

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра САПР

КУРСОВАЯ РАБОТА
по дисциплине «Элементная база цифровых систем»
Тема: Проектирование цифровых узлов

Студентка гр. 2308

Преподаватель

Буренёва О.И.

Санкт-Петербург

2025

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студентка

Группа 2308

Тема работы: «Проектирование цифровых узлов»

Исходные данные:

Разрядность входного кода, скорость передачи, область АП.

Содержание пояснительной записки:

«Содержание», «Введение», «Задание на проектирование узла», «Описание предлагаемых вариантов реализации узла», «Описание основных элементов библиотеки САПР Quartus II, необходимых для реализации узла», «Описание процесса синтеза и моделирования работы узла средствами САПР Quartus II», «Разработка генератора тактовых импульсов», «Разработка интерфейса сопряжения схемы узла с процессорной системой», «Заключение», «Список использованных источников».

Предполагаемый объем пояснительной записки:

Не менее 31 страницы.

Дата выдачи задания: __.__.____

Дата сдачи реферата: __.__.____

Дата защиты реферата: __.__.____

Студентка

Преподаватель

Буренёва О.И.

АННОТАЦИЯ

Данная курсовая работа посвящена разработке цифрового узла, который будет преобразовывать параллельный код в последовательный для асинхронного протокола передачи данных. Решение будет реализовано в программируемой логической интегральной схеме (ПЛИС) с использованием средств автоматизированного проектирования (САПР) Quartus II. В работе рассматриваются два варианта реализации узла: на основе параллельного регистра, а также с применением мультиплексора. Описаны ключевые элементы библиотеки САПР. Представлены этапы синтеза, моделирования и сравнительного анализа эффективности предложенных вариантов. Проведена разработка генератора тактовых импульсов и интерфейса сопряжения узла с процессорной системой. Курсовой проект будет включать в себя принципиальную электрическую схему, разработанную специально для данного узла, а также пояснительную записку.

SUMMARY

This course work is devoted to the development of a digital node that will convert parallel code into serial code for asynchronous data transmission protocol. The solution will be implemented in a programmable logic integrated circuit (FPGA) using Quartus II computer-aided design (CAD) tools. The paper considers two variants of node implementation: based on a parallel register, as well as using a multiplexer. The key elements of the CAD library are described. The stages of synthesis, modeling, and comparative analysis of the effectiveness of the proposed options are presented. A clock generator and interface for interfacing the node with the processor system have been developed. The course project will include a basic electrical circuit designed specifically for this node, as well as an explanatory note.

СОДЕРЖАНИЕ

| | |
|--|----|
| ВВЕДЕНИЕ | 5 |
| 1. ЗАДАНИЕ НА ПРОЕКТИРОВАНИЕ УЗЛА | 6 |
| 1.1 Преобразователь параллельного кода в последовательный для асинхронного протокола передач | 6 |
| 2. ОПИСАНИЕ ВАРИАНТОВ РЕАЛИЗАЦИИ УЗЛА | 7 |
| 2.1 Общие вспомогательные модули | 7 |
| 2.2 Схема на основе сдвигающего регистра | 8 |
| 2.3 Схема на основе мультиплексора | 9 |
| 3. ОПИСАНИЕ ОСНОВНЫХ ЭЛЕМЕНТОВ БИБЛИОТЕКИ САПР QUARTUS II, НЕОБХОДИМЫХ ДЛЯ РЕАЛИЗАЦИИ УЗЛА | 11 |
| 3.1 Логические примитивы | 11 |
| 3.2 Триггеры | 11 |
| 3.3 Мегафункции | 12 |
| 4. ОПИСАНИЕ ПРОЦЕССА СИНТЕЗА И МОДЕЛИРОВАНИЯ РАБОТЫ УЗЛА | 14 |
| 4.1 Синтез и моделирование узла на основе сдвигающего регистра | 14 |
| 4.2 Синтез и моделирование узла на основе мультиплексора | 18 |
| 4.3 Выбор лучшего варианта исполнения узла | 20 |
| 5. РАЗРАБОТКА ГЕНЕРАТОРА ТАКОВЫХ ИМПУЛЬСОВ | 22 |
| 5.1 Выбор варианта реализации генератора | 22 |
| 5.2 Инженерный расчет элементов генератора для получения необходимой частоты | 23 |
| 6. РАЗРАБОТКА ИНТЕРФЕЙСА СОПРЯЖЕНИЯ СХЕМЫ УЗЛА С ПРОЦЕССОРНОЙ СИСТЕМОЙ | 24 |
| 6.1 Синтез полной схемы преобразователя параллельного кода в последовательный для асинхронного протокола передач | 24 |
| ЗАКЛЮЧЕНИЕ | 27 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 28 |

ВВЕДЕНИЕ

Цель работы: приобрести знания о разнообразии возможностей современной цифровой техники при проектировании специализированных устройств, научиться эффективно применять современную элементную базу, освоить многовариантные подходы к синтезу цифровых узлов и устройств и выбор наилучшего варианта по заданным критериям качества; получить навыки в описании принципов функционирования разрабатываемых устройств и оформлении технической документации.

Задачи:

- Разработать функциональную схему проектируемого узла в нескольких вариантах;
- Разработать функциональную схемы интерфейса узла с ведущей микро процессорной системой;
- Ввод принципиальной электрической схемы узла с интерфейсным блоком для реализации в ПЛИС с помощью САПР;
- Компиляция полученного проекта;
- Тестирование разработанного узла в рамках САПР;
- Выбор наилучшего варианта узла с учетом заданного критерия качества;
- Разработка принципиальной электрической схемы типового элемента замены (ТЭЗа), содержащего спроектированный узел и вспомогательные схемы.

1. ЗАДАНИЕ НА ПРОЕКТИРОВАНИЕ УЗЛА

1.1 Преобразователь параллельного кода в последовательный для асинхронного протокола передач

Вариант 7.3:

Разработать узел, преобразующий параллельный код в последовательный с добавлением к нему старт-бита, стоп-бита и бита паритета.

Предусмотреть возможность передачи последовательного кода со скоростями, указанными в Таблице 1. Входной параллельный код, код выбора скорости передачи и сигнал пуска передаются из управляющего устройства (процессора). Критерий выбора варианта реализации схемы – минимум аппаратных затрат.

Таблица 1 – вариант задания

| Разрядность входного кода, количество бит | Скорость передачи, Кбит/с | Область АП |
|---|---------------------------|------------|
| 12 | 1,2; 2,4; 4,8 | 85h |

2. ОПИСАНИЕ ВАРИАНТОВ РЕАЛИЗАЦИИ УЗЛА

2.1 Общие вспомогательные модули

Для загрузки и сохранения состояний параллельного кода будет использоваться двенадцатиразрядный параллельный регистр. В данном модуле также предусмотрено формирование старт-бита – «0», стоп-бита – «1» и бита паритета – контрольный бит в вычислительной технике и сетях передачи данных, служащий для проверки общей чётности двоичного числа. Ставится в конец слова перед стоп-битом.

Блок-схема реализации функций и управления этим процессом приведена на Рисунке 2.1.

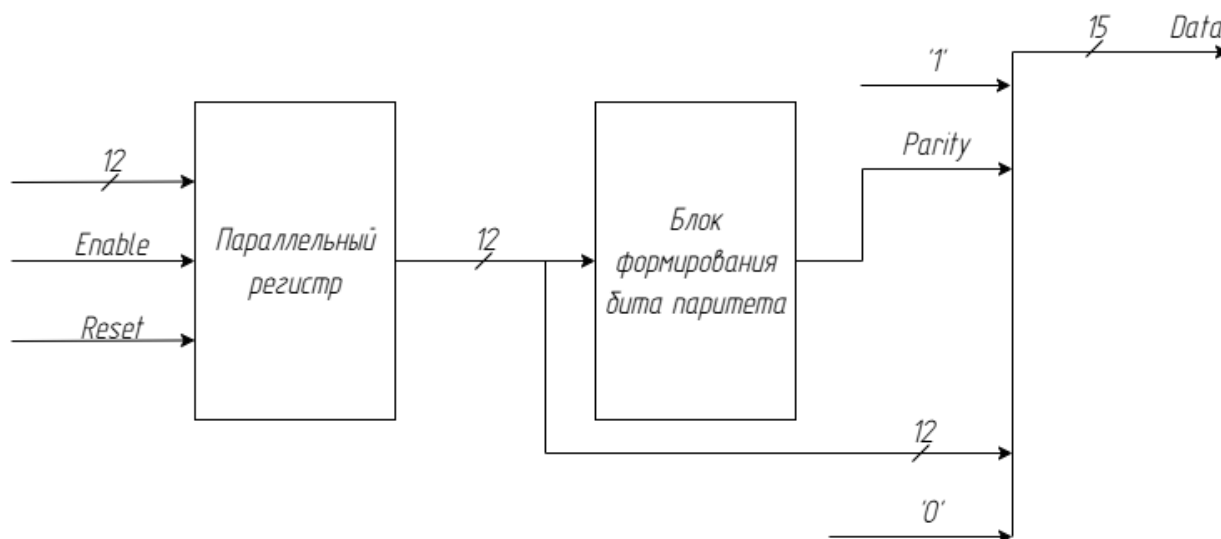


Рисунок 2.1 – Структурная схема (Э1) для хранения слова и дополнительных битов

Для загрузки и сохранения состояний кода скорости используется двухразрядный параллельный регистр. Понижение частоты реализуется через Т-триггеры в виду особенностей их работы. Выбор скорости выдачи слова осуществляется через мультиплексор.

Блок-схема реализации функций и управления этим процессом приведена на Рисунке 2.2.

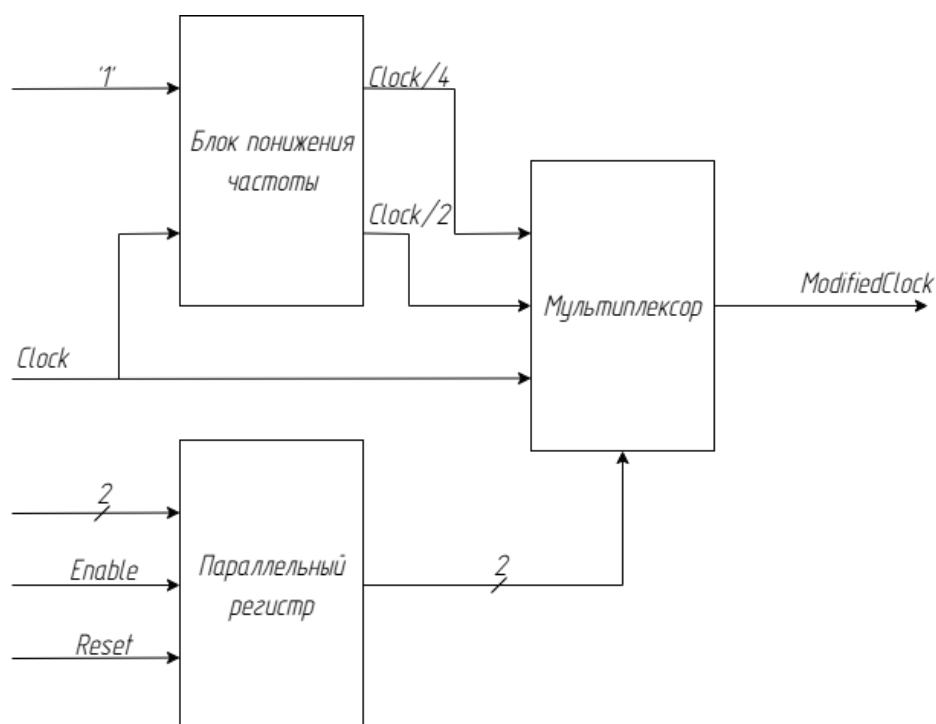


Рисунок 2.2 – Структурная схема (Э1) для хранения и формирования скорости выдачи слов

2.2 Схема на основе сдвигающего регистра

В данном способе используется параллельно-последовательный сдвигающий регистр, предназначенный для последовательной передачи данных, предварительно загруженных в него параллельно.

Управление процессом осуществляется с помощью управляющего устройства (УУ), которое принимает сигналы от программы, в том числе сигнал загрузки данных и сигнал задания скорости передачи слова.

Запуск передачи инициируется подачей сигнала Start, устанавливаемого в логическую единицу. После этого происходит параллельная загрузка 12-битного информационного слова в регистр с добавлением вспомогательных битов. Далее, в зависимости от установленной скорости, данные последовательно сдвигаются и подаются на выход схемы.

Блок-схема реализации функций и управления этим процессом от команд программы приведена на Рисунке 2.3.

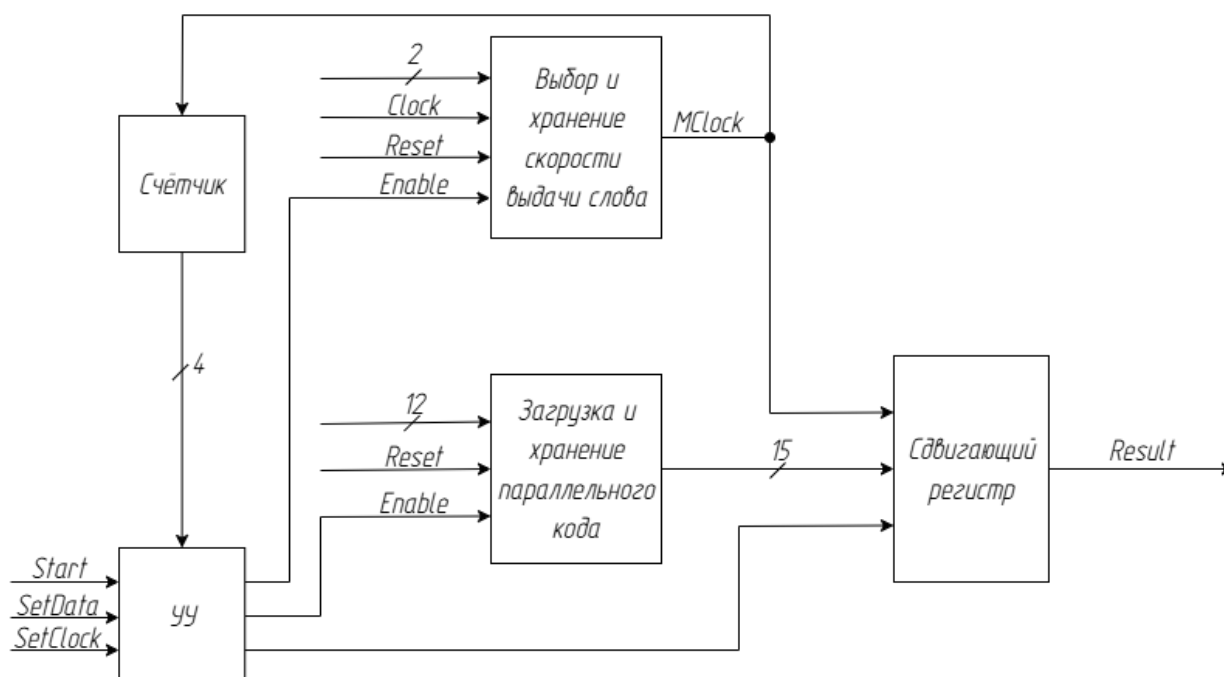


Рисунок 2.3 – Структурная схема (Э1) данного решения

2.3 Схема на основе мультиплексора

Данный способ формирования выходного потока данных заключается в использовании мультиплексора, коммутирующего разряды параллельно загруженного кода. В этой реализации каждый бит выходной последовательности формируется выборкой соответствующего разряда исходного слова по управляющему адресу.

Счётчик задаёт адресную последовательность, которая поочерёдно выбирает нужный разряд 12-битного входного слова, а также дополнительные сигналы, необходимые для формирования полного слова передачи: стартовый бит, стоповый бит и бит чётности.

УУ отвечает за загрузку данных, скорости выдачи слова и запуском передачи сигнала.

Блок-схема реализации функций и управления этим процессом приведена на Рисунке 2.4.

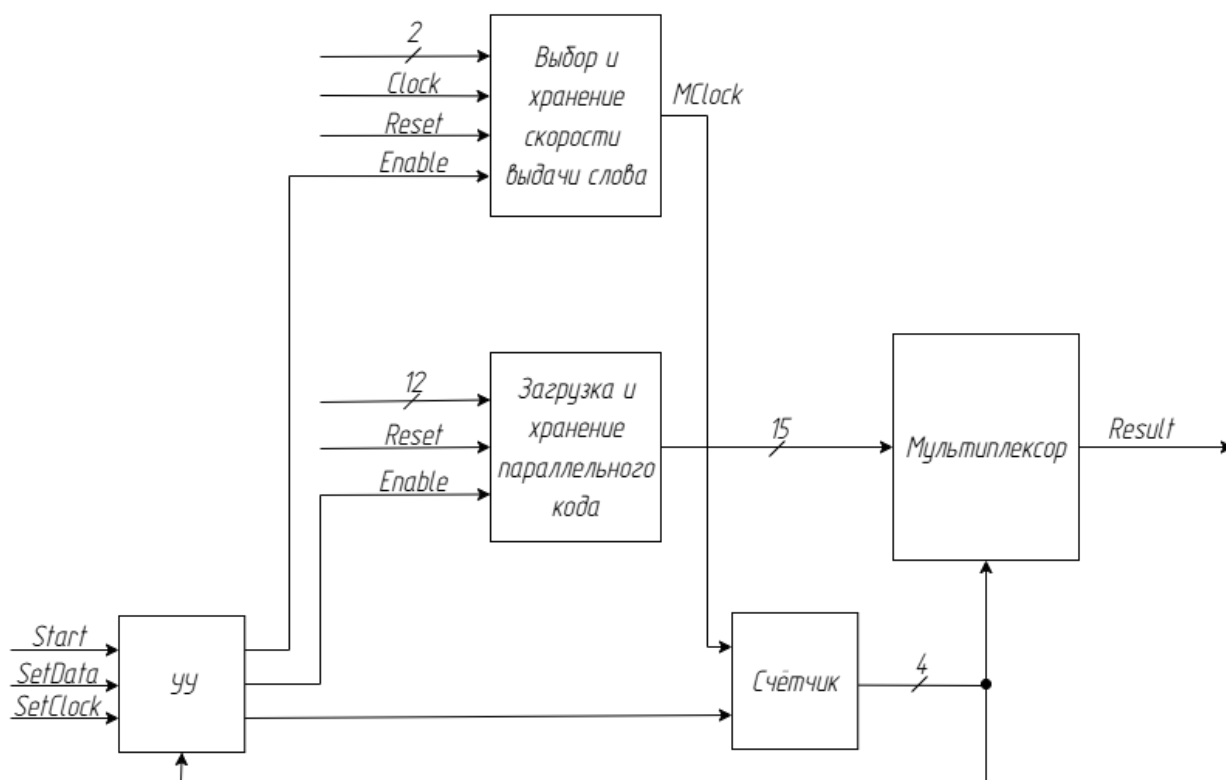


Рисунок 2.4 – Структурная схема (Э1) на основе мультиплексора

3. ОПИСАНИЕ ОСНОВНЫХ ЭЛЕМЕНТОВ БИБЛИОТЕКИ САПР QUARTUS II, НЕОБХОДИМЫХ ДЛЯ РЕАЛИЗАЦИИ УЗЛА

3.1 Логические примитивы

В данной работе были использованы следующие логические примитивы для возможности реализации схемы.

Элемент NOT служит для инверсии сигнала. Элемент OR2 применяется для контроля работы счетчика; OR4 – для счета количества переданных бит.

Элемент AND2 выполняет функцию установки входных данных в схему и установки скорости передачи данных.

Элемент XOR позволяет определить бит четности для входного слова. Графическое представление каждого из элементов приведено на Рисунке 3.1.

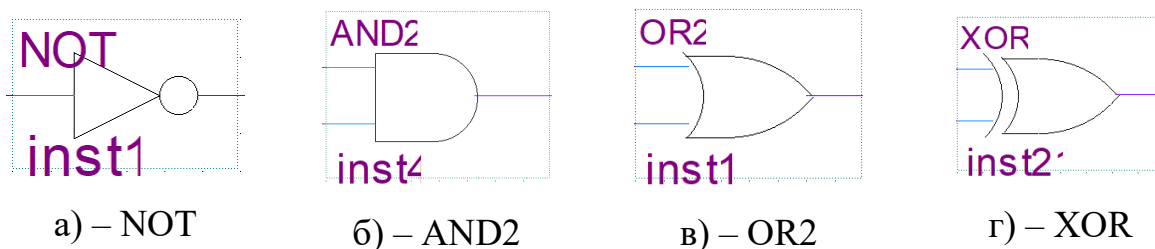


Рисунок 3.1 – Логические примитивы, использованные в процессе работы

3.2 Триггеры

Для хранения состояний параллельного кода слова и параметров скорости передачи данных используются параллельные регистры, реализованные на основе примитивов DLATCH. Такой выбор обусловлен более простой архитектурой D-защелки по сравнению с D-триггером (примитивом DFF). Кроме того, использование DLATCH обеспечивает более гибкую автономную работу схемы – установка данных может производиться в произвольный момент времени, без жёсткой привязки к синхросигналу. Графическое представление данного примитива в среде САПР Quartus II приведено на Рисунке 3.2.

Понижение частоты тактового сигнала осуществляется с помощью Т-триггеров. В текущей реализации используется примитив TFF, который содержит входы синхросигнала, асинхронного сброса/установки, а также вход данных. Его графическое отображение приведено на Рисунке 3.3.

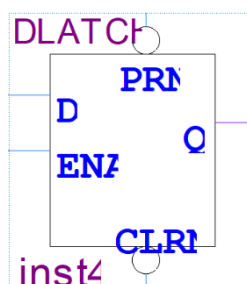


Рисунок 3.2 – Графическое обозначение примитива DLATCH D-защелки

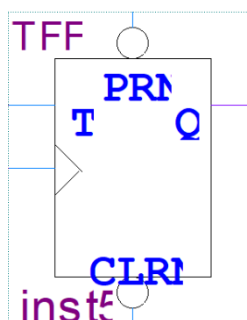


Рисунок 3.3 – Графическое обозначение примитива TFF Т-триггера

3.3 Мегафункции

В данной работе применены два мультиплексора: один – для выбора скорости передачи данных, другой – для формирования выходного сигнала. Конфигурация мультиплексоров зависит от задачи и может включать 3 или 16 информационных входов, а также 2 или 4 адресных входа. Графическое представление мультиплексора показано на Рисунке 3.4.

Контроль количества переданных бит осуществляется с помощью счётчика, оснащённого синхронным входом, входом разрешения счёта, асинхронным сбросом и четырьмя выходами, что позволяет выполнять счёт до 16. Его графическое изображение представлено на Рисунке 3.5.

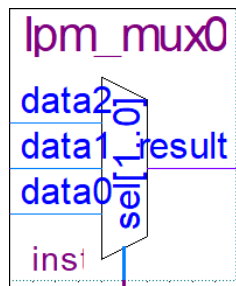


Рисунок 3.4 – Графическое обозначение мультиплексора для выбора скорости передачи данных

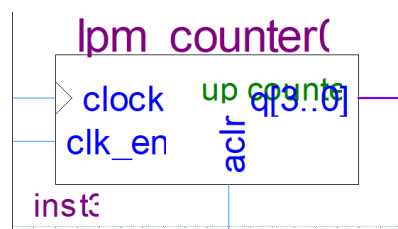


Рисунок 3.5 – Графическое обозначение счётчика

4. ОПИСАНИЕ ПРОЦЕССА СИНТЕЗА И МОДЕЛИРОВАНИЯ РАБОТЫ УЗЛА

4.1 Синтез и моделирование узла на основе сдвигающего регистра

Обе реализации преобразователя параллельного кода в последовательный для асинхронного протокола передач включают в себя наличие модуля для загрузки и хранения данных на основе параллельного регистра. Предусмотрено полное формирование необходимой последовательности битов для выходного канала. Первый бит является стартовым битом и реализован через «землю» (GND). Далее идет 12 битов параллельного кода, загруженного в регистр, после бит чётности, формируемый через цепочку XOR, и стоп-бит, который работает через VCC. Схема данной части, реализованной в САПР Quartus II, приведена на Рисунке 4.1.

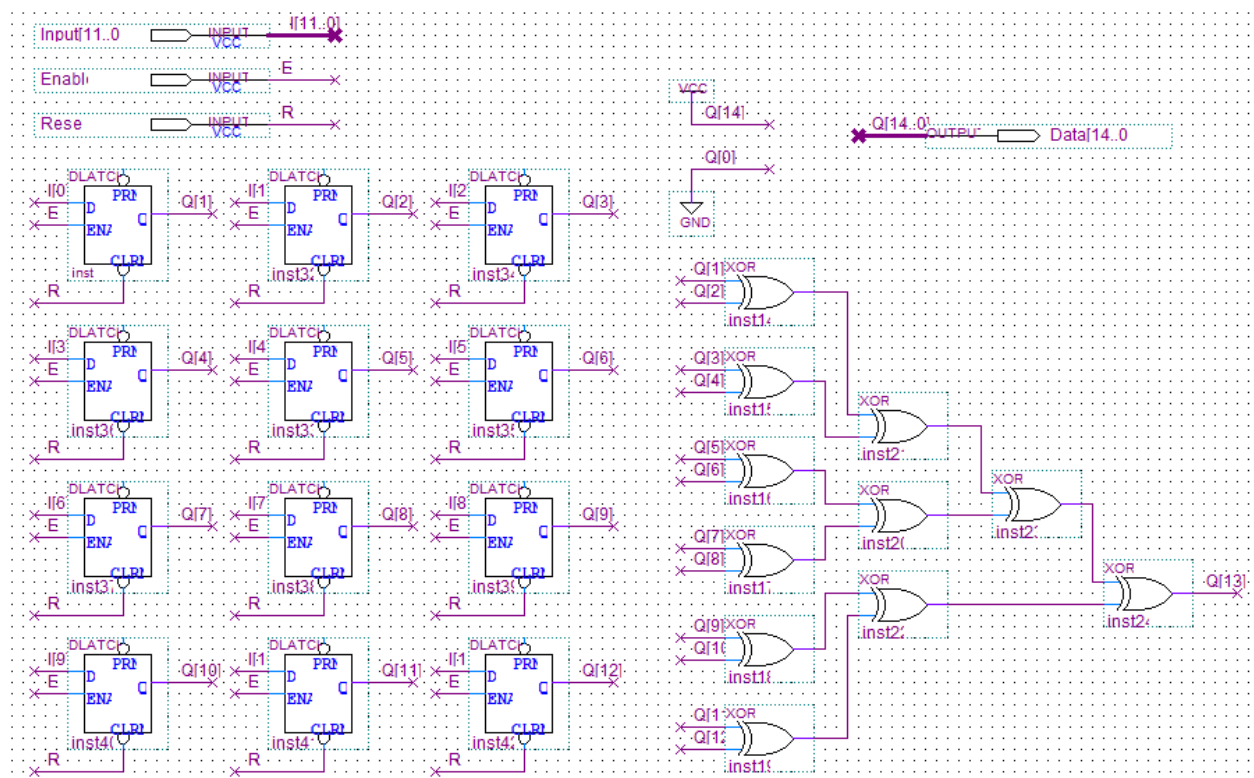


Рисунок 4.1 – Параллельный регистр для загрузки данных из шины и хранения последовательности битов для выходного канала

Кроме параллельного регистра для загрузки и хранения данных оба варианта реализации узла также включают блок выбора и хранения скорости передачи данных. Он содержит параллельный регистр для хранения состояния о выбранной скорости. Последовательно соединенные Т-триггеры формируют необходимые частоты для каждого из типов передачи данных. Выбор скорости осуществляется с помощью мультиплексора. Схема данного модуля узла, реализованного в САПР Quartus II, приведена на Рисунке 4.2.

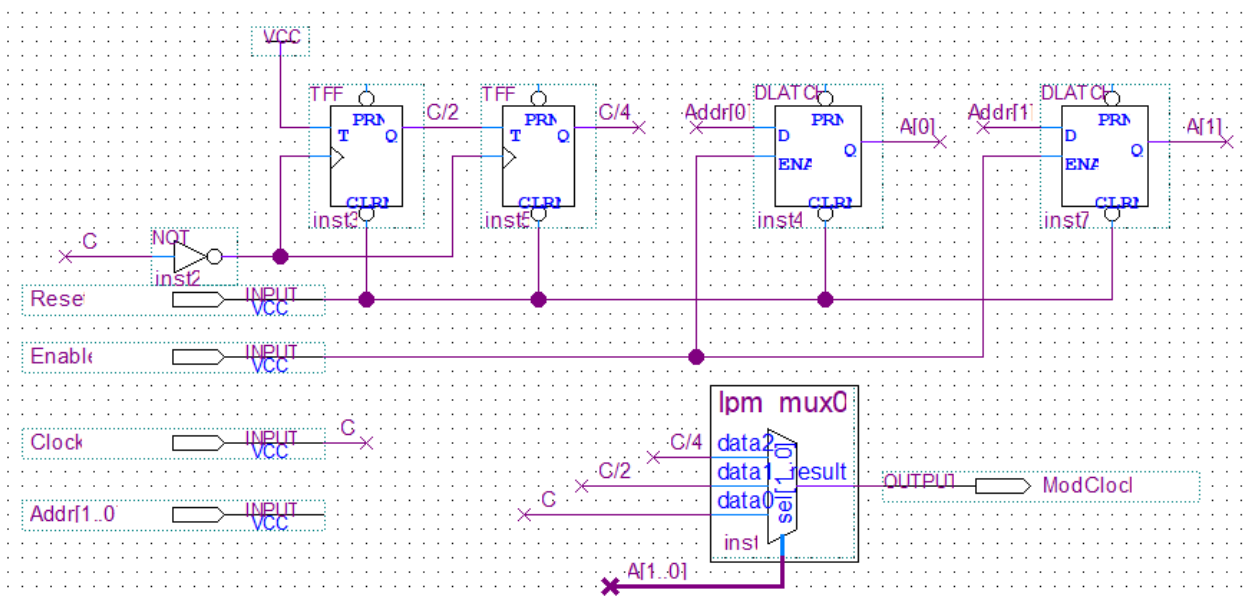


Рисунок 4.2 – Блок выбора и хранения скорости передачи данных

Общая схема преобразователя параллельного кода в последовательный для асинхронного протокола передач приведена на Рисунке 4.3. Подается управляющий сигнал Start для включения счетчика битов и начала загрузки данных в сдвигающий регистр. Два сигнала SetDATA и SetCLOCK выполняют соответственно загрузку данных параллельного кода и выбор скорости передачи данных. Шина данных In[11..0] содержит параллельный код, а Addr[1..0] данные для выбора типа частоты передачи. Для схемы предусмотрен асинхронный сброс с помощью входа Reset. Тактовый сигнал Clock задает изначальную частоту передачи данных.

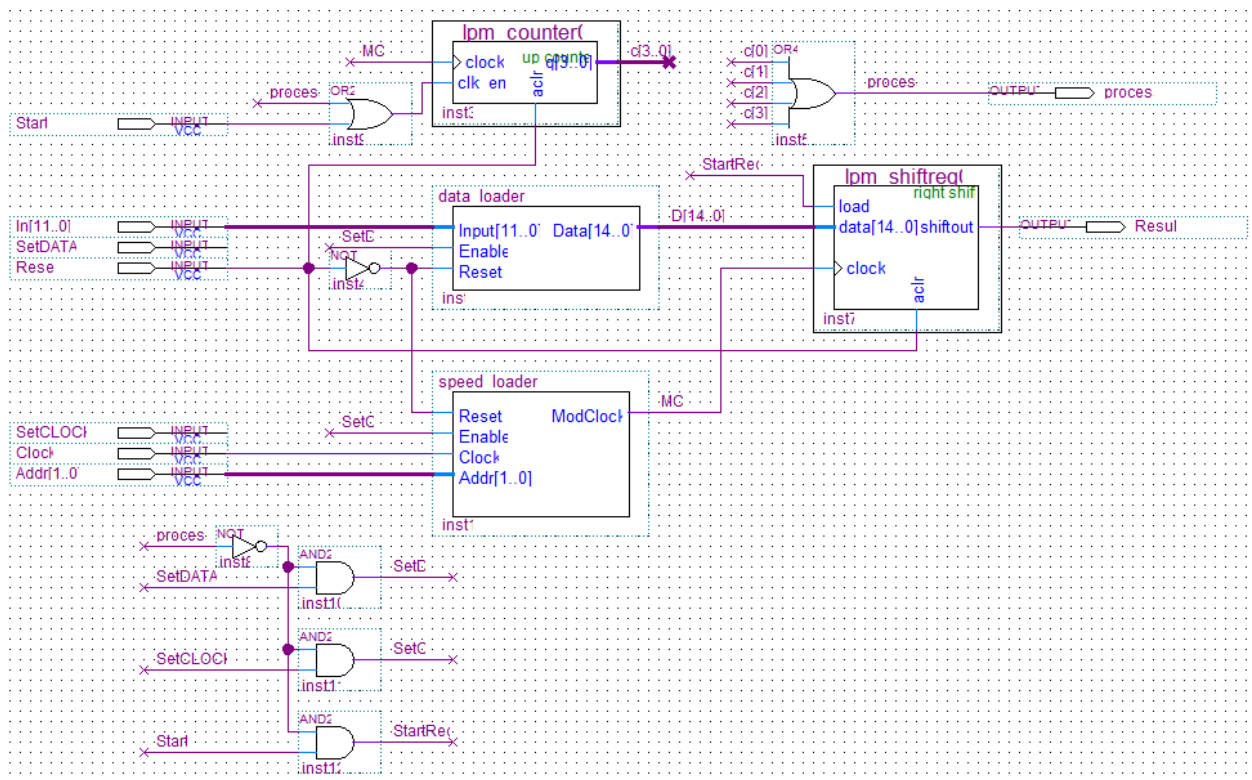


Рисунок 4.3 – Преобразователь параллельного кода в последовательный для асинхронного протокола передач на основе сдвигающего регистра

После синтеза схемы, проект был скомпилирован для поиска возможных ошибок и неточностей с целью их последующего устранения. Всего понадобилось 45 логических элемента. Результат компиляции представлен на Рисунке 4.4.

| Flow Summary | |
|------------------------------------|---|
| Flow Status | Successful - Wed May 07 19:37:58 2025 |
| Quartus II 64-Bit Version | 13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition |
| Revision Name | shift_reg |
| Top-level Entity Name | shift_reg |
| Family | Cyclone II |
| Device | EP2C5Q208C8 |
| Timing Models | Final |
| Total logic elements | 45 / 4,608 (< 1 %) |
| Total combinational functions | 45 / 4,608 (< 1 %) |
| Dedicated logic registers | 21 / 4,608 (< 1 %) |
| Total registers | 21 |
| Total pins | 21 / 142 (15 %) |
| Total virtual pins | 0 |
| Total memory bits | 0 / 119,808 (0 %) |
| Embedded Multiplier 9-bit elements | 0 / 26 (0 %) |
| Total PLLs | 0 / 2 (0 %) |

Рисунок 4.4 – Результат компиляции проекта на основе РТС

После успешной сборки проекта, было выполнено функциональное (Рисунок 4.5) и временное моделирование (Рисунок 4.6).

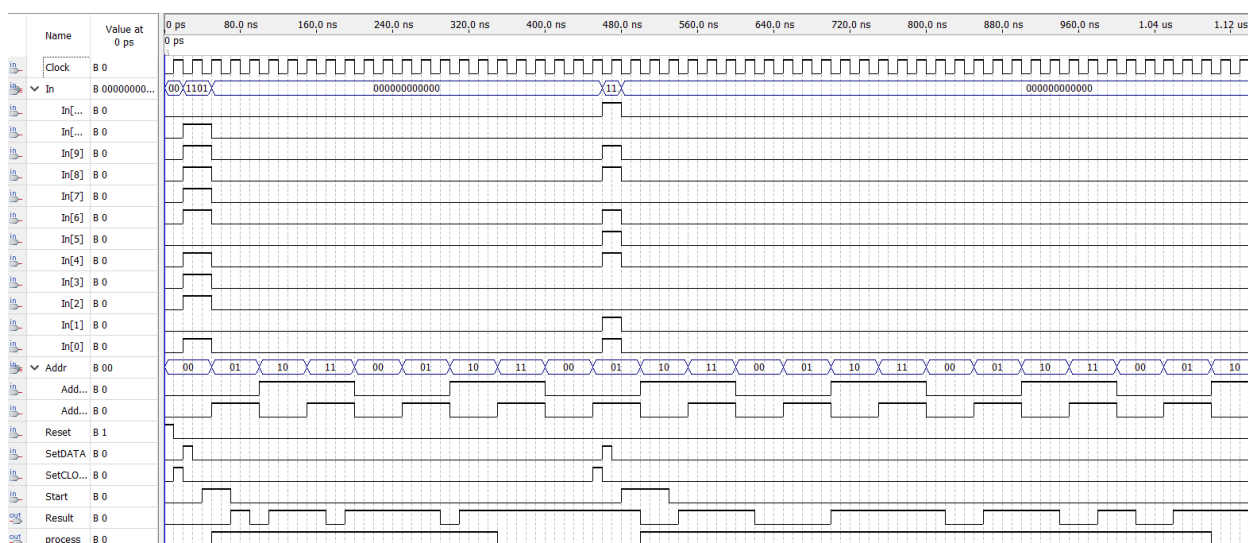


Рисунок 4.5 – Результат функционального моделирования

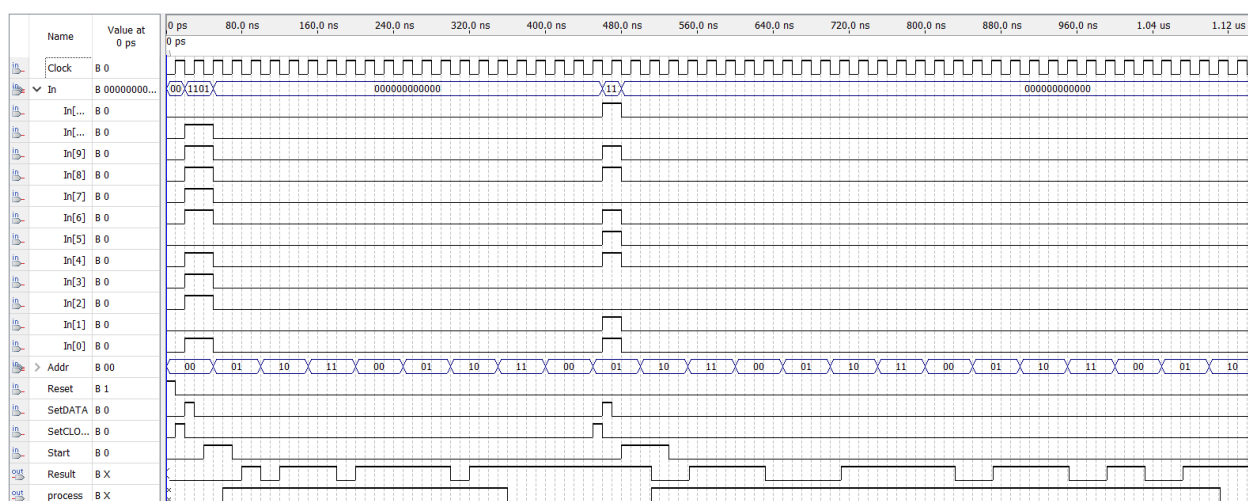


Рисунок 4.6 – Результат временного моделирования

По полученным временным диаграммам можно сказать, что схема работает корректно. Сигнал Start запускает работу счетчика и регистра, который начинает формировать битовую последовательность. Сигналы SetDATA и SetCLOCK правильно устанавливают необходимые параметры схемы. На временном моделировании можно заметить задержку выходного сигнала, связанную с задержкой загрузки данных на параллельный регистр.

4.2 Синтез и моделирование узла на основе мультиплексора

Общая схема преобразователя параллельного кода в последовательный для асинхронного протокола передач на основе мультиплексора приведена на Рисунке 4.7. Логика управления процессом работы схемы и загрузки данных аналогична узлу на основе сдвигающего регистра. Отличие состоит в замене сдвигающего регистра на мультиплексор. Такой подход позволяет выбирать нужные данные из параллельного в регистр в любой момент времени, выбор осуществляется нужного бита осуществляется счётчиком.

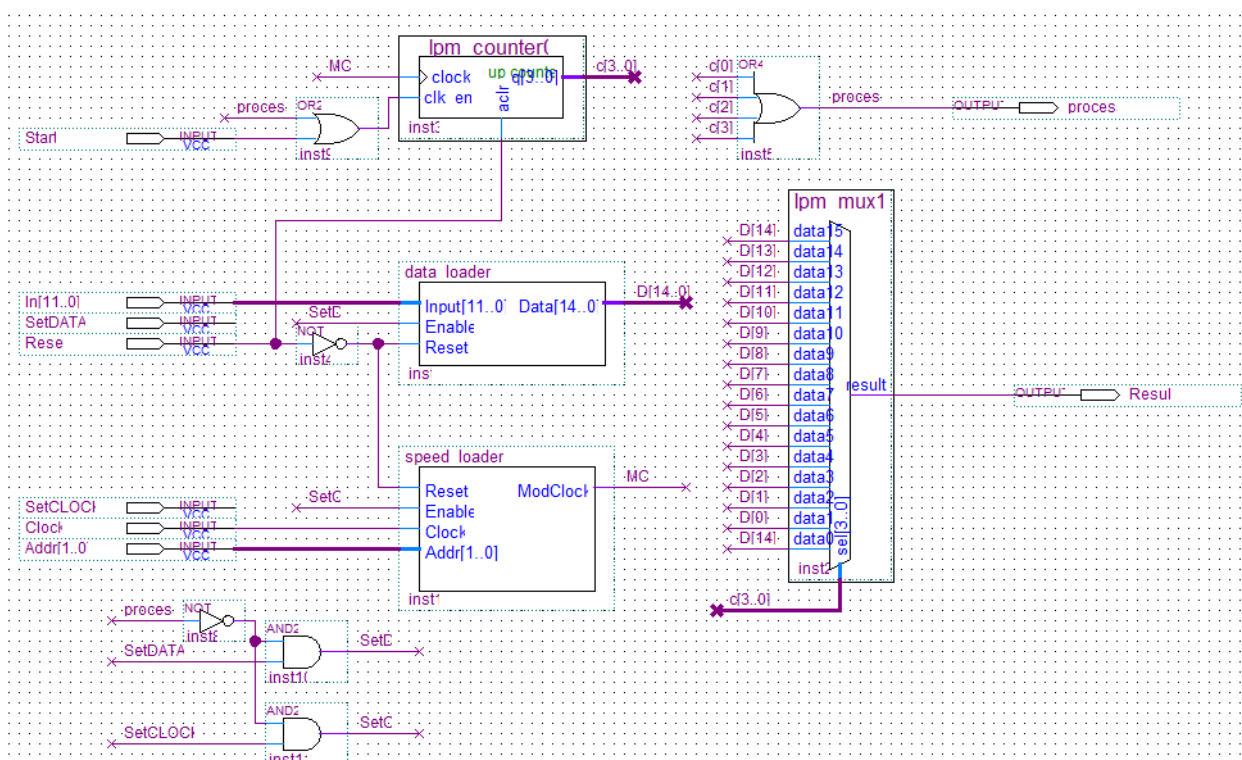


Рисунок 4.7 – Преобразователь параллельного кода в последовательный для асинхронного протокола передач на основе мультиплексора

После синтеза схемы, проект был скомпилирован для поиска возможных ошибок и неточностей с целью их последующего устранения. Всего понадобилось 40 логических элементов. Результат компиляции представлен на Рисунке 4.10.

| Flow Summary | |
|------------------------------------|---|
| Flow Status | Successful - Wed May 07 19:43:32 2025 |
| Quartus II 64-Bit Version | 13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition |
| Revision Name | scheme_mux |
| Top-level Entity Name | scheme_mux |
| Family | Cyclone II |
| Device | EP2C5Q208C8 |
| Timing Models | Final |
| Total logic elements | 40 / 4,608 (< 1 %) |
| Total combinational functions | 40 / 4,608 (< 1 %) |
| Dedicated logic registers | 6 / 4,608 (< 1 %) |
| Total registers | 6 |
| Total pins | 21 / 142 (15 %) |
| Total virtual pins | 0 |
| Total memory bits | 0 / 119,808 (0 %) |
| Embedded Multiplier 9-bit elements | 0 / 26 (0 %) |
| Total PLLs | 0 / 2 (0 %) |

Рисунок 4.10 – Результат компиляции проекта на основе мультиплексора

После успешной сборки проекта, было выполнено функциональное (Рисунок 4.11) и временное моделирование (Рисунок 4.12).

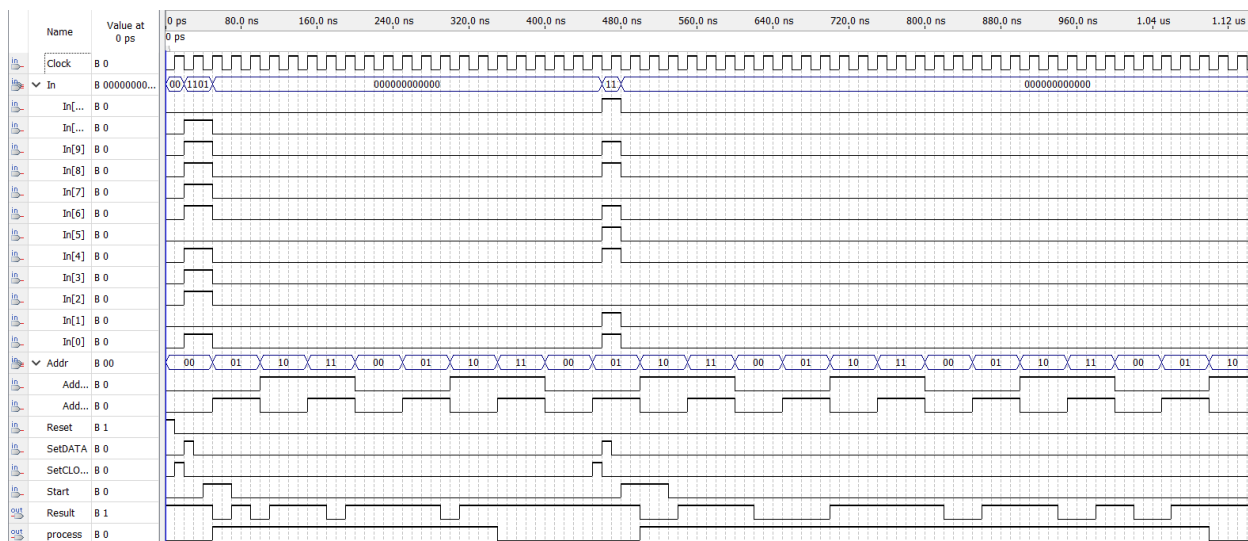


Рисунок 4.11 – Результат функционального моделирования

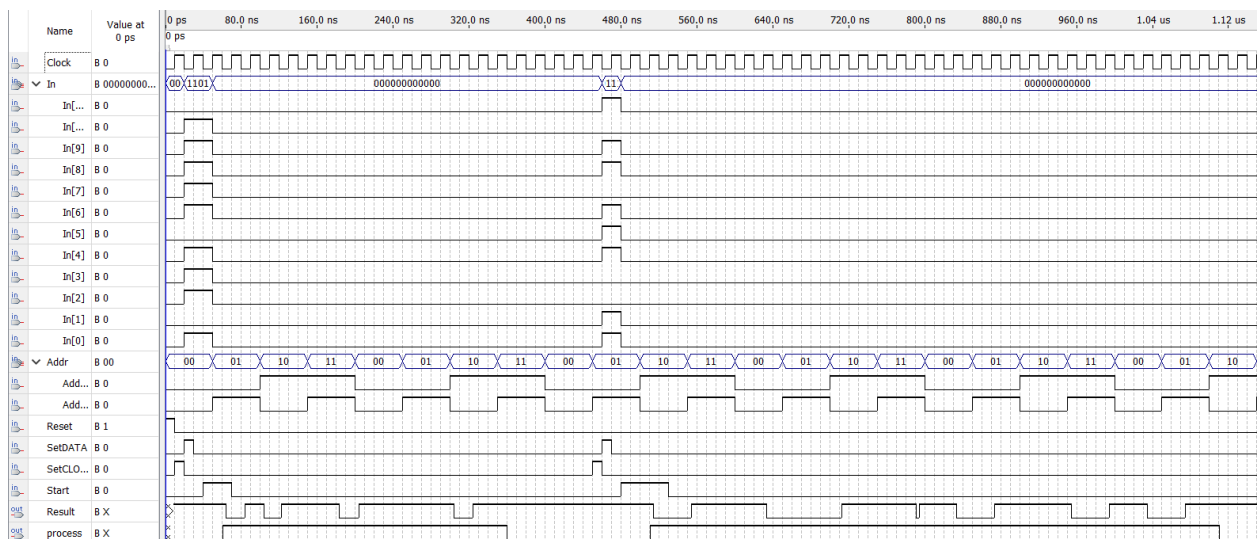


Рисунок 4.12 – Результат временного моделирования

По полученным временным диаграммам можно сказать, что схема работает корректно. Отличие между двумя вариантами на функциональном моделировании не наблюдается.

На временном моделировании помимо задержки выходного сигнала, связанной с работой параллельного регистра, имеется риск при передаче второго сообщения: в последовательности логических единиц сигнал на короткое время падает в ноль. Это, вероятно, зависит от работы счётчика и моментом переключения информационных входов на мультиплексоре.

4.3 Выбор лучшего варианта исполнения узла

Для каждого варианта реализации были собраны основные параметры их схем и способы получения итогового результата. Критерием выбора служит минимум аппаратных затрат. Данные представлены в Таблице 4.1.

Схема на основе мультиплексора явно лидирует по заданному критерию выбора, однако при такой реализации наблюдается риск, что в промышленном масштабе может быть критичным.

Таблица 4.1

| | Параметры схемы | Вариант реализации | |
|---|--|------------------------------------|---|
| | | Сдвигающий регистр | Мультиплексор |
| 1 | Аппаратные затраты (общее количество элементов) | 45 | 40 |
| 2 | Временные задержки (нс) | 8.98 | 9 |
| 3 | Простота реализации | 15 разрядный сдвигающий регистр | Работа мультиплексора проще, чем сдвигающего регистра |
| 4 | Наличие рисков | Не обнаружено | Имеется умеренная ошибка при передаче сообщения |

5. РАЗРАБОТКА ГЕНЕРАТОРА ТАКТОВЫХ ИМПУЛЬСОВ

5.1 Выбор варианта реализации генератора

Выбор конкретной схемы генератора тактовых импульсов (ГТИ) зависит от множества факторов, главным из которых является требуемая стабильность выходной частоты. Наибольшую стабильность обеспечивают схемы с кварцевыми резонаторами, для которых выпускается широкий ассортимент интегральных схем. Основным недостатком таких решений является относительно высокая стоимость.

Если допустима нестабильность частоты в пределах нескольких процентов, можно использовать более простые схемы – кольцевые или RC-генераторы, в которых задающие элементы реализованы с помощью резисторов и конденсаторов.

ГТИ будет выполнен по схеме, представленной на рисунке 5.1. Для стандартных ТТЛ-элементов, ввиду значительных входных токов, сопротивление R должно быть небольшим. В ТТЛ-схемах резистор R' не требуется. Частота такого генератора определяется следующим выражением:

$$f_0 \approx 0.53 \frac{1}{RC}$$

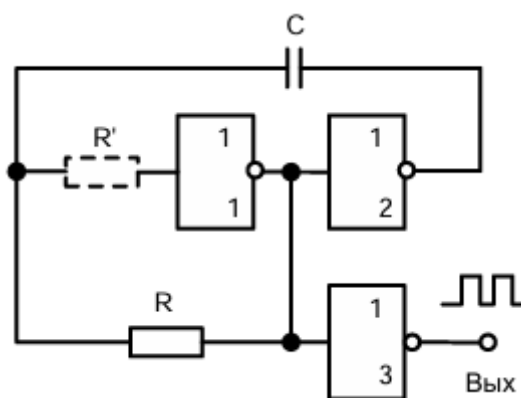


Рисунок 5.1 – Схема генератора тактовых импульсов

5.2 Инженерный расчет элементов генератора для получения необходимой частоты

В данном варианте необходимо добиться скорости передачи данных в 4.8 Кбит/с. Следовательно, частота сигнала должна быть 4.8 КГц.

Значения для резисторов R и конденсатора C будут взяты из ряда E24 номиналов. В Таблице 5.1 приведен сравнительный анализ допустимых значений R и C . Дополнительно рассчитано абсолютное отклонение.

Таблица 5.1

| № | C , мкФ | R , Ом | f , КГц | $ f - f_0 $, КГц |
|----|-----------|----------|-----------|-------------------|
| 1 | 0,62 | 180 | 4749,104 | 50,89606 |
| 2 | 0,56 | 200 | 4732,143 | 67,85714 |
| 3 | 0,51 | 220 | 4723,708 | 76,29234 |
| 4 | 0,47 | 240 | 4698,582 | 101,4184 |
| 5 | 0,39 | 270 | 5033,238 | 233,2384 |
| 6 | 0,36 | 300 | 4907,407 | 107,4074 |
| 7 | 0,33 | 330 | 4866,85 | 66,85032 |
| 8 | 0,3 | 360 | 4907,407 | 107,4074 |
| 9 | 0,27 | 390 | 5033,238 | 233,2384 |
| 10 | 0,27 | 430 | 4565,03 | 234,9699 |

По полученным данным можно сказать, что наиболее близкими вариантами стали 1 и 7. Они имеют самое маленькое отклонение от необходимой частоты.

Для итогового варианта будет выбрана пара $C = 0.33$ (мкФ) и $R = 330$ (Ом) в пользу чуть большей частоты, так как в реальных условиях есть паразитические емкости, и предполагаемый результат может быть ниже.

6. РАЗРАБОТКА ИНТЕРФЕЙСА СОПРЯЖЕНИЯ СХЕМЫ УЗЛА С ПРОЦЕССОРНОЙ СИСТЕМОЙ

6.1 Синтез полной схемы преобразователя параллельного кода в последовательный для асинхронного протокола передач

Разрабатываемое устройство взаимодействует с процессорной системой с помощью стандартной шины Microbus. Для работы схемы используются следующие сигналы: \overline{IOR} – считывание данных с вычислительного узла; \overline{IOW} – запись данных в ВУ, Clock – тактируемый сигнал генератора, Reset – сигнал сброса, Start – сигнал начала работы схемы. Также используется 8 разрядная шина адреса для подачи адреса области АП, и 8 разрядная шина данных для активации необходимых команд.

Работа схемы осуществляется при адресе 0x85 (1000 0101). Так как необходимо загрузить 12 разрядный параллельный код на вход ВУ, то будет необходимо разработать дополнительную логику для считывания данных с шины.

Идея реализации заключается в последовательной загрузке данных из шины по старшему биту. Описание такого способа приведено в Таблице 6.1. Данные будут храниться в параллельном регистре и загружаться постоянно на ВУ, пока сигнал готовности схемы Ready активен.

Дополнительно предусмотрена логика считывания данных с ВУ через элементы «И».

Таблица 6.1

| Маска | Загружаемые данные |
|-----------|---|
| 0xxx xxxx | xxx xxxx: первые 7 бит кодового слова полностью |
| 1yux xxxx | уу: данные для выбора скорости передачи х xxxx: старшие 5 бит кодового слова |

Разработанная схема интерфейса в САПР Quartus II приведена на Рисунке 6.1. Селектор адреса реализован через элемент NOR8.

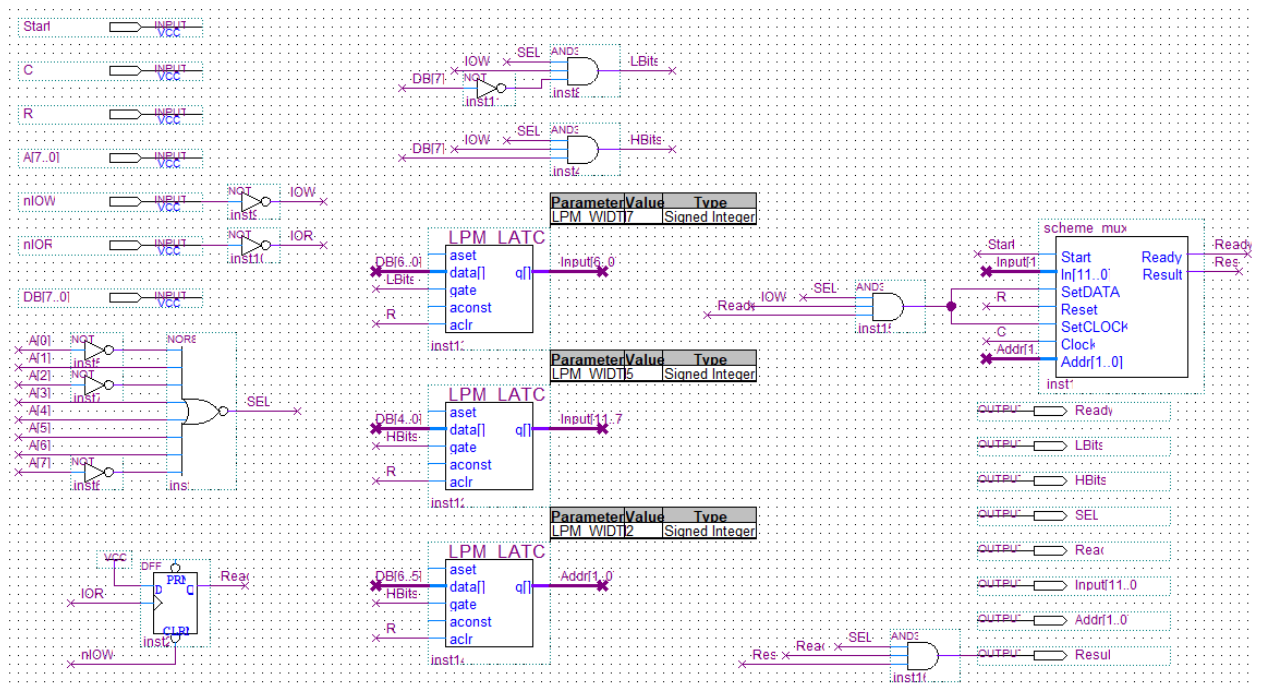


Рисунок 6.1 – Полная схема узла и интерфейса сопряжения с процессорной системой

После успешной сборки проекта, было выполнено функциональное (Рисунок 6.2) и временное (Рисунок 6.3) моделирование.

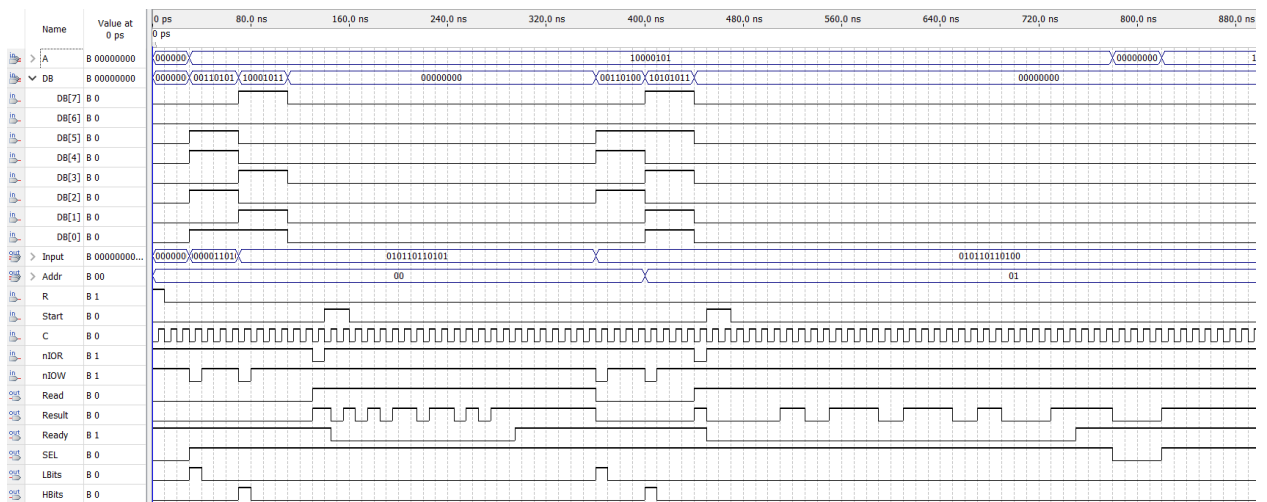


Рисунок 6.2 – Результат функционального моделирования

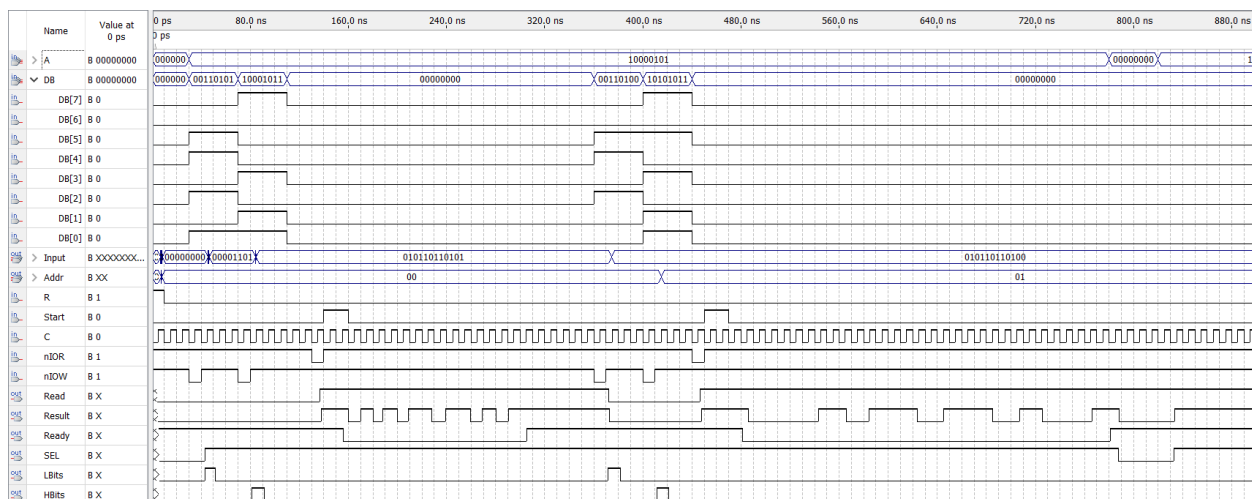


Рисунок 6.3 – Результат временного моделирования

Анализ временных диаграмм позволяет оценить корректность функционирования разработанного интерфейса. Сначала происходит последовательная загрузка входных данных двумя сигналами IOW в моменты появления нужных данных на шине. Затем активируется режим чтения с ВУ, и при подаче сигнала Start начинается побитовая загрузка параллельного кода с добавлением старт-бита, бита паритета и стоп-бита. Изменение скорости передачи данных функционирует корректно: при установке режима деления на 2 длительность выходного сигнала увеличивается.

На временном моделировании можно заметить задержку загрузки примерно в два тактовых сигнала.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы были приобретены практические навыки проектирования специализированных цифровых устройств с использованием САПР Quartus II. Были разработаны и реализованы два варианта преобразователя параллельного кода в последовательный для асинхронного протокола передачи данных: на основе сдвигающего регистра и мультиплексора. Проведён анализ схем с точки зрения сложности реализации и качества формируемых сигналов.

Также был спроектирован генератор тактовых импульсов, выполнены инженерные расчёты для обеспечения требуемой частоты. Разработана интерфейсная часть схемы для взаимодействия с процессорной системой и реализована логика загрузки большого объёма данных по одному адресу. Проведена компиляция и тестирование схемы устройства, подготовлена принципиальная электрическая схема ТЭЗа, включающая спроектированный узел и вспомогательные компоненты.

Таким образом, в процессе выполнения работы был получен практический опыт проектирования, синтеза, тестирования и документирования цифрового устройства.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Проектирование цифровых узлов: Методические указания к курсовому проектированию / сост.: Р.И. Грушвицкий, Е.П. Угрюмов. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2011. 00 с.
2. Угрюмов Е.П. Цифровая схемотехника: учеб. пособие для вузов. – БХВ-Петербург, 1-е изд. 2000 г., 2-е изд. 2006 г., 3-е изд. 2010 г. – 797 с.
3. Перечень доступных микросхем // ЧИП и ДИП – интернет-магазин приборов и электронных компонентов. URL: <https://www.chipdip.ru/catalog/ic-chip> (дата обращения 06.05.2025).
4. Ряд E24 номиналов конденсаторов // RadioLibrary – справочник радиолюбителя. URL: <https://www.radiolibrary.ru/reference/capacitorseries/e24.html> (дата обращения 06.05.2025).
5. Ряд E24 номиналов резисторов // RadioLibrary – справочник радиолюбителя. URL: <https://www.radiolibrary.ru/reference/resistorseries/e24.html> (дата обращения 06.05.2025).
6. Understanding UART // Rohde-schwarz – передовая технологическая компания. URL: https://www.rohde-schwarz.com/cac/products/test-and-measurement/essentials-test-equipment/digital-oscilloscopes/understanding-uart_254524.html (дата обращения 05.05.2025).
7. ГОСТ 2.743-91. Единая система конструкторской документации. Обозначения условные графические в электрических схемах. – М.: ИПК Издательство стандартов, 1992.
8. Конспекты лекций по дисциплине «Элементная база цифровых систем» за 6 семестр обучения.