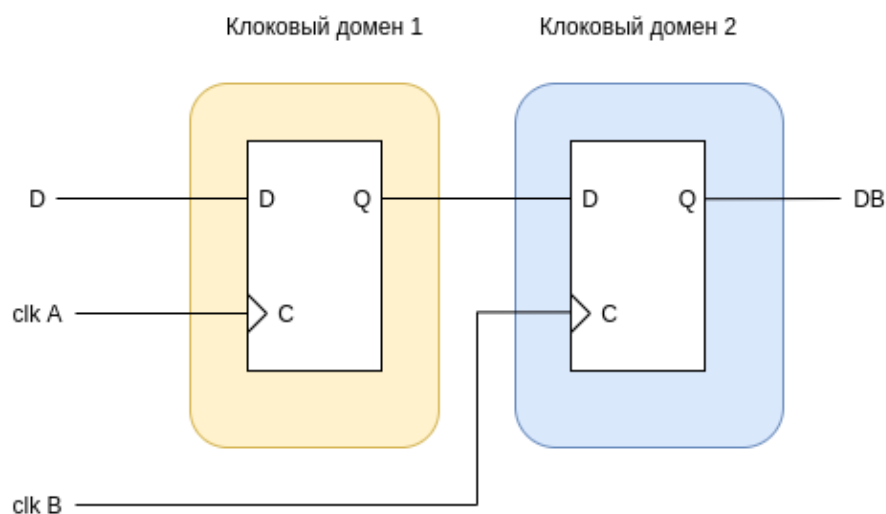


Рисунок 1.1 — Схема с одним доменом синхрочастоты

На Рисунке 1.2 показано несколько синхрочастот от различных источников. Ту часть проекта, которая управляется этими синхрочастотами, называют доменами синхрочастоты, и сигналы, осуществляющие передачу импульсов между этими асинхронными доменами синхрочастоты, называют путями пересечения домена синхрочастоты. Сигнал DA считают асинхронным сигналом в домене синхрочастоты, так как между генератором синхронизации A (clkA) и генератором синхронизации B (ClkB) не существуют постоянные соотношения фазы и времени.



**Рисунок 1.2 — Путь домена синхрочастоты**

## 1.2 Основные принципы

При разработке ультисинхрочастотных проектов следует уделять особое внимание стабильности сигнала. Когда сигнал пересекает домен синхрочастоты, то он появляется в новом домене синхрочастоты как асинхронный сигнал и должен быть засинхронизирован.

Синхронизация предотвращает в новом домене синхрочастоты метастабильное состояние первого запоминающего элемента схемы (триггера), и это позволяет в новом домене работать уже со стабильным сигналом.

*Метастабильность* — это неспособность триггера достигнуть известного состояния в определенный момент времени. Когда триггер входит в метастабильное состояние, то невозможно предсказать ни уровень выходного

напряжения элемента, ни период времени, за который этот выход перейдет к правильному уровню напряжения.

В течение этого переходного времени выход триггера будет находиться на некотором промежуточном уровне напряжения или колебаться и может передать этот недопустимый уровень сигнала со своего выхода к другим триггерам схемы.

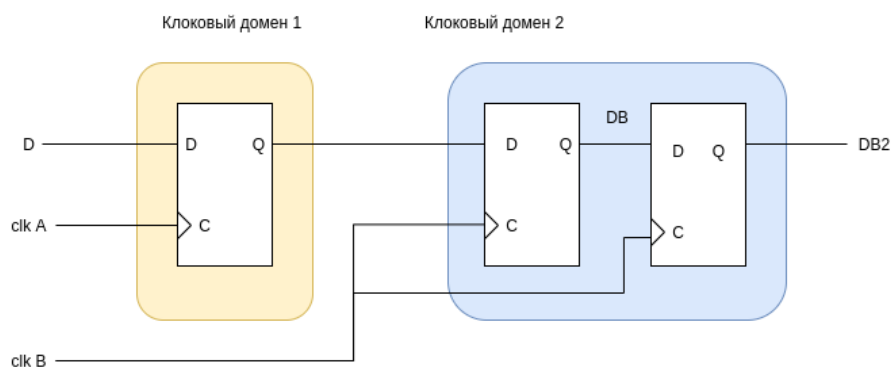
### **1.3 Решение проблемы метастабильности**

С целью решения проблемы метастабильности применяются каскады стабилизирующих триггеров, включенных последовательно. Рассмотрим это решение.

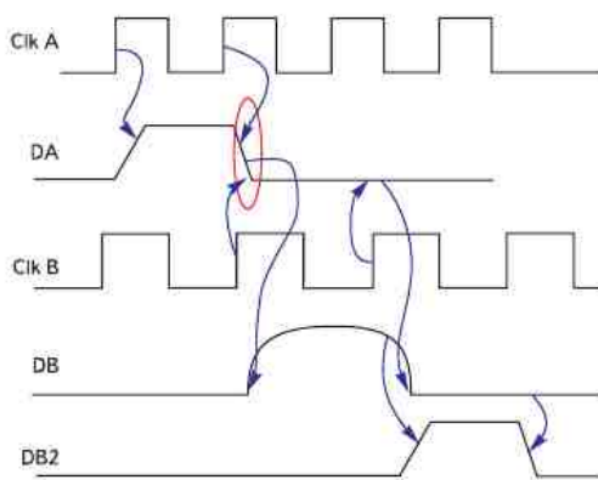
Простейший синхронизатор представляет собой два триггера, включенных последовательно без какой-либо комбинационной схемы между ними. Такая схема проекта гарантирует, что первый триггер выходит из своего метастабильного состояния, и его выход переходит в устойчивое состояние перед тем, как второй триггер сохраняет его. Необходимо также разместить эти триггеры, насколько возможно, ближе друг к другу для того, чтобы гарантировать наименьшую расфазировку тактовых сигналов между ними.

Другой тип ячейки синхронизатора представляет собой два близко расположенных триггера без какой-либо комбинационной логики между ними. Для того чтобы синхронизация работала должным образом, сигнал, пересекающий домен синхрочастоты, должен проследовать от триггера в домене синхрочастоты источника сигнала к первому триггеру синхронизатора, не пройдя через комбинационную логику.

Синхронизацию в домене называют привязкой входного сигнала к тактовой частоте домена, т. е. все изменения этого сигнала в домене будут происходить по фронту или срезу тактового сигнала домена, а не родительского тактового сигнала. На Рисунке 1.3) приведена схема традиционного синхронизатора из двух триггеров (сдвиговый регистр) для одиночного сигнала. Схема синхронизации приведена на Рисунке 1.4.



**Рисунок 1.3 — Триггеры синхронизации**



**Рисунок 1.4 — Схема синхронизации сигнала**

Данный способ борьбы с метастабильностью будет в дальнейшем использован при разработке узла ресинхронизации данных.

## 1.4 Асинхронная очередь

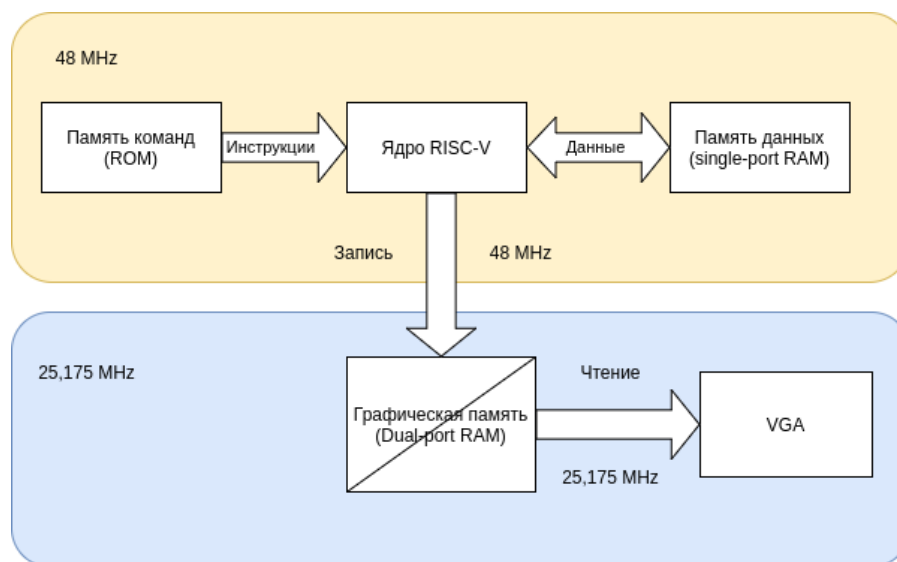
Во многих проектах необходимо передавать из одного домена в другой не только несколько одиночных сигналов, но и сигналы типа шин: шины данных, адреса и шины управления.

В этих задачах активно используются буферы типа FIFO и специальные протоколы для процедуры установления связи. Особенно полезной оказывается асинхронная очередь, позволяющая производить запись и чтение по синхросигналам двух клоковых доменов. Данная схема позволяет быстро



и удобно передавать данные в реальном времени между двумя разными областями синхронизации.

На Рисунке 1.5 приведена схема использования двухпортовой оперативной памяти, позволяющей осуществлять передачу данных между частями устройства, использующими синхросигналы разной частоты.



**Рисунок 1.5 — Двухпортовая оперативная память**

Двухпортовая память с независимыми тактовыми сигналами — аппаратный блок, используемый в современных FPGA и в технологических библиотеках для заказных микросхем. Чтение и запись производится совершенно независимо через два отдельных порта.

Единственным ограничением является одновременное обращение на запись и чтение по одному и тому же адресу памяти — оно может привести к неопределенному результату. На основе такого блока памяти зачастую создается модуль FIFO, который позволяет с одной стороны записывать данные из одного тактового домена, а с другой — забирать в другой тактовый домен. Заодно логика FIFO следит за тем, чтобы не происходило обращения к одной и той же ячейке памяти.

Данный узел работает по следующему принципу.

1. Источник данных загружает данные в асинхронную очередь. Тактирование производится по тактовому сигналу источника данных.

2. Потребитель данных считывает данные из асинхронной очереди в исходном порядке. Тактирование производится по тактовому сигналу потребителя данных.

## **2 РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ И СТРУКТУРНОЙ СХЕМ УЗЛА**

К разработке предлагается устройство ресинхронизации данных. Определим структурные элементы устройства, принцип и порядок его работы.

### **2.1 Организация взаимодействия с устройством**

Взаимодействие с устройством производится посредством передачи восьмибитного слова на вход `wdata`. Аппаратный блок осуществляет запись данных во внутреннюю память по сигналам тактового генератора домена источника данных `wclk`, основываясь на состоянии входных сигналов `wrst`, `wen` и служебных сигналов `wr_full`, `wptr`, `rprr`.

Чтение из устройства ресинхронизации производится посредством считывания тридцатидвухразрядного слова, генерируемого на выходе `rdata`. Аппаратный блок осуществляет чтение данных из внутренней памяти по сигналам тактового генератора домена получателя данных, основываясь на состоянии входных сигналов `rrst`, `ren`, `r_empty`, `wptr`, `rprr`, `rdredy`.

### **2.2 Организация памяти**

Устройство хранит данные в собственной памяти. Устройство может одновременно хранить шестнадцать восьмибитных слов, используя для их адресации пятиразрядный адрес, в котором старший бит призван реализовать возможность контроля заполнения и опустошения очереди.

Возможность асинхронного сброса памяти не реализована в силу того, что обычно память в устройствах, реализуемых на ПЛИС представлена в виде блочной оперативной памяти и выделенной оперативной памяти. Эти ресурсы обычно не поддерживают сброс. Сброс возможно реализовать в регистрах.

## 2.3 Тактирование

Управление устройством производится по двух синхросигналам `wclk` и `rclk`. С целью оптимальной работы цифрового устройства при передаче данных из одного домена синхросигнала в другой тактовые сигналы входного и выходного каналов сделаны независимыми, однако тактовая частота для выхода очереди составляет как минимум  $\frac{1}{4}$  от тактовой частоты входного интерфейса FIFO.

Для корректной передачи и предотвращения потери данных, а также с целью борьбы с метастабильностью в случае независимых тактовых сигналов используются группа синхронизирующих триггеров, которые передают текущее состояние адреса чтения и адреса записи между блоковыми доменами. Основываясь на значениях данных адресов делается вывод о заполненности и пустоте очереди.

## 2.4 Адресация

Адресация в памяти реализована с помощью двух пятиразрядных переменных, хранящих адрес ячеек, с которыми будут произведены операции чтения и записи при следующем синхросигнале.

Важной частью работы устройства является работа с памятью. Необходимо избежать потери и перезаписи не переданных данных. С этой целью на каждом шаге работы схемы осуществляется проверка и сравнение адресов чтения и записи.

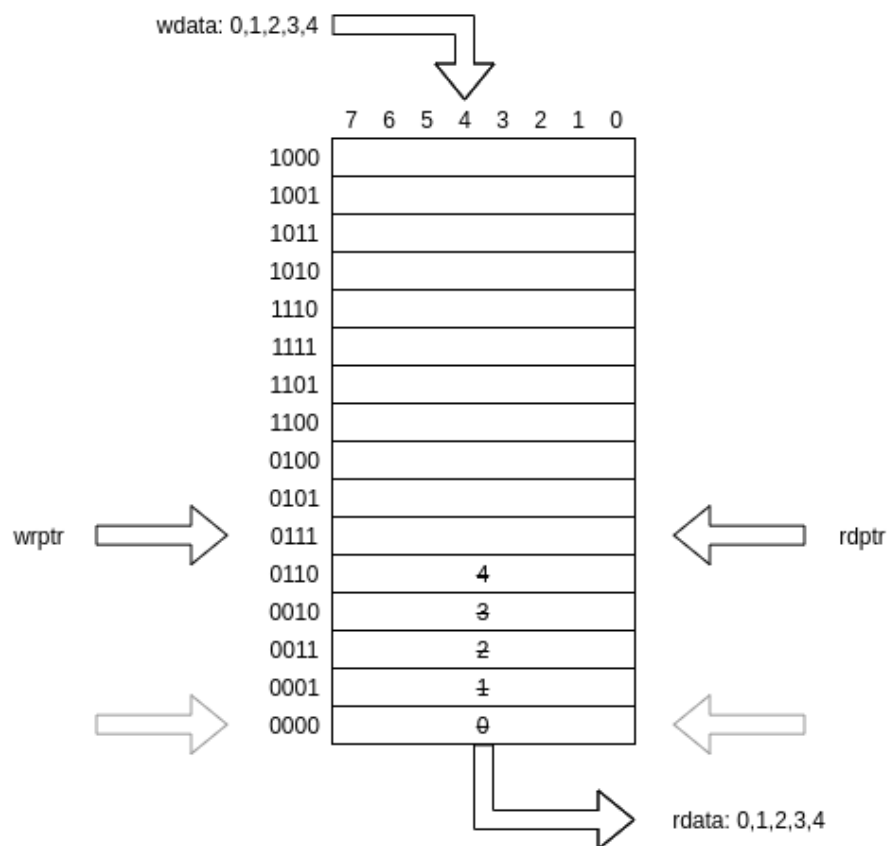
Во время работы схема хранит адреса в виде рефлексивного двоичного кода Грея. Выбор в сторону кода Грея сделан для снижения критичности ошибки определения адреса чтения или записи при его передаче.

В случае использования кода Грея при переходе к следующему адресу изменяется только один бит, и даже если схема не сможет верно определить значение адреса — посчитает, что адреса не изменился, это не приведет к нарушению работы системы.

В то время как при использовании прямого двоичного кода ошибка синхронизации неизбежна. При переходе могут изменяться все биты адреса

(например при переходе 01111 → 1000). В данном случае значение адреса является полностью неопределенным. Это может привести к нарушению в работе устройства в целом.

Когда указатели чтения и записи равны, это указывает на то, что FIFO пуст. Эта ситуация возникает во время операции сброса или когда было произведено чтение последнего слова в FIFO, как показано на Рисунке 2.1.

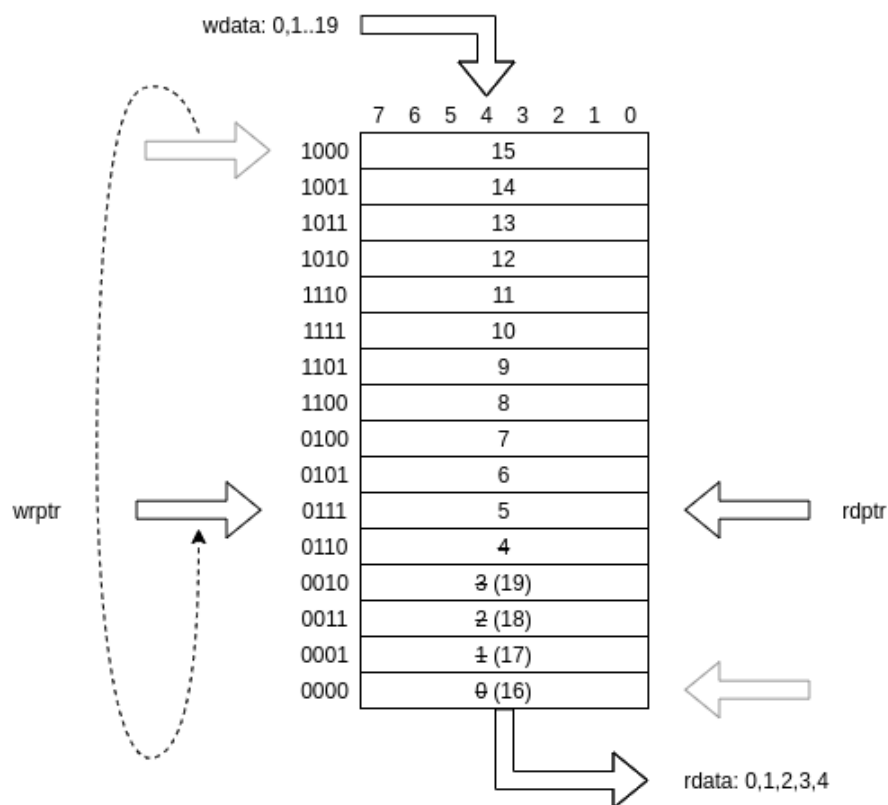


**Рисунок 2.1 — Все данные прочитаны. Очередь пуста**

Когда указатели чтения и записи снова равны, это означает, что очередь заполнена и дальнейшая запись не может быть произведена. Эта ситуация возникает, когда переменная адреса записи переполняется, как показано на Рисунке 2.2.

В обоих случаях адреса записи и чтения указывают на одну и ту же ячейку памяти, но при этом ситуации являются противоположными, что усложняет логику определения текущего состояния памяти цифрового устройства.

С целью определения текущего состояния асинхронной очереди необходимо использовать на один бит больше, чем требуется для адресации всех ячеек запоминающего устройства. Сравнение этого бита даст ответ о состоянии памяти в случае, когда указатели чтения и записи окажутся равны.



**Рисунок 2.2 — Очередь заполнена. Запись не допустима**

Пятиразрядные сигналы `wr_full` и `rd_empty`, позволяющие организовать корректную работу с устройством, формируются с помощью следующих логических условий.

$$wr\_full = (wptr == \{!rptr[4:3], rptr[aw - 2:0]\}); \quad (2.1)$$

$$rd\_empty = (wptr == rptr) \quad (2.2)$$

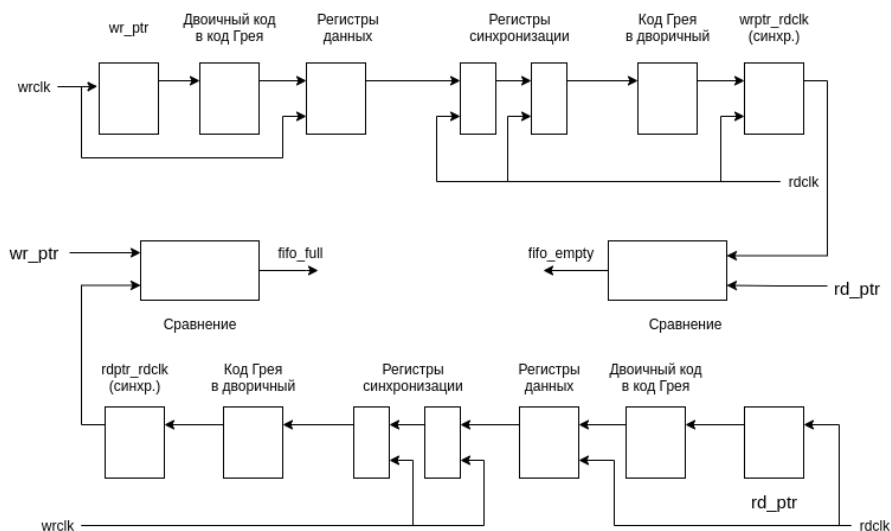
В устройстве необходимо реализовать проверку обращения к одному элементу памяти. Чтение по данному адресу должно быть заблокировано до тех пор, пока адрес записи не перестанет указывать на данную ячейку памяти.

При каждом приходящем тактовом сигнале устройство осуществляет сравнение адресов чтения и записи, предварительно производя с ними промежуточные действия. Приведем список преобразований и производимых операций.

- перевод из прямого двоичного кода в рефлексивный код Грея;
- сохранение текущего значения в регистре;
- передача значения адреса в синхронизационные регистры;
- перевод из рефлексивного кода Грея в прямой двоичный код;

- синхронизация приходящего адреса генератором соседнего домена синхронизации;
- сравнение адресов чтения и записи.

Данные операции производятся с адресом чтения и записи отдельно. Порядок следования этих операций можно пронаблюдать на Рисунке 2.3.

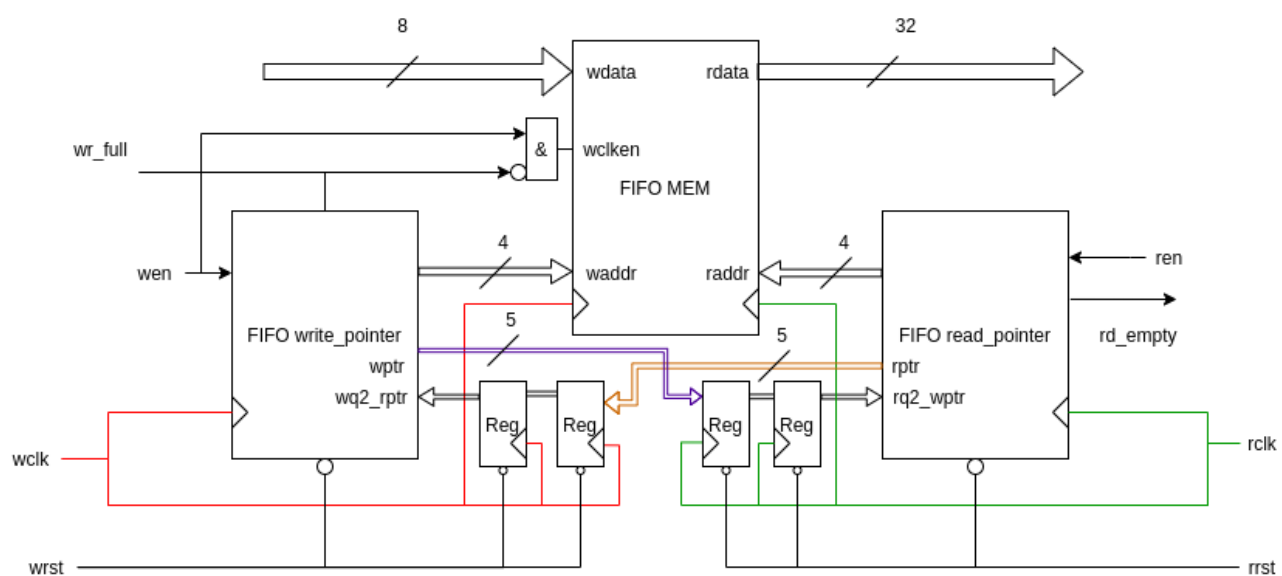


**Рисунок 2.3 — Обработка адресов чтения и записи**

## 2.5 функциональная схема узла

Предлагается для построения устройства ресинхронизации данных, работающего в режиме очереди с независимыми тактовыми сигналами, использовать данную функциональную схему (см. Рисунок 2.4).

Приведем описание входов, выходов и служебных сигналов устройства.



**Рисунок 2.4 — Функциональная схема устройства**

Таблица 2.1

Название	Назначение
asize	размер адреса данных с учетом адресации MSB
dsize	размер данных в битах
mem	ячейки памяти очереди
rbin	текущий адрес чтения в бинарном коде
rbinnext	адрес чтения в бинарном коде в следующем такте
rclk	тактирование на чтение данных
rdata	данные на чтение
rden	разрешение на чтение
rdredy	готовность выдать данные
rempty	сигнал пустоты очереди (прочтено все, что записано)
rempty_next	пустота очереди на следующем такте
rgray	текущий адрес чтения в коде Грея
rgraynext	адрес чтения в коде Грея в следующем такте
rq1_wgray	адрес чтения в первом синхро-триггере
rq2_wgray	адрес чтения во втором синхро-триггере
rrstn	сброс указателя чтения
wbin	текущий адрес записи в бинарном коде
wbinnext	адрес записи в бинарном коде в следующем такте
wclk	Тактирование на запись данных
wdata	данные на запись
wfull	сигнал заполнения очереди
wfull_next	заполненность очереди на следующем такте
wgray	текущий адрес записи в коде Грея
wgraynext	адрес записи в коде Грея в следующем такте
wq1_rgray	адрес записи в первом синхро-триггере
wq2_rgray	адрес записи во втором синхро-триггере
wren	разрешение на запись
wrstn	сброс указателя записи



## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения данных практических работ были получены знания по написанию запросов на диалекте MySQL, отработаны навыки работы в средах ER Assistant и ERwin Data Modeler, навыки работы с интерфейсом командной строки MySQL CLI Client, а также с инструментом для визуального проектирования баз данных MySQL Workbench.

Полученные знания были применены на практике для построения проектов баз данных, физические и логические модели баз данных. В последствии данные базы данных были реализованы в СУБД MySQL. Также было создано пользовательское веб – приложение, представляющее удобный интерфейс для взаимодействия с информационной системой и базой данных.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гончаровский, Олег Владленович. МАЖОРИТАРНАЯ СХЕМА С НЕСКОЛЬКИМИ ДОМЕНАМИ СИНХРОНИЗАЦИИ / Олег Владленович Гончаровский, Сергей Феофентович Тюрин // *Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии*. — 2017. — № 2. — С. 57–60.
2. Ушенина, Инна. Использование временных ограничений PERIOD и OFFSET при проектировании цифровых устройств на ПЛИС фирмы Xilinx / Инна Ушенина // *Компоненты и технологии*. — 2013. — № 5. — С. 97–106.
3. Hata, Hisashi. FPGA implementation of metastability-based true random number generator / Hisashi Hata, Shuichi Ichikawa // *IEICE TRANSACTIONS on Information and Systems*. — 2012. — Vol. 95, no. 2. — Pp. 426–436.
4. Cummings, Clifford E. Simulation and synthesis techniques for asynchronous FIFO design / Clifford E Cummings // SNUG 2002 (Synopsys Users Group Conference, San Jose, CA, 2002) User Papers. — 2002.

[1]. [2].[3], [4]