|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ |
| ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ высшего образования |
| **«Национальный исследовательский ядерный университет «МИФИ»** |
| ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ |
| КАФЕДРА «КОМПЬЮТЕРНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ» (№12) |

**КУРСОВАЯ РАБОТА**

**по дисциплине**

**СХЕМОТЕХНИКА ЦИФРОВЫХ УСТРОЙСТВ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Тема: Модуль протокольного обмена** | | | | |
| Студент | Птушко Владислав Сергеевич, Скороходов Данил Андреевич | | Группа | С20-501 |
|  | ФИО | |  |  |
| Руководитель | | Решетько Валерий Михайлович | | |
|  | | ФИО | | |

**Москва, 2022**

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc122953744)

[1. СПЕЦИФИКАЦИЯ 4](#_Toc122953745)

[1.1. Формат обмена данных 5](#_Toc122953747)

[2. СИГНАЛЫ ИНТЕРФЕЙСА И ИХ ОПИСАНИЕ 8](#_Toc122953746)

[2.1. Пояснения к сигналам 9](#_Toc122953747)

[2.1.1. Входные сигналы 9](#_Toc122953748)

[2.1.2. Выходные сигналы 9](#_Toc122953749)

[3. ОТЧЁТ О ВЕРИФИКАЦИИ 1](#_Toc122953750)1

[3.1. Верификация модуля 1](#_Toc122953751)1

[3.2. Проверка согласования с соседними модулями 1](#_Toc122953752)4

[4. СИНТЕЗ СХЕМЫ И ВРЕМЕННЫЕ ХАРАКТЕРИСТИКИ](#_Toc122953753) 15

[ЗАКЛЮЧЕНИЕ 17](#_Toc122953754)

# ВВЕДЕНИЕ

Модуль протокольного обмена, разработанный в курсовой работе, — часть функционального генератора сигналов и векторного анализатора сигналов. Он используется для получения данных от хоста и для их передачи на хост. Модуль протокольного обмена разделен на 2 части: 1 часть, отвечающая за приём данных от хоста и 2 часть, отвечающая за взаимодействие с функциональными модулями проекта.

# 1. СПЕЦИФИКАЦИЯ

Принцип функционирования модуля протокольного обмена представлен на рисунке 1.1 для анализатора и 1.2 для генератора. Сначала модуль считывает данные с эластичного буфера FIFO. Далее команды преобразуются в управляющие последовательности сигналов WISHBONE, а данные с помощью интерфейса WISHBONE передаются для обработки на конкретный ведомый модуль.

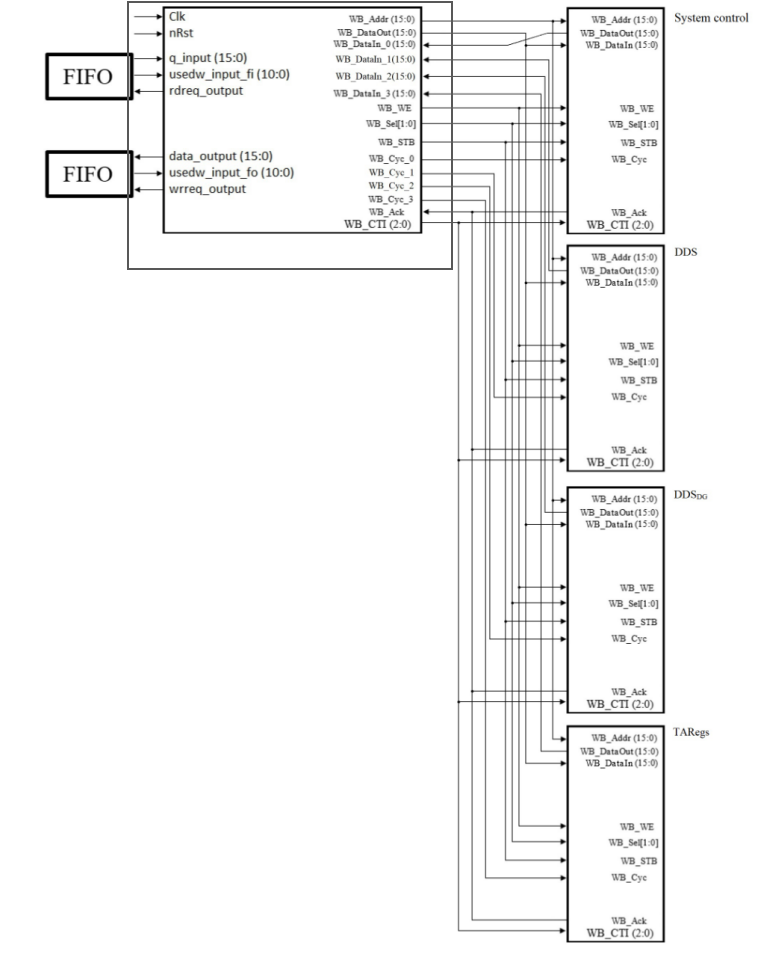


Рис. 1.1. УГП модуля протокольного обмена для векторного анализатора сигналов . (Часть, отвечающая за взаимодействие с функциональными модулями проекта)

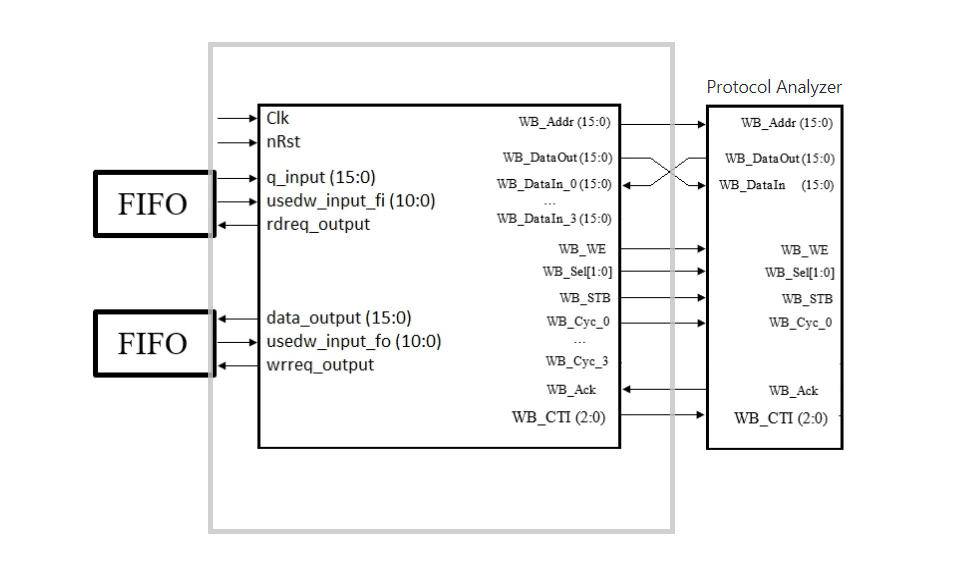


Рис. 1.2. УГП модуля протокольного обмена для функционального генератора сигналов . (Часть, отвечающая за взаимодействие с функциональными модулями проекта)

**1.1.** **Формат обмена данных**

Тип обмена данных — пакетный, обмен осуществляется 16-битными словами. Обобщенный формат пакета представлен в таблице 1.1.

Таблица 1.1— Обобщенный формат пакета

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 15 |  |  |  |  |  | 0 |  |
| Header\_W0 | | | | | | | 0H |
| Header\_W1 | | | | | | | 1H |
| Header\_W2 | | | | | | | 2H |
| Data\_W0 | | | | | | | 0D |
| … | | | | | | | … |
| … | | | | | | | … |
| Data\_WN-1 | | | | | | | ND-1 |

Пакет включает в себя две субъединицы — заголовок (Header) и данные (Data). Длина заголовка — 3 слова. Длина набора данных зависит от величины, указанной в заголовке пакета.

Заголовок пакета используется для описания задачи по обмену информации или управления. Размещение полей заголовка представлено в таблице 1.2.

Таблица 1.2 — Размещение полей заголовка

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 |  |  |  |  |  |  |  |  | 6 | 5 | 4 | 3 | 2 |  | 0 |
| BCount(9:0) | | | | | | | | | | R | | FB | Cmd(2:0) | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 31 |  |  |  | 27 | 26 |  | 24 | 23 |  |  |  |  |  |  | 16 |
| R | | | | | AddrValid | | | TID(7:0) | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 47 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 32 |
| Addr(15:0) | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Описание полей заголовка:

**Cmd(2:0)** — **C**o**m**man**d**, команда. От вида команды зависят дальнейшие операции с данными и последующие действия устройства в целом. Типы команд описаны в таблице 1.3.

Таблица 1.3 — Возможные типы команд

| Команда | Описание |
| --- | --- |
| “000” | reserved |
| “001” | Чтение конфигурационных данных |
| “010” | Запись конфигурационных данных |
| “011” | Чтение данных из порта ввода/вывода (FIFO) |
| “100” | Запись данных в порт ввода/вывода (FIFO) |
| “101” | Чтение данных из памяти |
| “110” | Запись данных в память |
| “111” | reserved |

**FB** — **F**eed**B**ack. Наличие «1» в этом поле означает, что заголовок должен быть немедленно отправлен на хост, для подтверждения приема и корректного анализа.

**R** — **R**eserved. Зарезервировано для возможного будущего использования.

**TID(7:0)** — **T**ransaction **ID**entifier. Идентификатор транзакции — номер транзакции циклически (после 255 следует 0) инкрементируется для каждой транзакции, инициированной хостом. Отклики на транзакции (для запросов на чтение, к примеру), инициированные хостом, должны иметь тот же идентификатор, что и исходный запрос хоста.

**AddrValid(2:0)** — **Addr**ess **Valid**. Поле, показывающее корректность выбора адреса. В случае возникновения ошибки в поле **Addr(15:0)** указывается размер области. Типы ошибок описаны в таблице 1.4.

Таблица 1.4 — Возможные типы ошибки выбора адреса

| Принимаемое значение | Описание |
| --- | --- |
| “000” | Адрес корректен |
| “001” | Ошибка в выборе нулевого базового адреса |
| “010” | Ошибка в выборе первого базового адреса |
| “011” | Ошибка в выборе второго базового адреса |
| “100” | Ошибка в выборе третьего базового адреса |
| “101” | Ошибка в выборе четвертого базового адреса |
| “110” | Reserved |
| “111” | Неизвестная ошибка в выборе адреса |

**BCount(9:0)** — **B**yte **Count**. Количество байт данных, которое будет передано в наборе данных или должно быть отправлено хосту при выполнении команды чтения. Количество данных *N*:



**Addr(15:0)** — **Addr**ess. Адрес назначения (источника) данных. Адресация осуществляется 16-битными словами.

# 2. СИГНАЛЫ ИНТЕРФЕЙСА И ИХ ОПИСАНИЕ

# На рисунке 2.1 представлена разрабатываемая часть модуля протокольного обмена, отвечающая за взаимодействие с функциональными модулями проекта

# 

Рис. 2.1. Разрабатываемая часть модуля протокольного обмена

Описание сигналов представлено в таблице 2.1.

Таблица 2.1. Описание сигналов

| № п.п. | Сигнал | Напр. | Описание |
| --- | --- | --- | --- |
| Системные сигналы | | | |
| 1 | Clk | in | Тактовый сигнал для BackEnd стороны шлюза. Активность — ↑ |
| 2 | nRst | in | Общий сигнал сброса. Активный уровень – ‘0’. |
| Сигналы **FIFO (входные)** | | | |
| 3 | q\_input | in | входные данные |
| 4 | usedw\_input\_fi | in | счётчик слов в fifo |
| 5 | rdreq\_output | out | запрос на чтение |
| Сигналы **FIFO (выходные)** | | | |
| 6 | data\_output | out | выходные данные |
| 7 | usedw\_input\_fo | in | счётчик слов в fifo |
| 8 | wrreq\_output | out | запрос на запись |
| Сигналы **интерфейса WISHBONE** (BackEnd – на стороне управляемых модулей) | | | |
| 7 | WB\_Addr (15:0) | out | Адрес смещения для целевого адресного пространства. Вычисляется как разность между адресом в заголовке и базовым адресом соответствующей адресной секции. |
| 8 | WB\_DataOut(15:0) | out | Поток данных, 16-битные слова. |
| 9 | WB\_DataIn (15:0) | in | Поток данных на хост, 16-битные слова. |
| 10 | WB\_WE | out | **W**rite **E**nable. Сигнал разрешения записи: высокий уровень — цикл записи в управляемые модули, низкий — цикл чтения из управляемых модулей. |
| 11 | WB\_Sel[1:0] | out | **Sel**ect. Сигналы разрешения приема/пересылки байта |
| 12 | WB\_STB | out | **St**ro**b**e. Строб-импульс, высокий уровень показывает корректность данных. |
| 13 | WB\_Cyc | out | **Cyc**le. Высокий уровень сигнала показывает выполнение корректного цикла обмена информацией. В данной реализации служит также сигналом выбора ведомого устройства. |
| 14 | WB\_Ack | in | **Ack**nowledge. Сигнал подтверждения штатного завершения пересылки элемента пакета. |
| 15 | WB\_CTI (2:0) | out | **C**ycle **T**ype **I**dentifier. В зависимости от значения сигнала возможны следующие режимы работы:   * “000” – обычный цикл; * “001” – пакетный цикл с фиксированным адресом; * “010” – пакетный цикл с инкрементируемым адресом; * “011-110” – зарезервировано; * “111” – последний пакет. |

В таблице 2.2 представлена общая карта памяти генератора сигналов.

Таблица 2.2 — адресное пространство

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Размер области, байт | Название | Описание |
| 0x0000-0x00FF | 256 | System control | Общее управление системой |
| 0x0100-0x01FF | 256 | DDS | Синтез периодических сигналов |
| 0x0200-0x02FF | 256 | QM (quadratic modulation) | Модуль организации пересылки данных средствами QPSK, 16-QAM, 8-PSK |
| 0x0300-0x03FF | 256 | DAC control | Модуль управления ЦАП |
| 0x0300-0x0FFF | 15K | reserved |  |
| 0x1000-0x17FF | 2K | DQM (data for quadratic modulation) | Данные, которые будут отправляться с помощью модуляции |
| 0x1800-0xFFFF | 46K | reserved |  |

В таблице 2.3 представлена общая карта памяти векторного анализатора.

Таблица 2.3 — адресное пространство

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Размер области, байт | Название | Описание |
| 0x0000-0x00FF | 256 | System control | Общее управление системой |
| 0x0100-0x01FF | 256 | DDS | Синтез частоты дискретизации |
| 0x0200-0x02FF | 256 | DDSDG | Синтез частот гетеродинирования |
| 0x0300-0x03FF | 256 | TARegs | Регистровое пространство анализатора трафика |
| 0x0400-0x0FFF | 15K | reserved |  |
| 0x1000-0x17FF | 2K | Data Buffers | Буферное пространство |
| 0x1800-0xFFFF | 46K | reserved |  |

Конечный автомат используется в данной схеме для контроля состояний представлен на рисунке 2.2. Обмен данными через шину WISHBONE производится только в состоянии Data transfer.

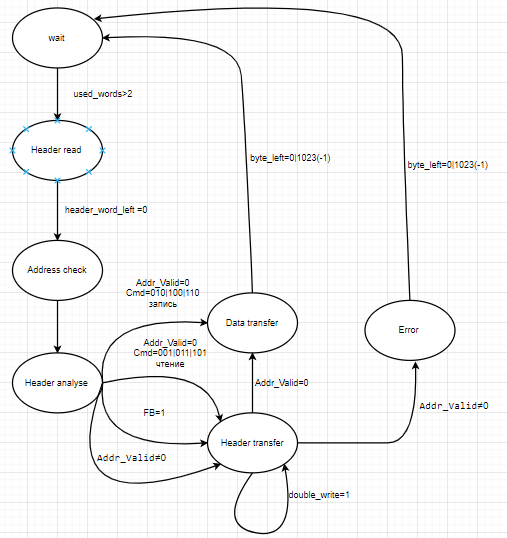


Рис. 2.2. Конечный автомат

**3. ОТЧЁТ О ВЕРИФИКАЦИИ**

# 3.1. Верификация модуля

Далее будут приведены примеры работы модуля с различными возможными типами команд (таб. 1.3.). Зарезервированные команды рассмотрены не будут так как не возможны.

А так же с **FB** и корректным **WB\_Addr (15:0)** с различными ведомыми устройствами**.**

На диаграмме (рис. 3.1.) представлен нормальный ход работы устройства в процессе обработки запроса на запись данных. В начале считывается заголовок, затем проверяется корректность адреса. Адрес соответствует модулю управления частотой дискретизации векторного анализатора сигналов. Затем формируется запрос и сигналы держатся до тех пор, пока не будет принят Ack.

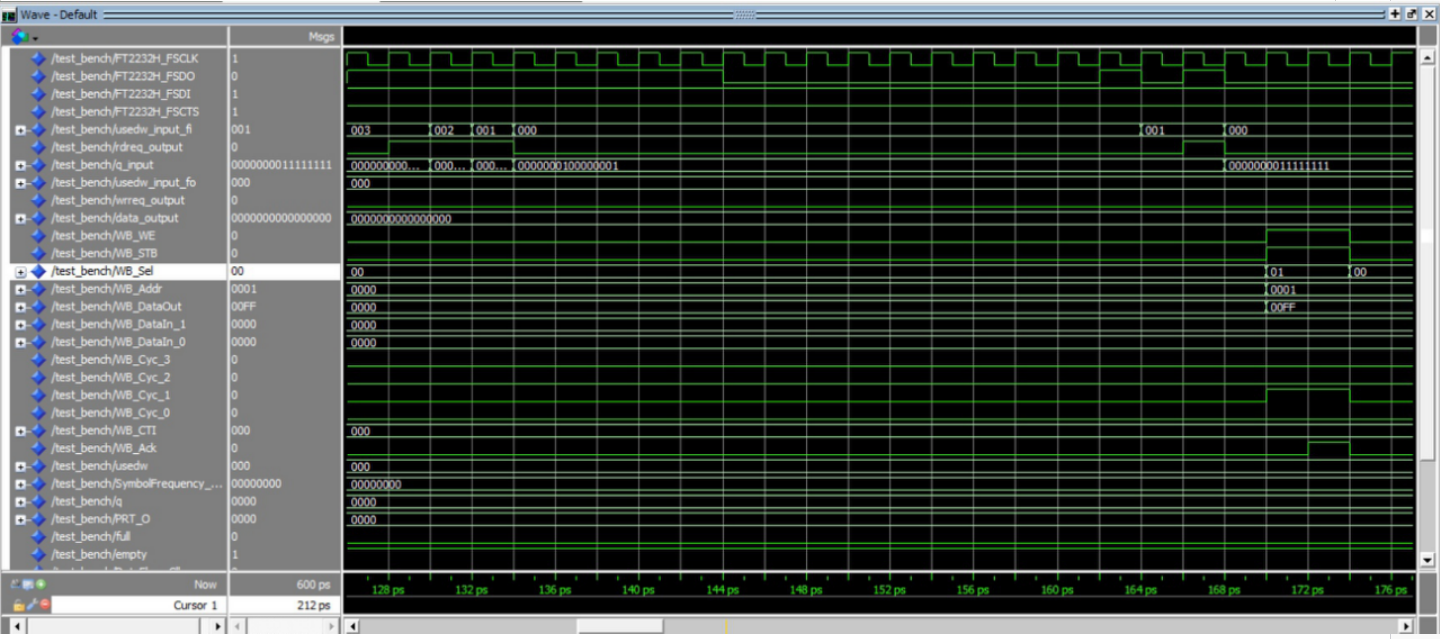


Рис. 3.1. Запись данных

На диаграмме (рис. 3.2.) представлен нормальный ход работы устройства в процессе обработки запроса на чтение данных. В начале считывается заголовок, затем проверяется корректность адреса. Адрес соответствует модулю управления частотой дискретизации векторного анализатора сигналов, в выходную fifo записывается заголовок, свидетельствующий о корректной обработке запроса. Затем формируется запрос и сигналы держатся до тех пор, пока не будет принят Ack.

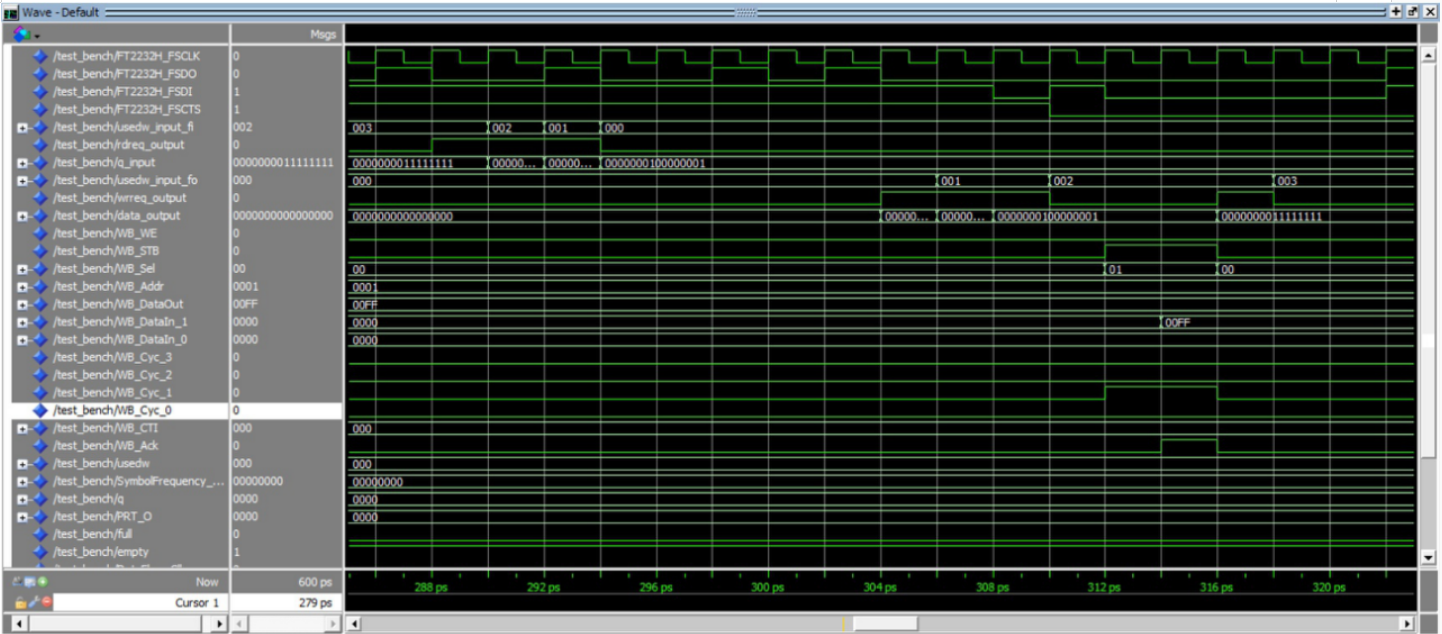


Рис. 3.2. Чтение данных

На диаграмме (рис. 3.3.) представлен нормальный ход работы устройства в процессе обработки запроса на запись данных в память. В начале считывается заголовок, затем проверяется корректность адреса. Адрес соответствует модулю управления частотой дискретизации векторного анализатора сигналов, в выходную fifo записывается заголовок, так как требовался фидбек. Затем формируется запрос и сигналы держатся до тех пор, пока не будет принят Ack.

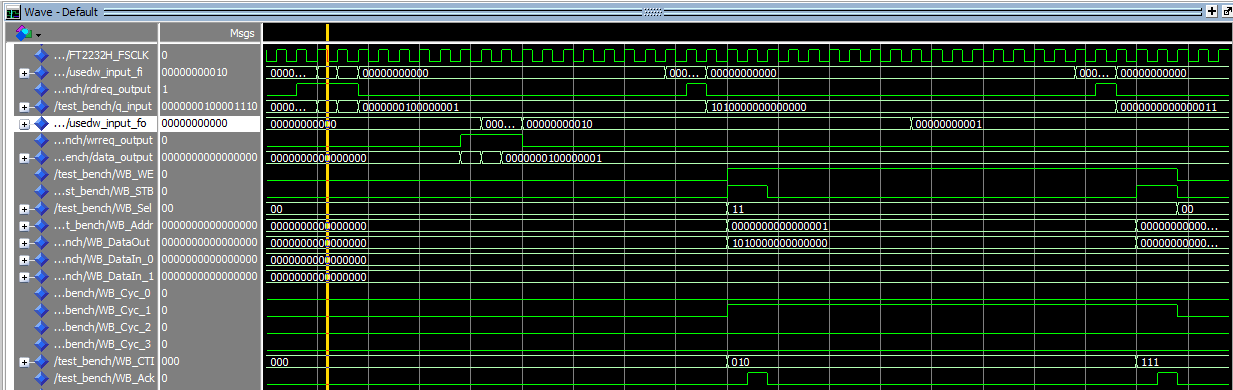


Рис. 3.3. Запись данных в память

На диаграмме (рис. 3.4.) представлен нормальный ход работы устройства в процессе обработки запроса на чтение данных из памяти. В начале считывается заголовок, затем проверяется корректность адреса. Адрес соответствует модулю управления частотой дискретизации векторного анализатора сигналов, в выходную fifo записывается 2 заголовка, так как это запрос на чтение и требовался фидбек. Затем формируется запрос и сигналы держатся до тех пор, пока не будет принят Ack.

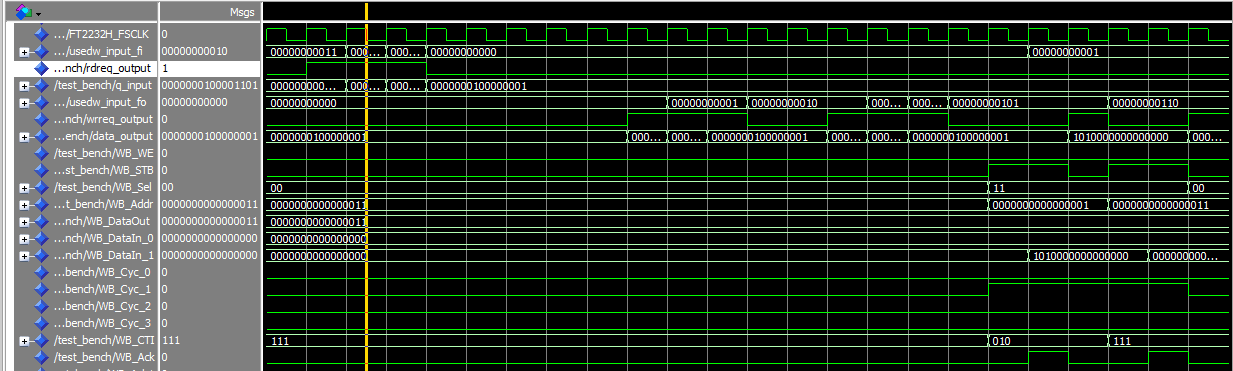


Рис. 3.4. Чтение данных из памяти

На диаграмме (рис. 3.5) представлен нормальный ход работы устройства в процессе обработки некорректного запроса на чтение данных из памяти. В начале считывается заголовок, затем проверяется корректность адреса. Адрес некорректный, в выходную fifo записывается заголовок, свидетельствующий о некорректном запросе.

# 

Рис. 3.5. Запрос с некорректным адресом(c фидбеком)

На диаграмме (рис. 3.6.) представлен нормальный ход работы устройства в процессе обработки некорректного запроса на чтение данных из памяти. В начале считывается заголовок, затем проверяется корректность адреса. Адрес некорректный, в выходную fifo записывается заголовок, свидетельствующий о некорректном запросе.

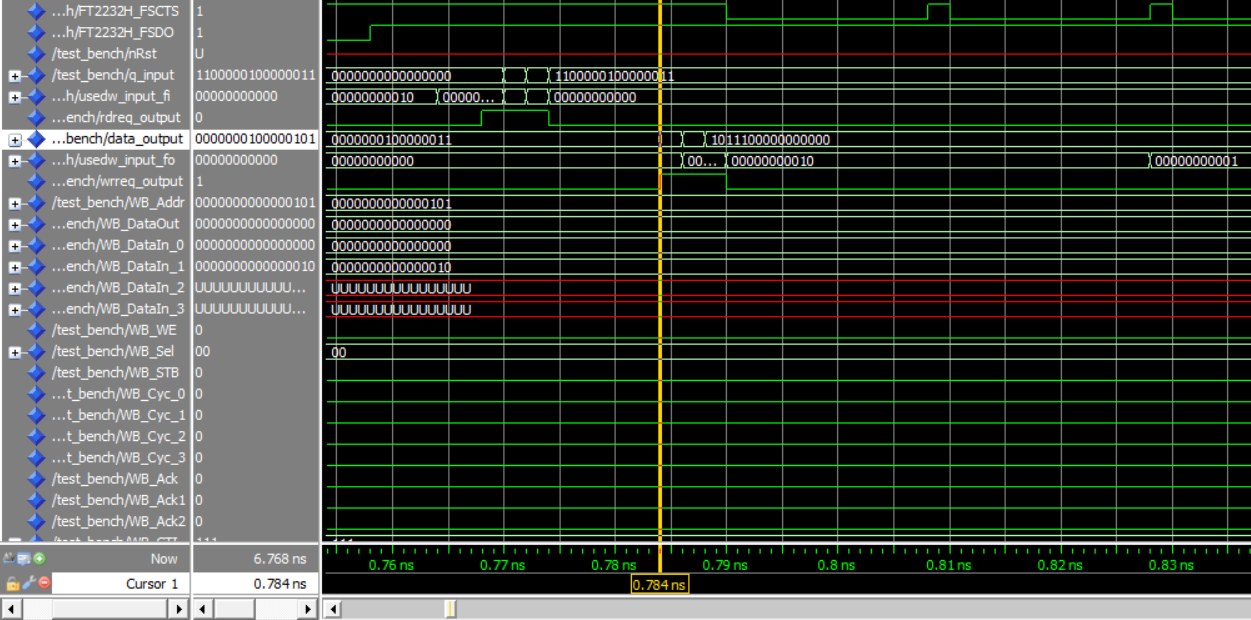


Рис. 3.6. Запрос с некорректным адресом(без фидбеком)

На диаграмме (рис. 3.7.) представлен нормальный ход работы устройства в процессе обработки запроса на запись данных в память с максимально допустимым количеством байт данных (256 байт) для модуля синтеза частоты дискретизации. В начале считывается заголовок, затем проверяется корректность адреса. Адрес соответствует модулю управления частотой дискретизации векторного анализатора сигналов, в выходную fifo записывается заголовок, так как требовалось подтверждение получения пакета. Затем выполняется обращение к ведомому модулю по шине WISHBONE для записи данных по указанному адресу.

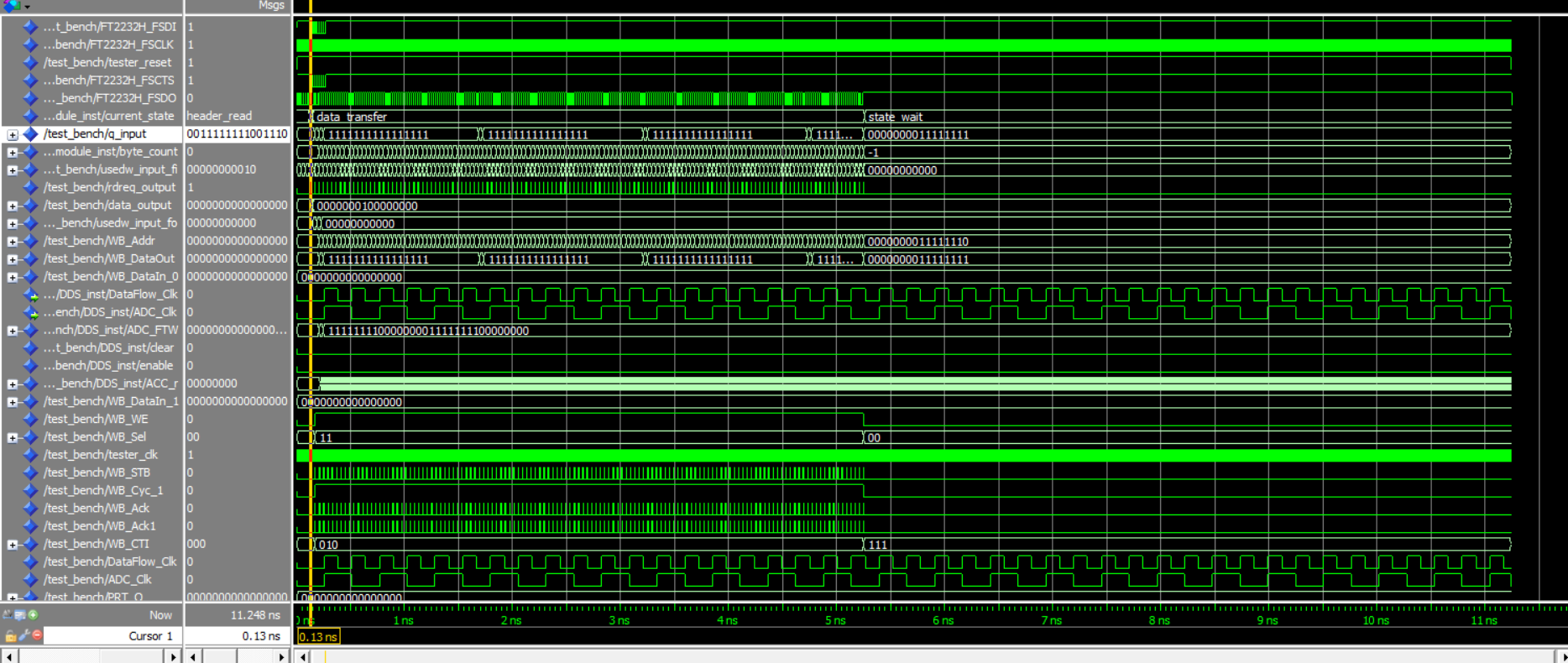


Рис. 3.7.Запись данных в память

На диаграмме (рис. 3.8.) представлен нормальный ход работы устройства в процессе обработки запроса на чтение данных в память с максимально допустимым количеством байт данных (256 байт) для модуля синтеза частоты дискретизации. В начале считывается заголовок, затем проверяется корректность адреса. Адрес соответствует модулю управления частотой дискретизации векторного анализатора сигналов, в выходную fifo записывается заголовок, свидетельствующий о корректной обработке запроса. Затем выполняется обращение к ведомому модулю по шине WISHBONE для записи данных по указанному адресу.

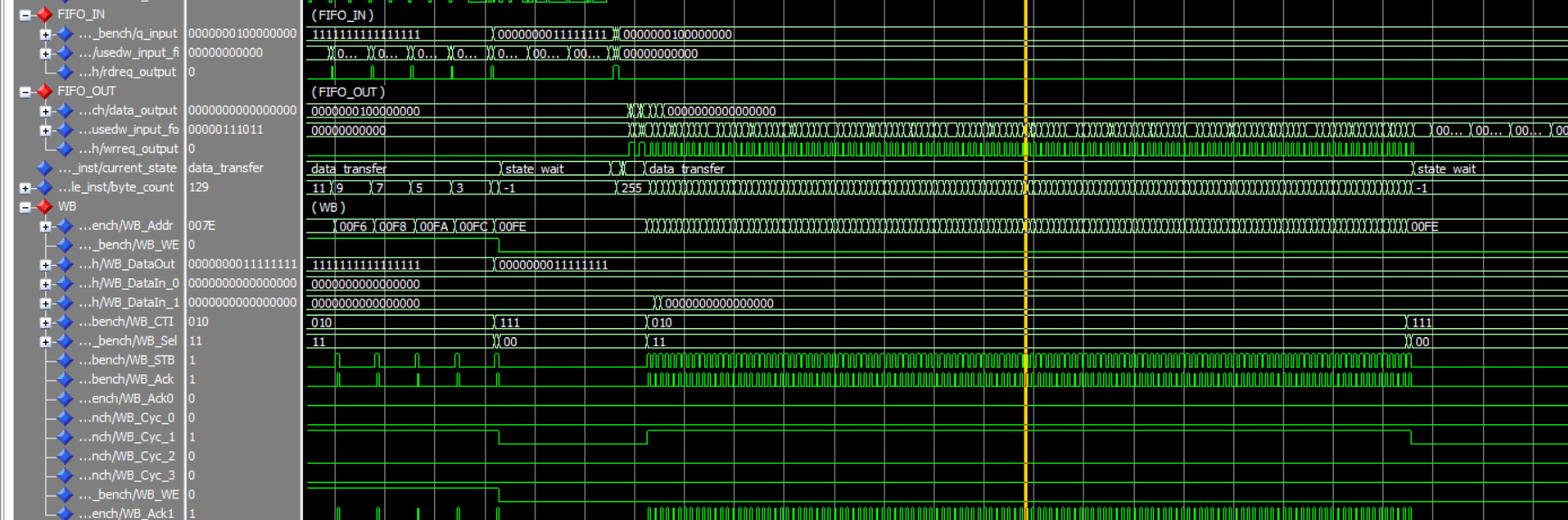


Рис. 3.8.Чтение данных в память

# 3.2. Проверка согласования с соседними модулями

Для проверки согласования и возможности интеграции в проект был собран специальный тестбенч,его ход работы представлен на диаграмме (рис. 3.9.). Он содержит три модуля: модуль протокольного обмена (Часть, отвечающая за прием данных от хоста), модуль протокольного обмена (Часть, отвечающая за взаимодействие с функциональными модулями проекта) и модуль управления частотой дискретизации. Структурная схема межсоединений показана на рисунке 3.10.

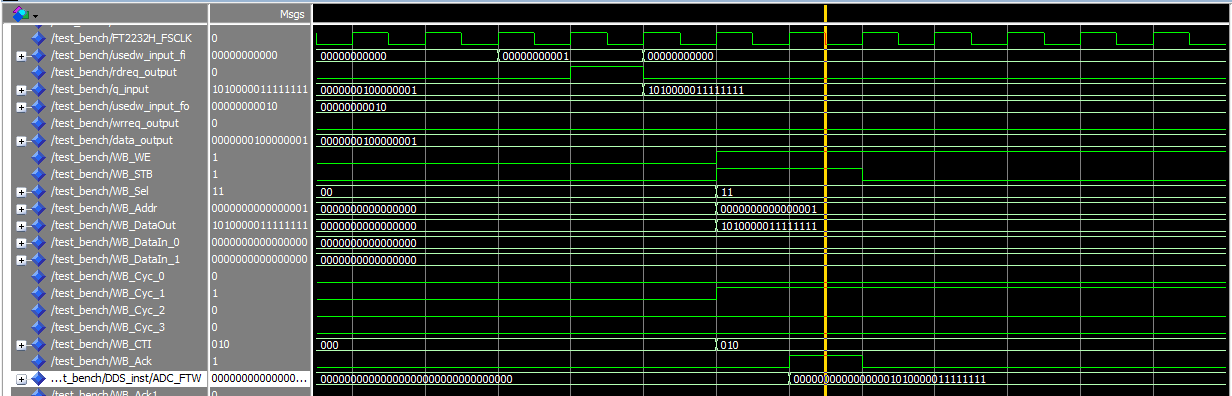


Рис. 3.9. Проверка согласования с соседними модулями.

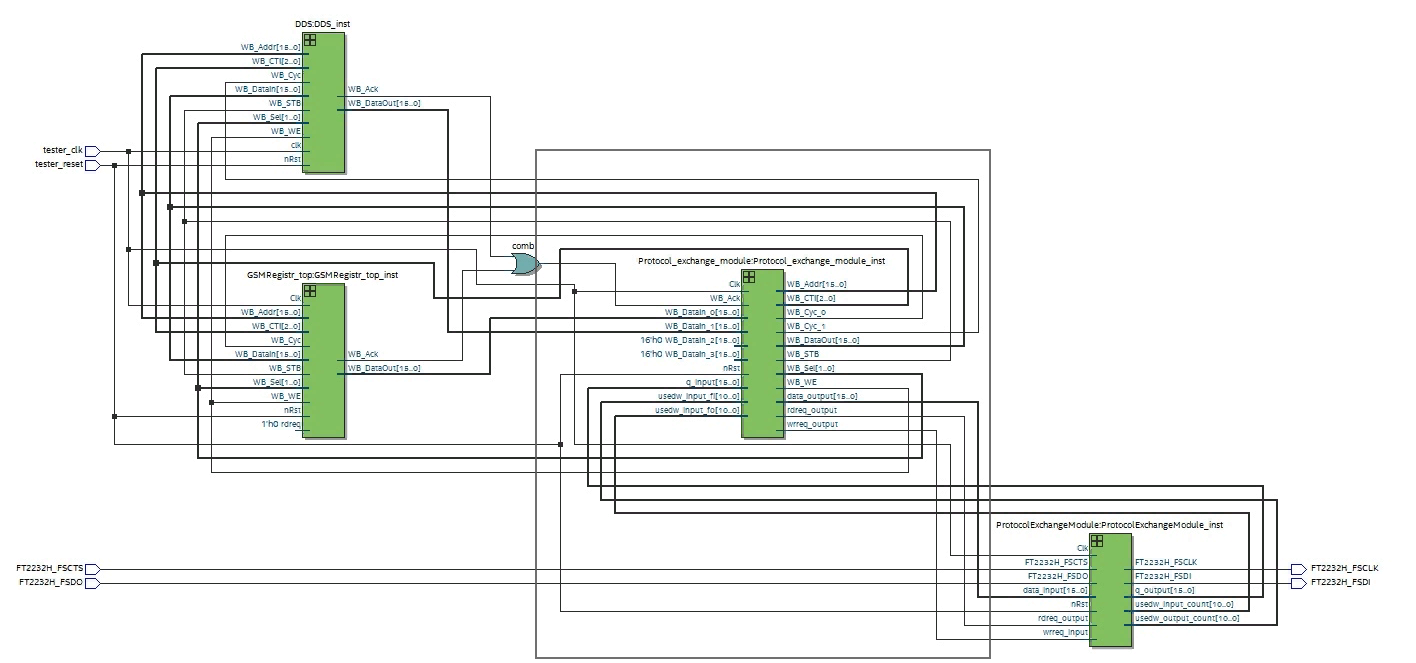


Рис. 3.10. Соединение соседних модулей в проекте.

# 4. СИНТЕЗ СХЕМЫ И ВРЕМЕННЫЕ ХАРАКТЕРИСТИКИ

Для проверки синтезируемости были проведены стадии Analysis & Synthesis, Place & Route, Generate programming files и Timing Analysis спроектированной схемы для ПЛИС семейства Cyclone 10 LP. На рисунках 4.1-4.5 представлены результаты проведённого синтеза. Все этапы синтеза пройдены успешно.

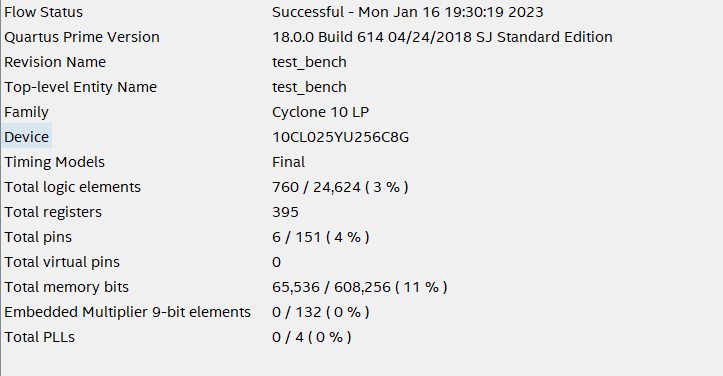


Рис. 4.1. Отчёт о занимаемых ресурсах.

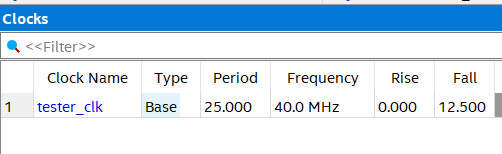


Рис. 4.2. Используемые тактовые сигналы.

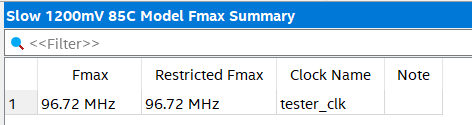


Рис. 4.3. Отчёт о временных характеристиках модуля при 85 С.

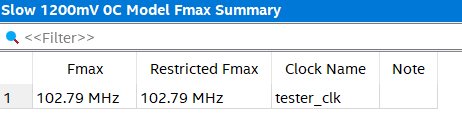


Рис. 4.4. Отчёт о временных характеристиках модуля при 0 С.

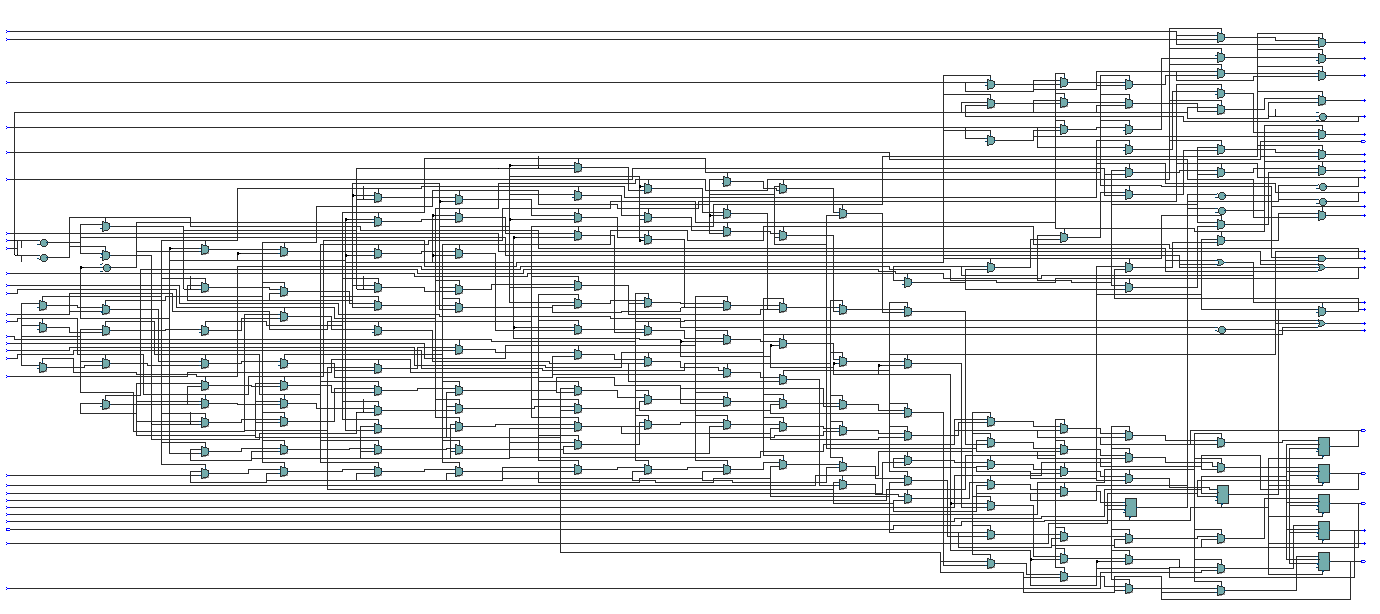


Рис. 4.5. Синтезированная схема модуля.

# ЗАКЛЮЧЕНИЕ

В процессе выполнения курсовой работы был разработан модуль протокольного обмена для получения данных от хоста и для их передачи на хост.

С помощью ModelSim 10.5b проведена симуляция работы устройства c различными входными данными, а также с соседними модулями.

Также с помощью Quartus Prime 18.0 была выполнена проверка проекта на синтезируемость для ПЛИС семейства Cyclone 10 LP.

В рамках выполнения курсовой работы были получены практические навыки в разработке проекта на языке описания оборудования VHDL, его тестирования в среде ModelSim 10.5b и синтеза в Quartus Prime 18.0.