

Аннотация

Дипломный проект посвящен разработке устройства, предназначенного для генерации тестовых пакетов на внутреннюю шину контроллера периферийных интерфейсов, сбора и передачи результатов тестов под управлением тестирующих программ. Проанализированы современные способы верификации микросхем. В проекте обоснован выбор архитектуры устройства. Рассмотрены принципы построения основных блоков. Проработана аппаратно-программная реализация устройства на базе ПЛИС. Выполнен расчет основных экономических показателей разработки и рассмотрен вопрос, связанный с охраной труда оператора-разработчика.

This diploma project is devoted to the development of the device designed for generating benchmark tests to the internal bus of the peripheral controller, collecting and transmitting the test results under the guidance of the testing programs. There have been analysed the ways of microcircuit verification. In the project there is substantiated the choice of the device architecture. There are considered the principles of building the main units. There has been developed firmware realization of the device based on FPGA. There has been also performed the calculation of the major economic indices of the development and regarded the designer's job safety issues.

Реферат

РПЗ 110 с., 33 рис., 26 табл., 23 ист., 3 прил.

ПРОЕКТ, КПИ, ПЛИС, ВЕРИФИКАЦИЯ, ТЕСТИРОВАНИЕ, RTL, ШИНА, ИНТЕРФЕЙС, QUARTUS, ЗАПРОС, РЕГИСТР, БАЗОВЫЙ АДРЕС

Дипломный проект посвящен разработке устройства, предназначенного для генерации тестовых пакетов на внутреннюю шину контроллера периферийных интерфейсов, сбора и передачи результатов тестов под управлением тестирующих программ.

В рамках дипломного проекта проанализированы современные способы верификации микросхем. На основании них разработана методика тестирования контроллера периферийных интерфейсов.

В процессе дипломного проектирования выделены основные блоки устройства, разработана структурная схема. Выбраны и обоснованы средства описания устройства.

В технологической части проработана аппаратно-программная реализация устройства на базе ПЛИС. Рассмотрены вопросы конфигурирования и отладки устройства.

В организационно-экономической части проведен расчет затрат на его разработку, составлен поэтапный план проведения работ.

В части дипломного проекта, посвященной охране труда, проведен анализ опасных и вредных факторов, влияющих на оператора-разработчика. Рассмотрен вопрос, связанный с утилизацией отработанных средств вычислительной техники.

Актуальность данной разработки заключается в том, что разработанное устройство позволит производить комплексное тестирование контроллера периферийных интерфейсов, тем самым повысить его надежность.

Содержание

1 Введение.....	9
2 Исследование способов верификации и тестирования микросхем.....	11
2.1. Анализ способов верификации микросхем.....	12
2.1.1 Верификация на уровне RTL описаний.....	13
2.1.2 Верификация полученного синтеза.....	14
2.1.3 Верификация после размещения и трассировки элементов.....	14
2.2. Методика тестирования микросхем.....	15
2.2.1 Тестирование с использованием тестового окружения.....	16
2.2.2 Отладка на макете.....	18
2.2.3 Проверка опытного образца.....	19
2.3. Прототип микросхемы КПИ.....	26
2.3.1 Прототипы КПИ на основе ПЛИС.....	26
2.3.2 Основной макет.....	27
3 Разработка компонентов АПК ТКПИ.....	30
3.1. Разработка структурной схемы АПК ТКПИ.....	30
3.2. Разработка аппаратных компонентов АПК ТКПИ.....	42
3.3. Логический анализатор.....	53
4 Реализация и верификация АПК ТКПИ.....	57
4.1. Выбор ПЛИС для реализации АПК ТКПИ.....	57
4.2. Использование мегафункций САПР Quartus.....	58
4.3. Сборка логической модели АПК ТКПИ.....	65
4.4. Конфигурирование ПЛИС.....	66
4.5. Тестирование АПК ТКПИ.....	73
5 Охрана труда и экология.....	75
5.1. Анализ опасных и вредных факторов на этапе проектирования логической модели АПК ТКПИ.....	75
5.2. Расчет системы естественного освещения на этапе проектирования.....	79
5.3. Утилизация отработанных средств вычислительной техники.....	85
6 Организационно-экономическая часть.....	88
6.1. Введение.....	88
6.2. Основные этапы разработки проекта.....	89
6.3. Расчет трудоемкости проекта.....	89
6.4. Определение численности исполнителей.....	91
6.5. Сетевая модель проекта.....	93

6.6. Анализ структуры затрат на разработку проекта.....	96
6.7. Определение возможной прибыли.....	101
6.8. Оценка экономических эффектов от разработки проекта.....	102
Заключение.....	106
Список использованных источников.....	107
Приложение А Техническое задание.....	109
Приложение Б Графические материалы.....	110
Приложение В Пространство регистров статуса и управления АПК ТКПИ.....	111

Определения, обозначения и сокращения

АПК ТКПИ – аппаратно-программный комплекс тестирования контроллера периферийных интерфейсов,

АПО – аппаратно программное обеспечение,

ВК – вычислительный комплекс,

ГС – граничное сканирование,

ПЛИС – программируемая логическая интегральная схема,

КВВ – канал ввода/вывода,

КПИ (IOHUB – Input/Output Hub) – контроллер периферийных интерфейсов,

МВК – многопроцессорный вычислительный комплекс,

ОЗУ – оперативное запоминающее устройство,

ОКР – опытно-конструкторская работа,

РГС – регистр граничного сканирования,

РИ – регистр инструкций,

РК – регистр команд,

РО – регистр обхода,

APIC (*Advanced Programmable Interrupt Controller*) - улучшенный программируемый контроллер прерываний,

ASIC (*Application-specific Integrated Circuit*) – заказная интегральная микросхема,

BAR (Base Address Register) – базовые адресные регистры в конфигурационном пространстве шины PCI,

CRC (*Cyclic Redundancy Code*) – циклический избыточный код,

DMA (*Direct Memory Access*) запросы – запросы прямого доступа в память,

DW (*Double Word*) – двойное слово (размерность 32 бита),

JTAG (*Join Test Access Group*) - специализированный аппаратный интерфейс, разработанный для тестирования собранных печатных плат и микросхем,

LVDS (*Low-voltage Differential Signaling*) - низковольтная дифференциальная передача сигнала,

PIO (*Program Input/Output*) запросы – программируемые запросы ввода/вывода,

PCI (*Peripheral Component Interconnect*) - шина ввода/вывода для подключения периферийных устройств к “материнской” плате компьютера,

PLL (*Phase-Locked Loop*) – фазовая автоматическая подстройка частоты,

RTL (*Register Transfer Level*) – уровень регистровых передач,

RDMA (*Remote Direct Memory Access*) – удаленный прямой доступ в памяти,

TAP (*Test Access Port*) – порт тестового доступа,

1 Введение

В настоящее время в высокопроизводительных и надежных вычислительных системах отечественного производства серии «Эльбрус», разрабатываемых компанией ЗАО «МЦСТ», в качестве контроллеров периферийных интерфейсов используются импортные микросхемы и контроллеры на основе ПЛИС.

Такой подход имеет следующие недостатки:

- Зависимость от зарубежных производителей;
- Большинство предлагаемых импортных микросхем предназначены для коммерческого использования;
- Контроллеры основанные на ПЛИС обладают невысокой производительностью для данной области применения.

При разработке вычислительных систем на базе микропроцессоров “Эльбрус-S”, “МЦСТ-4R” для достижения заданных характеристик и повышения производительности возникла необходимость разработки отечественного контроллера периферийных интерфейсов.

Контроллер периферийных интерфейсов разрабатывается в рамках ОКР “Повозка-Т”, которая выполняется на основании – постановление Правительства Российской Федерации от 10.11.07 г. № 771-35 «О государственном оборонном заказе на 2008-2010 гг.».

Он обладает следующими характеристиками:

- Полностью отечественная разработка;
- Отсутствие зависимости от зарубежных производителей;
- Широкий набор периферийных, сетевых и специальных контроллеров (Ethernet, USB2.0, RS-232, AC97 и д.р);
- Наличие канала доступа к подсистеме ввода/вывода процессора;
- Максимальный темп обмена с подсистемы ввода/вывода процессора – 1 Гбайт/с (0,5 Гбайт/с - прием, 0,5 Гбайт/с - передача).

Первоначально микросхема будет использоваться для комплектования ВК «Эльбрус-3С», разрабатываемого на основе системы на кристалле «Эльбрус-S». ВК «Эльбрус-3С»

предназначен для обеспечения перспективных систем и комплексов противоракетной и противовоздушной обороны (С-300, С-400) отечественными высокопроизводительными вычислительными системами для решения задач обработки больших объемов информации в режиме жесткого реального времени.

Так же КПИ будет использоваться с микропроцессорами «МЦСТ-4R» в различных вычислительных комплексах. На базе этих разработок планируется поэтапная замена выполнивших свои задачи МВК «Эльбрус-1» и МВК «Эльбрус-2».

Для вычислительных комплексов противоракетной и противовоздушной обороны одним из основных параметров является надежность. Одним из способов повышения надежности разрабатываемого КПИ - возможность проведения комплексного тестирования.

Целью данного дипломного проекта является разработка АПК ТКПИ, предназначенного для генерации тестовых пакетов на внутреннюю шину контроллера периферийных интерфейсов, сбора и передачи результатов под управлением тестирующих программ. Его функциональные характеристики:

- Объем памяти ОЗУ 256 Мбайт;
- Тип интерфейса с процессором шина PCI;
- Тип интерфейса с КПИ шина IO-Link;
- Режим работы: PIO запросы и DMA запросы;
- 46 программно доступных регистров;
- Наличие логического анализатора объемом 16384 выборок;
- Возможность запуска параллельных тестов для КПИ.

АПК ТКПИ позволяет провести комплексную верификацию КПИ, определить наибольшее количество ошибок, допущенных при проектировании КПИ.

2 Научно-исследовательская часть

2.1 Анализ способов верификации микросхем

Рассмотрим способы верификации КПИ в зависимости от выбранного маршрута физического проектирования.

Методика верификации заказных интегральных схем (ASIC) основным образом определяется из выбранного маршрута проектирования данной микросхемы. В тоже время процесс верификации задает ряд требований к создаваемому проекту, таких как возможность верифицирования разработанного проекта и его частей.

Взаимосвязь между данными процессами показана на рисунке 2.1 [1].

Рисунок 2.1 – V-диаграмма, взаимосвязь между проектированием и верификацией

Процесс проектирования показан на левой стороне данной диаграммы, он движется с постепенным возрастанием детализации до маленьких блоков, начиная с системного уровнем, а верификация (показана на правой стороне) идет в обратном направлении, т.е. до интеграции всей системы.

Как только собран весь проект, необходимо выполнять проверку всей системы. Поскольку система собирается из отдельных блоков, дальнейшая проверка выполняется до правильности всей системы целиком, что занимает верхнюю правую позицию на данной диаграмме — противоположно системному уровню проектирования.

Как только полная система интегрирована, необходима проверка правильности всей системы целиком.

Последний этап обладает двумя ключевыми проблемами:

- Так как вся система собрана, то происходит проверка функциональности каждого блока заново – это означает обширную перепроверку;
- Чем позже будет найдена ошибка, тем сложнее будет найти ее причину и исправить ее;

Поэтому при разработке методики верификации необходимо учитывать данные проблемы.

Если для логического моделирования использовать в основном модели системного уровня, то при этом моделируется вся система в целом, воздействия на систему подаются в терминах ее интерфейса (взаимодействие с памятью, snooping-запросы, внешние прерывания). Такой подход позволяет достаточно просто реализовать моделирование поведения системы и организовать его контроль и, при отсутствии технологических ограничений, вполне достаточен для целей верификации. К сожалению, у этого подхода существуют и определенные недостатки.

Главными из них являются:

- Низкая скорость моделирования, которая практически не растет со временем – увеличение вычислительных мощностей моделирующих стендов компенсируется увеличением сложности разрабатываемых систем;
- Невозможность прямого доступа к «внутренностям» аппаратуры и создания редких ситуаций, требующих сочетания нескольких одновременных событий – это может быть компенсировано либо созданием автоматических генераторов тестов и их длительными прогонами либо написанием вручную тестов, создающих какие-то совокупности тех или иных событий; понятно, что оба подхода требуют значительных временных затрат либо для создания тестов либо для их прогона.

От обоих недостатков свободен метод автономного моделирования отдельных функциональных блоков, входящих в систему. Малый объем тестируемого оборудования приводит к возрастанию скорости его моделирования в десятки и сотни раз. Воздействия

подаются непосредственно на интерфейс тестируемого блока, что в разы упрощает возможность создания тех или иных ситуаций. С учетом того, что тесты системного уровня после краткого этапа начальных проверок тестируют каждый какое-то отдельное устройство, а не всю систему в целом, выигрыш в производительности составляет сотни и тысячи раз, одновременно повышается и качество верификации. Недостаток автономного тестирования - плохая проверка собственно интерфейса функционального блока - преодолевается сочетанием автономного и системного тестирования (на уровне системы проверяются в основном интерфейсы).

Применение описанных методов должно сократить сроки верификации КПИ и повысить ее качество. В итоге это должно положительно сказаться на сроках верификации прототипа микросхемы и ее опытных образцов.

При проектировании физического дизайна КПИ используется методика сквозного проектирования [2]. Основные этапы, после которых необходимо проводить верификацию:

- Создание RTL описания;
- Синтез полученного RTL описания;
- Получения полного списка всех логических элементов проектируемой схемы вместе с их межсоединениями, после размещения и трассировки элементов.

2.1.1 Верификация на уровне RTL описаний

На данной стадии для проверки функциональности применяется динамическая верификация. Динамическую верификацию можно разделить на программную и аппаратную составляющую:

- В качестве программной составляющей используется тестовое окружение, с помощью которого выполняется RTL моделирование системы и отдельного устройства;
- В качестве аппаратной составляющей выступает прототипирование всей разрабатываемой системы;

2.1.2 Верификация полученного синтеза

На данном этапе для проверки эквивалентности полученного синтеза с исходным описанием применяется формальная верификация (САПР FormalPro Mentor Graphics) [3].

Формальная верификация позволяет обнаружить ошибки в комбинационной логике и получить для анализа дополнительную информацию по проверяемому объекту :

- Количество эквивалентных точек полученных при анализе RTL описания и его синтеза;
- Количество неэквивалентных точек и их местонахождение в схеме;
- Количество регистров, находящихся в фиксированном состоянии (константы 0 или 1) и их наименования;
- Определить висячие связи между элементами;
- Определить расхождение интерфейсов (несовпадение количества входных и выходных связей);

2.1.3 Верификация после размещения и трассировки элементов

На данном этапе для проверки может использоваться как моделирование системы с помощью тестового окружения на библиотечных элементах, т.е. с учетом задержек в схеме, так и применяться формальная верификация.

Согласно проведенным исследованиям [2], моделирование на данном этапе, требует больших временных и вычислительных затрат от разработчиков.

Таблица 2.1 – Сравнение времени проверки вентилей между моделированием и использованием формальной верификации

Размер проекта	Моделирование	Формальная верификация
1	2	3
200К вентиляей	4 дня	0.25 часа
500К вентиляей	14 дней	0.66 часа
2М вентиляей	60 дней	2 часа
9.6М вентиляей	200 дней	4.6 часа

Но отказываться от моделирования совсем нельзя, т.к. моделирование обеспечивает функциональную проверку, а формальная верификация логическое покрытие.

Правильность выбранных методов верификации подтверждает схема, показанная на рисунке 2.2. [4]

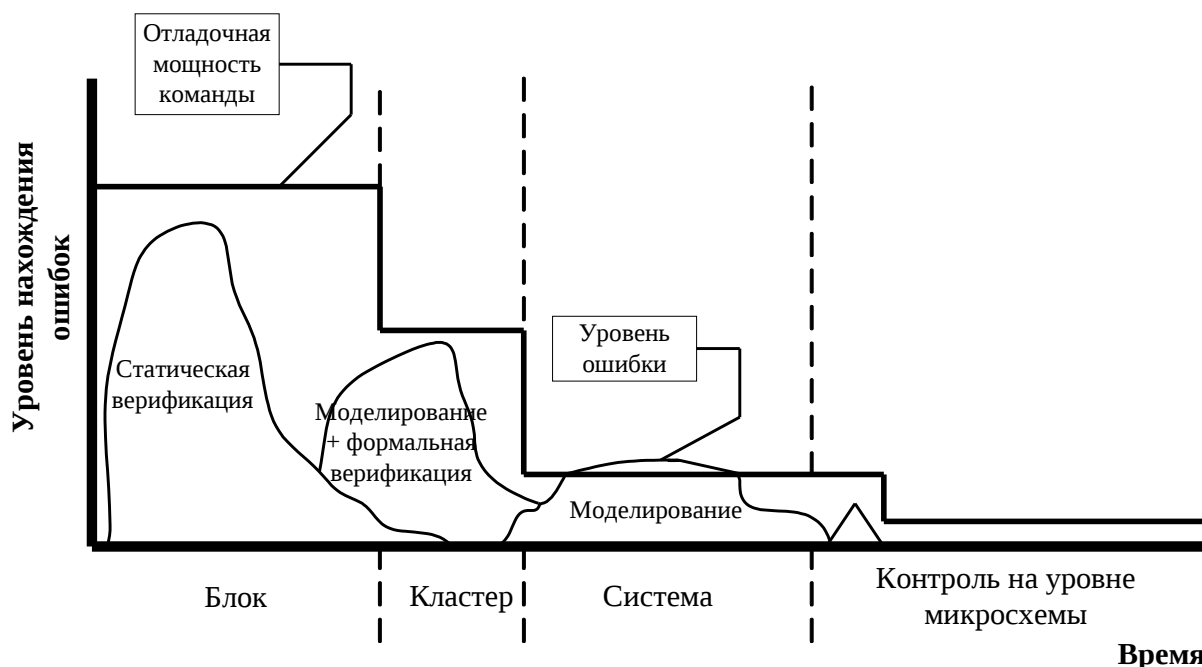


Рисунок 2.2 – Системный и подсистемный уровень моделирования, объединенный с формальной верификацией

2.2 Методика тестирования микросхем

Создание тестового аппаратно-программного обеспечения (АПО) – одна из основных задач в процессе проектирования микросхем. С ростом объема микросхемы трудоемкость разработки тестового АПО растет существенно быстрее трудоемкости разработки аппаратной части, где зависимость близка к линейной при хорошо структурированном проекте, и особенно высока для сложнофункциональных микросхем. Трудоемкость создания тестового АПО для таких микросхем может превышать трудоемкость всей аппаратной разработки.

Тестовое АПО в данном проекте должно разрабатываться независимыми коллективами разработчиков. Опыт разработок показал, что человек, пишущий независимые тесты, может найти больше ошибок, чем сам разработчик, спроектировавший соответствующий аппаратный узел.

Процесс тестирования КПИ можно разделить на три крупных стадии [5]:

- Тестирование с использованием тестового окружения;
- Отладка на макете;
- Проверка опытного образца.

Рассмотрим каждую из этих стадий отдельно.

2.2.1 Тестирование с использованием тестового окружения

Для начальной верификации КПИ и его компонентов используются специализированное тестовое окружение, позволяющее начать отладку сразу после написания RTL.

Тестовое окружение представляет собой ПО, написанное на языках C++, Verilog. Оно моделирует поведение КПИ согласно его спецификации. Для моделирования взаимодействия КПИ с внешними устройствами (жесткий диск, принтер и т.д.) должны быть разработаны соответствующие модели данных устройств, т.е. контроллеров.

На данном этапе могут использоваться следующие виды тестов:

- *Архитектурные тесты.*

Данные тесты проверяют соответствие модели разрабатываемой архитектуре (спецификации);

- *Тесты всех функциональных устройств, входящих в КПИ.*

Для КПИ в качестве функциональных тестов выступают тесты, проверяющие операции ввода/вывода. Например, для PCI устройств должен быть следующий набор тестов: проверка конфигурационного пространства устройства (чтение/запись в области конфигурационного пространства), проверка режима DMA передачи данных (в качестве параметров такого теста выступают - максимальный объем передачи данных, тип операции чтение или запись, максимальный объем одной операции чтение или запись, временные паузы между данными операциями, количество запуска данного вида теста, возможность создавать на шине различные ситуации, которые предусмотрены стандартом шины PCI), передачи данных в обычном режиме, работы

устройства в качестве ведущего (Master) и ведомого (Slave), доступ к внутренним регистрам данного устройства. В качестве функционального узла выступают контроллеры, входящие в состав КПИ;

- *Тесты взаимодействия между устройствами, входящими в состав КПИ.* Данными тестами проверяется работа системного коммутатора, обеспечивающего взаимосвязь между контроллерами периферийных устройств;

- *Тестирование устройства в целом.*

Должны быть созданы тесты, проверяющие взаимодействие всех узлов модели и функционирование микросхемы в целом;

- *Тесты взаимодействия модели микросхемы с внешними устройствами.*

Должна быть протестирована работа микросхемы с микросхемами окружения и внешними устройствами. Для возможности данного тестирования должны быть соответствующие модели окружения;

- *Соответствие логической модели поведенческой.*

Результаты (состояния регистров) тестирования поведенческой модели и модели уровня RTL должны совпадать.

Данный способ тестирования имеет свои достоинства и недостатки.

К достоинствам данного способа относится: возможность тестирования устройства сразу после написания RTL модели, быстрое изменение тестового окружения в зависимости от требуемой задачи.

К недостаткам данного способа относится: малый объем передаваемых данных при тестировании около 2-3 Кбайт, низкая скорость моделирования, для прогона большого количества тестов требуется значительные вычислительные мощности.

Преодолеть все данные недостатки позволяет отладка на макете.

2.2.2 Отладка на макете

Должен быть создан прототип микросхемы на базе программируемых пользователями микросхем ПЛИС. На данном прототипе должна быть возможность подключения внешних устройств: принтера, жестких дисков, флэш карт с разъемом USB, аудио устройств. В качестве управляющего устройства должен быть установлен процессор, для разрабатываемого комплекса Эльбрус-3С в качестве него используется микропроцессор "Эльбрус-S" и "МЦСТ-4R".

Данный прототип позволяет проводить следующие тесты:

- *Повторное выполнение всех архитектурных, функциональных тестов;*
- *Тесты проверки всей системы целиком, включая взаимодействие процессора с КПИ;*
- *Проверка работы микросхемы под управлением ОС Linux и кодов начальной загрузки;*
- *Проверка работы КПИ с реальными внешними устройствами (жесткий диск, Ethernet, принтер, аудио устройства и т.д.);*
- *Возможность запуска параллельных тестов.*

Т.е. одновременные запросы к нескольким устройствам. Таким способом можно изучить и проверить максимальную пропускную способность КПИ - какое количество запросов он одновременно может обрабатывать. Для этой цели может использоваться специальный буфер, в который будут накапливаться запросы и затем по команде они будут выполняться, или запросы будут поступать в КПИ в определенные моменты времени, которые задает операционная система.

Данный подход имеет свои преимущества по сравнению с предыдущим пунктом: позволяет достигнуть частоты близкие к рабочим (до 100 МГц), что позволяет создавать полностью работающую систему. Количество передаваемых данных при тестах близко к реальным, время выполнения данных тестов занимает несколько минут, часов (в зависимости от объема теста) по сравнению с несколькими днями при использовании тестового окружения.

Это очень эффективное тестирование, однако, для каждой микросхемы приходится разрабатывать свои технологические платы – прототипы требуемых плат. На первый взгляд, данное тестирование приводит к заметному повышению трудоемкости и увеличению времени проектирования. Однако при создании практически всех микросхем данное тестирование позволяло найти ошибки, которые привели бы к необходимости повторного запуска процедуры изготовления кристаллов. В конечном итоге создание тестовых плат приводит к сокращению стоимости проектов без увеличения времени проектирования.

Одновременно с данным видом тестирования, становятся доступны результаты топологического проектирования. Поэтому на данной стадии необходимо проводить *тестирование модели микросхемы транзисторного уровня после топологического проектирования*, по мере готовности частей проекта.

Данные тесты готовятся на базе ранее подготовленных архитектурных тестов и тестов внутренних узлов. Основная задача тестов состоит в соблюдении всех временных параметров функционирования микросхемы после перехода на транзисторный уровень и появлении паразитных элементов. Для достижения требуемых параметров иногда приходится возвращаться к изменению логической модели микросхемы. Данное моделирование не позволяет проводить детальное тестирование в силу большого времени выполнения тестирования, но выполнение его крайне необходимо для выявления ошибок топологического проектирования.

После выполнения данных стадий осуществляется отправка всей необходимой документации на завод для изготовления данной микросхемы. Срок изготовления данной микросхемы составляет 3-4 месяца. Затем начинаются испытания опытного образца.

2.2.3 Проверка опытного образца

При проверке опытного образца используются *тесты для отбраковки кристаллов и микросхем, которые направлены на поиск возможных дефектов при их изготовлении*.

Различают следующие средства тестирования: проверка массивов логических блоков, массивов памяти и аналоговых блоков. Основная задача средств тестирования – выявление максимального количества технологических дефектов. При этом необходима минимизация времени и площади элементов тестирования.

Для упрощения, ускорения тестирования и отбраковки кристаллов и микросхем используется введение в кристалл сканирующих цепочек.

Применение сканирующих цепочек описано стандартом IEEE 1149.1 (JTAG) и получило название – технология граничного сканирования (ГС). Она утвердилась как незаменимый инструмент при тестировании устройств с ограниченным доступом к выводам интегральных микросхем.

Обобщенная схема микросхемы [6], содержащей структуру ГС, приведена на рисунке 2.3. Центральный эллипс изображает функциональное ядро микросхемы. Вокруг ядра МС показаны контакты ввода/вывода (КВВ) микросхемы, их связи с функциональным ядром не показаны на рисунке для его упрощения.

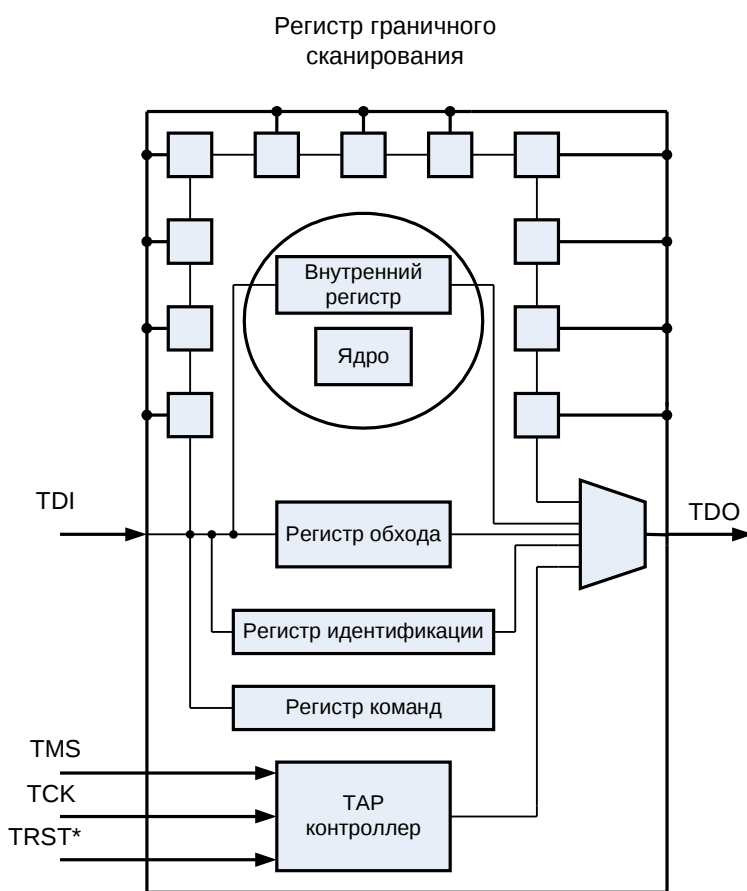


Рисунок 2.3– Обобщенная схема микросхемы, содержащей структуру ГС

Для управления элементами структуры ГС предусмотрены четыре контакта (а также необязательный пятый). Это контакты TDI, TDO, TMS, TCK, /TRST.

Функциональное назначение этих линий:

- **TDI** (вход тестовых данных) — вход последовательных данных периферийного сканирования. Команды и данные вдвигаются в микросхему с этого вывода по переднему фронту сигнала TCK;
- **TDO** (выход тестовых данных) — выход последовательных данных. Команды и данные выдвигаются из микросхемы с этого вывода по заднему фронту сигнала TCK;
- **TCK** (вход тестового тактирования) — тактирует работу встроенного автомата управления периферийным сканированием. Максимальная частота сканирования периферийных ячеек зависит от используемой аппаратной части и на данный момент ограничена 25...40 МГц;
- **TMS** (вход управления тестированием) — обеспечивает выбор режима тестирования;
- В некоторых случаях к перечисленным сигналам добавляется сигнал **TRST** для инициализации порта тестирования, что необязательно, так как инициализация возможна путем подачи соответствующей последовательности сигналов на вход TMS.

Контроллер TAP (Test Access Port) является центральным блоком управления протоколом ГС и всеми событиями в данной структуре. Контроллер управляется сигналом выборки тестового режима TMS, синхронизированным TCK, или асинхронно сбрасывается сигналом /TRST (если он есть). Сигналы выборки тестового режима синхронизированы передним фронтом пульса TCK.

Все внутренние регистры в структуре ГС являются сдвиговыми. Внутренние регистры в структуре ГС делятся на две группы: **группа IR** содержит только один обязательный регистр команд РК (Instruction Register), **группа DR** содержит все остальные регистры данных (Data Registers) — регистр обхода РО (Bypass), регистр идентификации РИ (Identification), регистр граничного сканирования РГС (Boundary-Scan). Обязательными являются наличие регистра команд и регистров граничного сканирования. Регистр граничного сканирования позволяет проверять соединения платы, обнаруживая типичные промышленные дефекты: висящие выводы, короткие замыкания. Он обладает доступом к

Тест на проверку пайки выводов микросхемы

Микросхема КПИ имеет 566 регистров граничного сканирования.

Она имеет следующие типы выводов:

- Входные (input) - для них достаточно одного РГС;
- Выходные (output) - для них необходимо 2 РГС (1 – сигнал разрешения (OE), 2 – выводимое значение);
- Двухнаправленные (inout) - для них необходимо два РГС.

Для проведения данной проверки необходимо наличие платы, соединяющей выходные выводы с входными и двухнаправленные выводы с двухнаправленными.

Данный тест заключается в следующем:

1. В регистр команд записывается команда *Execute*, которая отключает РГС от внутренней логики микросхемы;
2. В состоянии *ShiftDR* TAP контроллера через контакт TDI вводятся значения для выходных выводов;
3. В состоянии *UpdateDR* TAP контроллера фиксируем данные значения;
4. На следующей итерации в состоянии *CaptureDR* фиксируем данные, которые мы выводили в предыдущей итерации, на входах МС.
5. В состоянии *ShiftDR* TAP контроллера через контакт TDI вводим новые значения для выходных выводов, а через контакт TDO получаем данные для сравнения.

Операция сравнения полученных данных происходит по следующей формуле:

$$(TOD \wedge OD) \& Mask == 0 \quad (1)$$

где *TDO* — тест-вектор, содержащий ожидаемые выходные данные на контакте TDO;

OD — вектор реакций, полученные выходные данные на контакте TDO;

Mask — маска.

Маской называется вектор, состоящий из единиц и нулей, длина которого равна длине выходного вектора реакций, ожидаемого на контакте TDO в рамках данного входного тест-вектора. Разряд вектора реакций, маска которого равна 1, является тестовым и предназначен для сравнения с соответствующим разрядом выходного вектора реакций, получаемого на

TDO. Разряд вектора реакций, маска которого равна 0, не является тестовым и сравниваться с соответствующим разрядом выходного вектора реакций на TDO не будет.

Тест на проверку статических параметров микросхемы

Назначение данного теста в проверке электрических параметров микросхемы при приемке и поставке, т.е. они должны соответствовать заданным характеристикам, указанным в ТЗ.

Например, для основных выходов каналов:

- Выходное напряжение высокого уровня $V_{oh} > 2,4 \text{ В}$;
- Выходное напряжение низкого уровня $V_{ol} < 0,4 \text{ В}$.

Для проверки данных параметров на соответствующие PГС подается значение логической “1” или “0”, в зависимости от того, что требуется проверить.

Затем полученные выходные данные с контакта TDO подаются на определенные входы тестера и анализируются соответствующим программным обеспечением.

Обобщая, рассмотренные варианты верификации микросхем, с учетом специфики КПИ получим методику для его тестирования (рисунок 2.5).

Рассмотрим более подробно этап отладки на макете, его состав и функциональные возможности.

2.3 Прототип микросхемы КПИ

2.3.1 Прототипы КПИ на основе ПЛИС

Прототип КПИ будет выполнен на основе ПЛИС. Выбор ПЛИС обоснован тем, что данные микросхемы идеально подходят в процессе отладки на макете:

- Возможность переконфигурировать данные микросхемы в процессе отладке;
- Два варианта прошивки: временная (после подачи сигнала “сброса” данная конфигурация ПЛИС стирается), постоянная.

К недостаткам данного способа относится к ограничению возможности ПЛИС, невозможность достичь реальных рабочих частот на макете.

При проектировании в кратчайшие сроки набора контроллеров периферийных интерфейсов на разработку прототипа на основе ПЛИС предъявляются следующие требования:

1. Модуль должен позволять провести полное тестирование RTL описания заданных контроллеров;
2. Модуль должен позволять провести разработку и тестирование программ инициализации и драйверов заданных контроллеров;
3. Модуль должен позволять тестирование контроллеров, как по отдельности, так и всего набора одновременно.

Разработка единого прототипа на основе ПЛИС удовлетворяющего вышеуказанным требованиям без наличия микросхем процессоров для разрабатываемого набора контроллеров потребовала бы недопустимо большого временного периода. Чтобы обеспечить одновременную разработку как RTL моделей контроллеров, так и системного программного обеспечения, а также выполнить вышеуказанные требования были приняты следующие решения:

1. Реализовать каждый контроллер в виде устройства на шине интерфейса PCI в соответствии с программной моделью КПИ;
2. Разработать PCI карту – для прототипа контроллеров интерфейсов USB 2.0 и SATA вер. 2.x с целью обеспечения на раннем этапе одновременного тестирования имеющейся версии RTL модели аппаратуры и разработки системного программного обеспечения;
3. Разработать единый прототип – основной макет PIONUB (Prototype Input/Output Controller) для тестирования всего набора контроллеров периферийных интерфейсов при их подключении к общему интерфейсу с процессором (IOLink) через общий коммутатор.

2.3.2 Основной макет

Основной макет представляет из себя стенд, на котором находится ПЛИС с полным набором КПИ.

Так как пока не готовы микропроцессоры "Эльбрус-S", "МЦСТ-4R", то комплексная отладки всей системы вместе с КПИ не возможна. Поэтому разработан стенд для отладки только КПИ. Процессора имеющего IO-Link интерфейс, так же отсутствует. Поэтому, вместо данной конфигурации будет использоваться следующая: взят известный процессор с готовым программным обеспечением для него, для связи его с КПИ по IO-Link интерфейсу необходимо разработать связывающее устройство.

Для сокращения времени написания специализированных тестов для проверки подсистемы ввода – вывода КПИ на макете, на данном устройстве будут использоваться те же тесты, которые использовались на тестовом окружении КПИ.

К данному устройству предъявляются следующие требования:

- Возможность подключения выделенной памяти большого объема для проверки режима прямого доступа к памяти (DMA) и проверить работу системы на реальных соотношениях частот при обращении к памяти;
- Наличие интерфейса IOLink для связи с КПИ.

В качестве управляющего устройства может использоваться:

- Инструментальная машина (подсоединяется с помощью специализированного кабеля);
- Подсистема управления на базе известного процессора, расположенного непосредственно на этой же плате, либо отдельно.

В качестве управляющего устройства выбрана система на кристалле “МЦСТ R-500S”.

Микросхема R-500S представляет собой двухпроцессорную систему на кристалле с встроенными кэшем второго уровня, контроллером оперативной памяти и контроллерами периферийных каналов. Она является отечественной универсальной системой на кристалле с процессорным ядром архитектуры SPARC. Микросхема разработана по технологическим нормам 0,13 мкм с использованием библиотек стандартных элементов.

В качестве периферийной шины данный процессор использует шину PCI.

Поэтому тестирующее устройство должно обладать следующим интерфейсом:

- Наличие интерфейса PCI для взаимодействия с процессором R500S.

Для уменьшения временных и денежных затрат для разработки отдельной платы для тестирующего устройства принято решение расположить его непосредственно на макете в виде ПЛИС.

Функциональная схема макета КПИ показана на рисунке 2.6.

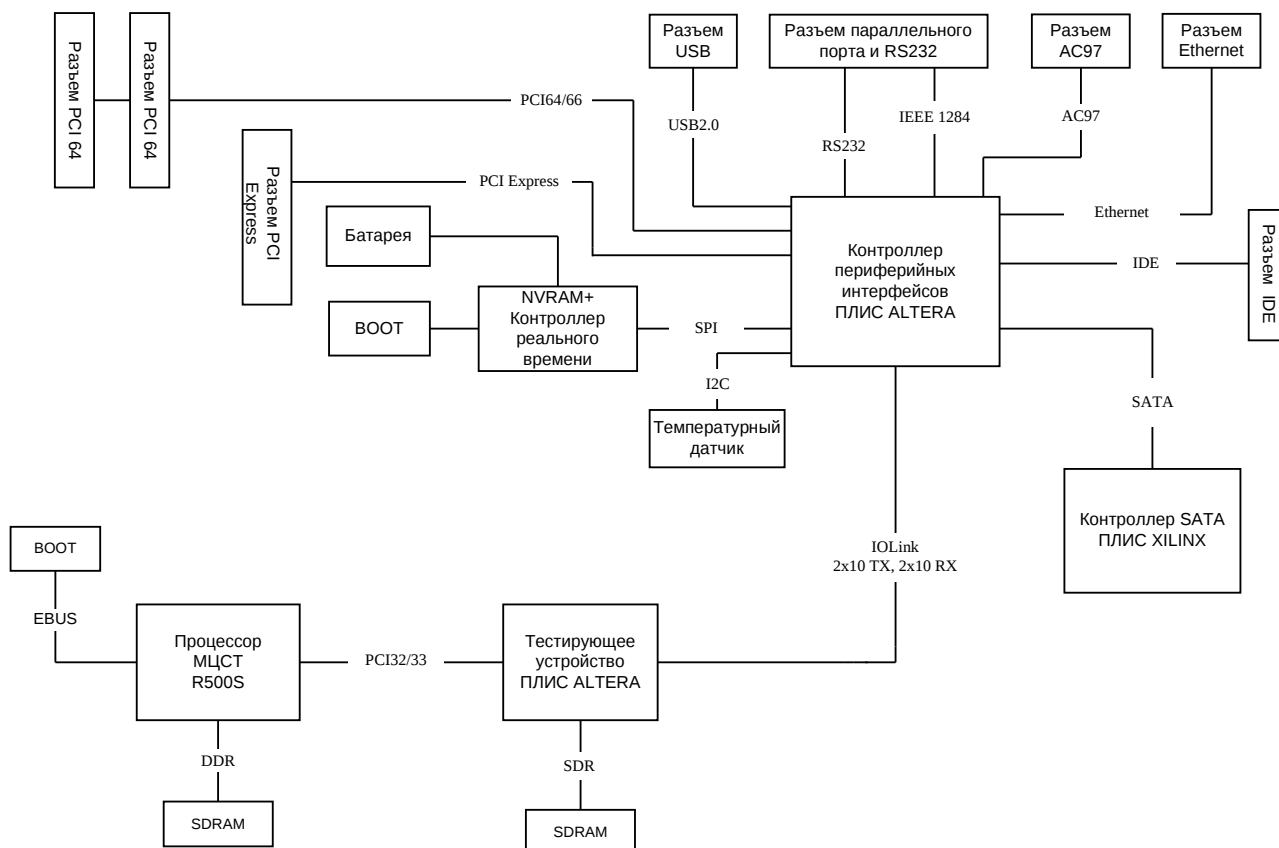


Рисунок 2.6 – Функциональная схема макета КПИ

В данной схеме можно выделить 4 части:

1. Управляющая часть – к ней относится процессор R500S и вся необходимая периферия для его полноценной работы (Boot, SDRAM и т.д.);
2. Тестирующее устройство, реализовано в виде ПЛИС компании Altera, Контроллер SATA, выполненный на ПЛИС компании Xilinx, так как особенностью данной ПЛИС является наличие встроенного физического уровня PHY интерфейса SATA вер. 2.x;
3. Контроллер КПИ, реализованный на ПЛИС компании Altera;
4. Наличие всех необходимых разъемов для подключения периферийных устройств.

Выводы

Разработана методика тестирования микросхемы КПИ, позволяющая провести тестирование проекта на всех этапах работ от разработки до производства микросхемы, включая проверку после получения опытных образцов микросхемы с фабрики.

АПК ТКПИ будет реализовано в виде ПЛИС в составе основного макета КПИ.

3 Проектно-конструкторская часть

3.1 Разработка структурной схемы АПК ТКПИ

Взаимодействие АПК ТКПИ с КПИ

АПК ТКПИ взаимодействует с КПИ посредством IO-Link интерфейса. Данный интерфейс служит для отправки в КПИ запросов в режиме программируемого ввода/вывода (PIO запросы) и получения ответов на них в зависимости от посылаемого запроса. Также осуществляется обмен данными в режиме DMA (прямой доступ к памяти) между DMA контроллером, одного из интерфейсов, входящего в состав КПИ и АПК ТКПИ.

Интерфейс IO-Link

IO-link интерфейс предназначен для передачи запросов и ответов на запросы [7]. Запросы и ответы передаются в виде пакетов, имеющих различный формат в зависимости от их параметров. Данный интерфейс представляет собой два канала, которые предназначены для одновременной и независимой передачи пакетов в противоположных направлениях. Протокол обмена в обоих каналах один и тот же. Канал представляет собой приемник и передатчик, находящиеся на противоположных концах линии связи.

Для сокращения времени разработки в АПК ТКПИ для работы с данным интерфейсом будем использовать тот же контроллер, который реализован в КПИ.

В КПИ связь по данному интерфейсу осуществляет системный контроллер высокоскоростного ввода/вывода (Input/Output Communication Controller) для организации межмашинного/внутрисистемного обмена.

Для сокращения времени разработки АПК ТКПИ для работы с данным интерфейсом будем использовать этот же контроллер.

ИОСС может работать в двух основных режимах: RDMA (Remote DMA) и IO (Input/Output). Режим RDMA предназначен для межмашинного обмена, а режим IO-внутрисистемного. Контроллер имеет внешний последовательный интерфейс (LVDS канал связи) и интерфейс с системой.

Внешний последовательный интерфейс

Таблица 3.1 - Интерфейс передатчика для LVDS канала связи

Название сигнала	Тип	Описание сигнала
1	2	3
lv_tclock	Выход	Сигнал стробирования
lv_tval	Выход	Сигнал пакетной синхронизации
lv_tdat [7:0]	Выход	Линии данных

Таблица 3.2 - Интерфейс приемника для LVDS канала связи

Название сигнала	Тип	Описание сигнала
1	2	3
lv_rclock	Вход	Сигнал стробирования
lv_rval	Вход	Сигнал пакетной синхронизации
lv_rdat [7:0]	Вход	Линии данных

Обмен по каналу связи производится пакетами. Все пакеты содержат заголовок размером: 8, 12 или 16 байт, защищенный 16-ти разрядным CRC-кодом. После заголовка может идти блок данных размером: 4-64 байта (с шагом 4 байта), защищенный 32-х разрядным CRC-кодом. Максимальный размер пакета — 84 байта, минимальный — 8 байт.

В поле заголовка передается тип, размер и номер пакета в последовательности. Транзакция обмена состоит из двух фаз: передача пакета и подтверждение приема. Для подтверждения успешного приема пакета или сообщения об ошибке, используются служебные пакеты, которые формируются автоматически. В случае отсутствия по какой — либо причине подтверждения приема, транзакция заканчивается по превышению лимита времени. В случае ошибки, либо превышения лимита времени, передатчик производит повтор пакета(ов) с номера последнего успешно переданного.

Интерфейс с системой

Интерфейс передатчика показан в таблице 3.3, а интерфейс приемника в таблице 3.4.

Таблица 3.3 – Интерфейс передатчика

Название сигнала	Тип	Описание сигнала
1	2	3
up_val	Выход	Строб значимости транзакции (ширина 1 бит)
up_dat [31:0]	Выход	Линия передачи пакета
up_rdy	Вход	Сигнал готовности принимающей стороны (Пакеты ответов передаются передатчиком независимо от состояния данного сигнала)

Таблица 3.4 – Интерфейс приемника

Название сигнала	Тип	Описание сигнала
1	2	3
dn_val	Выход	Строб значимости транзакции (ширина 1 бит)
dn_dat [31:0]	Выход	Линия передачи пакета
dn_rdy	Вход	Сигнал готовности приема пакета (Пакеты ответов принимаются всегда, независимо от состояния сигнала dn_rdy)

Транзакции (запросы и ответы) состоят из 2 блоков: заголовка и данных (если есть). Блок заголовка начинается одновременно с выставлением stroba значимости и может содержать от 2 до 4 четырех байтовых слов. Размер блока заголовка определяется в поле «PacketAttributes». В последнем слове блока заголовка старшие 2 байта не являются значимыми и должны быть нулями (место для CRC заголовка). Далее передается блок данных, если это определено в поле «PacketAttributes». Между транзакциями обязателен разрыв не менее такта. Адресация байт в блоке данных «Big Endian».

Описание типов пакетов обмена интерфейса IO-Link

Для общения с КПИ, он поддерживает следующие виды заголовков пакетов [8]:

- Запрос в пространство памяти (64 битная адресация);
- Запрос в пространство памяти (32 битная адресация);
- Запрос в пространство конфигураций;

- Пакет завершения (ответа);
- Пакет обслуживания очередей.

Формат данных заголовков показан в таблице 3.5.

Таблица 3.5– Форматы заголовков пакетов

Слова\биты	31-24	23-16	15-8	7-0		
Заголовок пакета запроса в пространство памяти (64-х битная адресация)						
0	ID	COP	PacketAttributes			
1	Адрес[31:0]					
2	Адрес[32:63]					
3	CRC		BEMask	Size		
Заголовок пакета запроса в пространство памяти (32-х битная адресация)						
0	ID	COP	PacketAttributes			
1	Адрес[31:0]					
2	CRC		BEMask	Size		
Заголовок пакета запроса в пространство конфигурации						
Слова\биты	31-24	23-16	15-8	7-0		
0	ID	COP	PacketAttributes			
1	BusNum	[23:19] DevNum	[18:16] FuncNum	[15:12] Reserved	[11:8] RegNum1	RegNum0
2	CRC		BEMask		Size	
Заголовок пакета завершения (ответа)						
Слова\биты	31-24	23-16	15-8	7-0		
0	ID	COP	PacketAttributes			
1	CRC		[15:12] Status	[11:8] Address[5:2]	Size	
Заголовок пакета обслуживания очередей						
Слова\биты	31-24	23-16	15-8	7-0		
0	Reserved	COP	PacketAttributes			
1	CRC				FreeNum	

Где: ID[7:4] – идентификатор устройства запроса;

ID[3:0] – Номер запроса. При формировании пакета запроса абонент помещает в данное поле уникальный номер запроса. При формировании пакета ответа абонент помещает в данное поле номер запроса, которому соответствует данный пакет.

COP – Коды операций:

000 0000 – чтение из памяти (32 битная адресация);

100 0000 – запись в память (32 битная адресация);

010 0000 – чтение из памяти (64 битная адресация);

110 0000 – запись в память (64 битная адресация);

000 1001 – чтение boot;

000 0010 – чтение из пространства ввода/вывода;

100 0010 – запись в пространство ввода/вывода;

000 0100 – чтение 0-го конф. пространства;

100 0100 – запись в 0-ое конф. пространство;

000 0101 – чтение 1-го конф. пространства;

100 0101 – запись в 1-ое конф. пространство.

PacketAttributes[15:5] – не используются;

PacketAttributes[4] – признак “Запрос от ioapic” (Активный уровень “1”);

PacketAttributes[3:0] – Атрибуты пакета:

- xx01 – размер заголовка 2 DW (заголовок не содержит адреса);
- xx10 - размер заголовка 3 DW (заголовок содержит 32-битный адрес);
- xx11 - размер заголовка 4 DW (заголовок содержит 64-битный адрес);
- x1xx – пакет содержит данные;
- x0xx – пакет не содержит данные;
- 1xxx – “почтовый” пакет;
- 0xxx – “не почтовый” пакет.

Status[2:0] – статус завершения выполнения запроса:

- 000 – нормальное завершение;
- 001 – не поддерживаемый запрос;
- 100 – устройство назначения отвергло запрос.

BEMask[7:4] – Значимость байтов в последнем слове блока данных пакета запроса.

BEMask[3:0] – Значимость байтов в первом слове блока данных пакета запроса записи.

Size[7:4] – зарезервировано.

Size[3:0] – Размер блока данных пакета (запроса или ответа) в 4-х байтовых словах.

- 0001 – 1 слово;
- 0010 – 2 слова;
- ...
- 0000 – 16 слов.

Работы с данными пакетами разделим на две стадии:

- Формирование IO-Link пакета;
- Обработка пакетов, поступающих со стороны КПИ.

Для формирования IO-Link пакетов выделим специальный FIFO буфер – последовательный буфер (Sequence buffer).

Для обработки пакетов, поступающих со стороны КПИ, потребуются следующие буферы:

- Буфер IO-Link пакетов завершений “*Буфер ответов*” для всех пакетов завершений, размером 512 двойных слов, что позволяет разместить в нем не менее 16 IO-Link пакетов ответов максимального размера (максимальный размер пакета ответов 18 двойных слов);
- Буфер IO-Link пакетов APIC-DMA запросов “*APIC буфер*” для всех пакетов с активным признаком “*APIC-запрос*” в заголовке пакета, размером 128 двойных слов, что позволяет разместить в нем не менее 16 IO-Link пакетов APIC-запросов максимального размера (максимальный размер 5 двойных слов);
- Буфер IO-Link, содержащий “потерянные” пакеты запросов/ответов при адресном промахе.

Взаимодействие АПК ТКПИ с процессором

Взаимодействие АПК ТКПИ с процессором происходит по шине PCI.

Она используется для формирования IO-Link запросов в КПИ, чтения полученных ответов на данные запросы, конфигурирования управляющих регистров, чтения регистров статусов.

Шина PCI должна предоставлять доступ в следующие буферы:

- В ОЗУ контроллер памяти АПК ТКПИ объемом 64 Мб;
- В буфер последовательной выдачи;
- В буфер IO-Link пакетов APIC-DMA-запросов;
- В буфер IO-Link пакетов завершений;
- В пространство регистров статуса и управления “*Блок регистров*”.

Для данной цели в конфигурационном пространстве PCI используем пять регистров базового адреса.

Краткое описание работы SDRAM-памяти и контроллера памяти

В АПК ТКПИ используется четыре микросхемы SDRAM памяти (синхронное динамическое ОЗУ - Synchronous Dynamic RAM) памяти производителя Micron типа

MT48LC32M16A2TG-75C, каждая из которых имеет объем 536 870 912 битов организацией 4x16x8 (“число банков”x”число слов в банке”x”разрядность слова”) [9]. Соответственно, каждая микросхема имеет объем 512 Мбит, а четыре – обеспечивают общий объем памяти равный 256 Мбайт. Данные микросхемы памяти работают на частоте 133 МГц и подключены таким образом, чтобы обеспечить 64-х разрядную организацию шины данных памяти, при этом работа с этими четырьмя микросхемами происходит как с одной микросхемой разрядностью слова 64 бита.

Для ускорения времени разработки АПК ТКПИ в качестве контроллера памяти выбран ранее разработанный в ЗАО “МЦСТ” контроллер памяти SDRAM, успешно используемый в других проектах.

Суммарный объем ОЗУ контроллера памяти равен 64 Мб. Ширина шины данных равна 64 бита. Частота 133 МГц. Данный контроллер имеет два канала со стороны внутреннего пользования для обмена данными <sdr0> и <sdr1>.

Канал <sdr0> используем для обмена данными со стороны шины PCI. Для этой цели потребуется модуль “*машина состояний PCI2Mem*”, отвечающая за организацию обмена данными. А канал <sdr1> - для обмена данными со стороны канала IO-Link потребуется “*машина состояний IOL2Mem*”.

Память АПК ТКПИ используется для проверки режима работы прямого доступа в память КПИ. Со стороны шины PCI в ОЗУ осуществляется запись данных, которые будут считаны контроллером прямого доступа памяти одним из устройств, входящих в состав КПИ. Так же возможно считывание полученных данных от контроллера прямого доступа памяти и дальнейший их анализ на инструментальной машине.

Основной алгоритм работы в данном режиме показан на рисунке 3.1.

Рисунок 3.1 – Алгоритм проверки в режиме DMA

Описание работы АПК ТКПИ

Структурная схема АПК ТКПИ показана на рисунке 3.2.

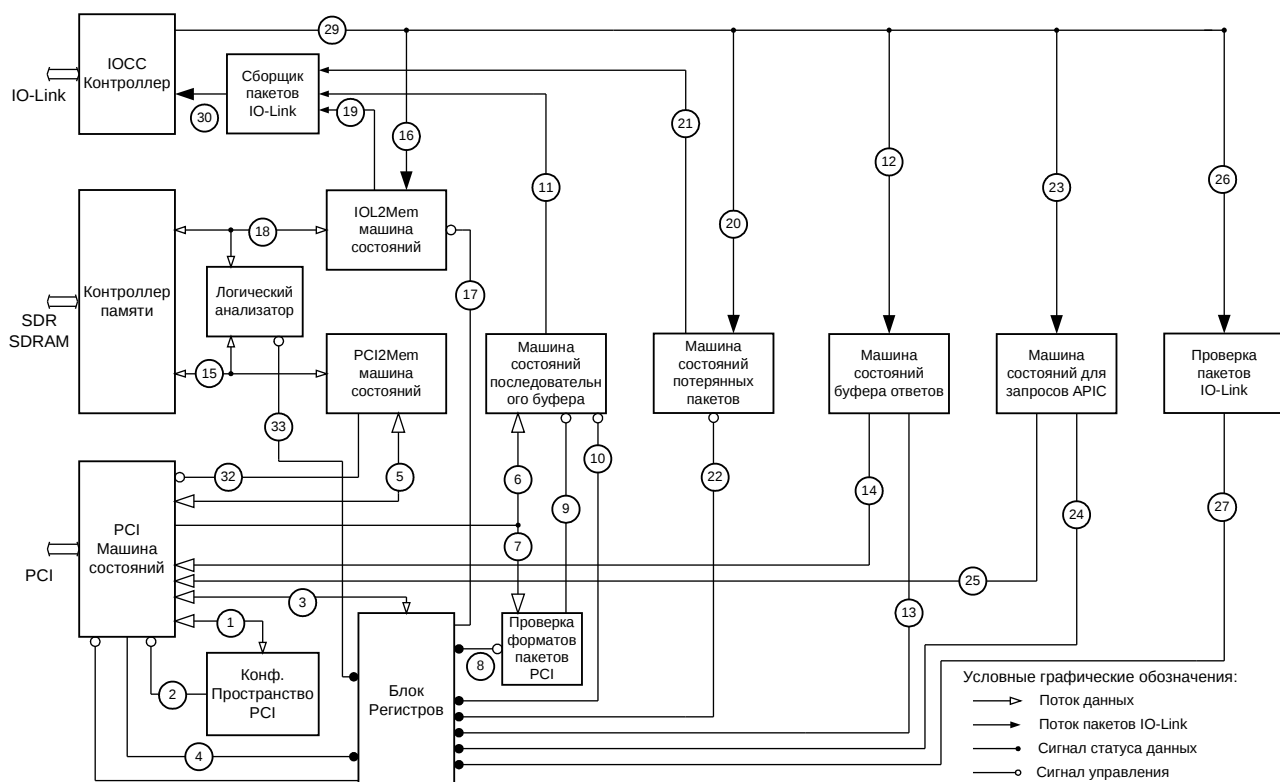


Рисунок 3.2 – Структурная схема АПК ТКПИ

Перед началом работы выполняется чтение и программирование регистров конфигурационного пространства (регистров управления доступом к PCI-устройству) линия 1. При нарушении требования на размер конфигурационного обращения (размер должен быть ≤ 4 байта) 1-я фаза обмена данными выполняется и завершается нормально (при операции записи данные заносятся в соответствующий регистр, при операции чтения на шину выдаются требуемые данные). В следующей фазе модуль “Машина состояний PCI” выполняет завершение типа “disconnect without data” и в модуль “Блок регистров” передается сигнал об ошибке доступа со стороны шины PCI (линия 4), где устанавливается в активное состояние соответствующий бит статуса. После выполнения процедуры программирования конфигурационных регистров выполняется инициализация, которая включает в себя сброс системы тестирования и программирование регистров управления в модуле “Блок регистров”.

Значения BARx – регистров передаются из модуля “Конфигурационное пространство PCI” в модуль “Машина состояний PCI” (линия 2), где они используются для маршрутизации текущего обращения со стороны шины PCI в один из указанных ниже модулей. Данные обращения должны удовлетворять требованиям на тип операции (чтение или запись) и на размер обращения (данные требования введены для сокращения ошибок при программировании АПК ТКПИ):

- “Машина состояний PCI2Mem” – чтение и запись любого размера;
- “Последовательный буфер” – запись размером 4 байта;
- “Буфер ответов” – чтение размером 4 байта;
- “Буфер APIC запросов” – чтение размером 4 байта;
- “Блок регистров” – чтение и запись с размерами 1, 2, 4 байта.

При нарушении приведенных выше требований, в модуль “Блок регистров” передается сигнал об ошибке доступа со стороны шины PCI (линия 4), где устанавливается в активное состояние соответствующий бит статуса. Сама операция обращения при этом аннулируется (при операциях записи данные не записываются, при операции чтения на шину возвращается “мусор”).

При обращении по чтению в ОЗУ АПК ТКПИ, модуль “Машина состояний PCI” выполняет проверку занятости модуля “Машина состояний PCI2Mem” (линия 32). В случае занятости на шине выполняется завершение транзакции типа “retry” (повторная попытка). В противном случае запрос направляется из модуля “Машина состояний PCI” в модуль “Машина состояний PCI2Mem”, при этом операция запроса не завершается, пока выполняется считывание 8 байт данных (линия 15) из ОЗУ посредством модуля “Контроллер памяти”. После получения данных от модуля “Машина состояний PCI2Mem” (линия 5) модуль “Машина состояний PCI” выполняет завершение типа “disconnect without data” (разъединение без данных). При этом данная ситуация не считается ошибочной и последующие фрагменты данных могут быть считаны PCI-ведущим посредством повторного обращения в ОЗУ АПК ТКПИ. После завершения приема данных записи с шины PCI модуль “Машина состояний PCI” выполняет передачу данных записи в модуль “Контроллер памяти” и на время данной операции устанавливает активный уровень сигнала занятости (линия 32).

Модуль “Машина состояний последовательного буфера” предназначен для приема со стороны шины PCI фрагментов IO-Link пакетов запросов (линия 6) и выдачи сформированных IO-Link пакетов в модуль “Мультиплексор пакетов IO-Link” (линия 11). Фрагменты, принимаемые по линии 6, помещаются в “Последовательный буфер”. Одновременно с этим принимаемые фрагменты направляются в модуль “Проверка форматов пакетов PCI” (линия 7), где выполняется проверка корректности параметров формируемого заголовка пакета. В случае обнаружения некорректности в модуль “Блок

регистров” передается сигнал обнаружения ошибки (линия 8), и там устанавливается в активное состояние соответствующий бит статуса. Если из модуля “Блок регистров” приходит сигнал запрета формирования пакетов с некорректными параметрами (линия 8), “Последовательный буфер” блокируется (линия 9), и дальнейшая работа с ним возможна только после повторной инициализации АПК ТКПИ. В противном случае формирование пакетов продолжается в штатном режиме.

При получении IO-Link пакета ответа из модуля “ИОСС контроллер” (линия 12) он помещается в “Буфер ответов” модуля “Машина состояний буфера ответов”. Модуль “Машина состояний буфера ответов” передает в модуль “Блок регистров” информацию о количестве принятых пакетов, о номере текущего пакета на выходе буфера и о попытках записи в буфер пакета на фоне его состояния “full” (линия 13). Пакет ответа может быть считан из “Буфера ответов” со стороны шины PCI последовательным чтением его 4-байтовых фрагментов (линия 14).

При получении IO-Link пакета APIC-запроса из модуля “ИОСС контроллер” (линия 23), он помещается в буфер “APIC буфер” модуля “Машина состояний APIC запросов”. Модуль “Машина состояний APIC запросов” передает в модуль “Блок регистров” информацию о количестве принятых пакетов и о номере текущего пакета на выходе буфера (линия 24). Пакет ответа может быть считан из буфера “APIC буфер” со стороны шины PCI последовательным чтением его 4-байтовых фрагментов (линия 25).

Модуль “Блок регистров” выполняет сбор информации о внутренних событиях. Данная информация может быть считана со стороны шины PCI (линия 3). Он осуществляет управление формированием прерываний на шине PCI (линия 28) и управлением внутренними процессами обмена, обработки данных. Регистры управления модуля программируются со стороны шины PCI (линия 3).

Поток IO-Link пакетов со стороны модуля “ИОСС контроллер” (линия 29) без маршрутизации направляется одновременно во все модули:

- “Машина состояний IOL2Mem”;
- “Машина состояний потерянных пакетов”;
- “Машина состояний буфера ответов”;
- “Машина состояний APIC запросов”;

- “Проверка интерфейса IO-Link”.

В модуль “Машина состояний IOL2Met” из модуля “Блок регистров” передается информация об адресном диапазоне ОЗУ со стороны интерфейса IO-Link и сигнал управления, задающий возможность направления пакета запроса в ОЗУ при адресном промахе (линия 17). На основании данной информации принятый пакет либо помещается в очередь запросов модуля “Машина состояний IOL2Met”, либо игнорируется. Запросы с вершины очереди запросов модуля “Машина состояний IOL2Met” обрабатываются и направляются в модуль “Контроллер памяти” (линия 18). Ответы на запросы чтения помещаются в очередь ответов модуля “Машина состояний IOL2Met”. Ответы с вершины очереди модуля “Машина состояний IOL2Met” обрабатываются и в виде IO-Link пакетов ответов направляются в модуль “Мультиплексор пакетов IO-Link” (линия 19).

В модуль “Машина состояний потерянных пакетов” из модуля “Блок регистров” передается информация об адресном диапазоне ОЗУ со стороны интерфейса IO-Link, сигнал управления, задающий возможность направления пакета запроса в ОЗУ при адресном промахе и сигнал управления, задающий поле статуса IO-Link пакета, который формируется в ответ на получение пакета запроса чтения, не попадающего в ОЗУ (линия 22). Принятый пакет, классифицируемый на основании данной информации как “потерянный”, помещается в очередь модуля “Машина состояний потерянных пакетов”. В противном случае пакет игнорируется. Запросы с вершины очереди обрабатываются и, в зависимости от сигнала управления, задающего поле статуса, формируется либо пакет ответа со статусом “нормальное завершение” с блоком данных размера, соответствующего параметру запроса, и с ‘1’ во всех разрядах блока данных, либо пакет ответа со статусом “неподдерживаемый запрос”. Сигнал адресного промаха передается в модуль “Блок регистров” (линия 22), где устанавливается в активное состояние соответствующий бит статуса.

Модуль “Проверка интерфейса IO-Link” выполняет:

- Анализ входящего потока IO-Link пакетов (линия 29) на предмет корректности параметров заголовка и на соответствие фактической длины пакета длине, задаваемой параметрами его заголовка;
- Контроль корректности работы подсистемы менеджмента очередей контроллера IOHub, подключенного к внешнему интерфейсу IO-Link.

При обнаружении ошибки формата пакета в модуль *”Блок регистров”* передается сигнал обнаружения ошибки формата и все параметры заголовка ошибочного пакета (*линия 27*). При обнаружении адресного промаха DMA-запросом из контроллера IOHUB в модуль *”Блок регистров”* передаются параметры заголовка пакета данного запроса (*линия 27*). При обнаружении ошибки подсистемы менеджмента очередей в модуль *”Блок регистров”* передается сигнал обнаружения ошибки (*линия 27*). По сигналам ошибки в модуле *”Блок регистров”* устанавливаются в активное состояние соответствующие биты статуса.

Модуль *”Мультиплексор пакетов IO-Link”* осуществляет учет количества свободных ресурсов приемной очереди контроллера IOHub, подключенного к внешнему интерфейсу IO-Link, а также выполняет арбитраж запросов на выдачу IO-Link пакетов со стороны модулей:

- *”Машина состояний IOL2Mem”*;
- *”Машина состояний последовательного буфера”*;
- *”Машина состояний потерянных пакетов”*;

Принятые от указанных модулей пакеты помещаются в буфер выдачи. В случае наличия свободных ресурсов приемной очереди контроллера IOHub и готовности контроллера IOCC очередной пакет из буфера выдачи передается в модуль *”IOCC контроллер”* (*линия 30*), из которого, в свою очередь, передается на внешний интерфейс IO-Link.

Модуль *”Логический анализатор”* выполняет запись трассы обмена на одном из каналов системного интерфейса контроллера ОЗУ под управлением сигналов из модуля *”Блок регистров”* (*линия 33*) .

3.2 Разработка аппаратных компонентов АПК ТКПИ

Выбор средства разработки

В настоящее время существует множество способов проектирования ПЛИС:

- Представление логической модели в виде графической принципиальной схемы. Затем с помощью специальных программ, данная схема преобразуется в описание на языке высокого уровня (язык описания аппаратуры), к таким языкам относятся Verilog, VHDL, AHDL, SystemC и д.р;
- Представление логической модели в виде автомата состояний, который

затем с помощью специальных программ преобразуется в описание на языке описания аппаратуры;

- Создание логической модели с помощью языков описания аппаратуры.

Первые два способа используются при разработке проектов небольших размеров и имеют следующие недостатки:

- Сложность в понимании графической схемы другими разработчиками;
- При внесении серьезных изменений происходит полная переработка графической части;

От всех данных недостатков освобожден третий способ — создание логической модели проекта непосредственно на языке описания аппаратуры.

Так как в ЗАО «МЦСТ» при разработке систем на ПЛИС и при проектировании заказных интегральных микросхем используется язык Verilog [10], то выберем его в качестве средства разработки.

Блок сопряжения

В АПК ТКПИ используются три опорных частоты:

1. Шина PCI с частотой синхронизации 33 МГц;
2. Шина памяти SDR SDRAM с частотой синхронизации 133 МГц;
3. Интерфейс IO-Link с частотой 100 МГц.

Поэтому при переходе сигнала из одного частотного домена в другой необходимо производить пересинхронизацию данного сигнала. Для этой цели необходимо создать блок сопряжения для пересинхронизации частот на границе частотных доменов [11]. Данную пересинхронизацию будет выполнять модуль «pov_tb_prs».

Для этого при небольшой разнице между опорными частотами применяется цепочка из двух регистров, что позволяет получить стабильный сигнал на выходе второго регистра (рисунок 3.3).

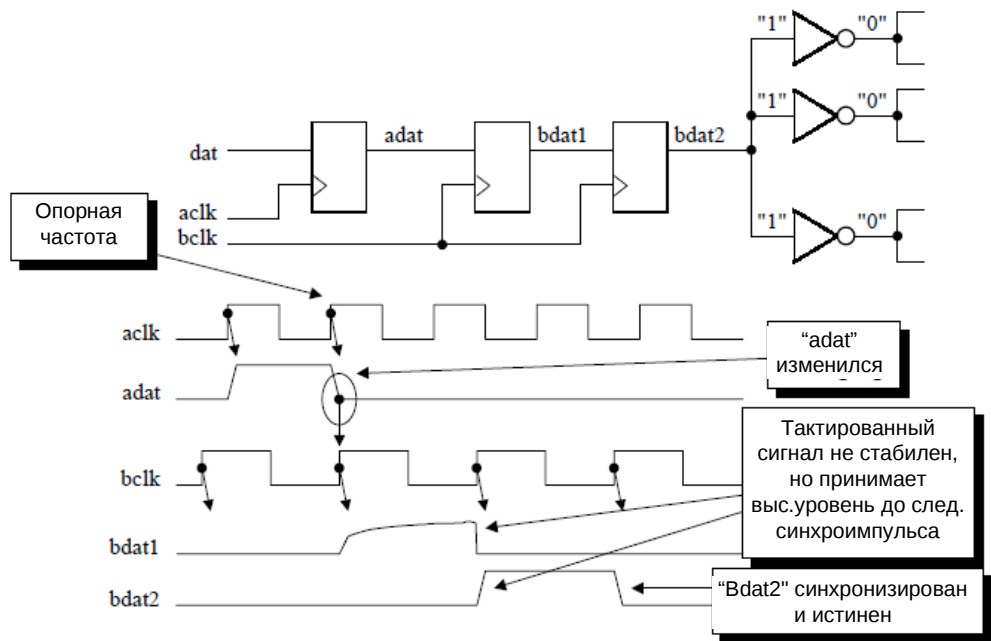


Рисунок 3.3 — Синхронизация частоты через два регистра

Так как в нашем случае разница между частотами существенна (частота синхронизации IO-Link больше частоты синхронизации PCI в 6 раз), то данная схема не подходит. Поэтому для фиксирования события прихода сигнала, требующего пересинхронизацию, введен однобитный счетчик. Он изменяет свое значение на противоположное с каждым приходом входного сигнала. Фрагмент RTL описания на Verilog показан ниже:

```
always @(posedge clk_0)
if(~rst_l_0)    i_cnt <= 1'b0;      // сброс счетчика при поступлении сигнала сброса
else if(i_pulse) i_cnt <= ~i_cnt;  // при возникновении сигнала счетчик меняет свое
                                   значение на противоположное
```

Сама синхронизация осуществляется на требуемой на выходе частоте, фрагмент RTL описания на Verilog показан ниже:

```
always @(posedge clk_1)
if(~rst_l_1) dline <= 2'b0;
else       dline <= {dline[0], i_cnt}; // Задержка сигнала на 1 такт.
```

Полное описание на Verilog данного модуля показано ниже:

```

module pov_tb__prs(      // заголовок модуля

input rst_0,           //сигнал сброса вх.частоты
input rst_1,           //сигнал сброса вых. частоты

input clk_0,           //входная тактовая частота
input clk_1,           //выходная тактовая частота

input i_pulse,        // сигнал требующий пересинхронизации на другую частоту

output o_pulse        // выходной пересинхронизированный сигнал
);
// Описание сигналов//
reg[1:0] rst_rt_0;
reg[1:0] rst_rt_1;

wire rst_l_0;
wire rst_l_1;

reg i_cnt;
reg o_cnt;
reg[1:0] dline;
//-----//
//-----//
wire rst_com = rst_0 & rst_1;

always @(posedge clk_0) rst_rt_0 <= {rst_rt_0[0], rst_rt_1[1]};      // для сигнала сброса также
always @(posedge clk_1) rst_rt_1 <= {rst_rt_1[0], rst_com};        // выполняется пересинхронизация
//-----//
assign rst_l_0 = rst_rt_0[1];
assign rst_l_1 = rst_rt_1[1];
//-----//
//-----//
always @(posedge clk_0)      //входной счетчик
    if(~rst_l_0) i_cnt <= 1'b0;
    else if(i_pulse) i_cnt <= ~i_cnt;

always @(posedge clk_1)      // задержка сигнала на такт
    if(~rst_l_1) dline <= 2'b0;
    else dline <= {dline[0], i_cnt};

always @(posedge clk_1)      //выходной счетчик
    if(~rst_l_1) o_cnt <= 1'b0;
    else if(dline[1] != o_cnt) o_cnt <= ~o_cnt;

assign o_pulse = (dline[1] != o_cnt);      // формирование выходного сигнала
//-----//
endmodule

```

Результат работы данного модуля показан на рисунке 3.4.

ошибочное состояние из-за которого могут ложно сработать сигналы статуса данного буфера и произойдет потеря данных.

Для избегания таких ситуаций используется счетчик в коде Грея. Особенностью кода Грея, является то, что его последовательные состояния различаются на один бит, что позволяет повысить помехоустойчивость преобразования кода.

Для того чтобы избежать обратного преобразования из кода Грея в двоичный код, сделана реализация, что при получении нового значения двоичного кода сразу же осуществляется его преобразование в код Грея.

Коды Грея легко получаются из двоичных чисел путём побитовой операции «Исключающее ИЛИ» с тем же числом, сдвинутым вправо на один бит. Следовательно, i -й бит кода Грея G_i выражается через биты двоичного кода B_i следующим образом:

$$G_i = B_i \oplus B_{i+1},$$

где \oplus – операция «исключающее ИЛИ»;

биты нумеруются справа налево, начиная с младшего.

Преобразование двоичного кода в код Грея на Verilog для домена записи показано ниже:

```
wire[3:0] wr_cnt_new = wr_cnt + 1'b1;    //увеличение дв. счетчика на 1
wire[3:0] wr_gcnt_new = {wr_cnt_new[3], wr_cnt_new[3:1] ^ wr_cnt_new[2:0]}; // преобразование в код
Грея
//-----//
always @(posedge w_clk) // Если выставлен сигнал подтверждения записи, то счетчик принимает новое
    if(~w_rst)           wr_cnt <= {3{1'b0}};    //значение
    else if(wr_cnt_I)     wr_cnt <= wr_cnt_new;
```

Таким образом, счетчики с кодом Грея используется для сравнения положения двух адресных указателей чтения и записи, формирования сигналов статусов данного буфера. А двоичные счетчики учитывают данные каждый в своем частотном домене.

Синхронизация выполнена так же как в блоке пересинхронизации, описанным выше.

Блок-схема асинхронного FIFO и его временная диаграмма показаны на рисунке 3.6 и 3.8 соответственно.

Рисунок 3.6 — Блок схема асинхронного FIFO

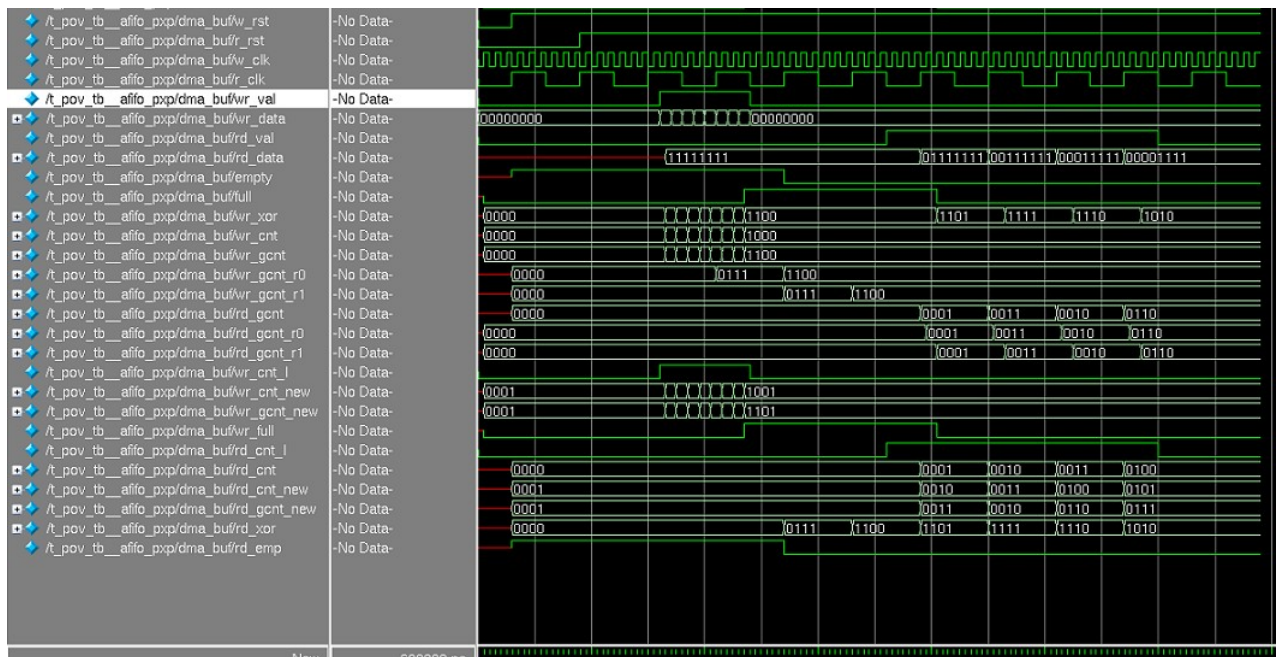


Рисунок 3.7 — Временная диаграмма работы асинхронного FIFO

На диаграмме показан пример работы такого буфера размером 8x8.

В начале, осуществляется запись 8 байт данных по сигналу <wr_clk> при выставленном сигнале <wr_val>. Значение двоичного счетчика записи при этом равно “1000” и счетчика записи в коде Грея “1100”. Что приводит к выставлению сигнала “Буфер полон” <wr_full>. Затем происходит чтение 4 байтов данных.

Разработанные модули и их назначение представлено в таблице 3.1

Таблица 3.7 – Разработанные модули и их назначение

Название модуля	Назначение
1	2
Pov_tb__afifo_pxp	Асинхронное FIFO для “последовательного буфера”, “буфера ответов” и “буфера APIC запросов”
Pov_tb__arb_a4	Реализация модуля “Мультиплексор пакетов IO-Link”
Pov_tb__dp_ram_pxp	Для временного хранения данных поступающих в режиме DMA со стороны IO-Link
Pov_tb__fifo_pxp	Синхронное FIFO для “буфера потерянных пакетов”
Pov_tb__pcnf	Конфигурационное пространство PCI
Pov_tb__prs	Блок пересинхронизации
Pov_tb__qmu	Осуществляет учет свободного места памяти в ОЗУ
Pov_tb__rb	Реализация модуля “блок регистров”
Pov_tb__sfifo_pxp	Реализует буфер трассы для логического анализатора.

Блок регистров

Данный модуль содержит в себе 46 программно доступных регистров.

Данные регистры можно разделить на 4 группы:

- Регистры управления буферами;
- Регистры статуса;
- Регистры, задающие адресное пространство памяти со стороны КПИ;
- Регистры сброса.

Каждый модуль, работающий в АПК ТКПИ, взаимодействует с блоком регистров, он получает от него необходимые данные и выставляет соответствующие биты в регистре статуса. В качестве примера рассмотрим принцип работы последовательного буфера.

Последовательный буфер

Данный буфер служит для формирования IO-Link пакетов PIO запросов в КПИ.

Его емкость равна 512 двойных слов, что позволяет разместить в нем не менее 16 IO-Link пакетов максимального размера.

Для формирования пакета необходимо последовательно выполнить N 4-х байтовых записей (N — полный размер IO-Link пакета), каждая из которых соответствует записи очередного 4-байтового фрагмента IO-Link пакета. Порядок записи фрагментов соответствует порядку передачи фрагментов пакета на IO-Link интерфейсе.

При записи очередного фрагмента заголовка в данный буфер, предусмотрена проверка на корректность формируемой комбинации параметров. В случае обнаружения некорректности, устанавливаются в активное состояние биты статуса `<sb_prog_status.pci_pkt_error0>`, `<sb_prog_status.pci_pkt_error0>` в зависимости от того в какой фазе приема заголовка пакета обнаружена некорректность.

Во время формирования заголовка пакета в буфере, пользователю доступна информация о текущей стадии через регистр статуса `<sb_prog_status>`, также при записи сохраняется фрагмент заголовка в регистре `<sb_header_X>`. Это позволяет определить параметры пакета, который не прошел проверку на корректность.

Конфигурационное адресное пространство PCI

Конфигурационное адресное пространство PCI состоит из 256 байт [12], которые можно адресовать, зная номер шины PCI, номер устройства и номер функции в устройстве. Первые 64 байта из 256 стандартизированы, а остальные регистры могут быть использованы по усмотрению изготовителя устройства.

Так как основная задача это анализ входного потока со стороны IO-Link, то в конфигурационное пространство PCI задействованы только необходимые регистры, остальные по умолчанию равны “0”

Основные регистры, используемые в PCI конфигурационном пространстве АПК ТКПИ, показаны в таблице 3.8.

Таблица 3.8 – PCI-конфигурационное пространство АПК ТКПИ

Адрес регистра	Наименование регистра	Режим доступа
1	2	3
01-00h	Vendor ID	RO
02-03h	Device ID	RO
04-05h	PCI Command Register	R/W, RO
- - -	- - -	- - -
0Eh	Header Type	RO
10h	Base Address Register 0	R/W, RO
14h	Base Address Register 1	R/W, RO
18h	Base Address Register 2	R/W, RO
1Ch	Base Address Register 3	R/W, RO
20h	Base Address Register 4	R/W, RO

PCI Command Register

Так как используется ОЗУ объемом 64 Мб для возможности DMA обменов, то бит разрешения приема запросов в пространство памяти [1] имеет доступ запись/чтение.

Устройство работает в режиме ведомого, поэтому бит “Мастера Шины” [2] имеет доступ только чтение.

Остальные биты данного регистра не используются, имеют доступ только чтение и по умолчанию равны “0”.

Base Address Register 0

Данный регистр определяет адресный диапазон для обращения к ОЗУ АПК ТКПИ.
Описание данного регистра показано в таблице 3.9.

Таблица 3.9 – Нулевой регистр базового адреса

Биты	Режим доступа	Описание
1	2	3
31:26	R/W	Биты сравниваются с битами [31:26] адреса запроса в пространство памяти
3:0	RO	Не используются

Base Address Register 1

Данный регистр определяет адресный диапазон для обращения к FIFO буферу последовательной выдачи. Описание данного регистра показано в таблице 3.10.

Таблица 3.10 – Первый регистр базового адреса

Биты	Режим доступа	Описание
1	2	3
31:4	R/W	Биты сравниваются с битами [31:4] адреса запроса в пространство памяти
3:0	RO	Не используются

Base Address Register 2

Данный регистр определяет адресный диапазон для обращения к к FIFO буферу APIC DMA запросов. Описание данного регистра показано в таблице 3.11.

Таблица 3.11 – Второй регистр базового адреса

Биты	Режим доступа	Описание
1	2	3
31:4	R/W	Биты сравниваются с битами [31:4] адреса запроса в пространство памяти
3:0	RO	Не используются

Base Address Register 3

Данный регистр определяет адресный диапазон для обращения к FIFO буферу ответов на PIO запросы. Описание данного регистра показано в таблице 3.12.

Таблица 3.12 – Третий регистр базового адреса

Биты	Режим доступа	Описание
1	2	3
31:4	R/W	Биты сравниваются с битами [31:4] адреса запроса в пространство памяти
3:0	RO	Не используются

Base Address Register 4

Данный регистр определяет адресный диапазон для обращения к блоку регистров статуса и управления. Описание данного регистра показано в таблице 3.13.

Таблица 3.13 – Четвертый регистр базового адреса

Биты	Режим доступа	Описание
1	2	3
31:7	R/W	Биты сравниваются с битами [31:7] адреса запроса в пространство памяти
6:0	RO	Не используются

3.3 Логический анализатор

Для возможности контроля над передаваемыми данными из АПК ТКПИ во внешнюю sdram память, потребовалось разработать логический анализатор для считывания диаграммы обмена на одном из каналов системного интерфейса контроллера памяти. Он позволяет сохранить до 16384 выборок.

Выход буфера трассы сделан программно доступным, для вывода диаграммы обмена непосредственно в файл для последующего ее анализа пользователем.

Анализатор состоит из следующих блоков:

- «Счетчик выборки» - счетчик операций обмена. Управляющие регистры, определяющие параметры операций, которые учитываются счетчиком:
 1. <analyzer mode control :: cmode_scale> (Адрес 0x80);
 2. <analyzer param0_count> (Адрес 0x84);
 3. <analyzer param1_count> (Адрес 0x88);
- «Блок Активизации» - блок активизации/останова процесса записи трассы в буфер «буфер трассы». Управляющие регистры, определяющие условия активизации/останова процесса записи трассы:

1. <analyzer mode control :: tmode> (Адрес 0x80);
 2. <analyzer mode control :: tmode_scale> (Адрес 0x80);
 3. <analyzer param0_trig> (Адрес 0x8C);
 4. <analyzer param1_trig> (Адрес 0x90);
 5. <analyzer param2_trig> (Адрес 0x94);
- «*Фильтр выборки*» - блок фильтрации потока операций обмена. Управляющие регистры, определяющие параметры, которым должен удовлетворять запрос для занесения его в буфер трассы:
 1. <analyzer mode control::stor_scale> (Адрес 0x80);
 2. <analyzer param0_stor> (Адрес 0x98);
 3. <analyzer param1_stor> (Адрес 0x9C);
 4. <analyzer param2_stor> (Адрес 0xA0);
 5. <analyzer param3_stor> (Адрес 0xA4);
 6. <analyzer param4_stor> (Адрес 0xA8);
 - «*Блок коммутации*» - коммутация входных потоков на вход анализатора. Управляющий регистр, задающий номер канала, на котором осуществляется анализ потока:
 1. <analyzer mode control::channel_num> (Адрес 0x80);
 - «*Буфер трассы*».

На рисунке 3.8 показана функциональная схема логического анализатора.

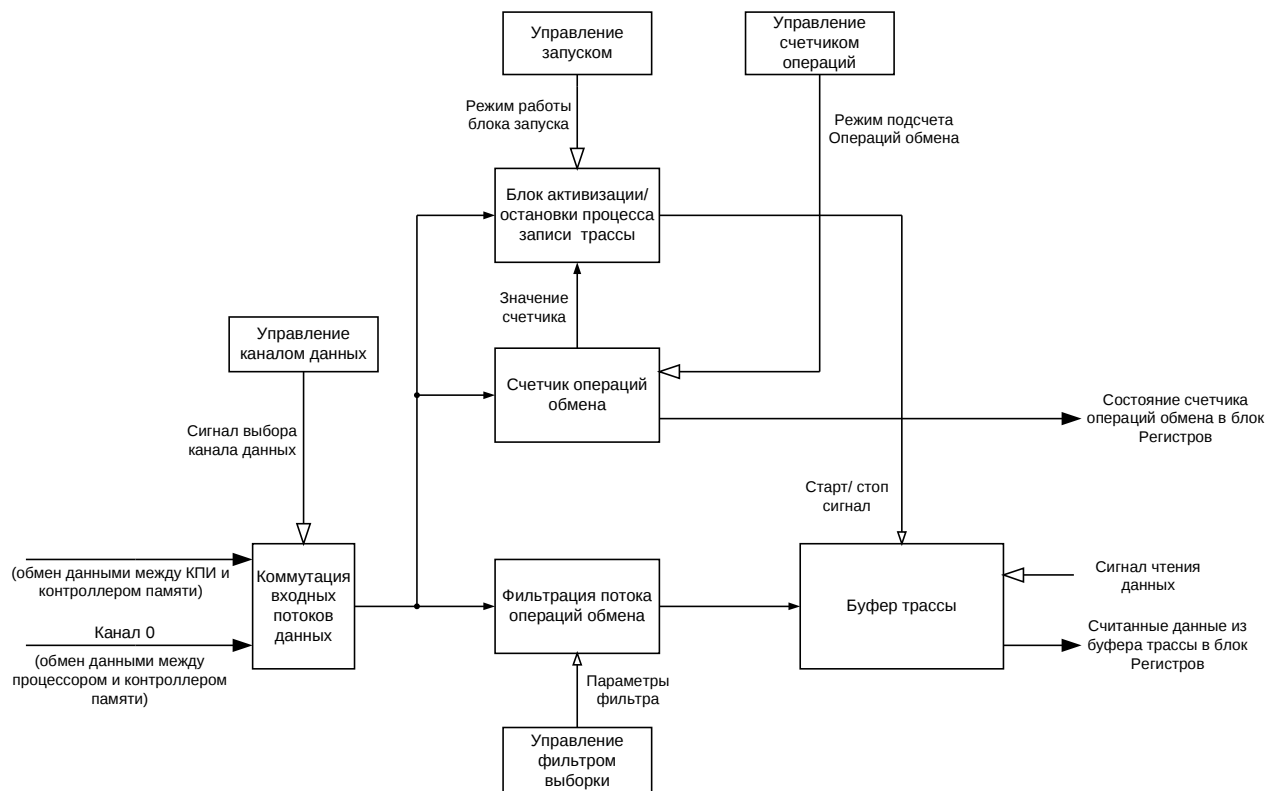


Рисунок 3.8 - Функциональная схема логического анализатора

Поток операций с обоих каналов поступает на вход блока «Коммутации входных потоков данных». Блок «Коммутации входных потоков данных» выбирает в соответствии с управляющим сигналом требуемый канал данных. С выхода данного блока поток данных направляется на входы блоков «Счетчик операций обмена», «Активизация», «Фильтр потока операций обмена».

«Счетчик операций обмена» выполняет подсчет операций, тип и адрес которых удовлетворяет условиям, заданным на управляющем входе блока «Управление счетчиком операций». Значение счетчика подается на вход блока «Активизация».

Данный блок, анализирует параметры очередной операции (тип операции, адрес, значение поля данных), значение счетчика операций обмена и значения на линиях управления, выполняет запуск/останов процесса записи элементов трассы в «Буфер трассы».

«Фильтр потока операций обмена» осуществляет анализ параметров текущей операции (тип, адрес, значение поля данных). При обнаружении соответствия параметров операции условиям, заданным на линиях управления «Управление фильтром выборки», данные пропускаются на вход «Буфер трассы», и в случае активности процесса записи трассы, заносятся в буфер.

Выводы

- 1) В результате проведенной проектно-конструкторской работы выделены основные блоки АПК ТКПИ (“Блок регистров”, “Последовательный буфер”, “Буфер для потерянных пакетов”, “Буфер ответов”, “Буфер для АРІС запросов”), разработана структурная схема АПК ТКПИ, описан принцип работы АПК ТКПИ;
- 2) АПК ТКПИ описано на языке описания аппаратуры Verilog на уровне регистровых передач (RTL);
- 3) Конфигурационное пространство АПК ТКПИ представляет 46 программно доступных регистров, предоставляющих широкие возможности для программирования АПК ТКПИ;
- 4) Для проверки пересылаемых данных в память и из нее реализован логический анализатор с буфером, позволяющим сохранять до 16384 выборок;

4 Технологическая часть

4.1 Выбор ПЛИС для реализации АПК ТКПИ

В настоящее время существует множество фирм производителей ПЛИС — “Xilinx”, “Altera”, “Atmel”, “Actel”, “Lattice semiconductor”.

Так как ЗАО «МЦСТ» имеет большой опыт работы с ПЛИС от компании “Altera”, принято решение реализовать АПК ТКПИ на Stratix II EP2S60F1020C3 [13], которая полностью удовлетворяет требованиям ТЗ.

Данная микросхема имеет следующие характеристики:

- Количество эквивалентных логических элементов (LEs) – 60 640;
- Количество адаптивных таблиц перекодировки (ALUT) – 48 352;
- Объем встроенного ОЗУ (Кбит) – 2 544;
- Кол-во блоков встроенного ОЗУ M512 (512 битов + 64 битов четности) - 329;
- Кол-во блоков встроенного ОЗУ M4K RAM Blocks (4 Кбит + 512 битов четности) – 255;
- Кол-во блоков встроенного ОЗУ M-RAM Blocks (512 Кбит + 65,536 битов четности) – 2;
- Количество PLL 12;
- Поддерживаемые уровни напряжения ввода-вывода (V) 3.3, 2.5, 1.8, 1.5;
- Поддерживаемые стандарты ввода-вывода PCI, LVDS, LVTTL и др;
- Поддерживаемые интерфейсы внешней памяти DDR2, DDR, SDR.

В качестве САПР используем Quartus II версии 9.0 [14] от компании “Altera”. Данный САПР позволяет поддерживать проект на всех этапах проектирования:

- Синтез;
- Размещение и трассировка элементов;
- Временной анализ;
- Программирование и конфигурирование устройства.

4.2 Использование мегафункций САПР Quartus

Для эффективного использования внутренних блоков ПЛИС (память, PLL) применяются мегафункции [15].

Фазовая автоматическая подстройка частоты

Для работы АПК ТКПИ необходимо формировать следующие опорные частоты:

- 33 МГц для модулей, работающих с шиной PCI;
- 133 МГц для внешней SDRAM памяти и контроллера памяти;
- 100 МГц для контроллера IO-Link интерфейса.

Для данной цели воспользуемся блоками фазовой автоматической подстройкой частоты или PLL (Phase-Locked loop), расположенными в ПЛИС. Данный модуль называется “pov_tb__pci_pll”. Для создания PLL используем мастер создания мегафункций – MegaWizard.

Все требуемые частоты получим из входной частоты, поступающей по шине PCI, равной 33 МГц.

В начале конфигурируем входную частоту для данного модуля: задаем период равный 30 нс, т.е. частота будет равна 33 МГц (рисунок 4.1).

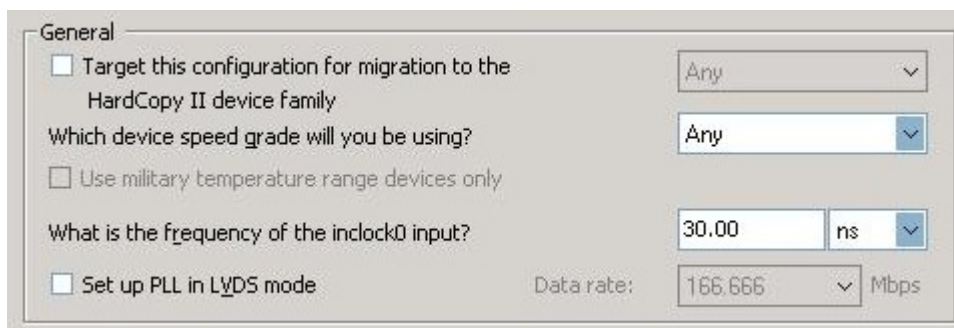


Рисунок 4.1 – Конфигурирование входной частоты для модуля “pov_tb__pci_pll”

Теперь конфигурируем требуемые выходные частоты (рисунок 4.2 – 4.3).

The screenshot shows the 'Clock Tap Settings' dialog box. The 'Requested settings' column shows a frequency of 100.0000000 MHz. The 'Actual settings' column shows a frequency of 33.333333 MHz. The 'Enter output clock parameters' option is selected. The 'Clock multiplication factor' is set to 1, and the 'Clock division factor' is set to 1. The 'Clock phase shift' is set to 0.00 deg. The 'Clock duty cycle (%)' is set to 50.00.

Parameter	Requested settings	Actual settings
Enter output clock frequency:	100.0000000 MHz	33.333333
Enter output clock parameters:		
Clock multiplication factor	1	1
Clock division factor	1	1
Clock phase shift	0.00 deg	0.00
Clock duty cycle (%)	50.00	50.00

Рисунок 4.2 – Конфигурирование частоты 33 МГц

The screenshot shows the 'Clock Tap Settings' dialog box. The 'Requested settings' column shows a frequency of 100.0000000 MHz. The 'Actual settings' column shows a frequency of 100.000000 MHz. The 'Enter output clock parameters' option is selected. The 'Clock multiplication factor' is set to 3, and the 'Clock division factor' is set to 1. The 'Clock phase shift' is set to 0.00 deg. The 'Clock duty cycle (%)' is set to 50.00.

Parameter	Requested settings	Actual settings
Enter output clock frequency:	100.0000000 MHz	100.000000
Enter output clock parameters:		
Clock multiplication factor	3	3
Clock division factor	1	1
Clock phase shift	0.00 deg	0.00
Clock duty cycle (%)	50.00	50.00

Рисунок 4.3 – Конфигурирование частоты 100 МГц

В результате получаем сгенерированный verilog файл “*pov_tb_pci_pll.v*” и необходимые файлы для дальнейшей работы САПР с данным модулем (рисунок 4.4).

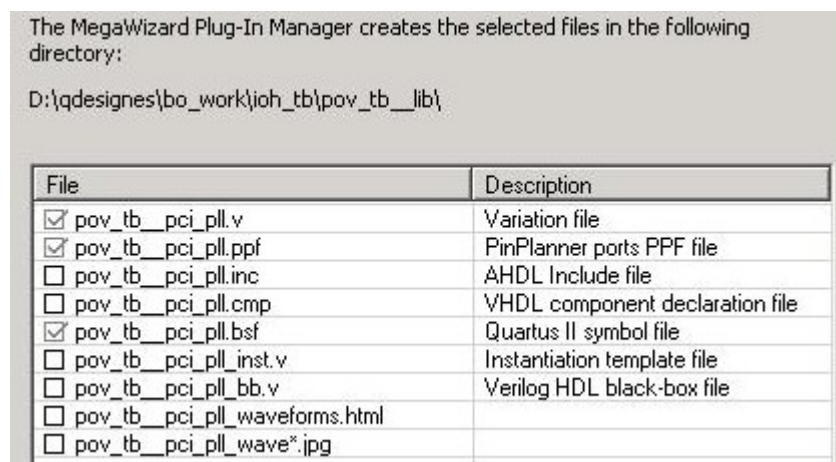


Рисунок 4.4 – Созданные файлы для “pov_tb__pci_pll”

В макете для отладки КПИ, с помощью АПК ТКПИ формируется опорная частота 100 МГц для SATA буфера из входной частоты равной 25 МГц.

Для этого создадим модуль “pov_tb__sata_pll”.

В начале конфигурируем входную частоту для данного модуля: задаем период равный 40 нс, т.е. частота будет равна 25 МГц (рисунок 4.5).

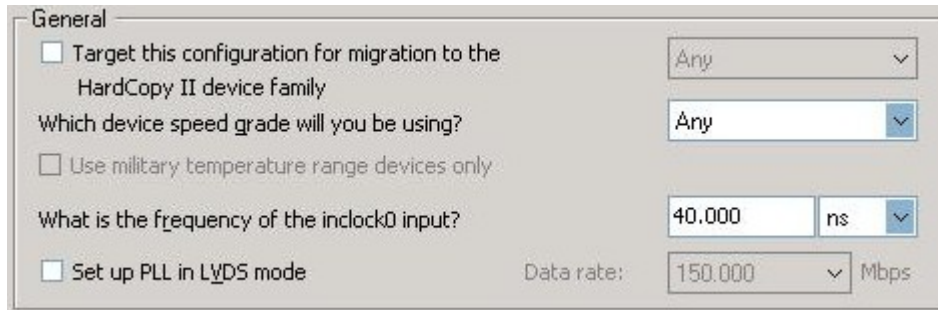


Рисунок 4.5 – Конфигурирование входной частоты для модуля “pov_tb__sata_pll”

Для выходной частоты поставим коэффициент умножения на 4, таким образом получим выходную частоту равную 100 МГц.

Clock Tap Settings

☐ Enter output clock frequency: 100.000000 MHz 100.000000

☒ Enter output clock parameters:

Clock multiplication factor: 4 4

Clock division factor: 1 1 << Copy

Clock phase shift: 0.00 deg 0.00

Рисунок 4.6 – Конфигурирование выходной частоты для модуля “ pov_tb__sata_pll”

Мегафункции для RAM блоков

В АПК ТКПИ используется большое количество буферов памяти (последовательный буфер, буфер ответов, буфер для логического анализатора) в различных модулях.

Таблица 4.1 - Использование модулей с буферами памяти

Название модуля	Размер	Количество
1	2	3
pov_tb__afifo_pxp	W33D512	2
	W33D128	1
	W1D32	2
	W5D16	1
	W33D16	1
pov_tb__fifo_pxp	W16D16	1
	W33D64	1
pov_tb__dp_ram	W32D128	4
	W16D16	1
	W4D128	2
pov_tb__sfifo_pxp	W96D16384	1
	W72D8	1
	W20D8	1
	W33D8	1

Для эффективного использования ресурса ПЛИС, необходимо, чтобы данные блоки памяти, при синтезе, определялись как блоки памяти, размещенные в ПЛИС.

Данная ПЛИС имеет следующие блоки памяти: M512 RAM, M4K RAM и M-RAM.

Блок памяти M512 RAM

Используется в качестве небольших FIFO буферов передачи доменов тактового генератора. Каждый блок содержит 576 битов. Общее количество блоков 329. Может использоваться в качестве простой двухпортовой RAM памяти, однопортовой RAM памяти, FIFO буфера, ПЗУ, сдвигового регистра.

Блок памяти M4K RAM

Используется в качестве буферов для различных приложений, таких как хранение процессорного кода, большого количества данных для приложений. Каждый блок содержит 4,608 бит. Общее количество блоков 255. Может использоваться в качестве истинной двухпортовой RAM памяти, простой двухпортовой RAM памяти, однопортовой RAM памяти, FIFO буфера, ПЗУ, сдвигового регистра.

Блок памяти M-RAM

Используется для различных приложений, для которых большое количество данных должно быть сохранено в микросхеме. Каждый блок содержит 589 824 бит. Общее количество блоков 2. Может использоваться в качестве истинной двухпортовой RAM памяти, простой двухпортовой RAM памяти, однопортовой RAM памяти, FIFO буфера.

Таким образом RTL код данных модулей при синтезе преобразуется в следующие блоки памяти:

- “pov_tb__fifo_pxp” – FIFO буфер;
- “pov_tb__dp_ram” – истинная двухпортовая RAM память;
- “pov_tb__afifo_pxp” – FIFO буфер;
- “pov_tb__sfifo_pxp” – FIFO буфер.

Структурная схема истинной двухпортовой RAM памяти в ПЛИС семейства Stratix 2 показана на рисунке 4.8.

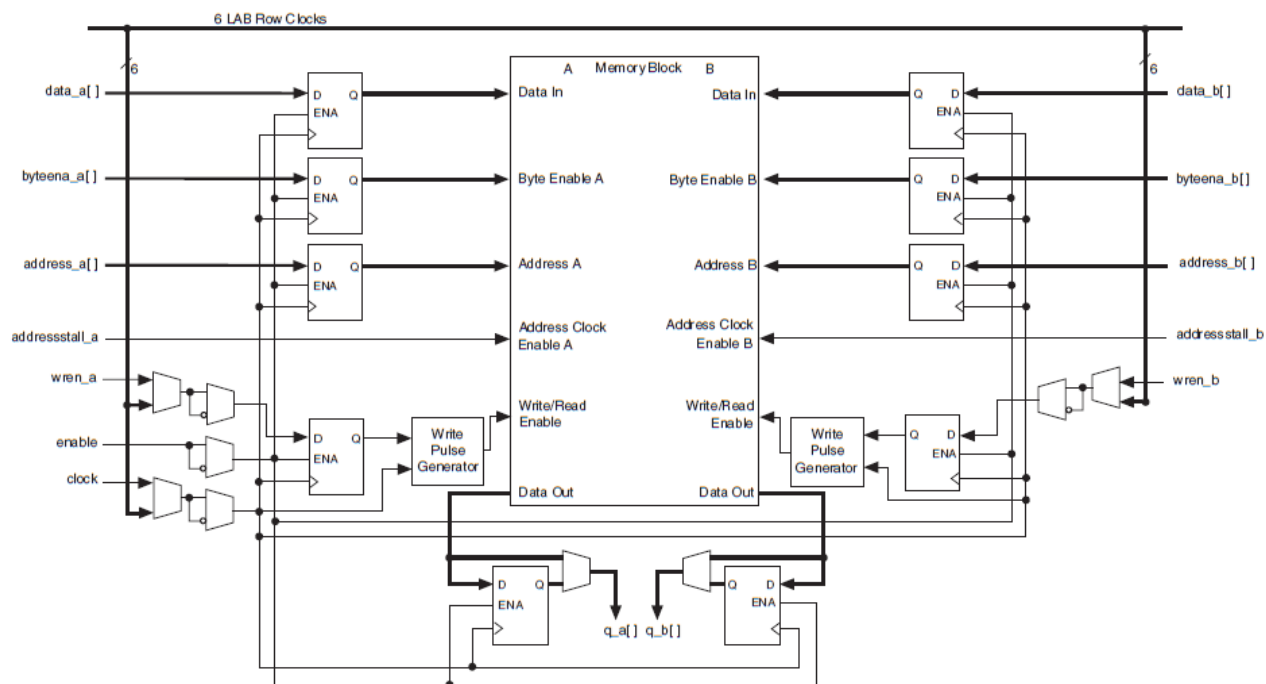


Рисунок 4.7 – Структурная схема истинной двухпортовой RAM памяти в ПЛИС семейства Stratix II

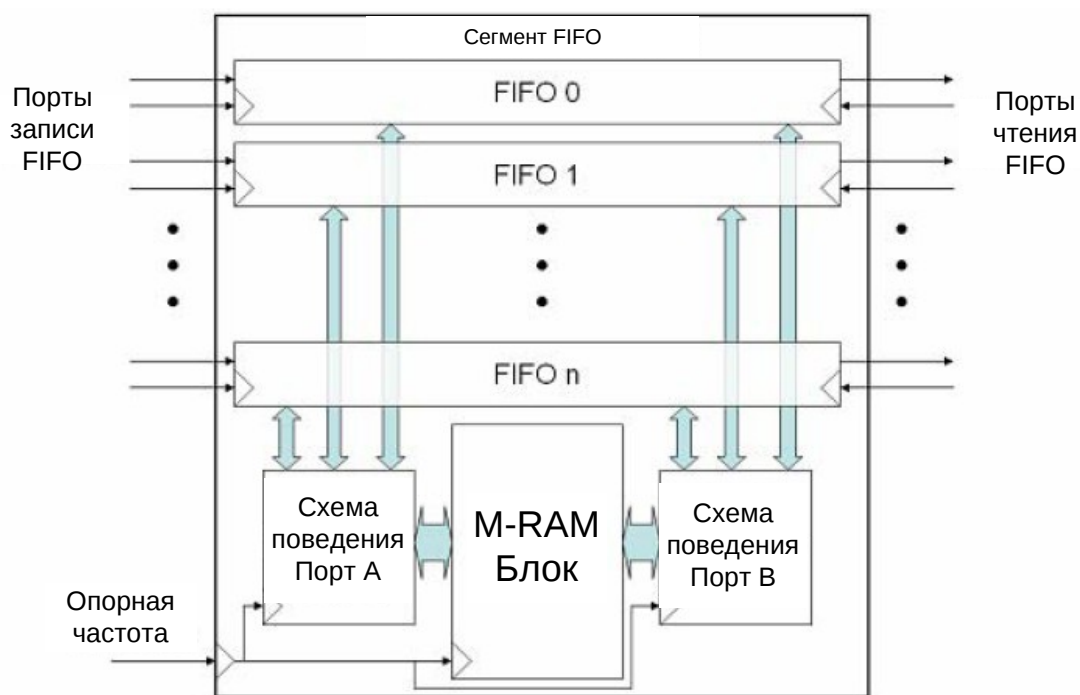


Рисунок 4.8 - Использование M-RAM блока памяти в качестве FIFO буфера

Так же для этого необходимо придерживаться определенного стиля описания RTL согласно [17].

Ниже показан пример такого описания для модуля “pov_tb_dp_ram_pxp”:

```

module pov_tb__dp_ram_pxp(

wclk,
rclk,

wr_val,
wr_address,
wr_data,

rd_address,
rd_data
);

parameter depth = 8;
parameter awidth = 3;
parameter width = 8;
//-----//
input          wclk;    // тактовый сигнал чтения
input          rclk;    // тактовый сигнал записи

input          wr_val;  // флаг записи
input[awidth-1:0] wr_address; // адрес записи
input[width-1:0] wr_data; // записываемые данные

input[awidth-1:0] rd_address; // адрес чтения
output[width-1:0] rd_data;    // считанные данные
//-----//
//-----//
reg [width-1:0] ram_cell [depth-1:0]; // ячейки памяти
reg [awidth-1:0] rd_address_r; // указатель адреса
//-----//
//          mem array
//-----//
always @(posedge wclk)
if (wr_val)          ram_cell[wr_address] <= wr_data; // запись данных

always @(posedge rclk)
          rd_address_r <= rd_address; // формирование адреса чтения данных
//-----//
assign rd_data =          ram_cell[rd_address_r]; // чтение данных
//-----//
//-----//
endmodule

```

Также необходимо включить соответствующие опции в настройках синтеза (рисунок 4.9).

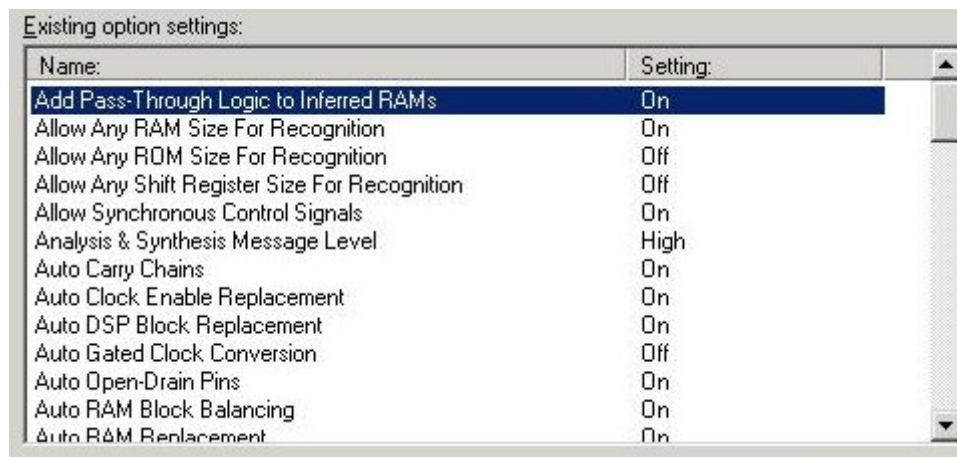


Рисунок 4.9 – Настройки синтеза

При этом в конфигурационном файле “pov_tb.qsf” добавятся соответствующие опции:

set_global_assignment -name AUTO_ROM_RECOGNITION OFF // *выключить распознавание для ROM блоков.*

set_global_assignment -name AUTO_RAM_RECOGNITION ON // *включить распознавания для RAM блоков.*

Также включим опцию при которой небольшие блоки RAM преобразуются в логические ячейки:

Auto RAM to Logic Cell Conversion on // *разрешить преобразование ячеек памяти RAM в логические ячейки.*

4.3 Сборка логической модели АПК ТКПИ

Синтез АПК ТКПИ

В результате синтеза исходное RTL описание проекта преобразуется в логические элементы ПЛИС согласно заданным параметрам синтеза.

Для улучшения синтеза включим следующие опции:

- Remove Redundant Logic Cells on – удаление излишних логических ячеек;
- Optimization Technique speed – оптимизация по скорости, для получения лучших временных характеристик;
- Auto Open-Drain Pins – включим использование выводов с открытым

стоком, это необходимо для PCI интерфейса и др.

В результате после анализа и синтеза проекта получаем следующие результаты (таблица 4.2).

Analysis & Synthesis Status	Successful - Mon Feb 15 14:38:42 2010
Quartus II Version	9.0 Build 235 06/17/2009 SP 2 SJ Full Version
Revision Name	pov_tb
Top-level Entity Name	pov_tb
Family	Stratix II
Logic utilization	N/A
Combinational ALUTs	6,643
Dedicated logic registers	9,675
Total registers	9675
Total pins	166
Total virtual pins	0
Total block memory bits	1,863,888
DSP block 9-bit elements	0
Total PLLs	4
Total DLLs	0

Рисунок 4.10 – Результаты синтеза

Таблица 4.2 – Результат синтеза АПК ТКПИ

Характеристики	Значение
1	2
Количество адаптивных таблиц перекодировки (ALUT)	6 647
Суммарное количество регистров	9 553
Суммарное количество PLL	4
Суммарное число битов памяти	1 863 888
Количество входных сигналов	24
Количество выходных сигналов	51
Количество двунаправленных сигналов	96
Суммарное число сигналов	166
Количество эквивалентных логических элементов	13 774
Задействованное число сегментов RAM памяти	1 345

Размещение и трассировка АПК ТКПИ

Так как IO-LINK интерфейс использует низковольтную дифференциальную передачу сигнала (low-voltage differential signaling или LVDS), то в конфигурационном файле (pov_tb.qsf) необходимо указать тип ввода/вывода для данного интерфейса - LVDS.

```
set_instance_assignment -name IO_STANDARD LVDS -to lv_tstr
set_instance_assignment -name IO_STANDARD LVDS -to lv_rdat[7]
set_instance_assignment -name IO_STANDARD LVDS -to lv_rdat[6]
set_instance_assignment -name IO_STANDARD LVDS -to lv_rdat[5]
set_instance_assignment -name IO_STANDARD LVDS -to lv_rdat[4]
set_instance_assignment -name IO_STANDARD LVDS -to lv_rdat[3]
set_instance_assignment -name IO_STANDARD LVDS -to lv_rdat[2]
set_instance_assignment -name IO_STANDARD LVDS -to lv_rdat[1]
set_instance_assignment -name IO_STANDARD LVDS -to lv_rdat[0]
set_instance_assignment -name IO_STANDARD LVDS -to lv_rfrm
set_instance_assignment -name IO_STANDARD LVDS -to lv_rstr
set_instance_assignment -name IO_STANDARD LVDS -to lv_tdat[7]
set_instance_assignment -name IO_STANDARD LVDS -to lv_tdat[6]
set_instance_assignment -name IO_STANDARD LVDS -to lv_tdat[5]
set_instance_assignment -name IO_STANDARD LVDS -to lv_tdat[4]
set_instance_assignment -name IO_STANDARD LVDS -to lv_tdat[3]
set_instance_assignment -name IO_STANDARD LVDS -to lv_tdat[2]
set_instance_assignment -name IO_STANDARD LVDS -to lv_tdat[1]
set_instance_assignment -name IO_STANDARD LVDS -to lv_tdat[0]
set_instance_assignment -name IO_STANDARD LVDS -to lv_tfrm
```

Для остальных задействованных выводов зададим тип 3.3 В.

```
set_global_assignment -name STRATIX_DEVICE_IO_STANDARD "3.3-V LVTTTL"
```

Неиспользуемые выходы переведем в третье состояние:

```
set_global_assignment -name RESERVE_ALL_UNUSED_PINS "AS INPUT TRI-
STATED"
```

В результате после размещения и трассировки элементов получаем следующие результаты (таблица 4.3).

Fitter Status	Successful - Mon Feb 15 14:51:01 2010
Quartus II Version	9.0 Build 235 06/17/2009 SP 2 SJ Full Version
Revision Name	pov_tb
Top-level Entity Name	pov_tb
Family	Stratix II
Device	EP2S60F1020C3
Timing Models	Final
Logic utilization	29 %
Combinational ALUTs	7,588 / 48,352 (16 %)
Dedicated logic registers	9,614 / 48,352 (20 %)
Total registers	9678
Total pins	186 / 719 (26 %)
Total virtual pins	0
Total block memory bits	1,863,888 / 2,544,192 (73 %)
DSP block 9-bit elements	0 / 288 (0 %)
Total PLLs	4 / 12 (33 %)
Total DLLs	0 / 2 (0 %)

Рисунок 4.11 – Результаты размещения и трассировки элементов

Combinational ALUT usage by number of inputs	
-- 7 input functions	234
-- 6 input functions	1984
-- 5 input functions	1508
-- 4 input functions	1125
-- <=3 input functions	2737

Рисунок 4.12 – Использование ресурсов ALUT

Таблица 4.3– Результат размещения и трассировки элементов АПК ТКПИ

Характеристики	Значение
1	2
Использование логики ПЛИС	13 562 / 48 352 (29 %)
Комбинационные ALUT и регистры, используемые в окончательном размещении	12892
Комбинационная логика без регистров	3397
Только регистры	5532
Комбинационная логика с регистрами	3963
Комбинационные ALUT и регистры без пар	1484
Использование адаптивных логических ячеек (ALUT)	7 360 / 48 352 (16 %)
Комбинационные ALUT, использующие 7 входов для логической функции	234
Комбинационные ALUT, использующие 6 входов для логической функции	1964
Комбинационные ALUT, использующие 5 входов для логической функции	1618

Таблица 4.3 - Продолжение

1	2
Комбинационные ALUT, использующие 4 входов для логической функции	1105
Комбинационные ALUT, использующие 3 входа и меньше для логической функции	2438
Использование логических регистров	9 495 / 48 352 (20 %)
Регистры ввода/вывода	64 / 4 154 (2 %)
Использование PLL	4 / 12 (33 %)
Суммарное число битов памяти	1 782 992 / 2 544 192 (70 %)
Суммарное число сигналов	186 / 719 (26 %)

Как видно из данной таблицы, благодаря использованию мегафункций для памяти RAM общая используемая площадь ПЛИС равна 29%, что дает возможность эффективно использовать возможности логического анализатора, встроенного в ПЛИС, и обеспечивать гибкость в случае добавлений и изменений в АПК ТКПИ.

4.4 Конфигурирование ПЛИС

В результате компиляции проекта получаем два файла “pov_tb.sof” и “pov_tb.pof”. Первый файл применяется для временной конфигурации ПЛИС, конфигурация сбрасывается после подачи сигнала сброса на ПЛИС, а второй конфигурационный файл для постоянной конфигурации ПЛИС.

В Stratix II используют ячейки SRAM памяти для хранения конфигурационных данных. Так как SRAM память энергозависимая, то конфигурационные данные должны быть загружены в ПЛИС Stratix II каждый раз при включении питания. Stratix II могут быть сконфигурированы, используя одну из пяти схем: быстрая пассивная параллельная (FPP), активная последовательная (AS), пассивная параллельная (PS), пассивная параллельная асинхронная (PPA) и используя интерфейс JTAG. Все данные схемы используют, либо внешний контроллер (микропроцессор), либо конфигурационное устройство.

Выберем активную последовательную схему, так как во всех пассивных схемах для конфигурации необходимо использование хост устройства или микропроцессора, который управляет процессом конфигурирования.

Для временной конфигурации ПЛИС в режиме отладки будем использовать JTAG интерфейс.

Активная последовательная схема конфигурации

В данной схеме конфигурации происходит с помощью последовательного конфигурационного устройства. Эти устройства используют энергонезависимую память и для конфигурации ПЛИС требуется четыре ввода/вывода. В качестве него используем статическое ОЗУ размером 16 Мбайт EPCS64SI16N.

Блок диаграмма EPCS64SI16N показана на рисунке 4.13.

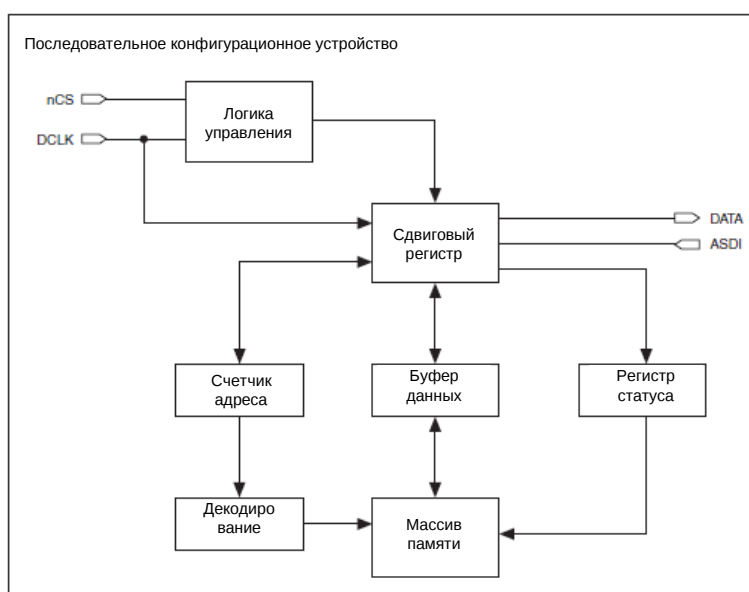


Рисунок 4.13- Блок диаграмма EPCS64SI16N

Данное устройство имеет четыре вывода:

- DCLK – вход опорной частоты;
- DATA – выход последовательных данных;
- ASDI – вход данных для конфигурирования;
- nCS – вход выбора микросхемы;

Активная схема конфигурирования показана на рисунке 4.14.

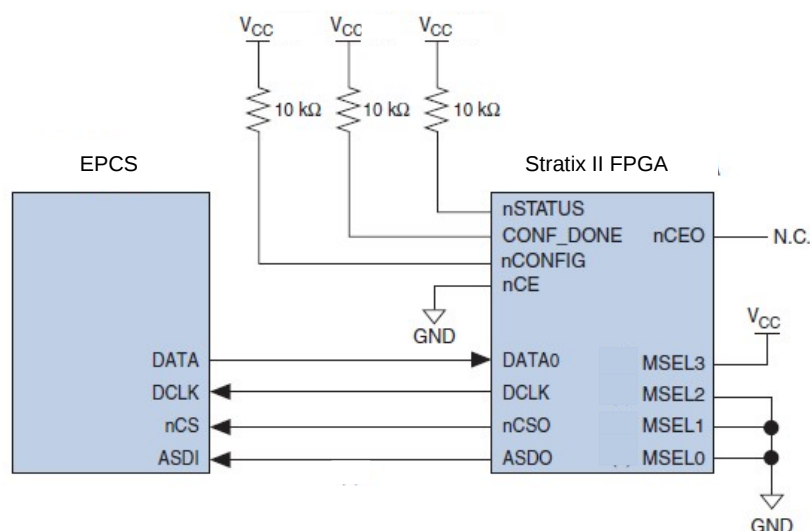


Рисунок 4.14 - Активная схема конфигурирования

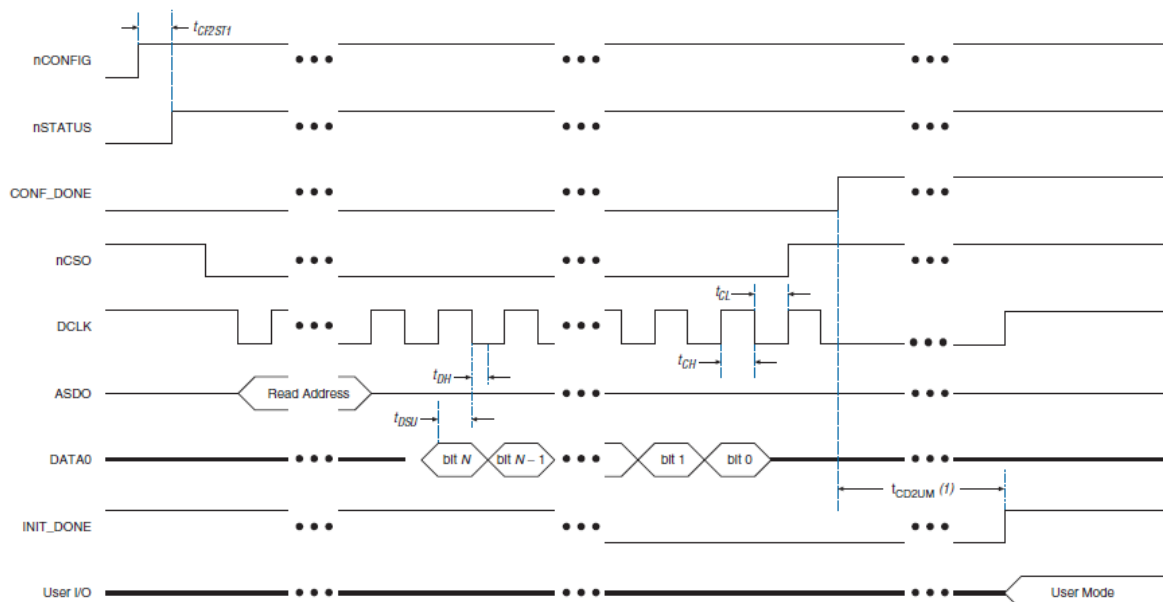
Конфигурационный цикл состоит из трех стадий: сброс, конфигурация и инициализация. Пока *nCONFIG* или *nSTATUS* находятся в низком состоянии, устройство в состоянии сброса. После сброса *nSTATUS* переключается в высокое состояние с помощью подтягивающего резистора номиналом 10 кОм и устройство переходит в режим конфигурирования.

Опорная частота *DCLK0* генерируется самой ПЛИС для всего конфигурационного цикла, задавая необходимые временные соотношения для последовательного интерфейса. Используя выводы MSEL, можно задать опорную частоту от 20 МГц до 40 МГц.

В конфигурационном режиме Stratix II активирует последовательное конфигурационное устройство, переводя *nCS0* в низкое состояние. Stratix II использует опорную частоту *DCLK0* и выход последовательных данных ASDO для отправки операционных команд. Конфигурационное устройство обеспечивает передачу данных от вывода *DATA* к выводу *DATA0*.

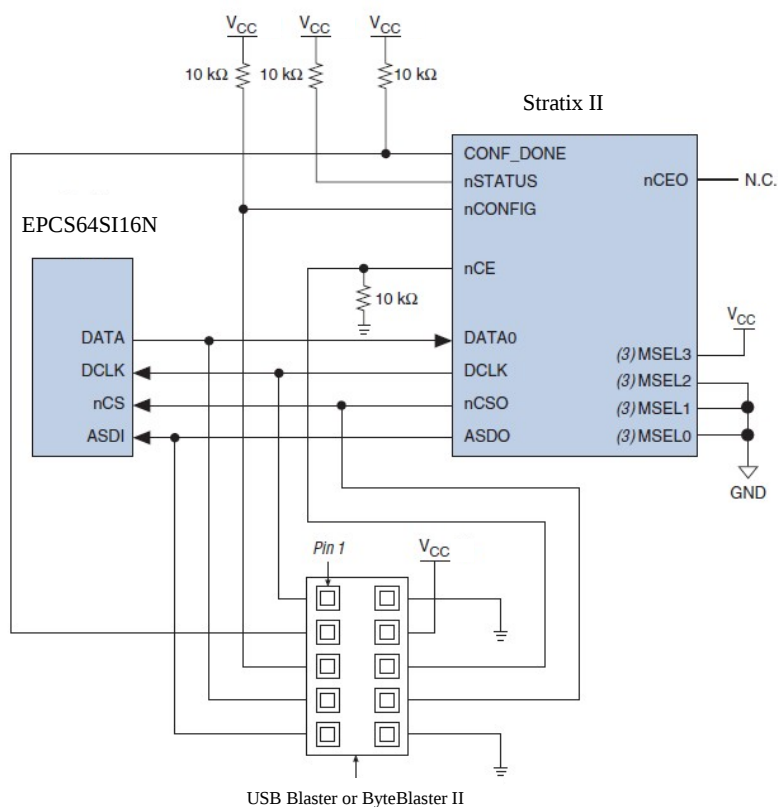
После завершения конфигурирования ПЛИС *CONF_DONE* переключается в высокое состояние с помощью подтягивающего резистора номиналом 10 кОм и устройство переходит в режим инициализации.

Инициализация происходит с помощью внутренней опорной частоты равной 10 МГц. Инициализация длится в течение 600 нс и устройству требуется 299 частотных циклов для инициализации и перехода в пользовательский режим.



Для программирования EPCS64SI16N используется загрузочный кабель USB-Blaster или ByteBlaster II.

Окончательная схема конфигурирования ПЛИС Stratix II показана на рисунке 4.16.



4.5 Тестирование АПК ТКПИ

Тестирование АПК ТКПИ можно разделить на два уровня - программный и аппаратный:

- На программном уровне используется программное тестовое окружение, с помощью которого выполняется RTL моделирование АПК ТКПИ;
- На аппаратном уровне проводится тестирование АПК ТКПИ в составе всего макета АПК ТКПИ.

Так как данное устройство относится к устройствам ввода/вывода, то основная задача тестирования проверка правильности выполнения операций ввода/вывода.

Отладка с помощью тестового окружения

С помощью тестового окружения выполняется минимальный набор тестов для проверки функционирования АПК ТКПИ, так как основная проверка будет проводиться на стенде КПИ.

Проверяется доступ в конфигурационное пространство шины PCI:

1. *Проверка инициализации регистров* - после снятия сигнала “сброса” проводится чтение конфигурационных регистров PCI и проводится сравнение полученных значений со значениями, указанными в документации на АПК ТКПИ.
2. *Проверка доступа к регистрам* (для регистров, у которых доступ “только запись” и “запись и чтение”) - осуществляется запись значения логической “1” (логического “0”) в данные регистры, затем проводится чтение их значений и производится сравнение полученных значений с ожидаемым результатом.

Аналогичные тесты применяются и ко всему конфигурационному пространству АПК ТКПИ.

Более серьезная проверка АПК ТКПИ выполняется с помощью следующего теста (рисунок 4.17)

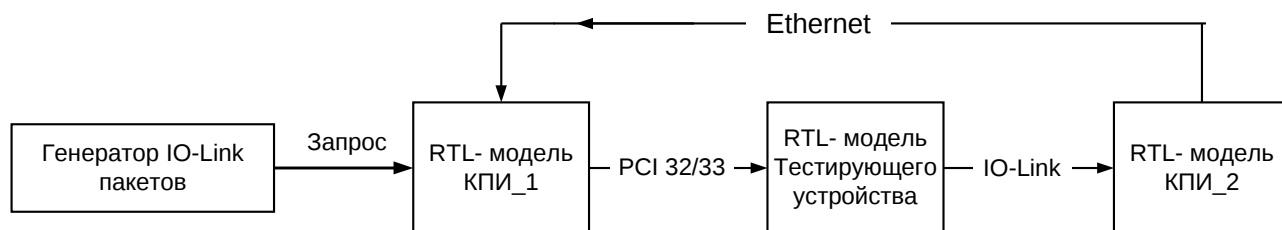


Рисунок 4.17 – Пример схемы тестирования АПК ТКПИ с помощью тестового окружения

К АПК ТКПИ с двух сторон присоединяется RTL модель КПИ. Две данные модели соединены через интерфейс Ethernet. Целью данного теста является формирование запроса от “КПИ 1” к “КПИ 2”, который активизирует Ethernet контроллер “КПИ 2” и тот передает, требуемые данные к “КПИ 1”.

С помощью генератора IO-Link пакетов формируется запрос. Запрос посредством “КПИ_1” преобразуется в формат шины PCI и поступает в АПК ТКПИ. В АПК ТКПИ происходит формирование выходных IO-Link пакетов и запись данных в ОЗУ АПК ТКПИ со стороны PCI. Затем данные пакеты поступают в “КПИ 2”, который обрабатывает их и выполняет действия согласно полученному запросу.

Таким образом, помимо проверки АПК ТКПИ проверяется правильность работы КПИ и Ethernet контроллера.

Аппаратная отладка

Аппаратная отладка АКП ТКПИ проводится на макете КПИ. Для ускорения времени отладки АКП ТКПИ было принято решение сразу отлаживать КПИ. Таким образом, в процессе отладки КПИ будет параллельно тестироваться АКП ТКПИ на правильность работы.

Одним из таких тестов является проверка контроллера шины PCIexpress в КПИ (рисунок 4.18)

Тест заключается в пересылки 64 Мб из памяти во флеш-память, расположенную на плате, которая подключается к макету КПИ через PCIexpress разъем, считывание полученных данных из флеш-памяти и их сравнение. В случае успешного выполнения исходный адрес данных смещается на 1 Мб и так далее, пока смещение не достигнет 64 Мб.

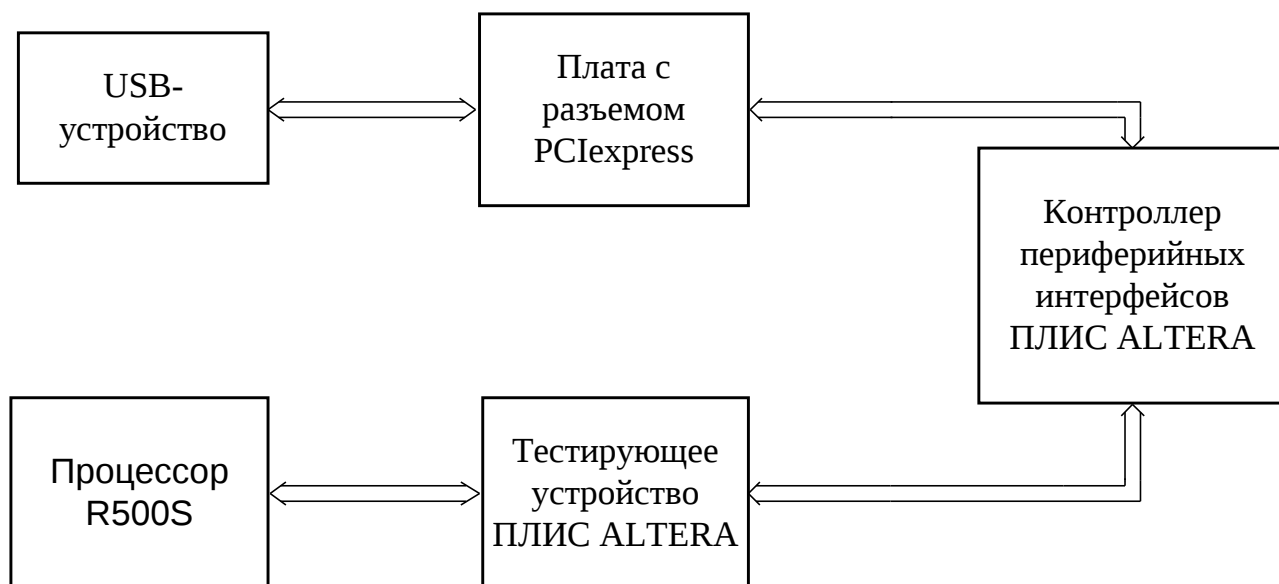


Рисунок 4.18 – Схема проверки контроллера PCIexpress

В начале выполнения теста проводится сканирование шины USB и определяются параметры флеш-карты:

```

EHCI found:      BUS[81] SLOT[0] FUNC[2]
* EHCI number:   0
* EHCI revision: 1.0

* ##### DEVICE DESCRIPTOR #####
* Descriptor Size:      18
* Descriptor Type:      1
* USB Specification:    2.0
* Device Class:         0x0
* Device Subclass:      0x0
* Device Protocol:      0x0
* Maximum Packet Size for Endpoint 0: 64
  
```

Затем выводится сообщение о найденной устройстве с шиной USB:

```

*****
*           USB MASS STORAGE DEVICES
*
*
* Mass Storage Device with Bulk-Only Transport Protocol found
* Location:   EHCI0 Port2
*           Configuration1 Interface0 Alternate Setting0
* Command Block: SCSI transparent command set
* Maximum Logical Unit Number: 0
  
```

После выполнения каждого теста выводится сообщение об успешном его завершении и начинается следующий запрос с новым смещением:

```
* TEST OK
* SHIFT: 1 bytesENVLIB: idle at 5286806 us <scope=test[0].thread, time=5286816

* TEST OK
* SHIFT: 43 bytesENVLIB: idle at 79567922 us <scope=test[0].thread, time=79567932 us>

* TEST OK
* SHIFT: 63 bytesENVLIB: idle at 115003457 us <scope=test[0].thread, time=115003467 us>
```

После выполнения всех обменов выводится успешный результат всего теста:

```
*
* TEST OK
ENVLIB: PASSED <scope=test[0].thread, time=116373621 us>
```

Выводы

- 1) АПК ТКПИ будет реализовано на ПЛИС Stratix II EP2S60F1020C3 фирмы Altera, для синтеза логической модели выбран САПР Quartus версии 9.0;
- 2) Основным параметром оптимизации синтеза является скорость, для получения работы АПК ТКПИ на требуемых частотах;
- 3) Благодаря использованию мегафункций для памяти RAM, общая используемая площадь ПЛИС равна 29%, что позволяет эффективно использовать возможности встроенного в ПЛИС логического анализатора и обеспечить гибкость при внесении изменений в АПК ТКПИ;
- 4) Для конфигурирования ПЛИС используется активная последовательная схема (ASP), настройки ПЛИС хранятся в статическом ОЗУ размером 16 Мбайт EPCS64SI16N с энергонезависимой памяти;
- 5) Отладка АПК ТКПИ проводилась на макете КПИ. В процессе отладки КПИ параллельно тестировался АПК ТКПИ.

5 Охрана труда и экология

5.1 Анализ опасных и вредных факторов на этапе проектирования логической модели АПК ТКПИ

В настоящее время проектирование происходит при помощи ПЭВМ.

На данном этапе на инженера могут оказывать влияние следующие опасные и вредные факторы:

- Микроклимат;
- Уровень шума (акустического);
- Освещение рабочего места (искусственное, естественное);
- Уровень электромагнитных полей на рабочем месте;
- Визуальные параметры ВДТ, контролируемые на рабочих местах.

Так же на производительность труда влияют следующие эргономические факторы: организация и оборудование рабочих мест с ПЭВМ.

Микроклимат

Микроклимат характеризуют следующие параметры:

- Температура воздуха, $^{\circ}\text{C}$;
- Относительная влажность воздуха, %;
- Скорость движения воздуха на рабочем месте, м/с;

Оптимальные параметры микроклимата для работы в теплом помещении в соответствии с СанПин 2.2.2/2.4.1340-03 приложение 2 показаны в таблице 5.1.

Таблица 5.1 – Оптимальные параметры микроклимата.

Температура, $^{\circ}\text{C}$	Относительная влажность, %	Абсолютная влажность, г/м ³	Скорость движения воздуха, м/с
1	2	3	4
19 - 21	62 - 55	10	<0,1

Уровень шума

При выполнении работы на ПЭВМ уровень шума согласно СанПин 2.2.2/2.4.1340-03 не должен превышать 50 дБА.

Освещение рабочего места

Естественное освещение должно осуществляться через световые проемы. Окна в помещениях, по возможности должны быть ориентированы на север и северо-восток.

Искусственное освещение в помещениях должно осуществляться системой общего равномерного освещения.

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300 – 500 лк. Освещение не должно создавать бликов на поверхности экрана.

В качестве источников света при искусственном освещении следует применять преимущественно люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ).

Для освещения помещений с ПЭВМ следует применять светильники с зеркальными параболическими решетками, укомплектованными электронными пускорегулирующими аппаратами (ЭПРА). Допускается использование многоламповых светильников с электромагнитными пускорегулирующими аппаратами (ЭПРА), состоящими из равного числа опережающих и отстающих ветвей.

Коэффициент пульсации не должен превышать 5%.

Уровень электромагнитных полей на рабочем месте

Электромагнитные поля характеризуются следующими параметрами: напряженность электрического поля, плотность магнитного потока, электростатический потенциал экрана монитора. Не следует размещать рабочие места вблизи силовых кабелей, вводов, высоковольтных трансформаторов.

Временные допустимые уровни электромагнитных полей (ЭМП) согласно СанПин 2.2.2/2.4.1340-03, создаваемых ПЭВМ, не должны превышать значений, представленных в таблице 5.2.

Таблица 5.2 - Временные уровни ЭМП, создаваемых ПЭВМ

Наименование параметров		ВДУ ЭМП
1		2
Напряженность электрического поля	в диапазоне частот 5 Гц - 2 кГц	25 В/м
	в диапазоне частот 2 кГц - 400 кГц	2,5 В/м
Плотность магнитного потока	в диапазоне частот 5 Гц - 2 кГц	250 нТл
	в диапазоне частот 2 кГц - 400 кГц	25 нТл
Электростатический потенциал экрана видеомонитора		500 В

Визуальные параметры ВДТ

К данным параметрам относятся: яркость белого поля, неравномерность яркости рабочего поля, контрастность, временная нестабильность изображения, пространственная нестабильность изображения.

Организация и оборудование рабочих мест с ПЭВМ

При размещении рабочего места следует учитывать положение рабочего стола, конструкцию рабочего кресла, расположение монитора, изолирование рабочих мест друг от друга:

- Расстояние между монитором и глазами пользователя составляет 700 мм;
- Рабочие места изолированы перегородками высотой 1,5 – 2,0 м.

5.1 Расчет системы естественного освещения на этапе проектирования

Расчет будем производить согласно требованиям СНиП 23-05-95.

Теоретическая часть

Естественное освещение подразделяется на боковое, верхнее и комбинированное.

Боковое естественное освещение – это освещение помещения через световые проемы в наружных стенах.

Верхнее естественное освещение – это освещение помещения через фонари, световые проемы в стенах в местах перепада высот здания.

Комбинированное естественное освещение – это сочетание верхнего и бокового естественного освещения.

Основным показателем для естественного освещения является значения коэффициента естественной освещенности (КЕО), %.

Коэффициент естественной освещенности (КЕО) — отношение естественной освещенности, создаваемой в некоторой точке заданной плоскости внутри помещения светом неба (непосредственным или после отражений), к одновременному значению наружной горизонтальной освещенности, создаваемой светом полностью открытого небосвода; выражается в процентах.

Нормируемые значения освещенности приводятся в точках ее минимального значения на рабочей поверхности внутри помещений.

Нормированные значения КЕО, e_N , для зданий, располагаемых в различных районах следует определять по формуле:

$$e_N = e_n m_N \quad (1)$$

где N — номер группы обеспеченности естественным светом;

e_n — значение КЕО по справочным таблицам;

m_N — коэффициент светового климата.

Система естественного освещения должна обеспечивать:

нормированные значения коэффициента естественной освещенности (КЕО) на рабочих местах или в расчетной точке помещения;

регламентируемые требования к равномерности распределения КЕО в рабочих зонах помещения;

нормированное значение коэффициента запаса;

максимальное время использования естественного света.

Для проверки расположения световых проемов в помещении и соблюдения требований норм естественного освещения помещений определяют предварительным и проверочным расчетами.

При предварительном расчете определяют площадь и размеры световых проемов.

Расчетное значение КЕО e_p определяется следующим образом:

а) при боковом освещении по формуле

$$e_p^b = \sum_{i=1}^L \varepsilon_{bi} q_i + \sum_{j=1}^M \varepsilon_{здж} b_{фj} k_{здж} \tau_0 / K_3 \quad (2)$$

б) при верхнем освещении по формуле

$$e_p^в = \sum_{i=1}^L \varepsilon_{vi} + \varepsilon_{ср} (r_2 k_{ф} - 1) \tau_0 / K_3 \quad (3)$$

в) при комбинированном (верхнем и боковом) освещении по формуле:

$$e_p^к = e_p^в + e_p^б \quad (5)$$

где L — количество участков небосвода, видимых через световой проем из расчетной точки;

ε_{bi} — геометрический КЕО в расчетной точке при боковом освещении, учитывающий прямой свет от i -го участка неба;

q_i — коэффициент, учитывающий неравномерную яркость i -го участка облачного неба МКО;

M — количество участков фасадов зданий противостоящей застройки, видимых через световой проем из расчетной точки;

$\varepsilon_{здj}$ — геометрический КЕО в расчетной точке при боковом освещении, учитывающий свет, отраженный от j -го участка фасадов зданий противостоящей застройки;

$b_{фj}$ — средняя относительная яркость j -го участка фасадов зданий противостоящей застройки;

r_0 — коэффициент, учитывающий повышение КЕО при боковом освещении благодаря свету, отраженному от поверхностей помещения и подстилающего слоя, прилегающего к зданию;

$k_{здj}$ — коэффициент, учитывающий изменения внутренней отраженной составляющей КЕО в помещении при наличии противостоящих зданий, определяемый по формуле

$$k_{здj} = 1 + (k_{зд0} - 1) \frac{\sum_{j=1}^M \varepsilon_{здj}}{\sum_{i=1}^N \varepsilon_{6i} + \sum_{j=1}^M \varepsilon_{здj}} \quad (6)$$

где:

$k_{зд0}$ — коэффициент, учитывающий изменения внутренней отраженной составляющей КЕО в помещении при полном закрытии небосвода зданиями, видимыми из расчетной точки;

τ_0 — общий коэффициент светопропускания, определяемый по формуле:

$$\tau_0 = \tau_1 \tau_2 \tau_3 \tau_4 \tau_5 \quad (7)$$

где:

τ_1 — коэффициент светопропускания материала;

τ_2 — коэффициент, учитывающий потери света в переплетах светопроема. Размеры светопроема принимаются равными размерам коробки переплета по наружному обмеру;

τ_3 — коэффициент, учитывающий потери света в несущих конструкциях (при боковом освещении $\tau_3 = 1$);

τ_4 — коэффициент, учитывающий потери света в солнцезащитных устройствах;

τ_5 — коэффициент, учитывающий потери света в защитной сетке, устанавливаемой под фонарями, принимаемый равным 0,9;

K_3 — коэффициент запаса, определяемый по табл. 3;

T — количество световых проемов в покрытии;

ε_{6i} — геометрический КЕО в расчетной точке при верхнем освещении от i -го проема;

$\varepsilon_{\text{ср}}$ — среднее значение геометрического КЕО при верхнем освещении на линии пересечения условной рабочей поверхности и плоскости характерного вертикального разреза помещения, определяемое из соотношения:

$$\varepsilon_{\text{ср}} = \frac{1}{N} \sum_{i=1}^N \varepsilon_{\text{в}i} \quad (8)$$

N — количество расчетных точек;

r_2 — коэффициент, учитывающий повышение КЕО при верхнем освещении благодаря свету, отраженному от поверхностей помещения;

$k_{\text{ф}}$ — коэффициент, учитывающий тип фонаря.

Практическая часть

Проведем расчет естественного освещения комнаты, в которой находится рабочее место инженера.

Исходные данные:

Глубина помещения $d_n = 4,8$ м, высота $h = 2,4$ м, ширина $b_n = 2,5$ м, площадь пола $A_n = 12,0$ м². Заполнение световых проемов двойным остеклением по спаренным деревянным переплетам. Уровень рабочей поверхности 0,7 м. Коэффициент отражения потолка $\rho_{\text{пот}} = 0,70$; стен $\rho_{\text{ст}} = 0,50$; пола $\rho_n = 0,40$. Затенение противостоящими зданиями отсутствует. Здание располагается в Москве. В качестве солнцезащитных устройств используется убирающиеся и регулируемые жалюзи. Верхнее освещение отсутствует.

Расчет проведем в 2-х рабочих точках находящихся от наружной стены на расстоянии 2 и 4 м соответственно.

Решение:

1. Согласно СП 23-102-2003 определяем, что здание располагается в 1 административном районе и нормированное значение КЕО e_n , равно 0,6 %.

2. По исходной глубине помещения и высоте верхней грани светового проема над условной рабочей поверхностью $h_{01} = 1,6$ м; определяем, что $d_n/h_{01} = 3,0$.

3. По рисунку 2 СП23-02-2003 на соответствующей кривой $e = 0,6$ % находим точку с абсциссой $d_n/h_{01} = 3,0$ и определяем, что необходимая относительная площадь светового проема $A_o/A_n = 21,5\%$.

4. Определим площадь светового проема A_o :

$$A_o = 0,215 \times 12 = 2,58 \text{ м}^2$$

Следовательно ширина светового проема будет равна $b_o = 1,6$ м.

Примем оконный блок размером 1,6 х 1,6 м.

5. Произведем проверочный расчет КЕО в точках $M1$, $M2$ по формуле (2).

Так как противостоящих зданий нет, то $\sum_{j=1}^M \varepsilon_{здj} \cdot b_{фj} \cdot k_{здj} = 0$, следовательно расчет будем

производить по формуле:

$$e_p^6 = \varepsilon_o q_o r_o \tau_o / K_s \quad (9)$$

Расчет КЕО будем производить методом А.М. Данилюка, поэтому нам потребуется начертить план и поперечный разрез помещения в одинаковых масштабах (рисунок 5.1). Для точки $M1$ был выбран масштаб 1:100, а для точки $M2$ – 1:50.

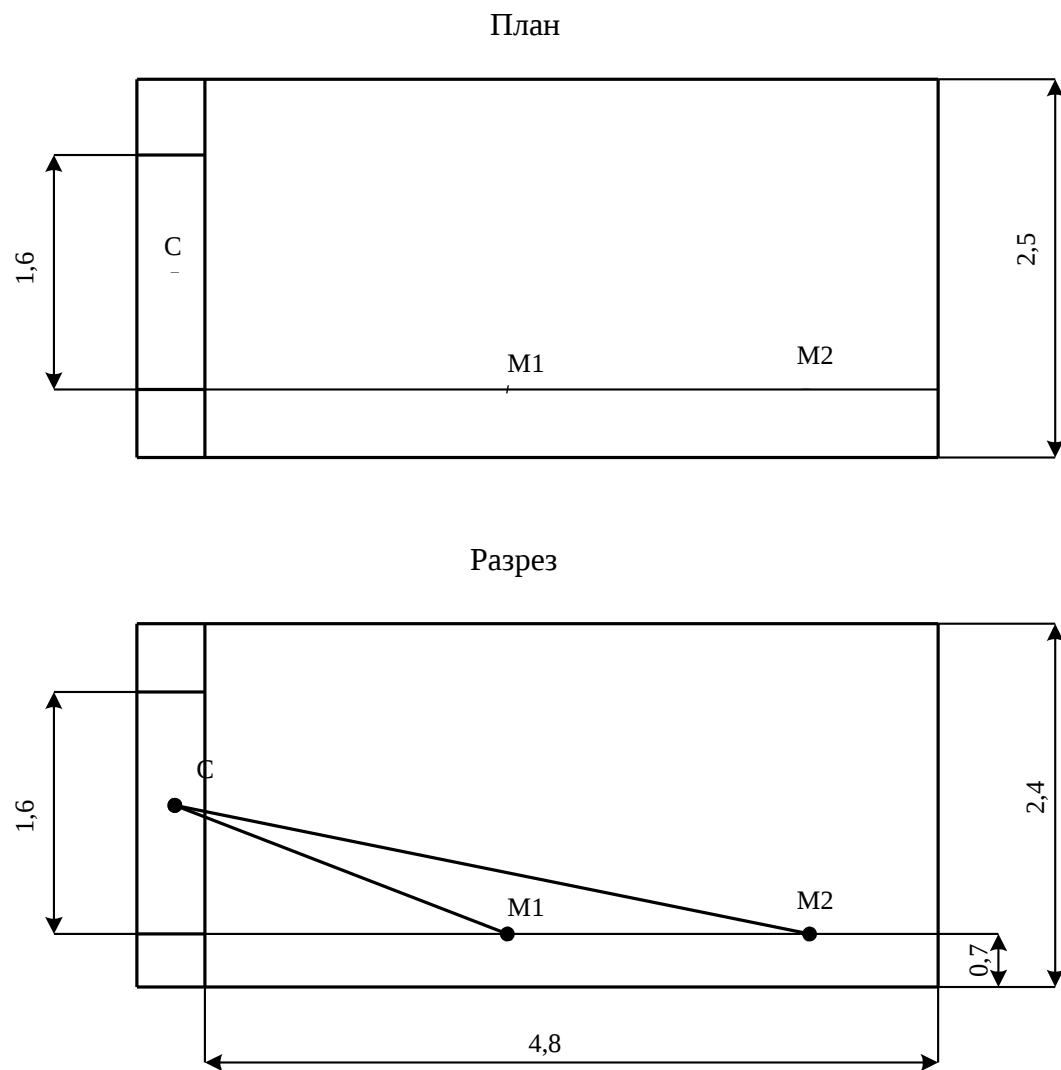


Рисунок 5.1 – План и разрез рабочего помещения

6. Накладываем график I для расчета КЕО методом А.М. Данилюка на поперечный разрез помещения, совмещая полюс графика I — 0 с точкой M1 и затем с точкой M2, а нижнюю линию — с условной рабочей поверхностью и подсчитываем число лучей по графику I, проходящих через поперечный разрез светового проема:

Для точки M1: $n_1 = 6$, через точку C на разрезе помещения проходит полуокружность с номером 9, световой проем, находится под углом $\theta = 20^\circ$ и по приложению Б СП23-02-2003 находим коэффициент, учитывающий неравномерную яркость облачного неба МКО: $q_i = 0,70$.

Для точки M2: $n_1 = 2,9$, через точку C на разрезе помещения проходит полуокружность с номером 34, световой проем, находится под углом $\theta = 12^\circ$, $q_i = 0,60$.

7. Накладываем график II для расчета КЕО на план помещения таким образом, чтобы его вертикальная ось и горизонталь (с номерами 9 и 34 соответственно) проходили через точку C; подсчитывают по графику II число лучей, проходящих от неба через световой проем:

Для точки M1: $n_2 = 31$.

Для точки M2: $n_2 = 26$.

8. Определяем значение геометрического КЕО по формуле

$$\varepsilon_6 = 0,01n_1n_2 \quad (10)$$

Для точки M1: $\varepsilon_6 = 0,01 \times 6 \times 31 = 1,86$;

Для точки M2: $\varepsilon_6 = 0,01 \times 2,9 \times 26 = 0,76$;

9. По размерам помещения и светового проема находят, что $d_n/h_{01} = 3,1$;

$b_n/d_n = 0,51$, для точки M1 $l_T/d_n = 0,41$, для точки M2 $l_T/d_n = 0,83$.

10. Найдем площади поверхностей потолка A_{nom} , стен A_{cm} и пола A_n и определим средневзвешенный коэффициент отражения ρ_{cp} по формуле

$$\rho_{cp} = (\rho_{nom} A_{nom} + \rho_{cm} A_{cm} + \rho_n A_n) / (A_{nom} + A_{cm} + A_n) \quad (11)$$

$$A_{nom} = 12 \text{ м}^2.$$

$$A_{cm} = S_{\text{всех стен}} - S_{\text{окна}} = 4,8 \times 2,4 \times 2 + 2 \times 2,5 \times 2,4 - 2,58 = 32,46 \text{ м}^2.$$

Тогда $\rho_{cp} = (0,7 \times 12 + 0,5 \times 32,46 + 0,3 \times 12) / (12 + 32,46 + 12) = 0,5$.

11. По найденным значениям d_n/h_{01} ; l_T/d_n ; b_n/d_n по таблице Б.4 СП23-02-2003 находим, что для точки $M1$ $r_o = 1,82$, для точки $M2$ $r_o = 3,92$.

12. Теперь определим общий коэффициент светопропускания τ_o .

По табличным значениям определяем:

Для двойного стекла $\tau_1 = 0,8$.

Для двойного деревянного спаренного переплета $\tau_2 = 0,7$.

При боковом освещении $\tau_3 = 1$.

Для убирающихся и регулируемых жалюзях $\tau_4 = 1$.

При расчете бокового естественного освещения коэффициент $\tau_5 = 1$.

Тогда по формуле (7): $\tau_o = 0,8 \times 0,7 \times 1 \times 1 \times 1 = 0,56$.

13. По СНиП 23-05-95 определяем, что коэффициент запаса для окон кабинетов и рабочих помещений с нормальными условиями среды равен $K_3 = 1,2$.

15. Определим геометрический КЕО в точках $M1$, $M2$, подставляя значения коэффициентов ε_b , q_i , r_o , τ_o и K_3 в формулу (9):

Для точки $M1$: $e_p^b = 1,86 \times 0,7 \times 1,82 \times 0,56 / 1,2 = 1,10 \%$.

Для точки $M2$: $e_p^b = 0,76 \times 0,6 \times 3,92 \times 0,56 / 1,2 = 0,83 \%$.

Из двух рассчитанных КЕО выбираем наименьший равный 0,83%.

Так как $e_p^b \geq e_n$, то можно сделать вывод, что помещение соответствует требуемым

нормам по естественному освещению рабочего кабинета.

5.2 Утилизация отработанных средств вычислительной техники

При работе с вычислительной техникой, изготовлении ее компонентов необходимо рассмотреть вопрос утилизации средств вычислительной техники, так как они содержат драгоценные металлы. К драгоценным металлам относятся: золото, серебро, платина, палладий, родий, иридий, рутений, осмий, а также любые химические соединения и сплавы каждого из этих металлов.

Утилизация проводится согласно “Методике проведения работ по комплексной утилизации вторичных драгоценных металлов из отработанных средств вычислительной техники” от 19 октября 1999 года.

Данный процесс проходит следующие стадии:

- Информационное обеспечение;
- Создание условий;
- Разборка изделий;
- Реализация партий.

На стадии “Информационное обеспечение” проводятся сбор информации об утилизируемом объекте: контроль соответствия элементной базы обследуемого объекта технической документации, расчёт ориентированного количества драгоценных металлов по паспортным данным элементной базы (разъёмов, соединителей, микросхем, корпусов), контроль соответствия расчётного количества драгоценных металлов, паспортному (формулярному), которое стоит на учёте в бухгалтерии предприятия.

На стадии “Создание условий” создаются условия для проведения работ по разборке изделий СВТ.

На стадии “Разборка изделий” происходит разборка изделия в зависимости от его типа, конструктивных особенностей и комплектации. Можно выделить следующие виды ВЭТ:

- Разборка универсальных ЭВМ;
- Разборка периферийных устройств;
- Разборка персональных компьютеров, рабочих станций, серверов;

При разборке изделий СВТ образуются материалы и изделия, которые имеют материальную ценность и подлежат реализации (таблица 5.3)

Таблица 5.3 - Виды материалов и изделий, подлежащих реализации при комплексной переработке СВТ.

№ п.п.	Вид материалов или изделий	Характеристика
1	2	3
1	Печатные платы, разъёмы и соединители, микросхемы	вторичные драгоценные металлы
2	Электрические провода и кабели, соединители	вторичная медь и её сплавы
3	Свинец и олово из печатных плат	вторичные припойные пасты (олово и

		свинец)
4	Танталовые конденсаторы	вторичный тантал
5	Некоторые корпуса компьютеров, дисковод и т.д	алюминиевые сплавы
6	Корпуса стоек, ячеек, шкафов, компьютеров	сталь

Продолжение Таблицы 5.3

1	2	3
7	Крепежные изделия	болты, гайки, винты
8	Вентиляторы и электромоторы	по паспорту СВТ
9	Пластиковая "фракция"	стеклотекстолит, пластмасса разъёмов и соединителей
10	Экраны компьютеров	стеклофаза, содержащая Pb, Cd, CdS, редкоземельные металлы

На стадии “Реализация партий” проводится последовательность действий направленная для завершения процедуры утилизации:

- Классификация отходов;
- Сертификация партий;
- Сдача на переработку;
- Соблюдение требований безопасности при работе с вторичными драгоценными металлами.

6 Организационно-экономическая часть

6.1 Введение

АПК ТКПИ разрабатывается в соответствии с планом работ, выполняемых ЗАО “МЦСТ” по ОКР “Повозка-Т”. Оно предназначено для генерации тестовых пакетов на внутреннюю шину контроллера периферийных интерфейсов, сбора и передачи результатов под управлением тестирующих программ.

Его функциональные характеристики:

- Объем внешней памяти 2048 Мб;
- Тип интерфейса с процессором шина PCI;
- Тип интерфейса с КПИ шина IO-Link;
- Режим работы: PIO запросы и DMA запросы;
- В конфигурационное пространство входят 46 программно доступных регистров;
- Наличие логического анализатора объемом 16384 выборок для считывания диаграммы обмена на одном из каналов системного интерфейса контроллера памяти.
- Возможность запуска параллельных тестов для КПИ.

ОКР “Повозка-Т” выполняется на основании – постановление Правительства Российской Федерации от 10.11.07 г. № 771-35 «О государственном оборонном заказе на 2008-2010 гг.». Целью выполнения ОКР является разработка микросхемы контроллера периферийных интерфейсов для высокопроизводительных систем на кристалле с архитектурой «Эльбрус».

Она обладает следующими характеристиками:

- широкий набор периферийных, сетевых и специальных контроллеров (Ethernet, USB2.0, RS-232, AC97 и др.);
- подключение к каналу доступа подсистемы ввода/вывода процессора;
- Максимальный темп обмена с подсистемы ввода/вывода процессора – 1 Гбайт/с (0,5 Гбайт/с - прием, 0,5 Гбайт/с - передача).

Первоначально микросхема будет использоваться для комплектования ВК «Эльбрус-3С», разрабатываемого на основе системы на кристалле «Эльбрус-S». ВК «Эльбрус-3С» предназначен для обеспечения перспективных систем и комплексов противоракетной и

противовоздушной обороны отечественными высокопроизводительными вычислительными системами для решения задач обработки больших объемов информации в режиме жесткого реального времени.

Так же КПИ будет применяться вместе с микропроцессорами «МЦСТ-R500S», «МЦСТ-4R» в различных вычислительных комплексах.

На базе этих разработок планируется поэтапная замена выполнивших свои задачи МВК (многопроцессорных вычислительных комплексов) «Эльбрус-1» и МВК «Эльбрус-2».

Оценим затраты на разработку АПК ТКПИ.

6.1 Основные этапы разработки проекта

Основными этапами разработки проекта являются:

- 1) Разработка структуры АПК ТКПИ;
- 2) Разработка функциональной схемы АПК ТКПИ;
- 3) Проектирование логической модели аппаратной части АПК ТКПИ;
- 4) Реализация аппаратных компонентов;
- 5) Тестирование аппаратной части АПК ТКПИ;
- 6) Разработка документации на устройство АПК ТКПИ.

6.1 Расчет трудоемкости проекта

Общие затраты труда на разработку и внедрение изделия (проекта) Q_p определяют следующим образом:

$$Q_p = t_1 + t_2 + t_3 + t_4 + t_5 + t_6 \quad (6.1)$$

где t_i - затраты труда на выполнение i -го этапа проекта.

Полный перечень работ с разделением их по этапам выполнения проекта показан в таблице 6.1. Для заполнения последних двух столбцов таблицы воспользуемся методом экспертных оценок. В таком случае ожидаемая продолжительность работ вычисляется по следующей формуле:

$$t_{o\mathcal{H}} = \frac{3T_{\min} + 2T_{\max}}{5} \quad (6.2)$$

Таблица 6.1 – Расчет трудоемкости этапов проекта

Этап, t_i	№ Работы	Содержание работы	Трудоемкость			
1	2	3	4			
			(чел-час)		(чел-дни)	
			t_{min}	t_{max}	$t_{ож}$	$t_{ож}$
1	1	Разработка структуры АПК ТКПИ	15	20	17	2,1
	2	Определение набора необходимого оборудования, выбор элементной базы	4	8	5,6	0,7
2	3	Определение состава основных блоков	25	40	31	3,8
	4	Разработка принципа работы устройства	10	16	12,4	1,5
3	5	Разработка алгоритмов работы основных блоков	35	55	43	5,3
	6	Разработка алгоритмов работы вспомогательных блоков	20	40	28	3,5
	7	Проектирование RTL – описаний основных блоков	30	45	36	4,5
	8	Проектирование RTL – описаний вспомогательных блоков	10	18	13,2	1,6
4	9	Сборка проекта	4	7	5,2	0,6
	10	Синтез, компоновка и трассировка АПК ТКПИ	2	6	3,6	0,4
5	11	Тестирование АПК ТКПИ с помощью тестового окружения	40	60	48	6
	12	Наладка АПК ТКПИ на макете	12	18	14,4	1,8
	13	Тестирование АПК ТКПИ на макете	80	120	96	12
6	14	Разработка документации	4	6	4,8	0,6

Общие затраты труда на разработку системы составляют $Q_p = 44,4$ человеко-дня.

6.2 Определение численности исполнителей

Средняя численность исполнителей N определяется как отношение затрат труда Q_p на выполнение проекта к величине фонда рабочего времени F :

$$N = \frac{Q_p}{F} \quad (6.3)$$

Величина фонда рабочего времени определяется соотношением (6.4):

$$F = T \cdot F_M \quad (6.4)$$

где T - время выполнения проекта в месяцах;

F_M - фонд времени в текущем месяце.

Фонд времени в текущем месяце рассчитывается из учета общего числа дней в году, числа выходных и праздничных дней:

$$F_M = \frac{t_p \cdot (D_K - D_B - D_P)}{12} \quad (6.5)$$

где t_p - продолжительность рабочего дня;

D_K - общее число дней в году;

D_B - число выходных дней в году;

D_P - число праздничных дней в году.

Тогда фонд времени в текущем месяце будет равен:

$$F_M = \frac{8 \cdot (365 - 104 - 12)}{12} = 166 \text{ часов.}$$

С учетом того, что срок выполнения проекта составляет 1 месяц (в соответствии с графиком работ, выполняемых ЗАО "МЦСТ" по ОКР "Повозка-Т"), фонд рабочего времени будет равен:

$$F = 1 \cdot 166 = 166 \text{ часов.}$$

Тогда, средняя численность исполнителей будет равна:

$$N = \frac{44,4 \cdot 8}{166} = 2$$

Таким образом, существует необходимость использовать 2-х исполнителей на отдельных работах.

Продолжительность отдельных работ при одновременном выполнении их несколькими исполнителями (t_i) определяется из соотношения:

$$t_i = \frac{t_{pp}}{W_{исп} \cdot K} \quad (6.6)$$

где t_{pp} - расчетная продолжительность работы;

$W_{исп}$ - количество исполнителей;

K_N - коэффициент выполнения нормы.

Продолжительность работ рассчитывается исходя из того, что одной работой занят один исполнитель, а коэффициент выполнения нормы равен единице

Продолжительность выполнения работ и этапов, при условии, что работа по тестированию АПК ТКПИ с помощью тестового ПО, предоставляемым отделом тестирования ЗАО “МЦСТ” для отладки КПИ, и его тестирование на макете будет выполняться двумя исполнителями, показана в таблице 6.2.

Таблица 6.2 - Продолжительность выполнения этапов

Этап, t_i	№ Работы	Содержание работы	Трудоемкость Чел-дни
1	2	3	4
1	1	Разработка структуры АПК ТКПИ	2,1
	2	Определение набора необходимого оборудования, выбор элементной базы	0,7
2	3	Определение состава основных блоков	3,8
	4	Разработка принципов работы устройства	1,5
3	5	Разработка алгоритмов работы основных блоков	5,3
	6	Разработка алгоритмов работы вспомогательных блоков	3,5
	7	Проектирование RTL – описаний основных блоков	4,5

	8	Проектирование RTL – описаний вспомогательных блоков	1,6
--	---	---	-----

Продолжение Таблицы 6.2

1	2	3	4
	9	Сборка проекта	0,6
	10	Синтез, компоновка и трассировка АПК ТКПИ	0,4
5	11	Тестирование АПК ТКПИ с помощью тестового окружения	0,8
	12	Наладка АПК ТКПИ на макете	3
	13	Тестирование АПК ТКПИ на макете	6
6	14	Разработка документации	0,6

6.3 Сетевая модель проекта

Сетевая модель проекта

Сетевой график устанавливает взаимосвязь между всеми работами проекта и позволяет определить продолжительность и трудоемкость как отдельных этапов, так и всего проекта в целом. Построение сетевого графика предполагает использование метода сетевого планирования, на базе которого разрабатывается информационно-динамическая модель процесса выполнения проекта.

В таблице 6.3 показаны основные события и работы проекта.

Таблица 6.3 – Основные события и работы проекта

№	Событие	Код работы	Работа	t	
				Чел-час	Чел-день
1	2	3	4	5	6
0	Начало работы	0-1	Разработка структуры АПК ТКПИ	17	2,1
1	Структура разработана	1-2	Определение состава основных блоков	31	3,8
		1-3	Определение набора необходимого оборудования, выбор элементной базы	5,6	0,7
2	Состав основных блоков определен	2-4	Разработка принципов работы устройства	12,4	1,5
3	Выбрана элементная база				
4	Разработан принцип работы устройства	4-5	Разработка алгоритмов работы основных блоков	43	5,3
		4-6	Разработка алгоритмов работы вспомогательных блоков	28	3,5

Продолжение Таблицы 6.3

1	2	3	4	5	6
5	Разработан алгоритм работы основных блоков	5-7	Проектирование RTL – описаний основных блоков	36	4,5
6	Разработан алгоритм работы вспомогательных блоков	6-7	Проектирование RTL – описаний вспомогательных блоков	13,2	1,6
7	Разработана логическая модель аппаратной части АПК ТКПИ	7-8	Сборка проекта	5,2	0,6
		7-10	Тестирование АПК ТКПИ с помощью тестового окружения	24	3
8	Проекта собран	8-9	Синтез, компоновка и трассировка АПК ТКПИ	3,6	0,4
9	АПК ТКПИ готово к установке на ПЛИС	9-11	Наладка АПК ТКПИ на макете	14,4	1,8
10	RTL модель протестирована				
11	АПК ТКПИ готова к запуску на макете	11-12	Тестирование АПК ТКПИ на макете	48	6
12	Устройство проверено	12-13	Разработка документации	4,8	0,6

Графическое отображение сетевой модели (сетевой график) содержит окружности, отображающие основные события проекта, и векторы, соединяющие эти окружности и определяющие необходимость выполнения соответствующих работ.

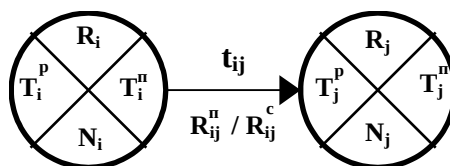


Рисунок 6.1 – Обозначение основных элементов сетевого графика

N_i , N_j - номер события, T_i^p - ранний срок наступления события i , T_i^n - поздний срок наступления события i , R_i - резерв времени события i , t_{ij} - продолжительность работы i - j , R_{ij}^n - полный резерв времени работы i - j , R_{ij}^c - свободный резерв времени работы i - j .

В соответствии с содержанием таблицы основных событий и работ проекта строится графическая модель сетевого графика. На рисунке 6.2 показана графическая модель сетевого графика.

После построения графической модели рассчитаем оставшиеся параметры элементов сети: сроки наступления событий, резервы времени, полный и свободный резервы времени.

Ранний срок совершения события определяет минимальное время, необходимое для выполнения всех работ, предшествующих данному событию и равен продолжительности наибольшего из путей, ведущих от исходного события (0) к рассматриваемому и рассчитывается по соотношению:

$$T_j^p = \max(T_i^p + t_{i-j}) \quad (6.7)$$

Критический путь - максимальный путь от исходного события (0) до завершения проекта. Его определение позволяет обратить внимание на перечень событий, совокупность которых имеет нулевой резерв времени.

Все события в сети, не принадлежащие критическому пути, имеют резерв времени R_i , показывающий на какой предельный срок можно задержать наступление этого события, не увеличивая сроки окончания работ (т.е. продолжительности критического пути).

Поздний срок совершения события - максимально допустимое время наступления данного события, при котором сохраняется возможность соблюдения ранних сроков наступления последующих событий. Поздние сроки вычисляются, начиная с последнего события - завершения проекта, по критическому пути (т.е. справа налево по графику). Они равны разности между поздним сроком совершения j -го события и продолжительностью $i-j$ работы. Поздний срок определяется соотношением:

$$T_i^n = \min(T_j^n - t_{i-j}) \quad (6.8)$$

Резерв времени события определяется следующим образом:

$$R_i = T_i^n - T_i^p \quad (6.9)$$

Полный резерв времени работы определяется, используя соотношение:

$$R_{i-j}^n = T_j^n - T_i^p - t_{i-j} \quad (6.10)$$

Свободный резерв времени можно определить, применяя соотношение:

$$R_{i-j}^c = T_j^p - T_i^p - t_{i-j} \quad (6.11)$$

В результате исследования определяется критический путь на сетевом графике - путь, имеющий наибольшую суммарную длительность работ.

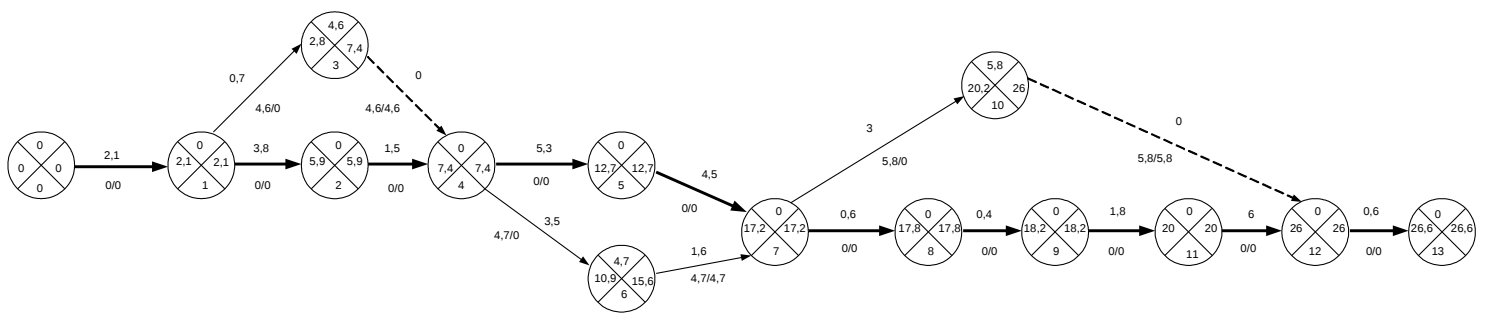


Рисунок 6.2 – Модель сетевого графика

Для рассматриваемого проекта критический путь проходит через вершины: 0-1-2-3-4-5-7-8-9-11-12-13 и имеет длину $T_{кр}=26,6$ дней. Округляя, определим $T_{кр}=27$ рабочих дней. Так как срок выполнения проекта составляет 1 месяц, то оптимизация сетевого графика не требуется.

6.4 Анализ структуры затрат на разработку проекта

Затраты на выполнение проекта состоят из затрат на заработную плату исполнителям, затрат на закупку или аренду оборудования, затрат на организацию рабочих мест, и затрат на накладные расходы:

$$K = C_{зарп} + C_{об} + C_{орг} + C_{накл} + C_{расх.матер} \quad (6.12)$$

Где: $C_{зарп}$ - заработная плата исполнителей;

$C_{об}$ -затраты на обеспечение необходимым оборудованием;

$C_{орг}$ - затраты на организацию рабочих мест;

$C_{накл}$ - накладные расходы;

$C_{расх.мат}$ - затраты на электроэнергию.

Затраты на выплату исполнителям заработной платы определяется следующим соотношением:

$$C_{зарп} = C_{з.осн} + C_{з.доп} + C_{з.отч} \quad (6.13),$$

Где: $C_{з.осн}$ - основная заработанная плата;

$C_{з.доп}$ - дополнительная заработная плата;

$C_{з.отч}$ - отчисление с заработной платы.

Расчет основной заработной платы при дневной оплате труда исполнителей проводится на основе данных по окладам и графику занятости исполнителей:

$$C_{з.осн} = T_{зан} \cdot O_{дн} \quad (6.14)$$

Где: $T_{зан}$ - число дней, отработанных исполнителем проекта;

$O_{дн}$ - дневной оклад исполнителя;

При 8-и часовом рабочем дне он рассчитывается по соотношению:

$$O_{дн} = \frac{O_{мес} \cdot 8}{F_M} \quad (6.15),$$

где $O_{мес}$ - месячный оклад, F_M - месячный фонд рабочего времени.

С учетом налога на доходы физических лиц размер месячного оклада увеличивается, что отражено в следующей формуле:

$$O_{мес} = O \left(1 + \frac{N_{дфл}}{100}\right) \quad (6.16),$$

где O - оклад, который позволит исполнителю заниматься проектом и который получен из информации кадровых агентств, $N_{дфл}$ - налог на доходы с физических лиц (13%).

Перечень исполнителей, их месячные и дневные оклады, а также время участия в проекте и рассчитанная основная заработная плата для каждого исполнителя (www.hh.ru) показаны в таблице 6.4. Требуются 2 исполнителя с опытом работы не менее 1 года, знанием языков Verilog, C++, опытом работы с САПР Quartus и системой моделирования Modelsim.

Таблица 6.4 – Основная заработная плата специалистов

№	Должность	Оклад, руб.	Дневной оклад, руб.	Трудовые затраты, ч	Заработная плата, руб.
1	2	3	4	5	6
1	Инженер	30000	1445,78	35,4	51180,61
2	Ведущий программист	44000	21206,48	10	21204,8

Расходы на дополнительную заработную плату учитывают все выплаты непосредственно исполнителям за время не проработанное на производстве, но предусмотренное законодательством, в том числе: оплата очередных отпусков, компенсация

за недоиспользованный отпуск, и др. Величина этих выплат составляет 20% от размера основной заработной платы:

$$C_{з.доп} = 0.2 \cdot C_{з.осн} \quad (6.17)$$

Для рассматриваемой разработки $C_{з.доп}$ составит 14 477,08 руб.

Отчисления с заработной платы состоят в настоящее время в уплате единого социального налога. С 2010 года единый социальный налог отменен, вместо него нынешние плательщики налога будут уплачивать страховые взносы в пенсионный фонд РФ, фонд социального страхования, фонды обязательного медицинского страхования (федеральный и территориальный фонды), согласно закону № 213-ФЗ от 24.07.2009.

Отчисления с заработной платы составят:

$$C_{з.отч} = (C_{з.осн} + C_{з.доп}) \cdot H_{соц} \quad (6.18)$$

где $H_{соц}$ - включает в себя следующие отчисления:

в фонд социального страхования РФ – 2,9%,

в фонд обязательного медицинского страхования – 3,1%,

в федеральный бюджет – 6%,

в пенсионный фонд РФ – 14%.

Исходя из этого, $H_{соц} = 26\%$ и отчисления на единый социальный налог равны

$C_{з.отч} = 22584,2$ руб. Тогда затраты на выплату заработной платы для рассматриваемой

разработки равны $C_{зарп} = 72385,41 \text{руб} + 14477,08 \text{руб} + 22584,2 \text{руб} = 109\,446,69$ руб.

Затраты на расходные материалы определяются перечнем минимально необходимых для организации работ расходных материалов. Данный перечень приведен в таблице 6.5. Цены даны в соответствии с информацией (www.komus.ru) для Москвы на 04.04.2010.

Таблица 6.5 – Расходные материалы

№	Наименование	Цена, руб.	Количество, шт.	Сумма, руб.
1	2	3	4	5

1	Бумага офисная КОМУС (500л, А4)	142	2	284
2	Карандаш (механический) Pilot Н-165	78,80	3	236,40
3	Ручка шариковая, синяя, черная Attache	5,55	4	22,2
4	Ластик Koh-I-noor 300	5,60	3	16,8
5	Кресло офисное Prestige	1399	2	2798
6	Стол письменный 1401 бук	1614	3	4842
				Итого: 8199,4 руб.

Итого $C_{\text{расх.матер}} = 8199,4$ руб.

Затраты, связанные с обеспечением работ оборудования, определяются составом оборудования и необходимостью его закупки или аренды. В данной разработке целесообразно закупить необходимое оборудование. Перечень оборудования, необходимого для организации работ и расчет затрат на его приобретение приведен в таблице 6.6. Цены даны в соответствии с информацией (www.altera.ru, megratec.ru, computers.wikimart.ru, deltel.ru) на 04.04.2010 (среднее значение)

Таблица 6.6 – Затраты на оборудование

№	Наименование	Цена, руб.	Количество, шт.	Сумма, руб
1	2	3	4	5
1	ПЭВМ в настольном исполнении на базе Intel Pentium IV 2,8GHz /266MHz /256K/ FC-PGA	22000	3	66000
2	Принтер HP LaserJet P1005n	4660	1	4660
3	Процессор MCST R500S	5842	1	5842
4	ПЛИС EP2S60F1020C3	42698	1	42698
5	ПЛИС EP2SGX130GF1508C3	197488	1	197488
6	ППЗУ EPCS64SI16N	1177	2	2354
7	SDRAM MT48LC32M16A2TG	1583	4	6332
8	ПЛИС XC5VLX50T-1FF1136C	18139	1	18139
9	Лицензия на САПР Quartus	85060	1	85060
10	Лицензия на САПР Modelsim	35052	2	70104
				Итого: 498677 руб.

Итого $C_{\text{об.мак}} = 498\,677$ руб, это затраты для всего макета (основные

комплектующие), а для нашего проекта $C_{\text{об}} = 276\,031$ руб. По экспертной оценке Попова

А.Ю. затраты в рамках всего ОКР “Повозка-Т” составляют около 42 млн. руб. Тогда затраты

на оборудование в данном проекте составляют 0.6% от всех затрат на выполнение ОКР. Поэтому для организации затраты на оборудование в данном проекте являются не существенными.

Расчет затрат, связанных с организацией рабочих мест для исполнителей проекта, проводится, ориентируясь на требования СанПин 2.2.2/2.4.1340-03 (санитарные правила и нормы) и на стоимость годичной аренды помещения требуемого уровня сервиса.

В соответствии с санитарными нормами, расстояние между рабочими столами с видеомониторами должно быть не менее 2 м., а между боковыми поверхностями видеомониторов - не менее 1,2 м. Площадь на одно рабочее место с терминалом или ПК должна составлять не менее 6 кв.м., а объем - не менее 20 куб.м. Площадь, предусмотренная для размещения одного принтера, соответствует 0,5 площади рабочего места исполнителя. Исходя из количества исполнителей и набора необходимой офисной техники получаем что необходимая площадь офисного помещения равна, приблизительно, 24-30 м². Цены даны в соответствии с информацией (www.amant-realty.ru) для Москвы на 04.04.2010.

В итоге для аренды было выбрано помещение (м. Третьяковская). Наиболее дешевое офисное помещение, удовлетворяющее требованиям. Расположенное около метро Третьяковская, с телефоном МГТС, интернетом, соответствующее необходимой рабочей площади.

Вычислим затраты на аренду помещения:

$$C_{орг} = \frac{C_{кв.м}}{12} \cdot S \cdot T_{AP} \quad (6.19),$$

где $C_{кв.м}$ - стоимость аренды одного кв. метра площади за год,

S - арендуемая площадь рабочего помещения,

T_{AP} - срок аренды.

$$\text{Таким образом затраты на аренду помещения} = \frac{19483}{12} \cdot 24 \cdot 4 = 77\,932 \text{ руб.}$$

Накладные расходы, связанные с выполнением проекта, следует вычислить, ориентируясь на расходы по основной заработной плате. Обычно они составляют от 60%

до 100% расходов на основную заработанную плату. По экспертной оценке Попова А.Ю. накладные расходы, составляют 60%. Таким образом:

$$C_{\text{НАКЛ}} = 0.6 \cdot C_{\text{З.ОСН}} \quad (6.20)$$

Для рассматриваемой разработки $C_{\text{НАКЛ}}$ равны 43 431 руб.

Тогда суммарные **затраты на выполнение проекта** равны:

$$C_{\text{З.ОСН}} + C_{\text{НАКЛ}} = 109446 \text{ руб.} + 43431 \text{ руб.} = 152887 \text{ руб.}$$

Для наглядности проиллюстрируем структуру затрат на выполнение проекта, используя круговую диаграмму, показанную на рисунке 6.3.

Рисунок 6.3 – Структура затрат на выполнение проекта

6.5 Определение возможной прибыли

АПК ТКПИ разрабатывается в рамках ОКР “Повозка - Т”, цель которой разработка КПИ. Поэтому будем определять возможную прибыль от данного проекта для конечной цели проектирования, т.е для КПИ.

Так как, ОКР “Повозка-Т” выполняются в соответствии с постановлением Правительства Российской Федерации от 10.11.07 г. № 771-35 «О государственном оборонном

заказе на 2008-2010 гг.», то срок окупаемости отсутствует, так как все затраты на проведение данного ОКР берет на себя государство.

Определим возможную прибыль от реализации данного проекта.

По экспертным данным Попова А.Ю. планируется выпускать по 4000 данных микросхем в год по стоимости 5800 руб. за одну микросхему, продолжительность выпуска 2 года, стоимость производства одной партии микросхем равняется 8.2 млн. руб.

Себестоимость одной микросхемы определим из соотношения, так как данные о производстве микросхем, являются закрытыми сведениями.

$$K_{\text{себ}} = \frac{C_{\text{произв. парт}}}{N_{\text{мс}}}, \quad (6.21)$$

где $C_{\text{произв. парт}}$ - стоимость производства одной партии микросхем,

$N_{\text{мс}}$ - количество микросхем в одной партии.

Таким образом, себестоимость одной микросхемы равна:

$$K_{\text{себ}} = \frac{8200000 \text{ руб}}{4000} = 2050 \text{ руб}$$

Процент прибыли и сумму расчетной прибыли от одной реализации данной микросхемы, определим по формулам:

$$D_{\text{приб}} = \left(\frac{K_{\text{пр}}}{\Delta K + K_{\text{вн}}} - 1 \right) \cdot 100\% \quad (6.22)$$

где $D_{\text{приб}}$ - процент прибыли от одной реализации микросхемы,

$K_{\text{пр}}$ - стоимость микросхемы, поставляемой на рынок,

ΔK - часть стоимости изготовления, приходящаяся на одну микросхему,

$K_{\text{вн}}$ - стоимость внедрения микросхемы.

$$C_{\text{приб}} = (\Delta K + K_{\text{вн}}) \cdot D_{\text{приб}} \cdot (1 - H_{\text{ндс}}) \quad (6.23)$$

где $H_{\text{ндс}}$ - процентная ставка налога на добавочную стоимость (18%).

Тогда получаем:

$$C_{\text{приб}} = (2050 \text{ руб} + 0) \cdot 2.32 \cdot 0.82 = 4740 \text{ руб.}$$

Расчетная прибыль от продажи одной микросхемы равна 4740руб.

6.6 Оценка экономических эффектов от разработки проекта

Данный проект разрабатывается на этапе тестирования модели микросхемы КПИ с использованием ПЛИС. Сам КПИ разрабатывается для систем на базе микропроцессоров “Эльбрус-S”, “МСЦТ-4R”, данные системы предназначены для обеспечения перспективных систем и комплексов противоракетной и противовоздушной обороны отечественными высокопроизводительными вычислительными системами для решения задач обработки больших объемов информации в режиме жесткого реального времени.

Каждая такая система уникальна и для них в процессе проектирования изготавливаются уникальные отладочные стенды и прототипы. Поэтому объекты сравнения отсутствуют.

На данном этапе основным фактором является надежность разрабатываемого КПИ и возможность его верификации.

Оценим выигрыш в сроках проектирования и стоимости разработки (без накладных расходов и зарплаты) при использовании отладочных стендов для микросхем, изготавливаемых с технологическими нормами 130 нм.

Стоимость изготовления пластин по экспертной оценке Попова А.Ю. для данной технологической нормы составляет около \$250 тысяч долларов или 7,3 млн. руб.

В случае обнаружения ошибки в изготовленных микросхемах, время повторного топологического проектирования по экспертной оценке составляет от 2 до 3-х месяцев, время изготовления кристаллов – 2 месяца, время доставки и корпусирования кристаллов – 1 месяц. Таким образом, стоимость повторного изготовления микросхемы $C_{\text{повт.изгот}}$ составит от 7,3 млн. руб. до 8,2 млн. руб., время изготовления – 5-6 месяцев.

Как видно из таблицы 6.6, если учитывать только затраты на основные комплектующие, то стоимость макета $C_{\text{макета}}$ составляет около 498.677руб, время

разработки всего макета по экспертным оценкам 2 месяца, изготовления (с учетом времени закупки комплектующих) 2-3 месяца. Итого 5 месяцев.

Как видно, время разработки не увеличивается, около 5 месяцев. Общее время разработки согласно техническому заданию на ОКР “Повозка-Т” составляет 26 месяцев (октябрь 2008 года – ноябрь 2010 года). Т.о данный этап составляет 19% от времени всей разработки, что является существенным временем.

Оценим теперь разницу стоимостей:

$$\text{Отношение стоимости} = \frac{C_{\text{повт.изгот.}}}{C_{\text{макета}}} = \frac{8.200.000 \text{ руб}}{498.677 \text{ руб}} = 16.4$$

Таким образом, если отладочный стенд позволяет найти дополнительные ошибки, не выявляемые на предыдущих этапах верификации, то выигрыш в стоимости разработки может превысить в 16 раз, при этом время разработки не увеличивается.

На рисунке 6.4 показана круговая диаграмма затрат на весь макет (учитываем только затраты на комплектующие) для отладки КПИ и на изготовление микросхемы на фабрике.

Рисунок 6.4 – Сравнение затрат на изготовление макета и на изготовление кристалла микросхемы КПИ.

Оценим эффективность от применения КПИ в вычислительных комплексах, выпускаемых ЗАО “МЦСТ”.

В качестве периферийного контроллера ввода – вывода («южный мост») и в качестве системного контроллера ввода – вывода «северный мост» в предыдущих проектах, использовались либо импортные микросхемы, либо контроллеры на основе ПЛИС. Например, в вычислительном комплексе “Эльбрус-3М1” в качестве системного контроллера ввода – вывода используется микросхема компании «Intel» взаимодействующая с периферийным контроллером ввода – вывода посредством интерфейса PCI, который обеспечивает производительность 133 Мбайт/сек. Стоимость такой микросхемы равняется 1015 руб.(www.xard.ru).

Такой подход имеет следующие недостатки:

- Зависимость от зарубежных производителей;
- Низкая производительность.

Микросхема КПИ является мультиконтроллером “южного моста”, подключаемая к каналу доступа подсистемы ввода/вывода процессора и содержащая широкий набор периферийных, сетевых и специальных контроллеров.

Она взаимодействует с процессором по дуплексному каналу межсистемного обмена IO-Link с производительностью 1 Гбайт/с (0,5 Гбайт/с - прием, 0,5 Гбайт/с - передача) и также является периферийным контроллером ввода – вывода с поддержкой множества интерфейсов.

Таким образом, при применении микросхемы КПИ производительность увеличивается в 8 раз, но при этом стоимость в 6 раз дороже по сравнению с используемыми импортными аналогами. Получаемые преимущества от ее изготовления:

- Полностью отечественная разработка;
- Отсутствие зависимости от зарубежных производителей;
- Подключение к каналу доступа подсистемы ввода/вывода процессора;
- Широкий набор периферийных, сетевых и специальных контроллеров.

При использовании ПЛИС в качестве такой микросхемы ее стоимость равна 197488 руб. Рыночная стоимость КПИ равна 5800 руб., при выпуске партии из 4000 микросхем КПИ, то ее использование дает экономию средств в 34 раза.

Выводы

- 1) Применение микросхемы КПИ с широким набором периферийных, сетевых и специальных контроллеров позволяет увеличить производительность в 8 раз, но в 6 раз дороже по сравнению с используемыми импортными аналогами;
- 2) Использование КПИ позволяет сократить расходы в 34 раза по сравнению с использованием ПЛИС в качестве такого контроллера;
- 3) Разработка КПИ позволяет избежать зависимости от зарубежных производителей микросхем;
- 4) Отладочный стенд позволяет найти дополнительные ошибки, не выявляемые на предыдущих этапах верификации, это позволяет в 16 раз снизить стоимость разработки, по сравнению с вариантом отказа проведения стендовых испытаний и повторного изготовления микросхемы КПИ, при обнаружении ошибок;
- 5) Возможная прибыль от продажи одной микросхемы равна 4740 руб.;
- 6) Суммарные затраты на выполнение проекта АПК ТКПИ равны 515 039 руб.

Заключение

1. В результате выполнения дипломного проекта разработано устройство, предназначенное для генерации тестовых пакетов на внутреннюю шину контроллера периферийных интерфейсов, сбора и передачи результатов под управлением тестирующих программ. Устройство имеет 43 программно-доступных регистра.
2. Результаты выполнения дипломного проекта полностью соответствуют требованиям технического задания. Устройство реализовано на современной вычислительной элементной базе – ПЛИС Stratix II EP2S60F1020C3 фирмы Altera.
3. В ходе исследования рассмотрены современные способы верификации и тестирования микросхем. На основании такого анализа разработана методика тестирования контроллера периферийных интерфейсов.
4. В ходе проектно-конструкторской работы разработаны компоненты АПК ТКПИ (“Блок регистров”, “Последовательный буфер”, “Буфер для потерянных пакетов”, “Буфер ответов”, “Буфер для APIС запросов”, “Логический анализатор”). Электрическая принципиальная схема представлена с помощью языка высокого уровня описания аппаратуры Verilog на уровне регистровых передач.
5. В технологическом разделе проведен синтез логической модели АПК ТКПИ, для этого применялась САПР Quartus II версии 9.0. В результате использования мегафункций Quartus II для размещения буферов памяти в ПЛИС общая используемая площадь ПЛИС равна 29%, что обеспечивает гибкость при внесении изменений в АПК ТКПИ и получения требуемых частот для работы АПК ТКПИ.
6. В организационно-экономической части проекта разработан поэтапный план проведения работ по разработке АПК ТКПИ, проведен расчет затрат на реализацию проекта и определена норма прибыли.
7. В части, посвященной охране труда, определены опасные и вредные факторы, влияющие на оператора-разработчика. Произведен расчет системы естественного освещения на этапе проектирования логической модели АПК ТКПИ и рассмотрен вопрос утилизации отработанной вычислительной техники.

8.Результат дипломного проекта применяется в отладочном стенде для проверки КПИ. Его использование позволяет в 16 раз снизить стоимость разработки КПИ, по сравнению с вариантом проведения стендовых испытаний и повторного изготовления микросхемы КПИ, при обнаружении ошибок.

Список использованных источников

1. Ian Burgess. Regression Testing: Gate-Level Functional Verification Is Imperative and Equivalence Checking Provides the Solution. White Paper. – Mentor Graphics Corporation, 2004. – 8p.
2. Инженерная записка по ОКР “Повозка-Т”. – ЗАО “МЦСТ”, 2009. – 27с.
 3. Маслов В.С. Методические указания к курсовому/дипломному проекту и квалификационной работе. – М.: МГТУ им. Н.Э. Баумана, 2007. – 8с. ил.
4. Богданов А.Ю. Отчет по преддипломной практике. – 2010. – 19с. ил.
5. FormalPro Reference Manual. Mentor Graphics Corporation, 2008. - 268p.
6. Harry D. Foster. Integrating Functional Formal Verification Into Your Flow. White Paper. – Mentor Graphics Corporation, 2006. – 16p.
7. Бобков С.Г. Методика тестирования микросхем для серии “Багет”. //Программные продукты и системы. – 2007. – №3.
8. IEEE Standard Test Access Port and Boundary-Scan Architecture. – USA: The Institute of Electrical and Electronics Engineers, 2001. – 208p.
9. Диденко А.Б. Системный контроллер высокоскоростного ввода/вывода. ЗАО “МЦСТ”, 2010. – 31с.
10. Иванов А.В. Системный коммутатор контроллера периферийных устройств ИОН. ЗАО “МЦСТ”, 2009. – 109с.
11. Synchronous DRAM. MT48LC32M16A2TG-75C. Data sheet- Micron, 2005. – 57p.
12. IEEE Standard Verilog Hardware Description Language. – The Institute of Electrical and Electronics Engineers, 2001. – 791p.
13. Clifford E. Cummings. Synthesis and Scripting Techniques for Designing Multi-Asynchronous Clock Designs. – San Jose, 2001. – 26p.
 14. PCI Local Bus Specification. - PCI Special Interest Group, 2002. - 328p.
 15. Stratix II Device Handbook - Altera, 2009. – 768p.
 16. Introduction to the Quartus II Software - Altera, 2009. – 238p.
 17. Quartus II Handbook Version 9.1- Altera, 2009. – 2454p.
 18. СанПиН 2.2.2/2.4.1340-03. “Гигиенические требования к персональным электронно-вычислительным машинам и организации работы”.
 19. СНиП 23-05-95. Строительные нормы и правила Российской Федерации. Естественное и искусственное освещение.

20. СП 23-102-2003. Свод правил по проектированию и строительству. Естественное освещение жилых и общественных зданий.
21. Меняев М.Ф., Бышовец Б.Д., Пряников И.Ф. Организационно-экономическая часть дипломных проектов, направленных на разработку программного обеспечения. Учебное пособие. –М.:Изд-во МГТУ им.Н.Э.Баумана, 2007. – 30 стр.
22. Steve Golson, State machine design techniques for Verilog and VHDL. - Trilobyte Systems, 2004. – 20p.
23. J. Bhasker, Verilog HDL Synthesis a Practical Primer. – Star Galaxy Publishing, 1998. – 230p.

ПРИЛОЖЕНИЕ А

Техническое задание

ПРИЛОЖЕНИЕ Б

Графические материалы

ПРИЛОЖЕНИЕ В

Пространство регистров статуса и управления АПК ТКПИ