

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
«Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

М.Н. Томчук

ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО ДИСЦИПЛИНЕ «СХЕМОТЕХНИКА ЭВМ»

Учебное пособие



Киров 2011

УДК 004.3

Томчук М.Н. Лабораторный практикум по дисциплине «Схемотехника ЭВМ». Учебное пособие. – Киров, изд. ВятГУ, 2011 – 62 с.

Предлагается цикл лабораторных работ по разработке и исследованию элементов и узлов ЭВМ. Приводятся методические указания по выполнению работ, варианты заданий, требования к выполнению работ и отчетам.

Учебное пособие подготовлено на кафедре ЭВМ ВятГУ и предназначено для студентов, обучающихся по направлению 230100 – «Информатика и вычислительная техника». Оно также может быть полезно студентам других направлений, изучающих основы цифровой схемотехники ЭВМ.

Содержание

1 Общие положения	4
2 Структура цикла лабораторных работ	4
3 Порядок выполнения лабораторной работы	5
4 Описание лабораторного стенда для выполнения лабораторных работ 4 семестра.....	6
5 Описание стенда miniDiLaB-СШ для выполнения лабораторных работ 5 семестра.....	9
6 Порядок выполнения лабораторных работ 4 семестра	14
7 Порядок выполнения лабораторных работ 5 семестра	19
8 Задания на лабораторные исследования.....	50
Приложение А. Примеры вариантов заданий на лабораторные исследования.....	57
Приложение Б. Таблица подключений пользовательских выводов ПЛИС	58
Приложение В. Пример оформления титульного листа отчета	61
Библиографический список.....	62

1 Общие положения

Учебное пособие предназначено для студентов направления 230100 – «Информатика и вычислительная техника» и содержит указания по выполнению цикла лабораторных работ по дисциплине «Схемотехника ЭВМ», а также задания и требования к отчетам.

Целью лабораторных работ является получение и закрепление знаний и навыков, необходимых для разработки, отладки и ремонта цифровых устройств, знакомство с номенклатурой отечественных цифровых микросхем.

Для успешного выполнения лабораторных работ необходимо знать основы электротехники и электроники, основы двоичной логики, правила составления и минимизации логических функций, уметь составлять логические функции в различных логических базисах. Кроме того, для выполнения очередной лабораторной работы необходимо ознакомиться с лекционным материалом по теме лабораторной работы.

2 Структура цикла лабораторных работ

Цикл лабораторных работ по дисциплине «Схемотехника ЭВМ» рассчитан на два семестра и выполняется за 34 академических часа (по 17 академических часов в семестре). В таблице 1 приведен список лабораторных работ с указанием количества часов, необходимых для выполнения каждой лабораторной работы.

Таблица 1 – Структура цикла лабораторных работ

Семестр	№	Название работы	Час.
4	1	Исследование логических элементов	4
	2	Исследование триггеров	4
	3	Исследование регистров	4
	4	Исследование счетчиков	4
		Сдача долгов, доработка и защита отчетов по лабораторным работам	1
5	5	Разработка АЛУ для выполнения операции умножения на базе ПЛИС	4
	6	Разработка АЛУ для выполнения отдельных операций на базе ПЛИС	4
	7	Разработка АЛУ для выполнения набора операций на базе ПЛИС	8
		Сдача долгов, доработка и защита отчетов по лабораторным работам	1

Цикл лабораторных работ делится на две части. Первая ориентирована на изучение основ схемотехники на реальных микросхемах.

При ее выполнении студентом разрабатывается функциональная схема узла, затем принципиальная схема с учетом заданных или выбранных микросхем. Узел собирается на макетной плате стенда: в панельки устанавливаются выбранные микросхемы, а контактные площадки соединяются проводами при помощи пайки. Далее производится тестирование и отладка узла.

Вторая часть цикла лабораторных работ направлена на изучение основ работы со СБИС программируемой логики. Студенты используют САПР для разработки функциональной схемы устройства, моделирования и компиляции ее, а затем программируют ПЛИС и проверяют работу устройства.

3 Порядок выполнения лабораторной работы

Выполнение каждой лабораторной работы состоит из следующих частей:

- домашняя подготовка;
- разработка функциональной схемы узла в соответствии с заданием;
- разработка принципиальной схемы и сборка ее на макетной плате стенда (4 семестр) или разработка функциональной схемы средствами САПР, компиляция ее и программирование ПЛИС (5 семестр);
- тестирование устройства;
- при необходимости – отладка устройства;
- демонстрация работы устройства преподавателю;
- оформление отчета;
- защита отчета.

Домашняя подготовка включает повторение лекционного материала по теме лабораторной работы, самостоятельную работу с литературой (в том числе справочной).

Разработка функциональной схемы выполняется в соответствии с заданием на лабораторную работу и (для семестра 4) с учетом заданных микросхем или микросхем, имеющих в наличии. Часть работы можно выполнять в процессе домашней подготовки.

Разработка принципиальной схемы подразумевает выбор конкретных микросхем для реализации элементов и узлов функциональной схемы и разработку схемы, в которой указаны типы микросхем и номера выводов для всех элементов.

Сборка устройства на макетной плате заключается в установке микросхем в панельки и выполнении соединений в соответствии с принципиальной схемой при помощи пайки проводов к контактным площадкам макетной платы.

Реализация разработанной функциональной схемы в ПЛИС предполагает создание проекта в САПР, разработку функциональной схемы, задание назначения выводов ПЛИС, компиляцию схемы и программирование ПЛИС. Часть этой работы можно выполнять в процессе домашней подготовки (при наличии САПР).

Тестирование устройства заключается в проверке его работоспособности для всех допустимых вариантов входных воздействий. При обнаружении ошибок в работе устройства необходимо эти ошибки исправить.

После успешной демонстрации работы преподавателю оформляется отчет, содержащий **титульный лист, задание, описание всех этапов разработки, временные диаграммы** (при необходимости – см. требования к отчету по лабораторным работам), **выводы**.

Защита отчета заключается в ответах на вопросы преподавателя и выполнении индивидуальных заданий по теме лабораторной работы. После демонстрации работы и защиты отчета лабораторная работа считается сданной.

4 Описание лабораторного стенда для выполнения лабораторных работ 4 семестра

Лабораторный стенд предназначен для сборки и тестирования узлов, собранных из логических микросхем, и позволяет:

- собирать узлы, содержащие до восьми микросхем в стандартных корпусах DIP4-DIP16;
- подавать на входы узла до 16 постоянных логических сигналов, задаваемых пользователем (тумблеры В0-В15);
- подавать на входы узла до 2 импульсных логических сигналов без дребезга (кнопки);
- отображать состояния до 16 логических сигналов (светодиоды L0-L15: зеленый цвет – единица, красный – ноль);
- использовать в качестве устройства вывода информации 7-сегментный индикатор (подключен параллельно L8-L15, низкий уровень зажигает сегмент);
- отображать в шестнадцатеричном коде состояния тумблеров В0-В7 и входов L0-L7;
- использовать два выхода генератора тактовой частоты.

Стенд обеспечивает макетную плату напряжением питания 5 В.

Основная плата стенда схематично изображена на рисунке 1.

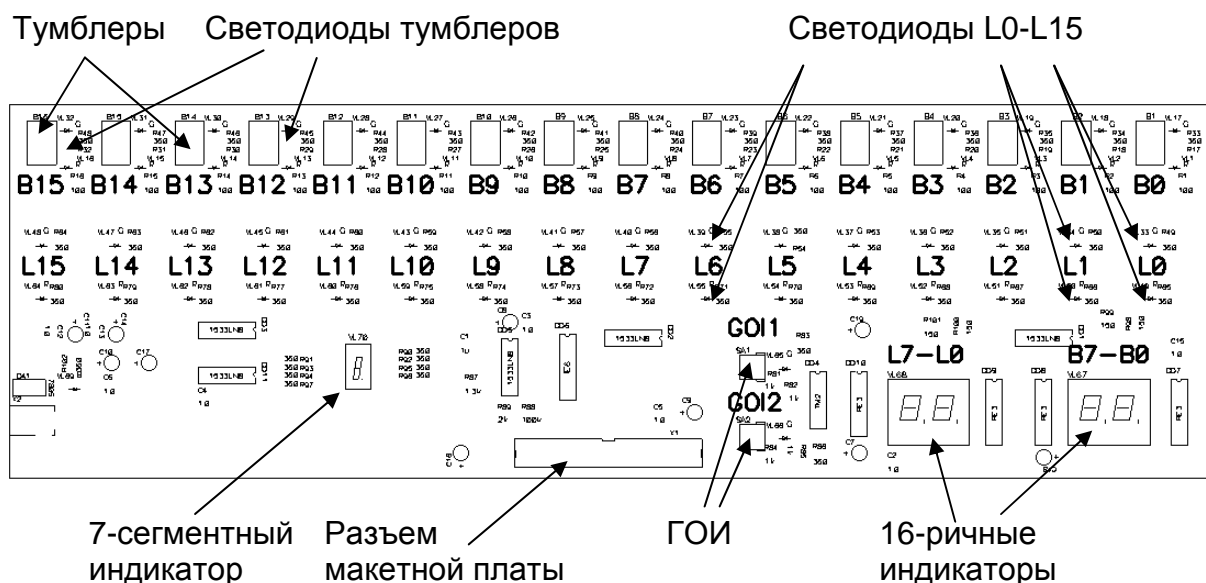


Рисунок 1 – Основная плата стенда

Тумблеры предназначены для подачи логических сигналов на макетную плату: если переключатель находится в нижнем положении, выдается ноль, если в верхнем – единица. При переключении возможендребезг контактов, поэтому выходы В0-В15 не следует подключать к динамическим входам Т-триггеров, счетчиков и сдвиговых регистров во избежание нескольких повторных срабатываний. Каждый тумблер связан со своей контактной площадкой макетной платы. Справа от каждого тумблера расположены светодиоды, которые отображают состояние выхода тумблера: красный – ноль, зеленый – единица. Если тумблер в верхнем положении, должен гореть зеленый светодиод. Но если при этом выход тумблера связан с другим выходом (например, выходом другого тумблера или выходом логического элемента), и на другом выходе ноль, будет гореть красный светодиод. Эта ситуация не является нормальной (ТТЛ-конфликт) и ее желательно избегать. При этом ток логической единицы тумблера ограничен, поэтому такая ситуация не приводит к повреждению микросхем и стенда. Ток логического нуля тумблера не ограничен, поэтому **подключать к выходу тумблера шину питания напрямую запрещается.**

Светодиоды L0-L15 отображают уровень логического сигнала на соответствующих контактных площадках макетной платы: красный цвет – «ноль», зеленый – «единица».

Параллельно светодиодам L8-L15 подключен 7-сегментный индикатор, что позволяет использовать входы L8-L15 для отображения на нем цифры или иного символа (каждый сегмент загорается нулем на соответствующем ему входе).

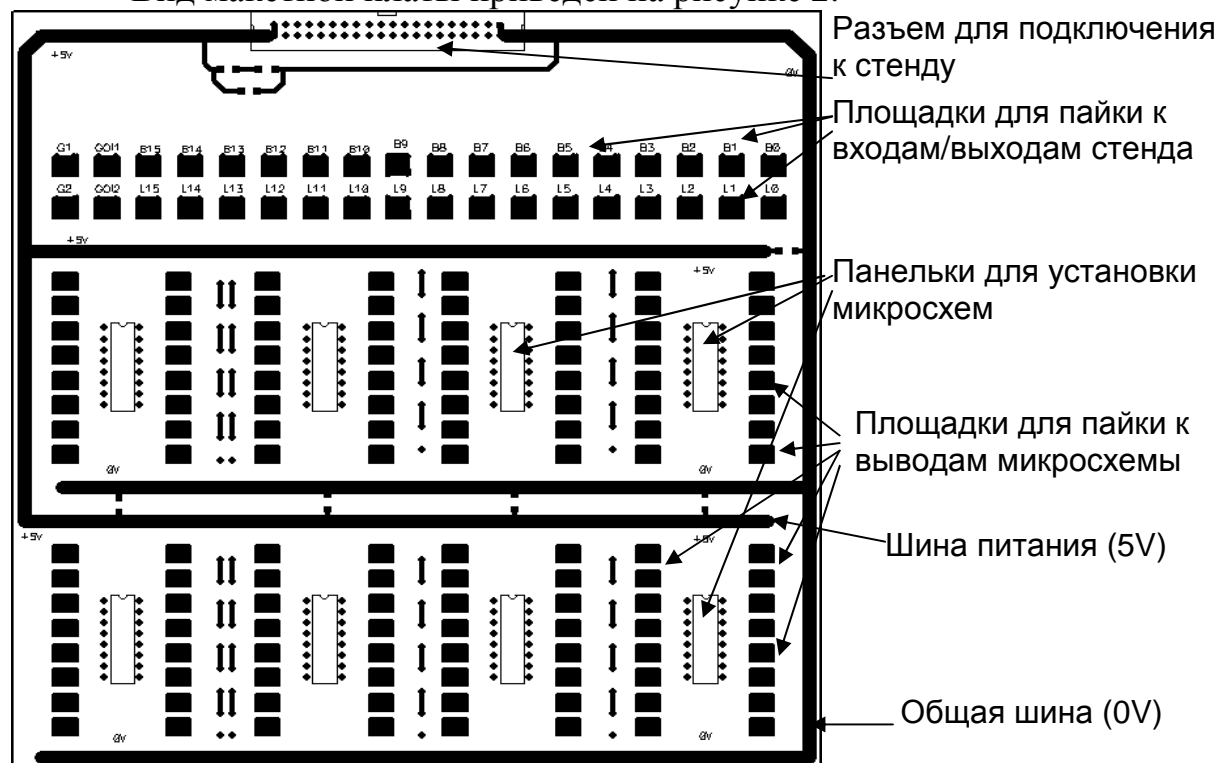
Состояние младших восьми разрядов выходов дублируется в шестнадцатеричном виде на индикаторе В7-В0, а состояние младших восьми разрядов входов – на индикаторе L7-L0.

Для выдачи одиночных импульсов на макетную плату служат генераторы одиночных импульсов ГОИ1 и ГОИ2. При отпущенных кнопках ГОИ1 и ГОИ2 состояние выходов, соответственно, 0 и 1. При нажатии на кнопку состояние выхода меняется на противоположное, а при отпускании кнопки – на исходное. Текущее состояние выхода ГОИ отображается зеленым светодиодом рядом с кнопкой: «1» – если светодиод горит, и «0» - если не горит. Особенность этих выходов в том, что при нажатии и отпускании кнопок отсутствуетдребезг контактов, что позволяет подключать эти выходы к любым динамическим входам микросхем.

Кроме того, на плате стенда расположен генератор тактовой частоты (около 100 Гц) и два выхода: выход генератора и выход делителя на 10 (частота около 10 Гц). Выходы генератора можно использовать для подачи серии периодических импульсов на схему. При этом генератор невозможно отключить или остановить. Если это необходимо, на макетной плате собирается схема, разрешающая или запрещающая прохождение импульсов от генератора.

Слева от главной платы расположен тумблер питания. **ВНИМАНИЕ! При пайке и демонтаже проводов на макетной плате, подключенной к главной плате, питание необходимо отключать!** (при этом отключать макетную плату от главной не рекомендуется во избежание ускоренного износа шлейфа).

Вид макетной платы приведен на рисунке 2.



На макетной плате расположены восемь панелек для установки микросхем. В них необходимо устанавливать микросхемы в корпусах DIP4-DIP16, необходимые для сборки схемы. Установку и извлечение микросхем необходимо производить очень аккуратно! **Недопустимо извлекать микросхему из панельки без использования инструмента (руками)**, так как при этом, как правило, гнутся и отламываются выводы микросхемы. Лучше доверить это преподавателю.

Каждая панелька имеет вокруг себя 16 контактных площадок, соединенных с выводами панельки и предназначенных для пайки к ним проводов. Перед установкой микросхемы и пайкой проводов необходимо убедиться в отсутствии короткого замыкания между соседними контактными площадками, и при его наличии – устранить его, а если это не удастся – воспользоваться другой панелькой.

Сборка схемы осуществляется путем соединения контактных площадок отрезками провода длиной, как правило, 10-20 см, при помощи пайки. Концы проводов должны быть зачищены от изоляции (примерно на 5 мм) и залужены. Пайку следует производить аккуратно, избегая замыкания соседних контактных площадок припоем и концами проводов.

Если на вход ТТЛ-микросхемы необходимо подать логический ноль, соответствующая входу контактная площадка соединяется с общей шиной (0V). При необходимости подать на вход ТТЛ единицу можно использовать следующие варианты подключения соответствующей контактной площадки:

- подключить через резистор сопротивлением 1-20 кОм к шине питания;
- подключить к выходу логической микросхемы, на который при работе схемы всегда выдается единица (например, к выходу инвертора, вход которого подключен к общей шине);
- при выполнении лабораторной работы допускается оставлять входы ТТЛ-микросхем неподключенными; в этом случае они будут восприниматься как входы, на которые подана логическая единица, но при этом необходимо понимать, что такое решение недопустимо при разработке устройств.

Не следует подключать напрямую шину питания к логическим входам. Шина питания должна быть подключена только ко входам питания микросхемы.

5 Описание стенда miniDiLaB-CIII для выполнения лабораторных работ 5 семестра

Стенд miniDiLaB-CIII предназначен для изучения основ построения устройств на базе программируемых логических интегральных схем (ПЛИС). Ниже приведено описание платы в соответствии с документацией разработчика.

Структура платы приведена на рисунке 3, а ее внешний вид – на рисунке 4.

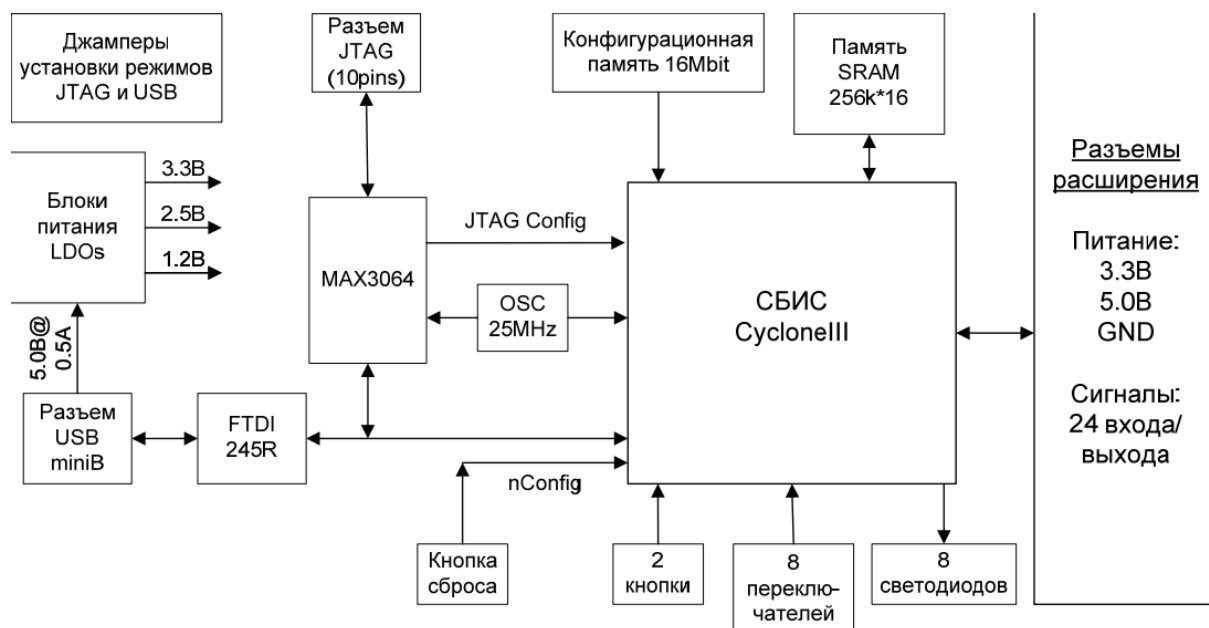


Рисунок 3 – Структура стенда miniDiLaB-СІІІ

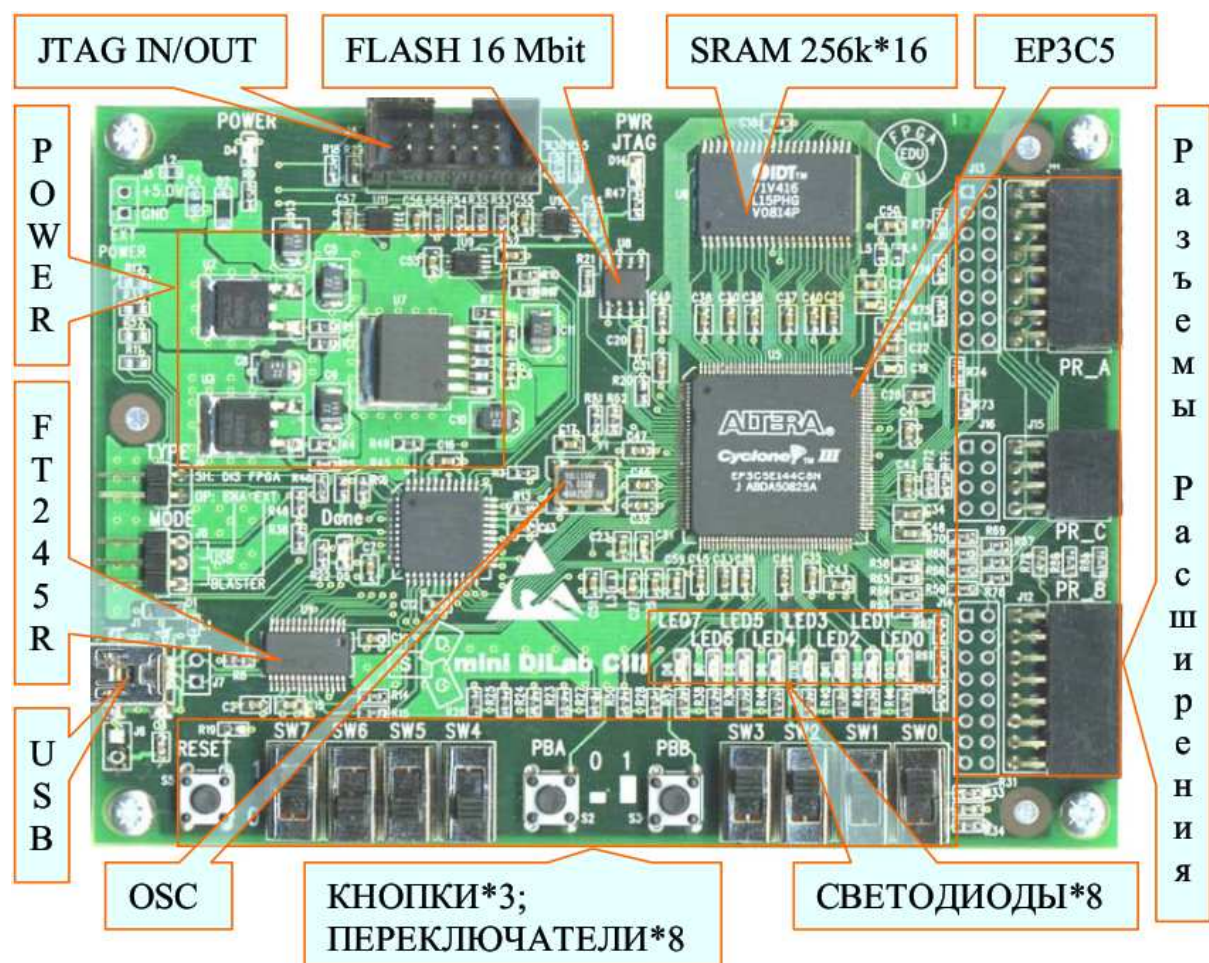


Рисунок 4 – Внешний вид платы miniDiLaB-СІІІ

В состав стенда входят следующие компоненты:

- СБИС программируемой логики Cyclone III EP3C5E144C8N;
- энергонезависимая память для хранения конфигурации;
- генератор тактовой частоты 25 МГц (OSC), подключенный к одному из выводов ПЛИС (таблица подключений пользовательских выводов СБИС приведена в приложении Б);
- восемь переключателей SW0-SW7, предназначенных для подачи на входы ПЛИС логических уровней нуля (переключатель в нижнем положении) или единицы (верхнее положение);
- две кнопки PBA и PBB, подключенные к выводам ПЛИС; при нажатии кнопки на соответствующий вывод подается логический ноль, при отпускании – единица (ВАЖНО! кнопки и переключатели не имеют схем устранениядребезга контактов, поэтому подключать их непосредственно к динамическим входам функциональных устройств не следует);
- восемь светодиодов LED0-LED7, подключенных к выводам ПЛИС и индицирующих их состояние: при низком уровне на соответствующем выводе светодиод горит, при высоком – не горит;
- микросхема статической КМОП-памяти 256к*16 бит, подключенная к ПЛИС;
- USB-интерфейс, построенный на микросхеме FTDI FT245R – служит для программирования ПЛИС и конфигурационной памяти (к нему подключен встроенный программатор); микросхема FT245R имеет очереди FIFO на прием и передачу, подключенные к ПЛИС, что позволяет использовать USB интерфейс для обмена данными между ЭВМ и устройством, реализованным в ПЛИС (использование FT245R описано в документации [11]);
- встроенный программатор с интерфейсов JTAG и внешним разъемом, позволяющим программировать ПЛИС внешним программатором или использовать встроенный программатор для конфигурирования внешний по отношению к плате устройств;
- разъемы расширения, на которые выведены линии питания 3,3 и 5 В, а также 20 свободных выводов ПЛИС и выводы, управляющие светодиодами LED0-LED3;
- источники питания на напряжения 3,3, 2,5, 1,2 В.

СБИС программируемой логики **EP3C5E144C8N** (подробное описание приведено в документации производителя [8]) имеет следующие характеристики:

- число выводов СБИС – 144;
- число пользовательских выводов (входов/выходов) – 94;
- число логических элементов – 5136;
- объем встроенной памяти – 414 кбит;
- число встроенных умножителей (разрядностью 9*9) – 46;
- число встроенных умножителей/делителей тактовых сигналов (PLL) – 2;
- объем файла конфигурации (без сжатия) – 3 Мбит.

Каждый логический элемент представляет собой 4-входовую логическую функцию, настраиваемую произвольным образом, и D-триггер с динамической синхронизацией.

Выводы ПЛИС поддерживают следующие стандарты:

- одиночные выводы: LVTTTL, LVCMOS, SSTL, HSTL, PCI, PCI-X;
- дифференциальные выводы: SSTL, HSTL, LVPECL, BLVDS, LVDS, mini-LVDS, RSDS, PPDS.

На плате также располагаются светодиод POWER, который горит, когда на плату подано питание, и кнопка RESET, которая позволяет сбросить текущую конфигурацию ПЛИС.

Программирование ПЛИС и конфигурационной памяти осуществляется встроенным программатором USB Blaster или внешним JTAG-программатором. Также встроенный программатор можно использовать для программирования внешних устройств. Режим работы платы задается джамперами TYPE и MODE (рисунок 5).

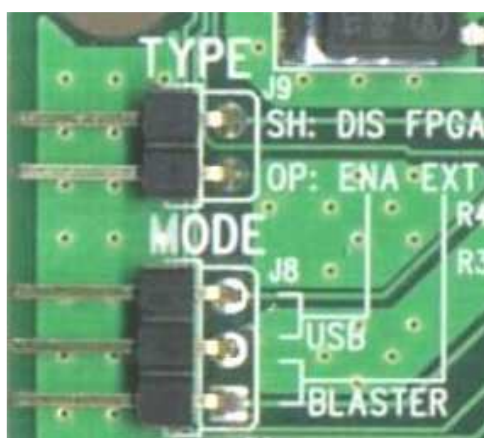


Рисунок 5 – Выбор режима работы платы

Ниже приведена таблица выбора режима.

Таблица 1 – Выбор режима работы платы

Джампер		Режим работы
MODE	TYPE	
Соединены выводы 1 и 2 (режим Blaster)	Джампер закорочен (SH)	FPGA – программирование ПЛИС на плате. USB-интерфейс используется для встроенного на плате USB Blaster. Встроенный USB Blaster осуществляет программирование ПЛИС на плате. Если в пользовательском проекте используется обмен по USB-интерфейсу, то он должен быть отключен (выводы USB_xxxx должны быть переведены в Z-состояние при единице на входе MODE_USB_N) (см. приложение Б).
	Джампер открыт (OP)	EXTERNAL – программирование внешней ПЛИС. USB-интерфейс используется для встроенного на плате USB Blaster. Встроенный USB Blaster осуществляет JTAG-программирование (через разъем J4) внешних (относительно платы) устройств, содержащих СБИС программируемой логики. Светодиод PWR JTAG будет включен, когда внешнее устройство выдаст на вывод 4 разъема J4 питающее напряжение. ПЛИС, расположенная на плате, загружает конфигурацию из конфигурационной памяти. Если в пользовательском проекте используется обмен по USB-интерфейсу, то он должен быть отключен (выводы USB_xxxx должны быть переведены в Z-состояние при единице на входе MODE_USB_N).
Соединены выводы 1 и 2 (режим USB)	Джампер закорочен (SH)	DISABLE – USB-интерфейс не используется. Программирование СБИС, расположенной на плате, может осуществляться внешним программатором, подсоединенным к разъему J4. При установке данного режима на выход 4 разъема J4 выдается питающее напряжение +3.3В, светодиод PWR JTAG включен.
	Джампер открыт (OP)	ENABLE – USB-интерфейс используется для пользовательского проекта в СБИС Cyclone III, расположенной на плате. Программирование FPGA может осуществляться программатором, подсоединенным к разъему J4. При установке данного режима на вывод 4 разъема J4 выдается питающее напряжение +3.3В, светодиод PWR JTAG включен.

6 Порядок выполнения лабораторных работ 4 семестра

В 4 семестре лабораторные работы выполняются на лабораторном стенде, описанном в разделе 4.

Сначала необходимо проанализировать задачу, выбрать способ ее решения. Если вариантом задана микросхема, необходимо найти ее описание в справочнике и изучить его. Далее необходимо составить функциональную схему устройства с использованием условно-графических обозначений (УГО) заданных микросхем. Обычно для решения задачи требуется использовать дополнительные микросхемы (как правило, логические). В этом случае необходимо выбрать микросхемы из числа имеющихся, позволяющие реализовать требуемые функции (при этом следует минимизировать число используемых корпусов).

Минимизация функции выполняется следующим образом. Сначала для функции составляется диаграмма Вейча-Карно. При этом входные переменные делятся на две группы, содержащие примерно одинаковое число разрядов. Допустимые значения одной из групп заносятся в заголовки столбцов, а значения второй группы – в заголовки строк (порядок значений определяется кодом Грея – соседние коды должны отличаться только одним разрядом).

Далее таблица заполняется требуемыми значениями функции. Если для каких-либо значений входных переменных значение функции не определено, в клетку диаграммы ставится специальный знак («*»). Далее, если необходимо получить функцию в виде конъюнктивной нормальной формы (КНФ), в диаграмме выделяются контуры, содержащие нули, а для дизъюнктивной нормальной формы (ДНФ) – единицы. Контур представляет собой прямоугольную область, каждое значение в которой – «*» или 0 для КНФ (1 для ДНФ). Количество ячеек в контуре должно быть 2^n , где n – целое. При этом контур может прерываться на краю диаграммы и продолжаться с противоположного края (диаграмма представляет собой отображение тора). При этом для КНФ каждый ноль должен быть включен хотя бы в один контур (для ДНФ – каждая единица). Следует стремиться делать как можно меньше контуров, рисуя контуры как можно большего размера.

Далее составляются логические формулы. Для каждого контура выделяются входные переменные, которые не меняют своего значения в пределах всего контура. Эти переменные составляют конъюнкты для ДНФ и дизъюнкты для КНФ. Переменные, содержащие в контуре единицу, записываются для ДНФ без инверсии, а для КНФ с инверсией. Переменные, содержащие в контуре ноль, – наоборот – записываются для КНФ без инверсии, а для ДНФ с инверсией.

Например, пусть функция задана таблицей 2.

Таблица 2

X1	X2	X3	X4	F
0	0	0	0	1
0	0	0	1	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	0
0	1	1	1	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	1	0

Таблица содержит четыре переменные. Количество комбинаций значений этих переменных составляет $2^4=16$, однако в таблице всего 10 строк. Это означает, что для шести комбинаций значения выходной функции не определены, что позволяет самостоятельно выбирать такие значения, которые наиболее удобны для получения минимальной логической формулы. Это обеспечивается включением символа «*» в контуры. Диаграмма Вейча-Карно, соответствующая таблице 2, приведена на рисунке 6.

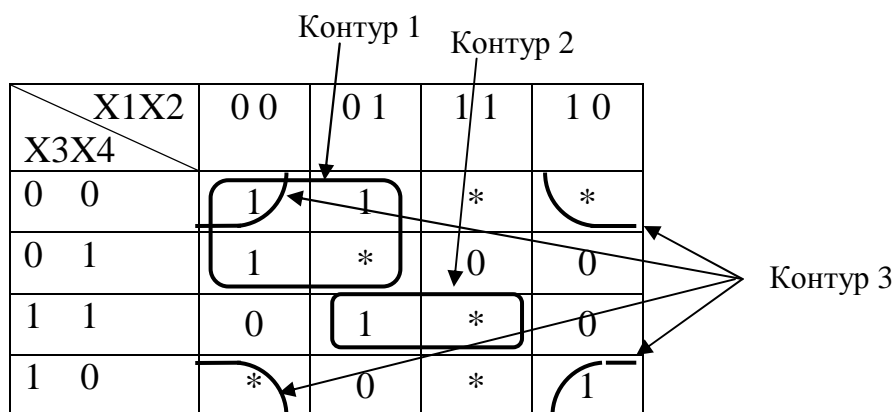


Рисунок 6 – Диаграмма Вейча-Карно

На диаграмме для построения формулы ДНФ контурами обведены единицы. Для укрупнения контуров и уменьшения их числа в контуры включены символы «*». Так получено два контура на четыре ячейки каждый и один контур на две ячейки. Далее составляются формулы конъюнктов. Для первого контура: не меняют своего значения переменные X1 и X3. Значение обеих – ноль, поэтому они включаются в формулу с инверсией: $\overline{X1} \& \overline{X3}$. Для контура 2: X2=1, X3=1, X4=1; формула: X2&X3&X4. Контур 3: X2=0, X4=0; формула: $\overline{X2} \& \overline{X4}$.

Таким образом, получена следующая формула:

$$F = \overline{X1} \cdot \overline{X3} \vee \overline{X2} \cdot \overline{X4} \vee X2 \cdot X3 \cdot X4.$$

Для составления принципиальной схемы необходимо выбрать микросхемы. Если реализовывать предложенную функцию в основном логическом базисе, потребуются: микросхема, содержащая три логических элемента «3И» (ЛИЗ), микросхема, содержащая четыре элемента «2ИЛИ» (ЛЛ1), микросхема, содержащая шесть инверторов (ЛН1). При этом остается два свободных инвертора и два элемента «2ИЛИ». Возможная схема приведена на рисунке 7.

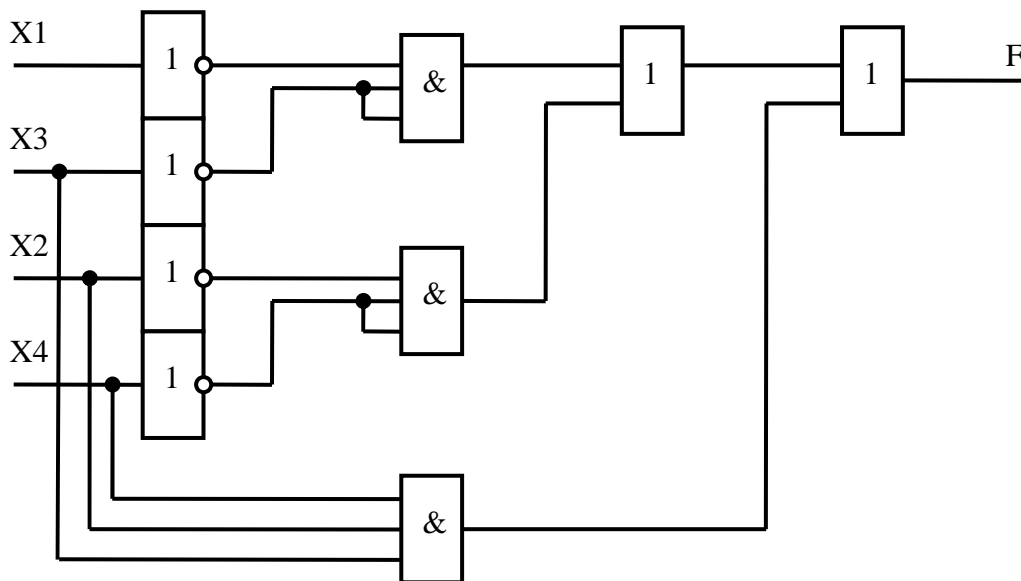


Рисунок 7 – Пример решения задачи в основном логическом базисе

Для реализации схемы потребуется три микросхемы, 24 проводника (48 паек). Задержку условно можно оценить в четыре задержки на логических элементах.

Для минимизации схемы преобразуем формулу следующим образом: $F = \overline{X1} \cdot \overline{X3} \vee \overline{X2} \cdot \overline{X4} \vee X2 \cdot X3 \cdot X4 = \overline{\overline{\overline{X1} \cdot \overline{X3} \vee \overline{X2} \cdot \overline{X4} \vee X2 \cdot X3 \cdot X4}} = \overline{\overline{\overline{X1} \cdot \overline{X3} \cdot \overline{X2} \cdot \overline{X4} \cdot X2 \cdot X3 \cdot X4}} = \overline{(X1 \vee X3) \cdot (X2 \vee X4) \cdot \overline{X2 \cdot X3 \cdot X4}}$

Функцию в таком виде можно реализовать, используя две микросхемы: ЛА4 (три элемента «3И-НЕ») и ЛЛ1 (четыре элемента «2ИЛИ»). При этом останутся свободными один элемент «3И-НЕ» и два элемента «2ИЛИ». Схема приведена на рисунке 8.

Для реализации схемы нужны две микросхемы, 15 проводников (30 паек). Задержку можно оценить как две задержки элементов.

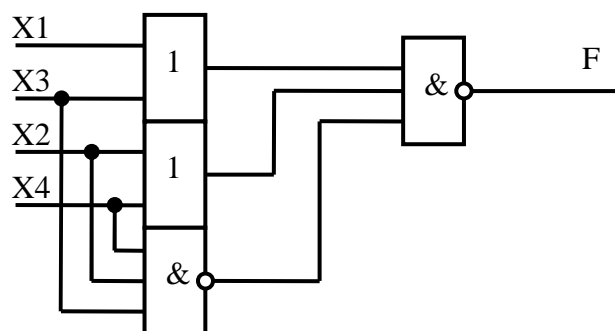


Рисунок 8 – Модифицированная функциональная схема

Функциональная схема (в соответствии с названием) демонстрирует функции элементов и их взаимосвязи. Но когда требуется собрать физическое устройство, функциональной схемы недостаточно. Нужна принципиальная схема, которая отличается наличием более полной информации: типы микросхем, номера выводов, разводка питания. Принципиальная схема, составленная в соответствии с требованиями ГОСТ [2.743-91], приведена на рисунке 9.

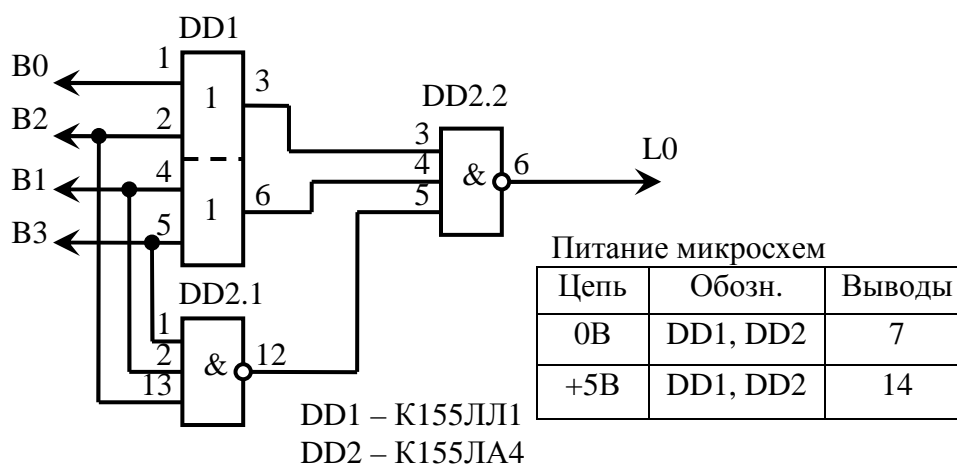


Рисунок 9 – Принципиальная схема

После составления и проверки принципиальной схемы можно приступать к сборке устройства. Для этого выбранные микросхемы (аккуратно!) размещаются в панельках монтажной платы, затем контактные площадки соединяются проводами в соответствии с принципиальной схемой. Например, необходимо соединить: площадку вывода 1 микросхемы ЛЛ1 с площадкой выхода стенда В0, площадки выводов 7 обеих микросхем с общей шиной и т.д.

Выводы микросхемы нумеруются от ключа, обозначаемого точкой или полукруглой выточкой с края корпуса микросхемы, начиная с единицы, против часовой стрелки. Пример нумерации выводов для микросхемы K155ЛЛ1 показан на рисунке 10.

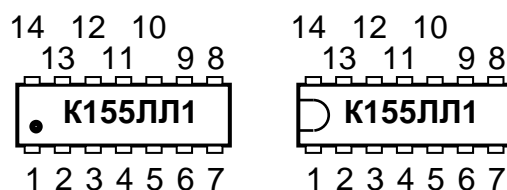


Рисунок 10 – Нумерация выводов микросхем

После сборки схемы осуществляется визуальная проверка правильности сборки. Особое внимание нужно уделять цепи +5В: нужно убедиться, что эта цепь подключена только к выводам питания микросхем и между контактными площадками этих выводов и соседними контактными площадками нет замыканий. После этого можно включать питание стенда и проверять правильность функционирования собранного устройства.

Если устройство работает неправильно, необходимо найти и устранить неисправность. Для этого при включенном питании выполняются следующие действия. Сначала необходимо измерить напряжение питания непосредственно на выводах каждой микросхемы (оно должно быть не менее 4 Вольт). Затем измеряется напряжение на логических входах микросхем по схеме, начиная со входов, подключенных к выходам стенда Вх. Логической единице должно соответствовать напряжение не ниже 2,4 В, а логическому нулю – не выше 0,8 В. Затем измеряется напряжение на выходах микросхем. Если напряжение не соответствует логическому значению, которое должно быть у данной цепи, необходимо проверить схему, убедиться в качестве соединений (возможен обрыв проводов, в том числе внутренний), убедиться в отсутствии коротких замыканий (замыкание может быть под панелькой микросхемы, тогда обнаружить его можно только тестером). Также нужно проверить тип выходов выбранной микросхемы (схемы с открытым коллектором не способны обеспечить требуемый уровень напряжения для логической единицы). При необходимости заменить микросхему.

Для измерения электрических параметров поступают следующим образом. Если нужно измерить напряжение, черный щуп вольтметра ставится на общую шину, а красный – на контактную площадку, соответствующую цепи, на которой измеряется напряжение. Для измерения тока в какой-либо цепи нужно выпаять провод, по которому течет измеряемый ток, и подключить амперметр вместо выпаянного провода.

7 Порядок выполнения лабораторных работ 5 семестра

Лабораторные работы 5 семестра предназначены для отработки схем и алгоритмов, разрабатываемых в ходе курсового проектирования по дисциплине «Схемотехника ЭВМ».

Целью курсового проектирования является разработка операционной части АЛУ ЭВМ, способного выполнить 5-6 заданных операций над данными заданного типа. В ходе курсового проектирования разрабатываются функциональные схемы (ФС) и алгоритмы выполнения отдельных операций, а затем объединенные ФС и ГСА. Для облегчения проверки корректности предложенных схем на различных вариантах исходных данных разработанные схемы реализуются с использованием ПЛИС в ходе выполнения лабораторных работ.

Структура АЛУ

Структура АЛУ, реализуемого в ходе лабораторных работ и курсового проекта, показана на рисунке 11.

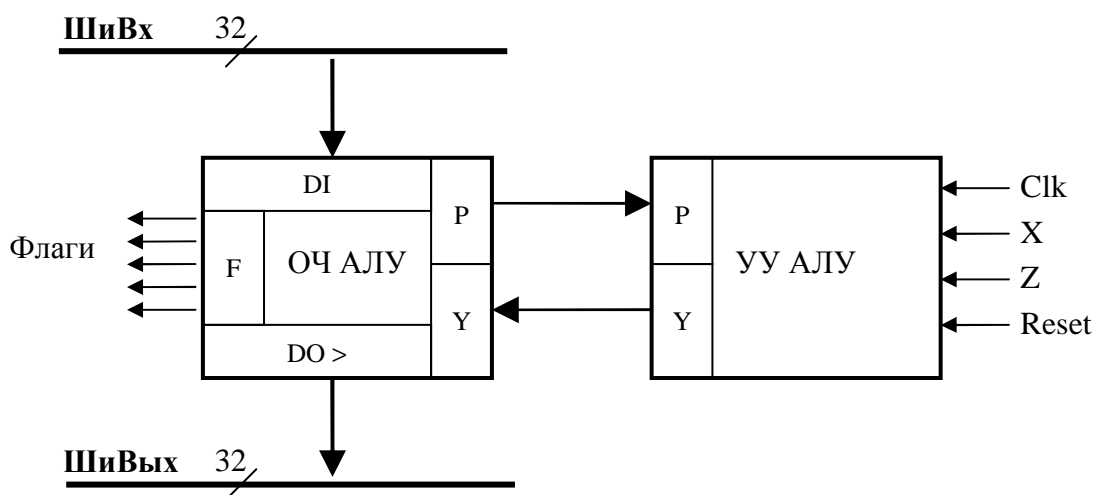


Рисунок 11 – Структура АЛУ

В процессе работы ОЧ АЛУ принимает данные с шины ШиВх, обрабатывает их и выдает на шину ШиВых. По окончании выполнения операции формируются флаги:

- готовности результата (RDY). Данный сигнал говорит о том, что операция выполнена и можно подавать сигнал Z для чтения результата;
- равенства результата нулю (ZF);
- знака результата (SF);
- переноса (CF);
- переполнения разрядной сетки (ПРС);
- деления на ноль (DIV0).

Выполняемая в ОЧ в текущем такте микрокоманда определяется набором управляющих сигналов Y , поданных в данном такте, а результат выполнения микрокоманды подается в виде набора осведомительных сигналов P в устройство управления АЛУ (УУ АЛУ).

Из центрального устройства управления (ЦУУ) подаются следующие сигналы:

- Clk – тактовый сигнал;
- X – сигнал присутствия на ШиВх операнда (в текущем такте операнд должен быть записан во внутренний регистр АЛУ);
- Z – сигнал чтения результата операции (в текущем такте результат должен выдаваться на ШиВых);
- Reset – сигнал сброса.

Временные диаграмма работы АЛУ показана на рисунке 12.

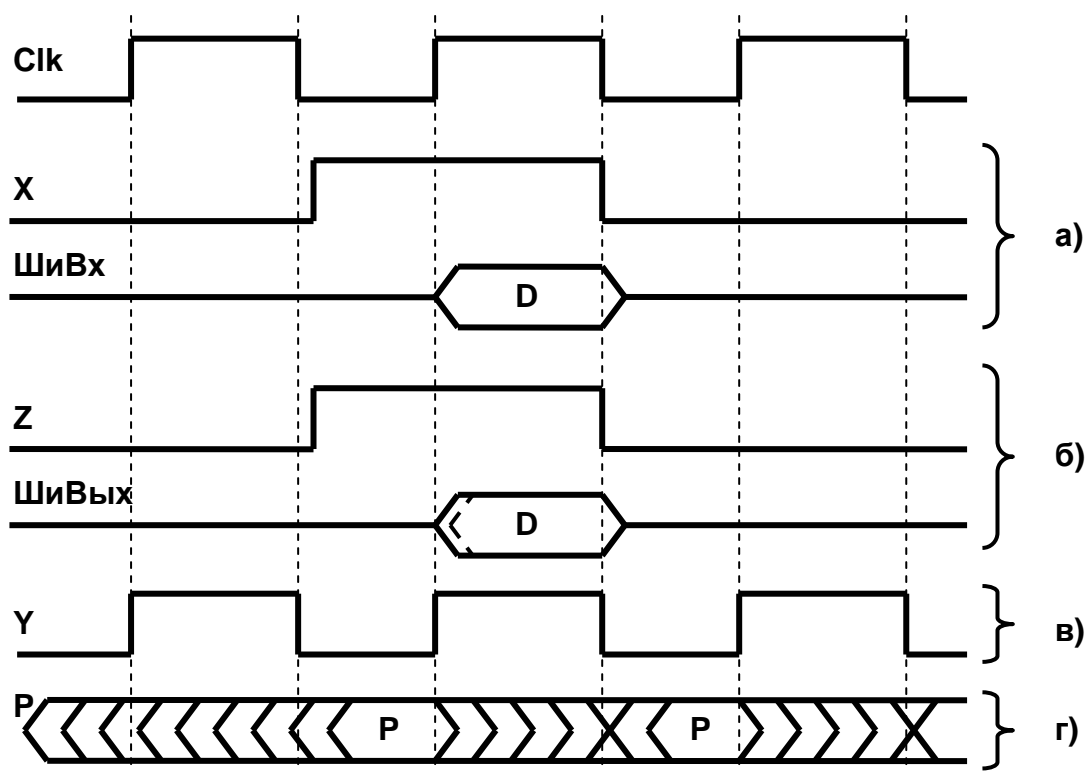


Рисунок 12 – Временные диаграммы работы АЛУ

На рисунке показаны следующие временные диаграммы:

а) поступление операнда на шину ШиВх. Перед подачей операнда после очередного спада Clk ЦУУ формирует сигнал X . В следующем такте на входную шину подается операнд. Причем операнд подается (и снимается) одновременно с сигналами Clk и Y , что делает невозможным его фиксацию в ОЧ АЛУ по подъему управляющего сигнала (фиксация должна осуществляться по уровню или по спаду управляющего сигнала);

б) выдача результата на шину ШиВых. Перед чтением результата после очередного спада Clk ЦУУ формирует сигнал Z . В следующем такте

на выходную шину должен быть выдан результат. Допускается выдача результата с небольшой задержкой относительно подъема Clk, но корректный результат должен держаться на шине до окончания Clk, так как в регистре-приемнике результат фиксируется по спаду Clk;

в) управляющие сигналы Y подаются в ОЧ АЛУ синхронно с сигналом Clk;

г) после снятия управляющих сигналов в ОЧ АЛУ начинают формироваться осведомительные сигналы R; через некоторое время их формирование заканчивается и далее они держатся до начала следующего такта. В течение этого времени управляющий автомат переходит в новое состояние и готовится к формированию управляющих сигналов следующего такта.

Лабораторная установка

Для проверки работоспособности разработанных функциональных схем эти схемы необходимо будет реализовать на базе ПЛИС. В процессе тестирования схемы потребуется в определенные моменты времени подавать на ФС исходные данные, управляющие сигналы, снимать результат и флаги. Для этого необходимо организовать не менее полусотни входных и столько же выходных линий. Плата стенда имеет всего восемь светодиодов для отображения состояния линий, восемь переключателей и две кнопки для задания значений входных сигналов, что недостаточно для тестирования ФС. Поэтому при выполнении лабораторных работ придется использовать специальный интерфейс (рисунок 13).

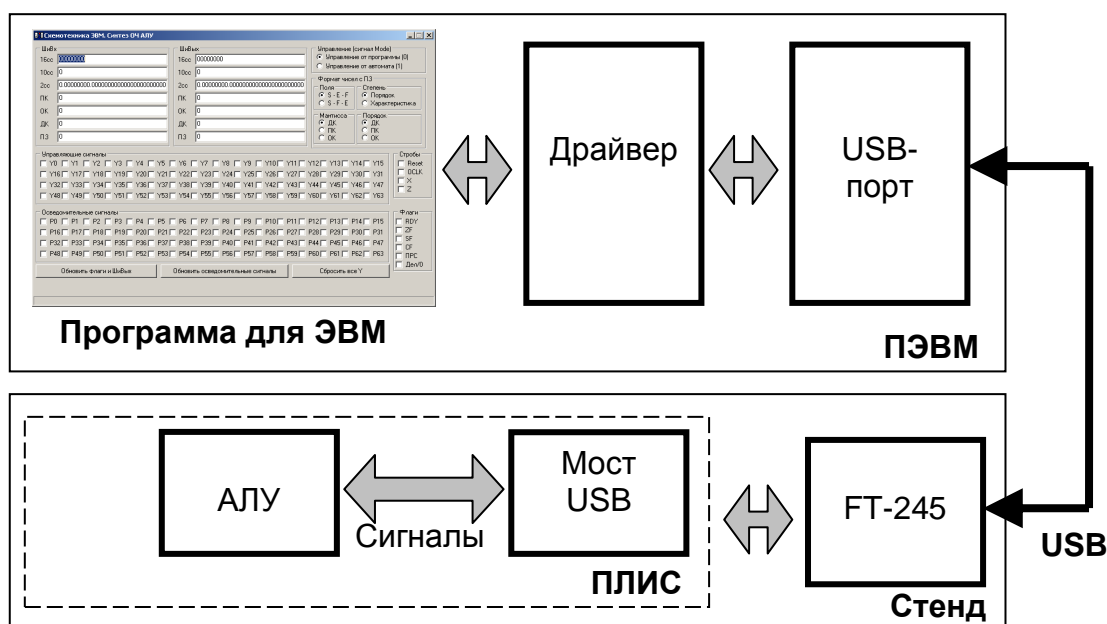


Рисунок 13 – Интерфейс лабораторной установки

Интерфейс работает следующим образом. В программе для ЭВМ пользователь задает значения сигналов, которые требуется подать на вход АЛУ. Эти значения посредством специального протокола передаются по шине USB в микросхему буфера FIFO FT-245, находящуюся на плате стенда. Из буфера данные извлекаются модулем моста USB, который программируется в ПЛИС вместе с АЛУ. Извлеченные данные модуль подает на входы АЛУ. Чтение информации с АЛУ осуществляется подобным же образом.

Таким образом, для работы с предложенным интерфейсом в конфигурацию ПЛИС (в функциональную схему) необходимо включать модуль моста USB. Данный модуль представляет собой набор исходных файлов на языке описания аппаратуры VHDL и файл УГО модуля. Обозначение модуля показано на рисунке 14.

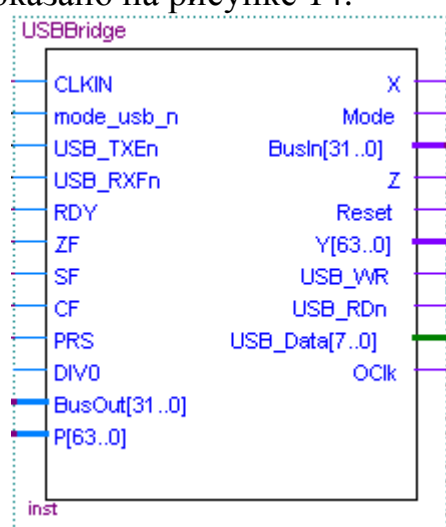


Рисунок 14 – Обозначение USB-моста в САПР Quartus II

Назначение входов и выходов модуля приведено в таблице 3.

Таблица 3 – Назначение входов и выходов модуля моста USB

Наименование сигнала	Функция
CLKIN	Входной тактовый сигнал. Может использоваться в качестве тактового сигнала АЛУ. Рекомендуется подавать на этот вход сигнал со скважностью 2 (заполнение 50%) и частотой не более 1 МГц.
mode_usb_n	Вход выбора режима работы USB. Высокое значение на этом входе запрещает работу USB. Это необходимо для перевода стенда в режим программирования ПЛИС (джампер MODE на плате стенда). Этот вход должен быть подключен к выводу 23 ПЛИС.

Продолжение таблицы 3

Наименование сигнала	Функция
USB_TXEn	Вход разрешения передачи по шине USB. Низкий уровень разрешает передачу. Должен быть подключен к выводу 42 ПЛИС
USB_RXFn	Вход разрешения чтения. Низкий уровень говорит о наличии данных в буфере и разрешает их читать. Должен быть подключен к выводу 39 ПЛИС.
RDY	Вход флага готовности. Служит для отображения состояния флага готовности в программе на ЭВМ. Подключается к соответствующему выходу ОЧ АЛУ. Рекомендуется также подключать к одному из светодиодов стенда, чтобы наглядно демонстрировать готовность результата.
ZF	Вход флага нулевого значения результата.
SF	Вход флага знака результата.
CF	Вход флага переноса.
PRS	Вход флага ПРС.
DIV0	Вход флага ошибки при делении на ноль. При отсутствии в АЛУ соответствующего выхода на этот вход подается ноль (GND).
BusOut	Вход результата (32 разряда). К этому входу подключается выход ШиВых АЛУ. Данные, поданные на этот вход, защелкиваются в регистре (входящем в состав USB-моста) по спаду сигнала Z. Далее из регистра данные могут быть переданы в программу на ЭВМ.
P	64 входа осведомительных сигналов. Подключаются к соответствующим выходам ОЧ АЛУ. На неиспользуемые входы следует подать низкий уровень (GND).
X	Выход сигнала готовности операнда. Устанавливается по команде с ЭВМ и держится в течение одного такта (см. рисунок 12). Подключается к соответствующему входу УУ АЛУ.
Mode	Выход режима работы (задается с ЭВМ). Низкий уровень – пошаговый режим (сигнал CLK изменяется в программе на ЭВМ), высокий уровень – нормальный режим (в качестве CLK используется сигнал, поданный на вход CLKIN).
BusIn	32-разрядный выход операнда. Соединяется со входом ШиВх ОЧ АЛУ. На этот выход во время действия сигнала X при высоком уровне сигнала CLK подается операнд (см. рисунок 12).

Продолжение таблицы 3

Наименование сигнала	Функция
Z	Выход сигнала готовности к приему результата. Результат должен быть подан на вход BusOut до следующего спада CLK (см. рисунок 12). Выход Z подключается в соответствующему входу УУ АЛУ.
Reset	Сигнал общего сброса АЛУ. Формируется в программе для ЭВМ. Подключается к соответствующему входу УУ АЛУ для установки начального состояния управляющего автомата.
Y	64 управляющих сигнала. Формируются в программе для ЭВМ. Подаются синхронно с CLK, поэтому изменять рекомендуется при низком уровне тактового сигнала. Подключаются к соответствующим входам ОЧ АЛУ.
USB_WR	Сигнал записи в очередь FIFO USB. Соединяется с выводом 98 ПЛИС.
USB_RDn	Сигнал чтения очереди FIFO USB. Соединяется с выводом 43 ПЛИС.
USB_Data	8-разрядная шина данных для обмена с FIFO USB. Разряды 0..7 подключаются к выводам 38, 32, 44, 50, 30, 28, 31 ПЛИС.
OClk	Тактовый сигнал (CLK) для АЛУ. Подключается к соответствующему входу УУ АЛУ (если необходимо, то и к ОЧ АЛУ). Формируется в зависимости от режима (см. MODE) либо вручную, либо генератором.

Окно программы для ЭВМ показано на рисунке 15.

В разделе ШиВх задается значение операнда. При этом 32-разрядный операнд можно вводить в одном из семи полей ввода в зависимости от формата операнда:

16сс – операнд в 16-ричной системе счисления. Например, **0ABCDE12** (соответствует двоичному коду 0000101010111100110111100010010);

10сс – операнд в десятичной системе счисления. Например, **12345** (соответствует двоичному коду 000000000000000000011000000111001);

2сс – операнд в двоичном коде;

ПК – операнд в виде десятичной дроби, представленной в прямом коде. Например, **0,75** (соответствует двоичному коду 0.11000000000000000000000000000000);

ОК – операнд в виде десятичной дроби, представленной в обратном коде. Например, **-0,25** (соответствует двоичному коду 1.10111111111111111111111111111111);

ДК – операнд в виде десятичной дроби, представленной в дополнительном коде. Например, **-0,25** (соответствует двоичному коду 1.11000000000000000000000000000000);

ПЗ – операнд в виде десятичной дроби в экспоненциальной записи. Двоичное представление зависит от настроек формата числа (см. ниже). Например, **1,23E+5** (для настроек по умолчанию – формат S-E-F, мантисса в ДК, порядок в ДК – соответствует двоичному значению 0.00010001.111100000011110000000000).

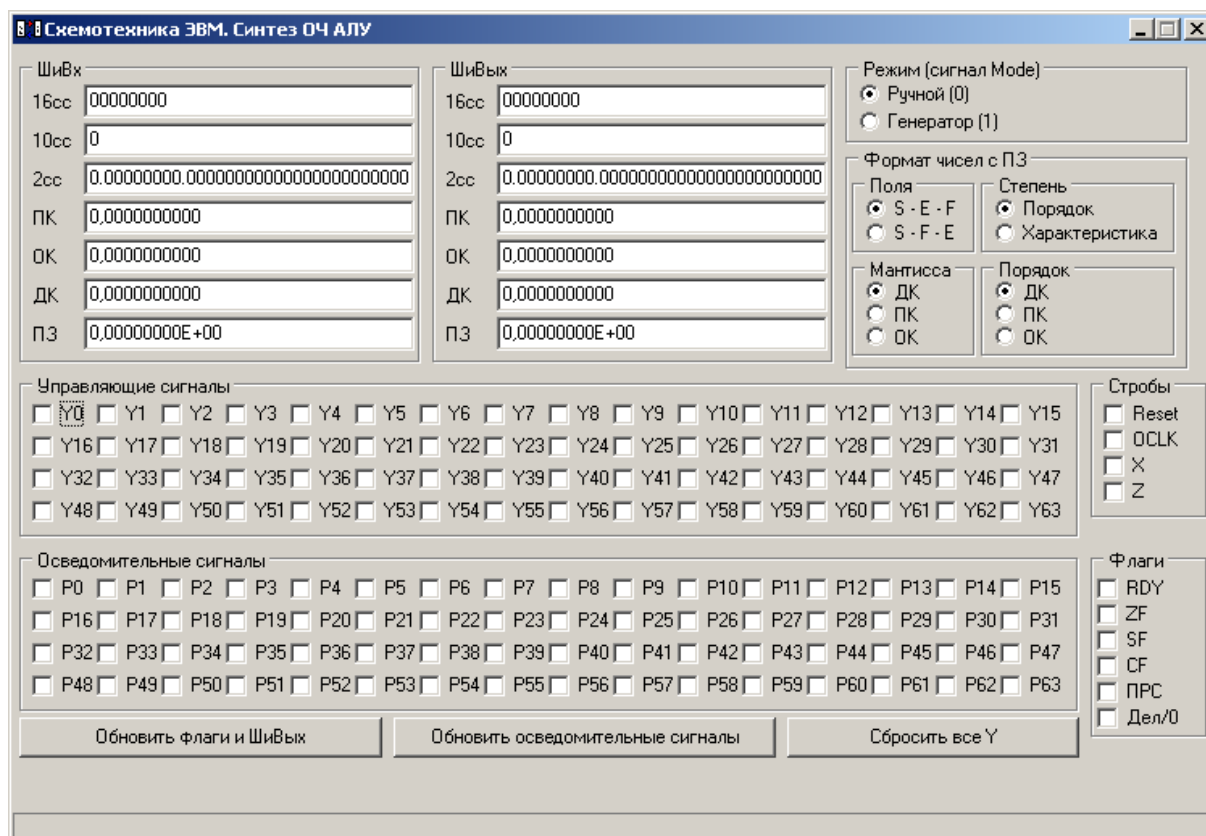


Рисунок 15 – Окно программы для ЭВМ

Раздел **ШиВх** содержит те же семь полей. В них отображается значение, считанное с ШиВх. Чтение значения производится из регистра по нажатию кнопки [Обновить флаги и ШиВх] или автоматически по сбросу сигнала Z.

Переключатель «**Режим**» предназначен для задания режима тактового сигнала. При выборе режима «Ручной» значение сигнала задается флажком OCLK. В режиме «Генератор» флажок OCLK становится недоступным. В качестве тактового сигнала используется сигнал от генератора, подаваемый на вход CLKIN моста USB.

Раздел «**Формат числа с ПЗ**» позволяет настроить представление чисел с плавающей запятой:

- переключатель «Поля» задает порядок полей: S-E-F (1 бит – знак, 8 бит – степень, 23 бита – мантисса) или S-F-E (1 бит – знак, 23 бита – мантисса, 8 бит – степень);
- переключатель «Степень» задает представление степени в виде порядка или характеристики;
- переключатель «Мантисса» задает формат представления отрицательных мантисс: в ДК, ПК или ОК.
- переключатель «Порядок» задает формат представления отрицательных порядков: ДК, ПК или ОК.

Раздел «**Управляющие сигналы**» включает 64 флажка для задания значений сигналов Y0 – Y63 в ручном (пошаговом) режиме. Нужно помнить, что сигналы выдаются синхронно с тактовым сигналом, поэтому изменять их нужно во время действия низкого уровня на выходе OCLK.

Кнопка [**Сбросить все Y**] устанавливает низкий уровень на выходах Y0 – Y63.

Раздел «**Осведомительные сигналы**» содержит 64 флажка, отражающие текущее состояние сигналов P0 – P63. Состояние флажков обновляется в ручном режиме по спаду сигнала OCLK либо по нажатию кнопки [Обновить осведомительные сигналы].

Раздел «**Стробы**» позволяет управлять выдачей сигналов Reset (сброс состояния УУ АЛУ), OCLK (тактовый сигнал – только в ручном режиме), X (сигнал из ЦУУ о готовности операнда), Z (запрос из ЦУУ на чтение результата). Состояние флажков X и Z сбрасывается автоматически по следующему спаду OCLK независимо от режима.

Флажки раздела «**Флаги**» показывают текущее состояние флагов результата выполнения операции в АЛУ. Состояние флажков обновляется по нажатию кнопки [Обновить флаги и ШиВых] или автоматически по сбросу сигнала Z.

Для контроля корректности алгоритмов требуется также устройство управления. На первом этапе его роль может выполнять пользователь, анализируя отображаемые в программе состояния осведомительных сигналов и задавая соответствующие управляющих сигналов. Структурная схема конфигурации ПЛИС для этого случая показана на рисунке 16.

В соответствии с рисунком на листе схемы размещаются разработанная пользователем ОЧ АЛУ и модуль USB-моста, затем они связываются друг с другом: выход данных АЛУ со входом данных USB-моста (BusOut), вход данных АЛУ с выходом данных USB-моста (BusIn), соответствующие входы-выходы сигналов Y, P, флагов (RDY, ZF, CF, SF, PPC, DIV0). Если в варианте задания операция деления отсутствует, выход DIV0 в АЛУ не реализуется, а на соответствующий вход USB-моста подается ноль.

На вход CLKIN необходимо подать сигнал с частотой не более 1 МГц. Для этого в схему необходимо будет включить делитель частоты (PLL), так как стенд имеет только один встроенный генератор, работающий с частотой 25 МГц.

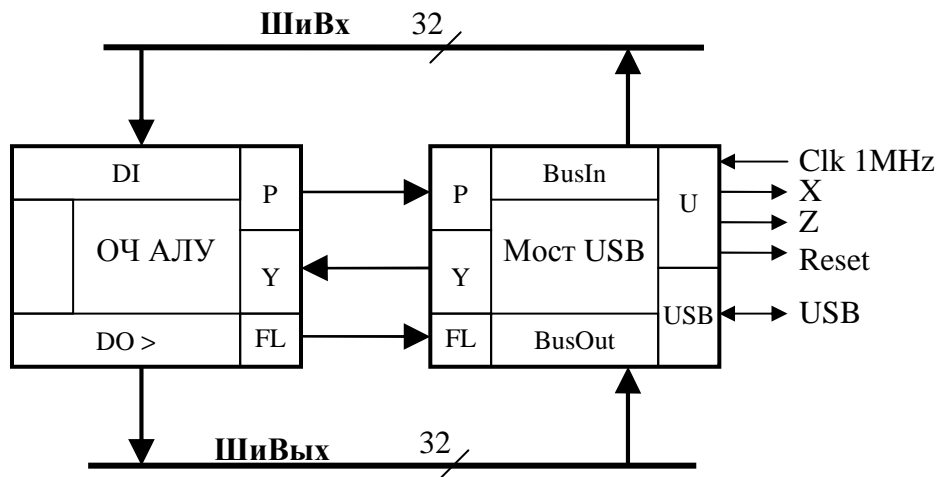


Рисунок 16 – Структурная схема АЛУ без УУ

В случае, если АЛУ вырабатывает менее 64 осведомительных сигналов, на неиспользуемые входы Р USB-моста подается низкий уровень. Неиспользуемые выходы Y, X, Z, Reset, OCLK, Mode никуда не подключаются.

Такой схемы достаточно для начальной отладки АЛУ. Но позднее, когда потребуется проверять работу АЛУ на различных вариантах исходных данных, в структуру АЛУ нужно будет включить устройство управления.

Поскольку реализация устройства управления не входит в задачи курсового проекта, в лабораторный комплекс включен готовый вариант реализации микропрограммного УУ в виде модулей на языке VHDL. Нужно отметить, что предложенный вариант УУ отличается простотой; при его разработке не преследовались цели оптимизации производительности и других характеристик: УУ предназначено только для проверки корректности разработанных алгоритмов.

Графическое обозначение УУ показано на рисунке 17.

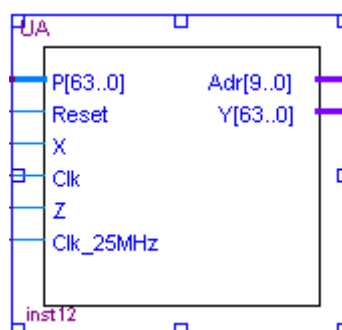


Рисунок 17 – УГО УУ АЛУ

- Р – 64 входа осведомительных сигналов;
- Reset – вход сброса для установки начального состояния управляющего автомата;
- X – вход сигнала от ЦУУ о наличии на ШиВх операнда;
- Clk – вход тактового сигнала; рекомендуется подавать тактовый сигнал со скважностью 2 (заполнением 50%) с частотой не более 1 МГц;
- Z – вход запроса результата;
- Adr – отладочный выход, содержащий 10-разрядный адрес текущей микрокоманды; рекомендуется подключать к светодиодам стенда для отслеживания хода выполнения микропрограммы;
- Y – 64 выхода управляющих сигналов.

УУ содержит следующие основные узлы:

- ПЗУ МПР – постоянная память микропрограмм; в ней хранится алгоритм работы УУ, представленный в форме микропрограмм;
- СТ – счетчик микрокоманд; хранит адрес текущей микрокоманды;
- MS КУ – мультиплексор кода условия – позволяет выбрать одно из 67 условий перехода;

- схема поразрядного «И» («&») – предназначена для синхронизации выдачи управляющих сигналов Y с тактовым сигналом Clk (представляет собой 64 двухвходовых элемента «И», на один из входов которых подается сигнал с выхода ПЗУ, на другой вход – сигнал Clk , а с выхода снимается соответствующий сигнал Y);
- БФС – блок формирования стробов; предназначен для формирования двух вспомогательных сигналов $S1$ и $S2$.

Данное УУ реализует принцип микропрограммного управления. Алгоритм управления, разработанный для реализации операции в АЛУ, представляется в виде микропрограммы, которая представляет из себя последовательность микрокоманд. Все микрокоманды имеют одинаковый формат, представленный на рисунке 19.

81	72 71	8	7	6	0
Адрес перехода (10 бит)	Набор микроопераций (управляющие сигналы Y)	Бит инверсии кода условия	Код условия (7 бит)		

Рисунок 19 – Формат микрокоманды

Микрокоманды включает следующие поля:

- «Адрес перехода» – 10-разрядный адрес (номер) микрокоманды, на которую будет сделан переход после выполнения текущей микрокоманды в случае, если заданное условие перехода будет выполнено;
- «Набор микроопераций» – 64-разрядное поле, которое определяет, какие сигналы Y будут выданы в ОЧ АЛУ при выполнении текущей микрокоманды (по высокому уровню сигнала CLK); каждому сигналу Y (с $Y0$ по $Y63$) соответствует свой бит;
- «Бит инверсии кода условия» – предназначен для инвертирования выбранного условия перехода. Если этот бит содержит ноль, то переход производится, если выбранное условие имеет значение «1», в противном случае выполняется микрокоманда со следующим адресом (номером). Если бит инверсии кода условия содержит единицу, то переход по адресу, указанному в первом поле, осуществляется, если выбранное условие имеет значение «0», в противном случае будет выполняться следующая микрокоманда;
- «Код условия» – 7-разрядное поле, предназначенное для выбора условия перехода. Коды условий приведены в таблице 4.

Таблица 4 – Коды условий перехода

Код условия	Сигнал	Описание
0	P0	Переход при истинности сигнала P0
1	P1	Переход при истинности сигнала P1
...	...	
63	P63	Переход при истинности сигнала P63
64	X	Переход при истинности сигнала X
65	Y	Переход при истинности сигнала Y
66	«0»	Нет перехода (выполняется следующая микрокоманда)

Например, для реализации цикла ожидания сигнала X нужно указать код условия 64 и установить бит ИКУ (переход по X=0), а в поле адреса указать адрес текущей микрокоманды. Для перехода к следующей микрокоманде – код 66 и ИКУ=0, а для безусловного перехода по указанному адресу – код 66 и ИКУ=1 (в первом поле указать адрес перехода).

Временная диаграмма работы УУ представлена на рисунке 20.

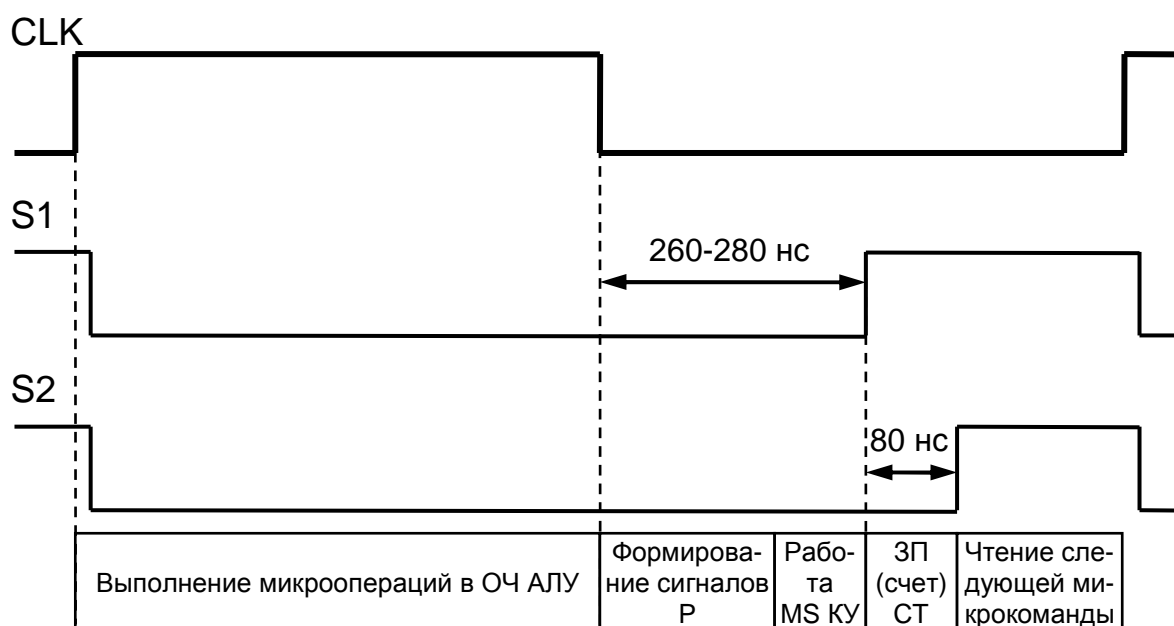


Рисунок 20 – Временная диаграмма работы УУ АЛУ

Во время действия высокого уровня на входе CLK в УУ не происходит никаких изменений. На выход выдаются сигналы Y, заданные в поле микроопераций текущей микрокоманды. После спада CLK операционной части АЛУ дается время для формирования осведомительных сигналов. Затем сформированные сигналы поступают на мультиплексор выбора условия перехода. По коду, заданному последним полем микрокоманды, выбирается одно из условий, при необходимости

инвертируется и подается на вход разрешения записи счетчика. Если на этом входе ноль, то по приходу импульса на вход С счетчика последний увеличит текущий код на единицу. Если на входе EW единица, то по фронту сигнала на входе С произойдет запись в счетчик нового адреса, который берется из поля адреса микрокоманды.

Далее в счетчике устанавливается новое состояние, которое через 80 нс фиксируется в регистре-защелке адреса, после чего из ПЗУ считывается новая микрокоманда.

Указанные задержки обеспечиваются блоком формирования стробов БФС, который формирует сигнал S1 через 260-280 нс после спада сигнала CLK и сигнал S2 через 80 нс после фронта сигнала S1. Для формирования задержек в УУ подается дополнительный тактовый сигнал с частотой 25 МГц. Если частота дополнительного тактового сигнала будет отличаться от 25 МГц, пропорционально изменятся и указанные задержки.

Структурная схема АЛУ с УУ показана на рисунке 21.

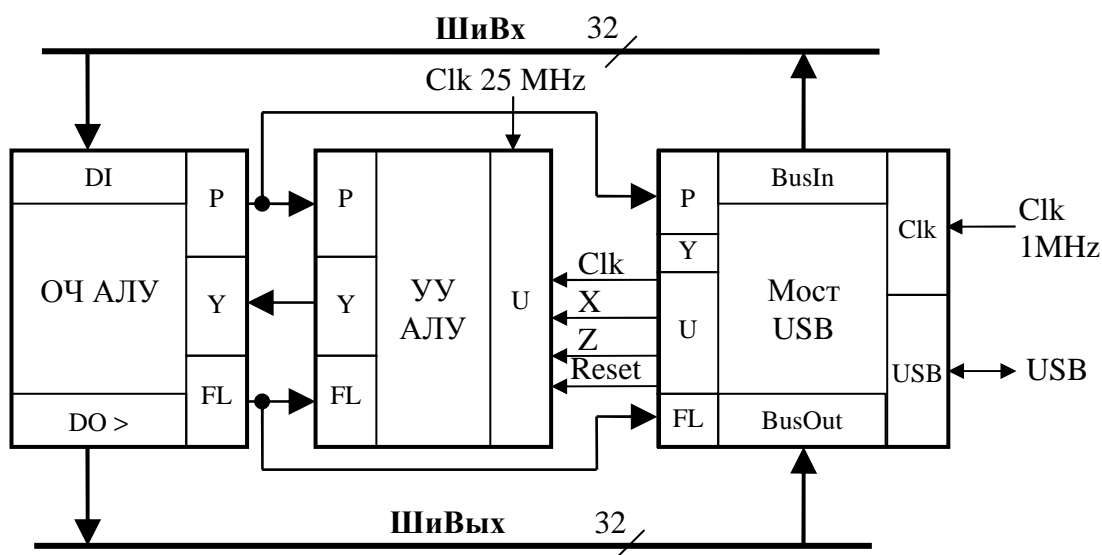


Рисунок 21 – Структурная схема АЛУ с УУ

Микропрограмма хранится в ПЗУ УУ. При компиляции проекта в САПР Quartus II данные, хранящиеся в ПЗУ, должны находиться в файле инициализации (MIF). Файл должен располагаться в каталоге проекта и называться **pzumpr.mif**.

Файл содержит 1024 82-разрядных слова в 16-ричной записи, содержащих микропрограмму. Для упрощения записи микропрограммы следует использовать программу редактор (рисунок 22).

№	Управляющие сигналы	Условие перехода	Адрес	Комментарий
0		~X	0	Цикл ожидания операнда 1
1	Y0			Чтение операнда 1
2		~X	2	Цикл ожидания операнда 2
3	Y1			Чтение операнда 2
4		~Z	4	Ожидание запроса результата
5	Y2			Выдача результата
6		1	0	Переход на начало
7				

Рисунок 22 – Окно редактора микропрограмм

Для каждой микрокоманды заполняются четыре поля:

«Управляющие сигналы» – содержит список управляющих сигналов, следующих друг за другом («Y0Y1Y3»), через пробел и/или запятую («Y0 Y1 Y3», «Y0, Y1, Y3»); если поле оставить пустым, то во время выполнения микрокоманды ни один управляющий сигнал выдан не будет;

«Условие перехода» – содержит условие, при выполнении которого будет осуществляться переход по указанному адресу (если условие не выполняется, то далее будет исполняться следующая микрокоманда). В поле можно указать один из сигналов P0-P64, X или Z (например, «P3»). При необходимости перехода по низкому уровню сигнала перед его обозначением ставится знак «~» (например, «~P5»). Единица в этом поле обозначает условный переход, а ноль или пустое поле – отсутствие перехода;

«Адрес» – десятичный адрес (номер) микрокоманды, на которую должен осуществляться переход в случае выполнения условия перехода (можно оставлять пустым, если поле условия перехода пустое или содержит ноль);

«Комментарий» – любой текст, поясняющий работу текущей микрокоманды (можно оставлять пустым).

Лабораторная работа №5

«Разработка АЛУ для выполнения операции умножения на базе ПЛИС»

Главной целью выполнения этой работы является знакомство с лабораторной установкой и последовательностью действий, выполняемых в ходе реализации АЛУ на базе ПЛИС. Для этого по шагам выполняется тестовый проект (все студенты выполняют один вариант работы).

Приведенные ниже схемы, экранные формы могут отличаться от таковых при выполнении проекта из-за различий в версиях САПР Quartus II (в том числе, различия в составе библиотек компонентов).

Ниже по шагам приведено описание порядка выполнения первой лабораторной работы пятого семестра.

1. Создайте каталог для нового проекта. Скопируйте туда файлы из каталога «--files--» (исходные файлы модулей USB-моста и устройства управления, УГО этих модулей, файл с назначением выводов ПЛИС). Запустите САПР Quartus II Web Edition. Нажмите кнопку «Create a New Project (New Project Wizard)» или выполните команду главного меню File → New Project Wizard. В появившемся окне укажите рабочий каталог проекта (первая строка) и название проекта (вторая строка). Название элемента верхнего уровня проекта (третья строка) автоматически заполняется названием проекта из второй строки (рисунок 23). Нажмите кнопку [Next].

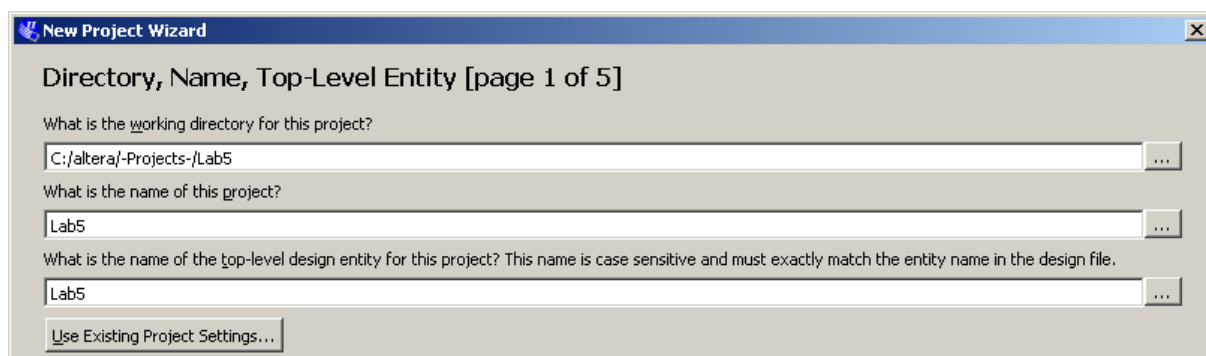


Рисунок 23 – Окно «New Project Wizard»

2. В следующем окне необходимо включить в состав проекта скопированные файлы. Для этого нажмите кнопку [...] напротив имени файла, найдите каталог проекта, войдите в него, выделите все файлы и нажмите кнопку [Открыть]. Файлы будут добавлены к проекту (рисунок 24). Нажмите кнопку [Next].

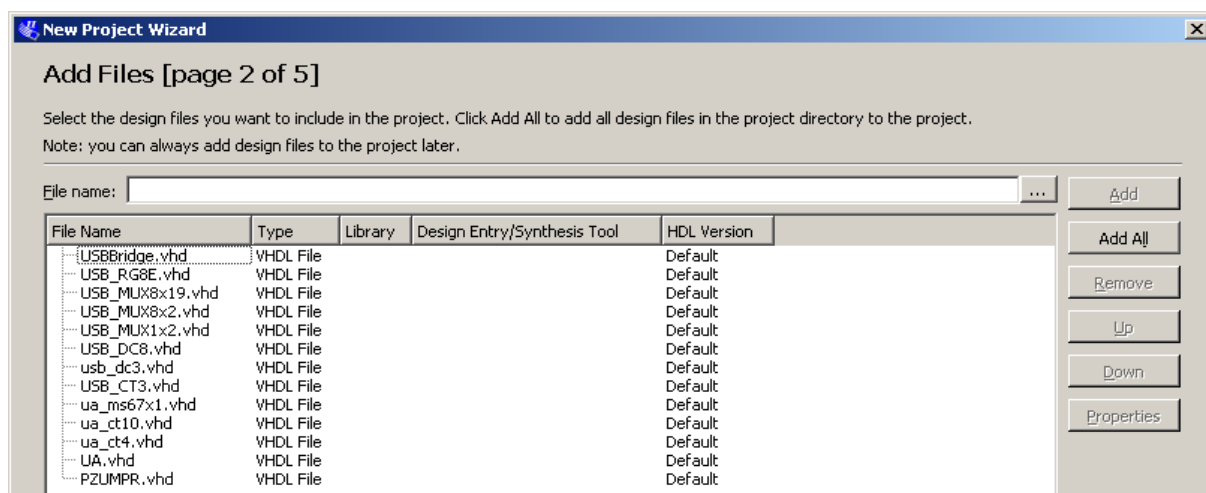


Рисунок 24 – Добавление файлов к проекту

3. Выберите ПЛИС: в списке «Family» выберите элемент «Cyclone III», в списке «Available devices» выберите ПЛИС EP3C5E144C8 (рисунок 25). Нажмите [Next].

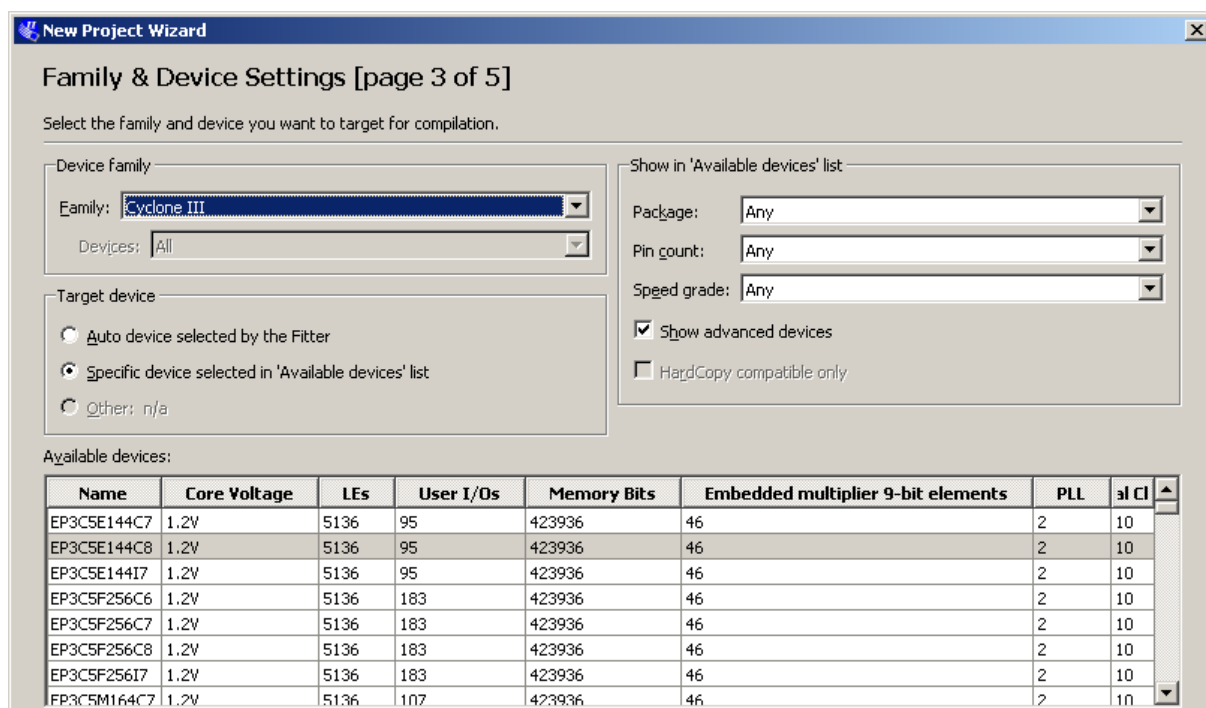


Рисунок 25 – Выбор ПЛИС

4. В окне «EDA Tool Settings» оставьте значения параметров без изменения и нажмите кнопку [Next].

5. В окне «Summary» нажмите кнопку [Finish].

5. Выполните команду главного меню Assignments → Import Assignments... В появившемся окне напротив поля ввода «File» нажмите кнопку [...], выберите файл assignments.qsf в каталоге проекта и нажмите [OK].

6. Выполните команду главного меню File → New... В появившемся окне в разделе «Design Files» выберите пункт «Block Diagram/Schematic File» и нажмите кнопку [OK]. Выберите пункт меню File → Save as..., в появившемся окне оставьте без изменения имя файла «Lab5» и нажмите кнопку [Сохранить].

7. Дважды щелкните левой кнопкой мыши в рабочей области. В появившемся окне в левой части раскройте список «Project», выберите элемент «USBBridge» и нажмите [OK]. Разместите элемент на схеме, щелкнув левой клавишей мыши в рабочем поле.

8. Снова дважды щелкните левой клавишей мыши в свободном месте рабочей области. Раскройте в левой части окна второй список (он может называться, например, «C:/altera/10.1/quartus/libraries/»), в нем – список «primitives», затем «pin». Выберите элемент «input», нажмите [OK] и разместите элемент в рабочей области. Дважды щелкните на имени

входа «pin_name» и замените имя на «clk_25mhz». Аналогичным образом поместите на схему элементы «output», «bidir», «not» (из списка «logic»), «gnd» (из списка «other»).

9. Дважды щелкните левой кнопкой мыши в рабочей области. В появившемся окне нажмите кнопку [MegaWizard Plug-In Manager]. В появившемся окне выберите пункт «Create a new custom megafunction variation» и нажмите [Next].

10. В появившемся окне в списке слева выберите элемент I/O → ALTPLL, справа, не удаляя путь, введите имя файла «PLL1» (рисунок 26). Нажмите кнопку [Next].

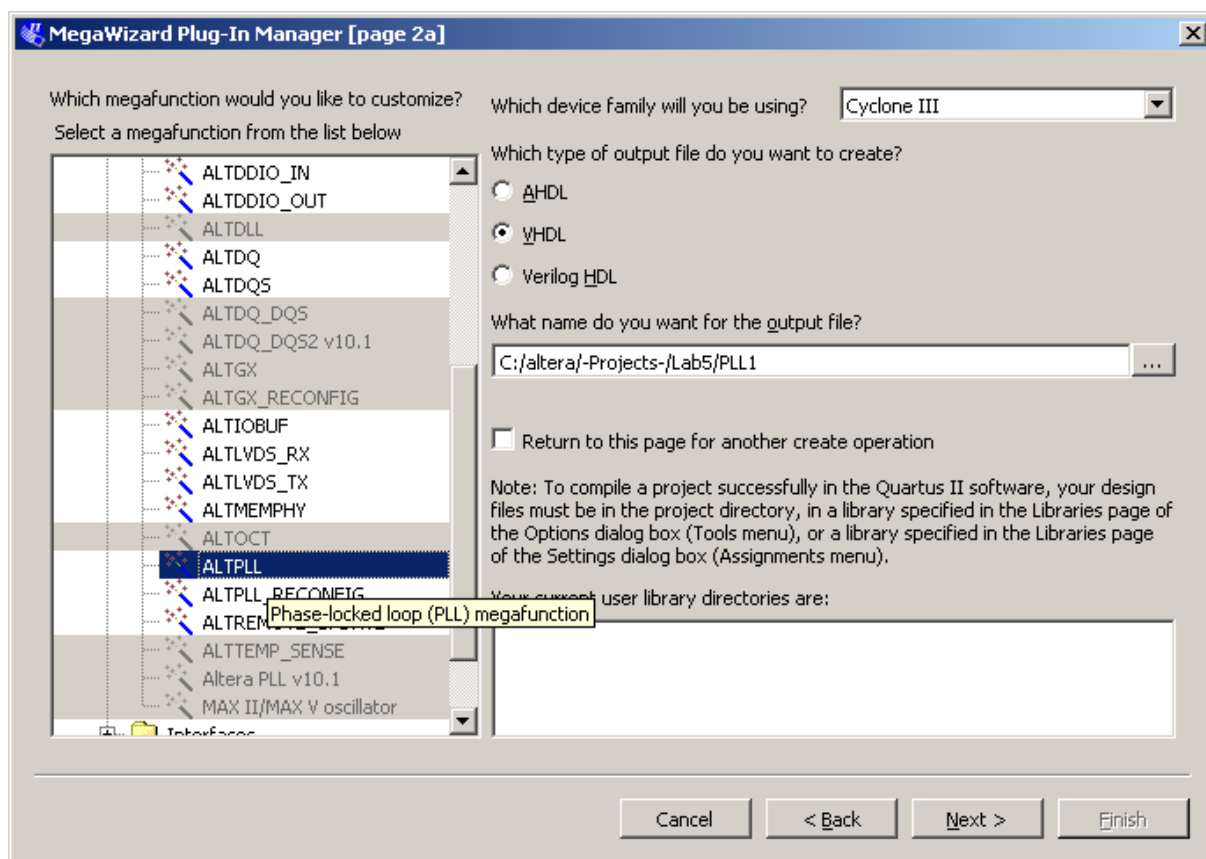


Рисунок 26 – Создание элемента ALTPLL

11. В появившемся окне в разделе «General» в поле ввода «What is the frequency of the inclk0 input?» укажите значение 25, оставив единицу измерения «MHz». Не меняя остальных параметров, нажмите [Next].

12. В следующем окне снимите флажки «Create an `areset` input to asynchronously reset the PLL» и «Create `locked` output». Нажмите [Next]. В следующих трех окнах, не меняя никаких значений, нажимайте [Next].

13. В окне этапа «c0 – Core/External Output Clock» установите переключатель «Enter output clock frequency» и в соседнем поле ввода укажите значение 1, оставив единицу измерения без изменения («MHz»)

MegaWizard Plug-In Manager [page 8 of 14]

inclk0 frequency: 25.000 MHz
Operation Mode: Normal

Clk	Ratio	Ph (deg)	DC (%)
c0	1/25	0.00	50.00

Cyclone III

☒ Use this clock

Clock Tap Settings

☒ Enter output clock frequency:
☐ Enter output clock parameters:

Clock multiplication factor

Clock division factor

Clock phase shift

Clock duty cycle (%)

Requested Settings

1 MHz 1.000000

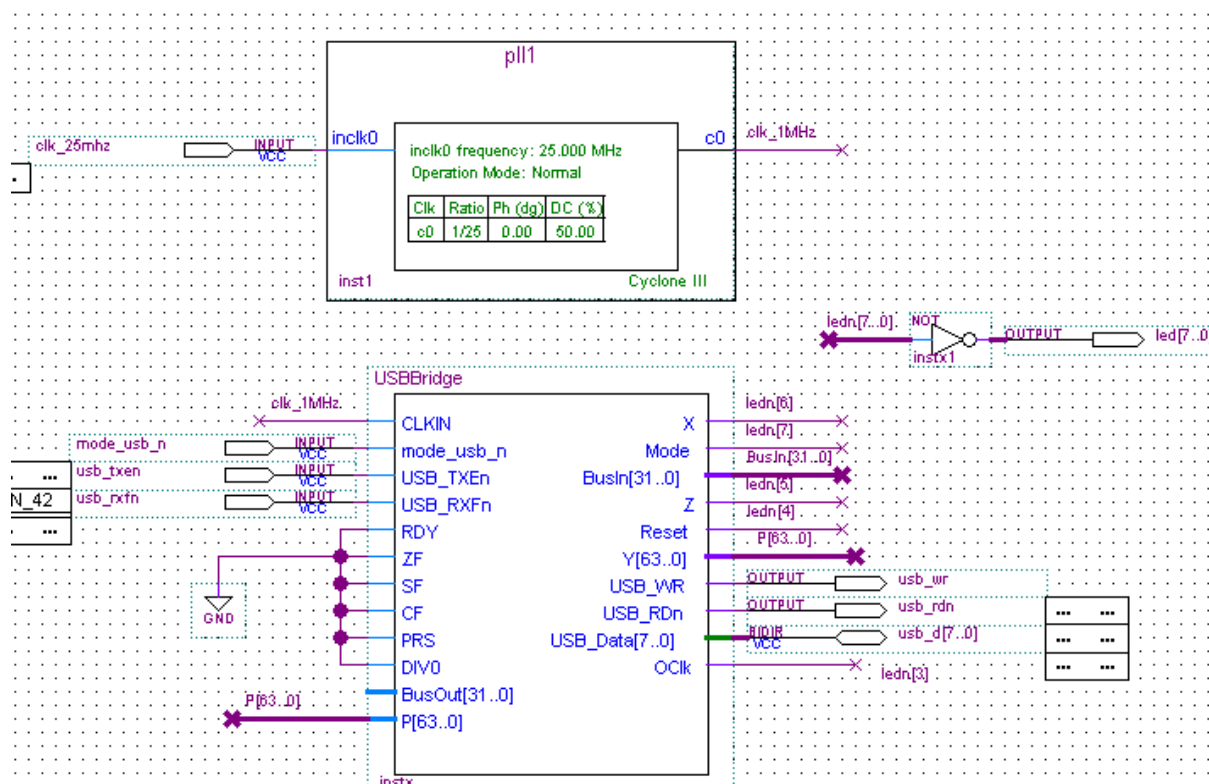
1 1


0.00 deg 0.00

50.00 50.00

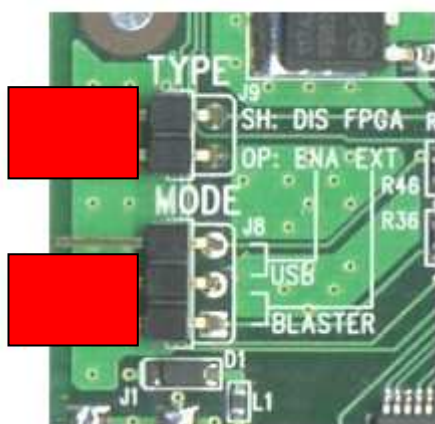
<< Copy

14. В окне «Symbol» нажмите [OK] и разместите полученный элемент на схеме.
15. Соберите схему, показанную на рисунке 28.

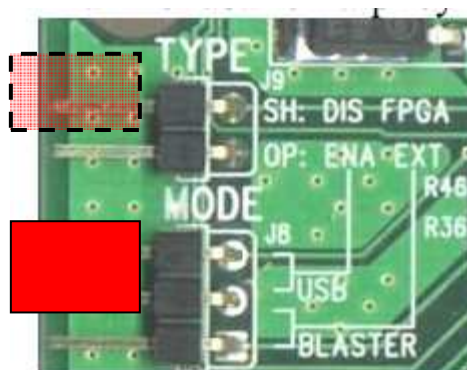


16. Нажмите кнопку  («Start Compilation»). Дождитесь завершения компиляции проекта.

17. Подключите плату стенда к ЭВМ. Установите переключку «TYPE». Переключку «MODE» переместите в положение, ближнее к разъему USB (режим «BLASTER») (см. рисунок 29 а). Нажмите кнопку («Programmer»). В появившемся окне должен быть выбран программатор USB-Blaster (если нет, выберите его вручную, нажав кнопку [Hardware Setup...]). Нажмите кнопку [Start]. Дождитесь завершения процесса программирования и закройте окно программатора (рисунок 30).



а) Положение переключек для программирования ПЛИС



б) Положение переключек для обмена данными по USB

Рисунок 29 – Расположение переключек на стенде

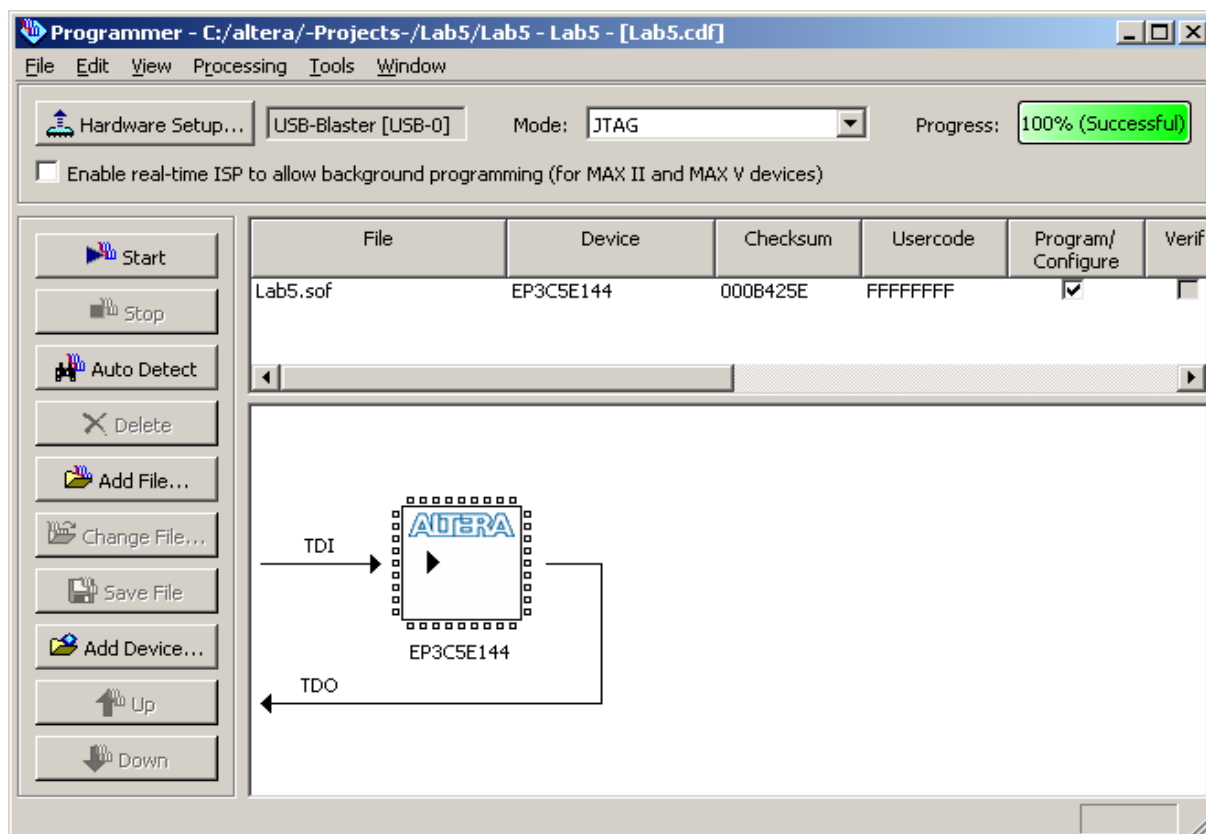


Рисунок 30 – Окно программатора

18. Переставьте переключики на стенде в положение, показанное на рисунке 29 б). Чтобы джампер «TYPE» не потерялся, можно одеть его на одну ножку (показано на рисунке пунктиром).

19. Запустите программу SchemKP.exe. Проверьте работу стенда и ПО (по завершении проверок закройте программу SchemKP):

- установите флажок «Reset». Убедитесь, что загорелся светодиод led4. Снимите флажок. Убедитесь, что светодиод погас;
- проверьте, что светодиод led3 подобным же образом реагирует на флажок OCLK;
- установите режим «Генератор». Должны загореться светодиоды led7 и вполнакала led3; установите режим «Ручной». Светодиоды должны погаснуть;
- установите флажок X. Дважды установите и снимите флажок OCLK. От момента первого снятия OCLK до второго снятия OCLK должен гореть светодиод led6 – в соответствии с временной диаграммой, показанной на рисунке 12;
- аналогичным образом проверьте флажок Z (должен загореться светодиод led5);
- установите несколько флажков управляющих сигналов, например, как показано на рисунке 31. Нажмите кнопку [Обновить осведомительные сигналы] и убедитесь в том, что все флажки Р сняты. Установите флажок OCLK и снова нажмите кнопку [Обновить осведомительные сигналы]. Флажки раздела «Осведомительные сигналы» должны установиться в том же порядке, что и флажки раздела «Управляющие сигналы» (так как по схеме выходы управляющих сигналов мы соединили со входами осведомительных). Снимите флажок OCLK. Флажки осведомительных сигналов должны сброситься.

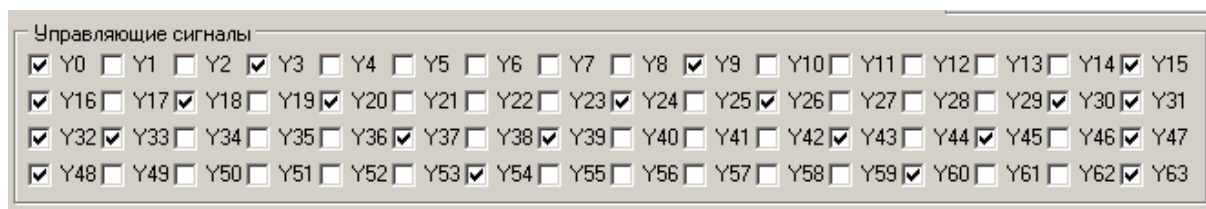


Рисунок 31 – Вариант расстановки управляющих сигналов

Далее нужно выполнить реализацию схемы умножения с младших разрядов множителя и сдвигом суммы частичных произведений вправо для двоичных чисел с фиксированной запятой в прямом коде.

Функциональная схема, разработанная для решения предложенной задачи, показана на рисунке 32.

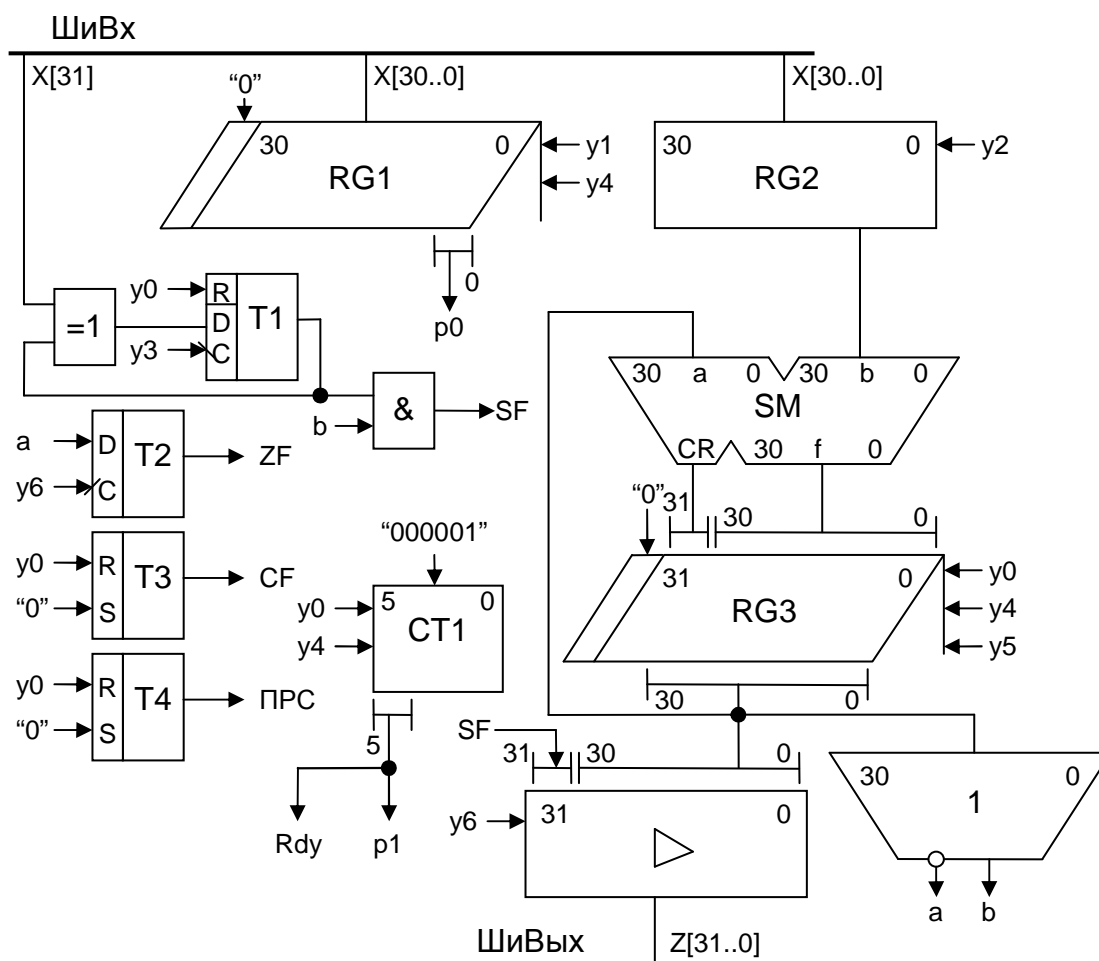


Рисунок 32 – ФС реализации операции умножения

Схема содержит регистр множителя RG1, регистр множимого RG2, регистр частичных произведений RG3, триггеры T1-T5 для хранения значений флагов, счетчик циклов суммирования CT1, схема проверки результата на ноль.

В таблице 5 приведены управляющие, а в таблице 6 осведомительные сигналы, используемые схемой.

Таблица 5 – Управляющие сигналы

Обозначение	Микрокоманда	Микрооперации
y0	Инициализация	T1 := 0 T3 := 0 T4 := 0 CT1 := 0 RG3 := 0
y1	Запись операнда в RG1	RG1 := X[30..0] (по спаду y1)
y2	Запись операнда в RG2	RG2 := X[30..0] (по спаду y2)
y3	Запись знака в T1	T1 := T1 \oplus X[31] (по спаду y3)
y4	Сдвиги	RG1 := 0.R1(RG1) RG3 := 0.R1(RG3) CT1 := CT1 + 1
y5	Запись суммы в RG3	RG3 := SMcr.CMf (по спаду y5)
y6	Выдача результата	Z = SF.RG3[30..0] T2 := ZF

Таблица 6 – Осведомительные сигналы

Обозначение	Событие	Условие
p0	Младший разряд множителя единица	RG1[0] = 1
p1	Умножение завершено	CT1[5] = 1

В соответствии с приведенной на рисунке 32 функциональной схемой строится схема в САПР Quartus II.

20. Создайте в текущем проекте новый файл схемы (см. шаг 6). Сохраните файл под именем OA.bdf.

21. Используя MegaWizard Plug-In Manager, создайте 31-разрядный сдвиговый регистр (функция LPM_SHIFTREG, направление сдвига (direction) – (Rght), выходы (outputs) – выходы данных (Data output), дополнительные входы (optional inputs) – вход последовательного занесения (Serial shift data input) и вход параллельного занесения (Parallel data input). Разместите на листе схемы два экземпляра регистра (для экономии времени этот регистр будем использовать и в качестве RG1, и в качестве RG2).

22. Аналогичным образом создайте 32-разрядный сдвиговый регистр (все параметры по аналогии с предыдущим шагом, кроме разрядности (32) и флажка входа асинхронного сброса (установить)). Разместите на схеме один экземпляр регистра.

23. С помощью того же мастера создайте и разместите на схеме 6-разрядный счетчик (LPM_COUNTER, 6 бит, асинхронные входы: вход установки в значение 1 – см. рисунок 33), 31-разрядный сумматор (LPM_ADD_SUB, 31 бит, выход переноса – carry output), 31-входовый элемент «ИЛИ».

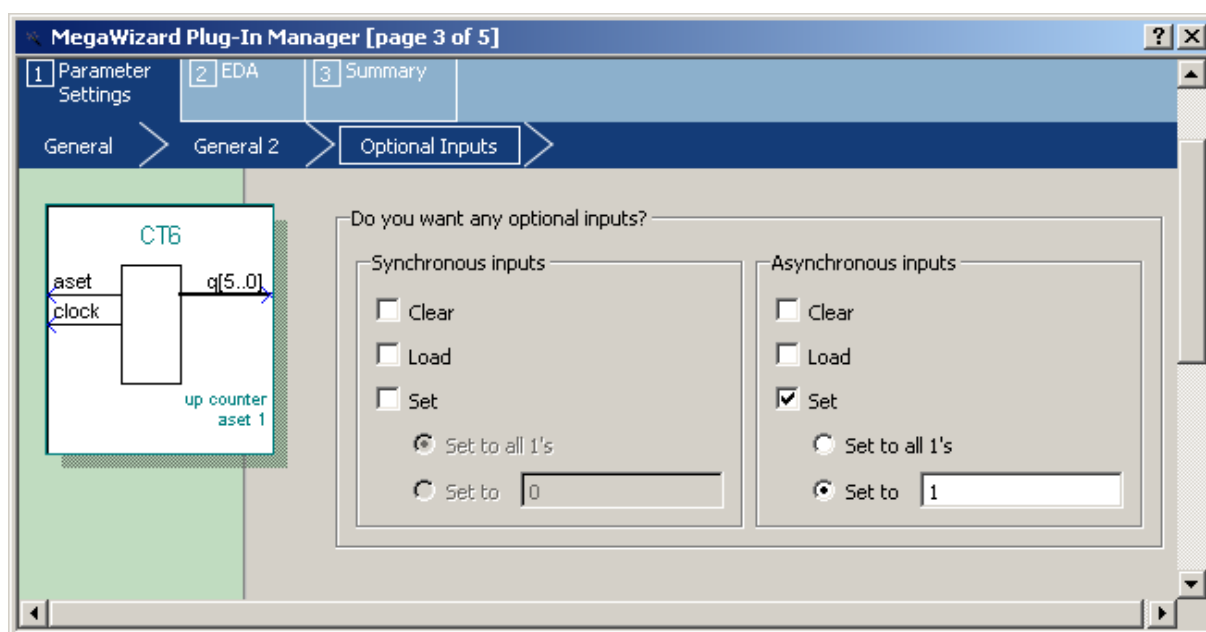
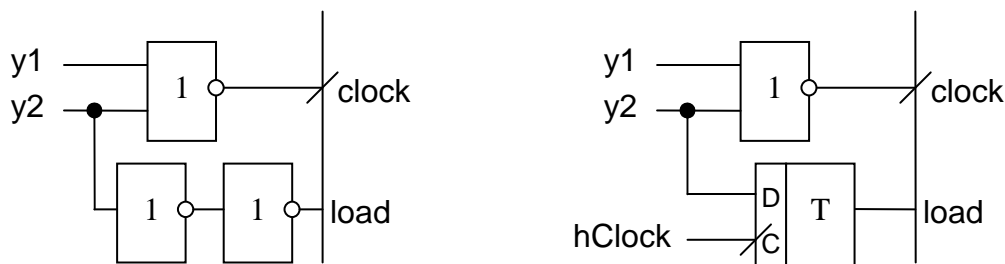


Рисунок 33 – Настройка параметров счетчика

Некоторую проблему представляет разница в управлении регистрами, заложенном в функциональной схеме и реализацией этого управления в ПЛИС (это касается сдвиговых регистров). В функциональной схеме запись и сдвиг осуществляются разными сигналами *y*: подъем, а затем спад одного из этих сигналов (при низком уровне второго) приводит к сдвигу, а подъем и спад второго сигнала (при низком уровне первого) – к записи. Реализация регистра в ПЛИС отличается: у регистра есть тактовый вход (clock) и вход загрузки. Обе микрооперации производятся по фронту тактового сигнала, а сама микрооперация определяется состоянием входа load **во время фронта сигнала clock** (низкий уровень – сдвиг, высокий – загрузка). В данном случае без искусственной задержки не обойтись. И здесь проявляется разница в создании принципиальной схемы из логических микросхем и разработке схемы для ПЛИС: если в первом случае сигнал load, скорее всего, достаточно было задержать двумя инверторами (рисунок 34 а), то во втором случае такой вариант не проходит из-за оптимизаций схемы (два инвертора, идущих подряд, компилятором будут просто исключены).

Задержку придется сделать, используя дополнительный триггер и тактовый сигнал высокой частоты (рисунок 34 б).

Еще раз обратите внимание: **создание схем задержек (и оценка задержек!)** СУЩЕСТВЕННО различаются в схемах для ПЛИС и в схемах на основе логических микросхем!



а) Схема на основе ИМС

б) Схема для ПЛИС

y1 – сдвиг

y2 – загрузка

hClock – тактовый сигнал высокой частоты

Рисунок 34 – Организация задержек

24. Добавьте на лист схемы входы и выходы: входы данных BusIn[31..0], входы управляющих сигналов Y[6..0], выходы данных BusOut[31..0], выходы осведомительных сигналов P[1..0], выходы флагов ZF, CF, SF, PRS, RDY, дополнительный тактовый вход hClock.

25. Соберите схему, показанную на рисунке 35.

26. Создайте графическое обозначение (символ) для текущей схемы (OA.bdf): выполните команду File → Create/Update → Create Symbol Files for Current File. Нажмите кнопку «Сохранить» в диалоге сохранения файла.

27. Перейдите на вкладку Lab5.bdf и добавьте на схему только что созданный символ: сделайте двойной щелчок левой кнопкой мыши в свободном месте, в появившемся окне слева раскройте список Project, выберите OA и нажмите [OK].

28. Измените схему, как показано на рисунке 36 (на рисунке видна только нижняя часть схемы, без PLL).

29. Скомпилируйте проект и запрограммируйте ПЛИС (не забудьте перед программированием переставить джамперы на плате стенда в положение, показанное на рисунке 29 а), а после завершения программирования закрыть окно программатора и переставить джамперы в положение, показанное на рисунке 29 б). Если впоследствии программа для ЭВМ будет отображать некорректные сведения, может потребоваться перепрошить ПЛИС.

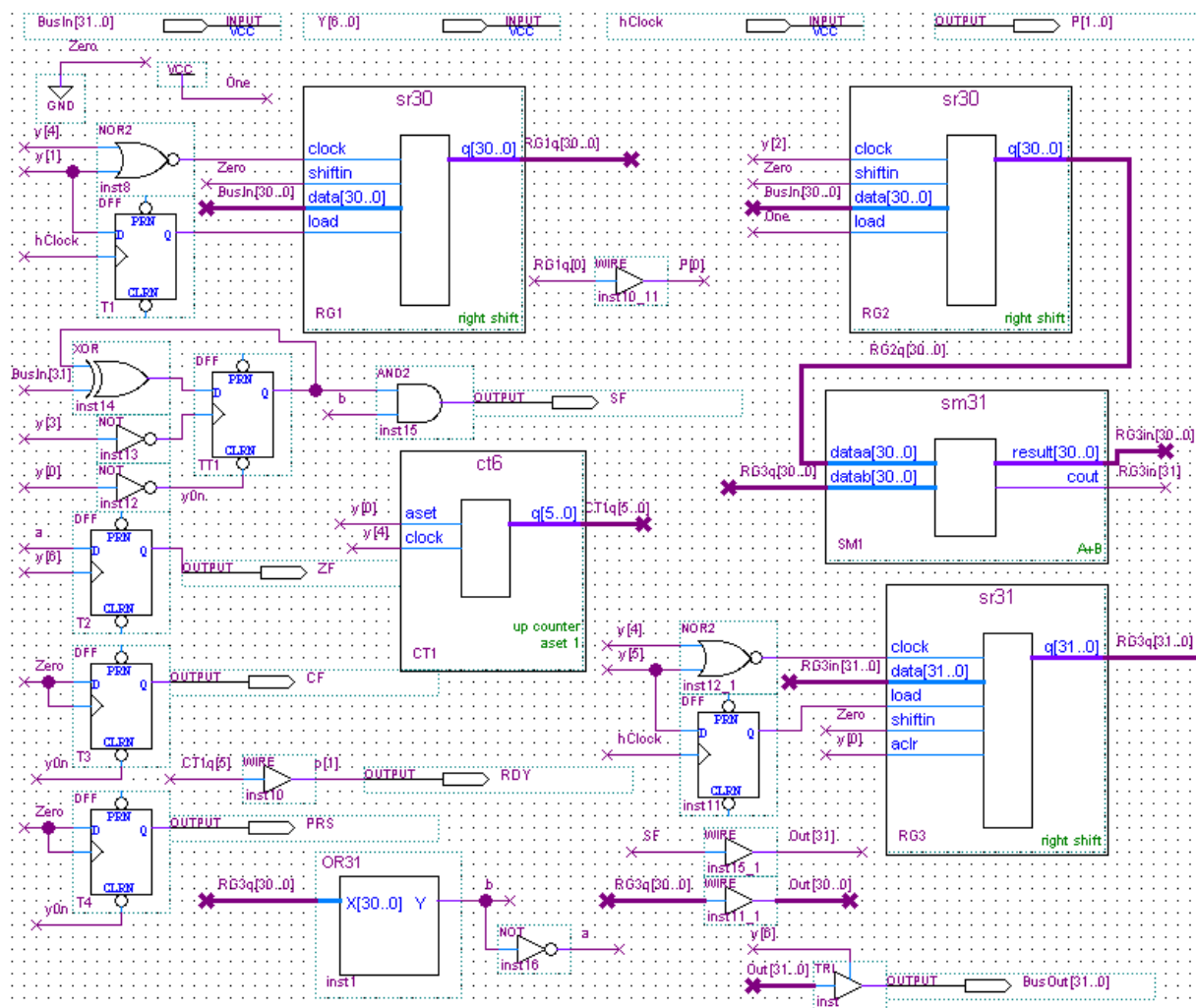


Рисунок 35 – Схема ОЧ АЛУ в САПР Quartus II

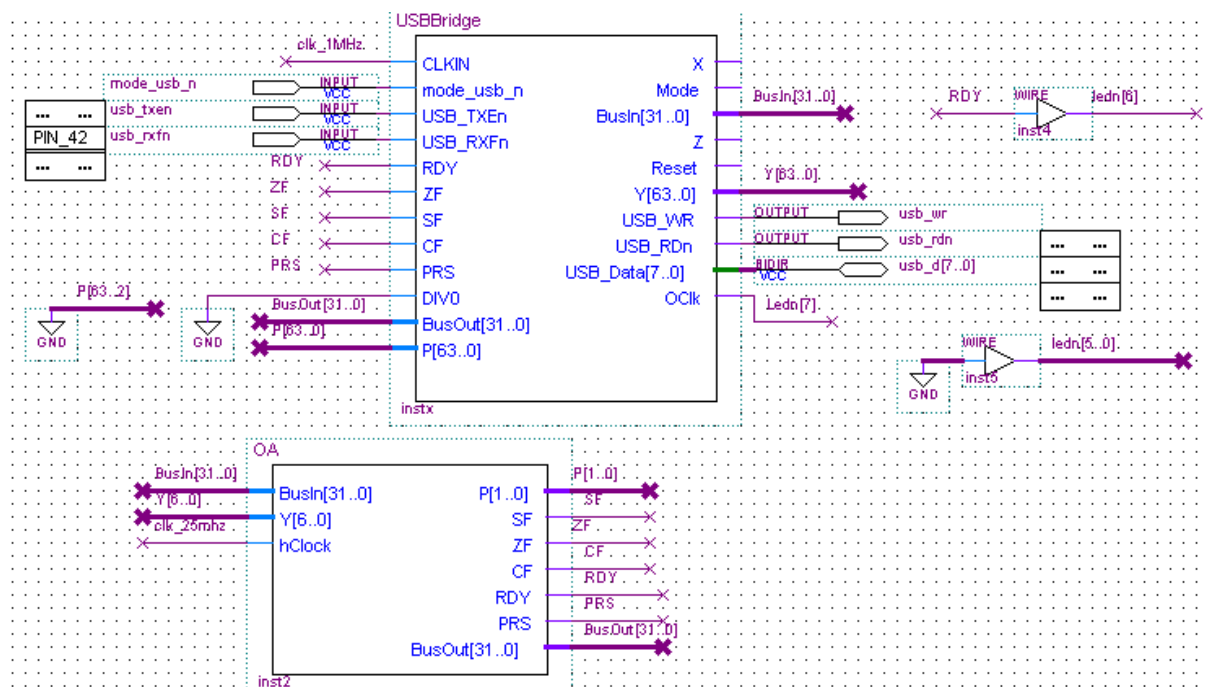


Рисунок 36 – Схема Lab5.bdf

30. Запустите программу SchemKP.exe и проверьте работу схемы на любом варианте исходных данных. Допустим, нужно умножить 0,5 на -0,75. Выполните следующие действия:

- в поле ПК раздела ШиВх введите число -0,75 и нажмите [Enter];
- установите флажок Y0 и подайте тактовый импульс (установив и сбросив флажок OCLK);
- сбросьте флажок Y0, установите флажок X и подайте тактовый импульс;
- установите флажки Y1 и Y3 и подайте тактовый импульс;
- сбросьте флажки Y;
- в поле ПК раздела ШиВх введите число 0,5 и нажмите [Enter];
- установите флажки Y2 и Y3 и подайте тактовый импульс;
- сбросьте флажки Y2 и Y3, установите Y4 и подавайте тактовые импульсы, пока не появится флажок P0 (29 раз);
- снимите Y4, установите Y5 и подайте тактовый импульс;
- снимите Y5, установите Y4 и подайте тактовый импульс;
- если флажок P0 остается (должен остаться), снимите Y4, установите Y5 и подайте тактовый импульс;
- снимите Y5, установите Y4 и подайте тактовый импульс; должен появиться флажок P1 (умножение завершено) и загореться светодиод led6. Нажмите кнопку [Обновить флаги и ШиВых] и убедитесь в том, что флажок RDY установлен;
- сбросьте Y4, установите Z и подайте тактовый импульс;
- установите Y6 и подайте тактовый импульс.

После выполнения последнего действия обновляется результат. Проверьте, что поле ПК результата содержит значение -0,375, а из флагов установлены RDY и SF.

Процесс проверки вручную достаточно сложен, поэтому необходимо будет добавить в схему устройство управления и на основе алгоритма составить микропрограмму.

ГСА выполнения операции умножения приведена на рисунке 37.

В соответствии с предложенной ГСА составим микропрограмму для устройства управления (таблица 7).

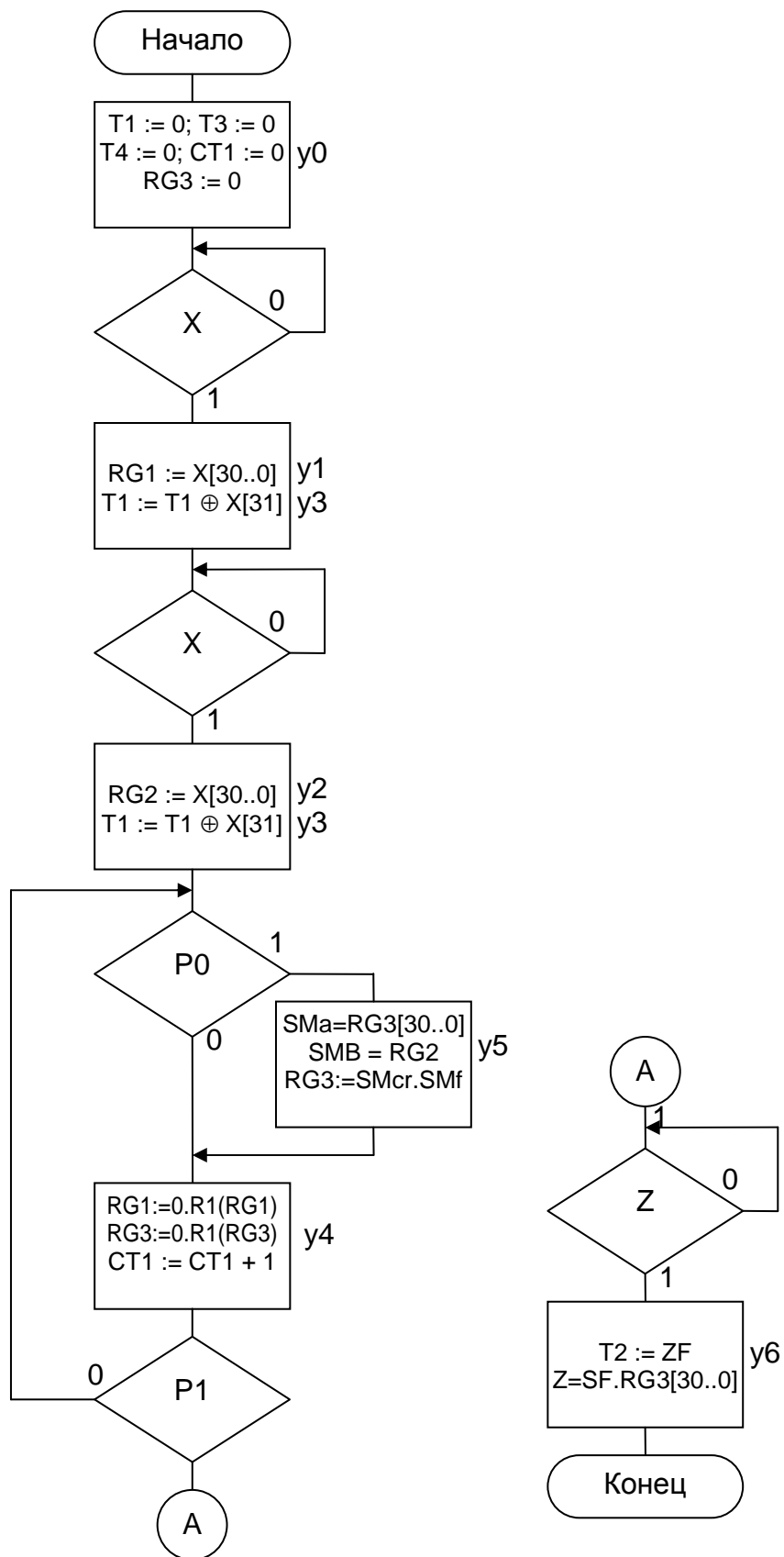
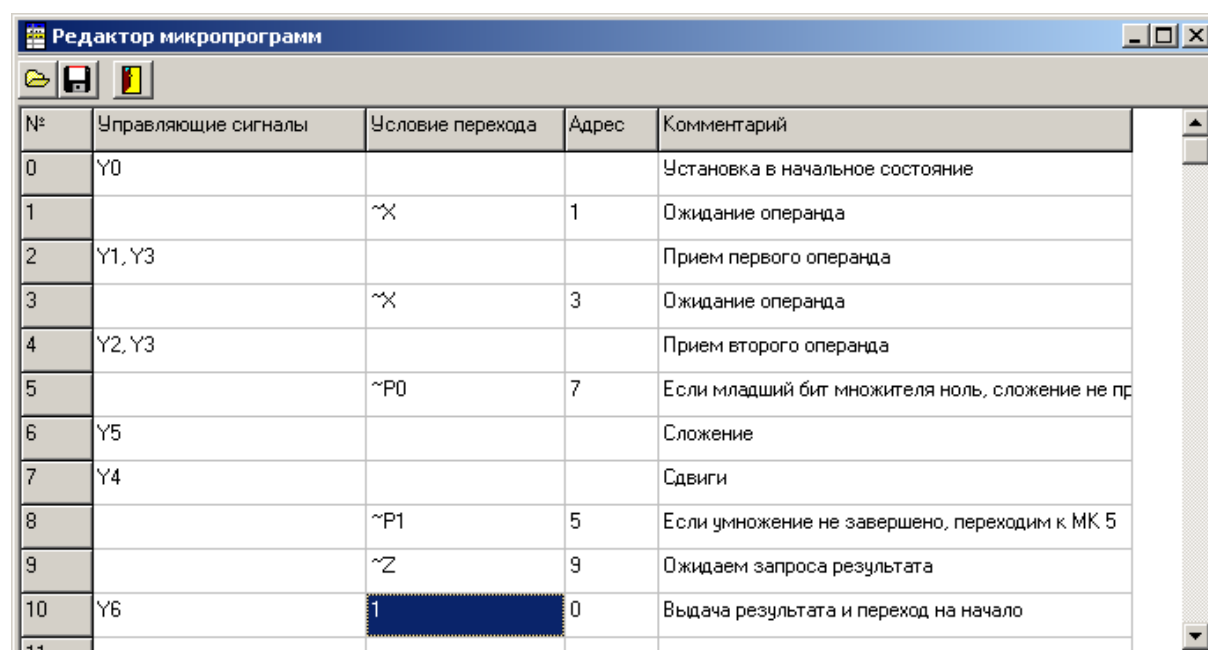


Рисунок 37 – ГСА операции умножения

Таблица 7 – Микропрограмма для устройства управления

№ МК	Управляющие сигналы	Условие и адрес перехода	Комментарий
0	Y0	–	Установка в начальное состояние
1	–	$\sim X \rightarrow 1$	Ожидание операнда
2	Y1, Y3	–	Прием первого операнда
3	–	$\sim X \rightarrow 3$	Ожидание операнда
4	Y2, Y3	–	Прием второго операнда
5	–	$\sim P0 \rightarrow 7$	Если младший бит множителя ноль, сложение не производим
6	Y5	–	Сложение
7	Y4	–	Сдвиги
8	–	$\sim P1 \rightarrow 5$	Если умножение не завершено, переходим к МК 5
9	–	$\sim Z \rightarrow 9$	Ожидаем запроса результата
10	Y6	$\rightarrow 0$	Выдача результата и переход на начало

31. Запустите программу MIFEdit.exe и введите предложенную микропрограмму (рисунок 38). Сохраните файл rzumpr.mif в каталоге проекта.



№	Управляющие сигналы	Условие перехода	Адрес	Комментарий
0	Y0			Установка в начальное состояние
1		$\sim X$	1	Ожидание операнда
2	Y1, Y3			Прием первого операнда
3		$\sim X$	3	Ожидание операнда
4	Y2, Y3			Прием второго операнда
5		$\sim P0$	7	Если младший бит множителя ноль, сложение не пр
6	Y5			Сложение
7	Y4			Сдвиги
8		$\sim P1$	5	Если умножение не завершено, переходим к МК 5
9		$\sim Z$	9	Ожидаем запроса результата
10	Y6	1	0	Выдача результата и переход на начало

Рисунок 38 – Микропрограмма в окне редактора

32. Добавьте в схему Lab5.bdf устройство управления (Project → UA) и выполните соединения, как показано на рисунке 39. Обратите внимание, что выходы Adr устройства управления подключены к

светодиодам led5 – led0. Таким образом, эти светодиоды будут показывать текущее состояние управляющего автомата в двоичном коде (точнее, номер текущего шага по модулю 64, – это нужно иметь ввиду, если количество микрокоманд в микропрограмме превысит 64).

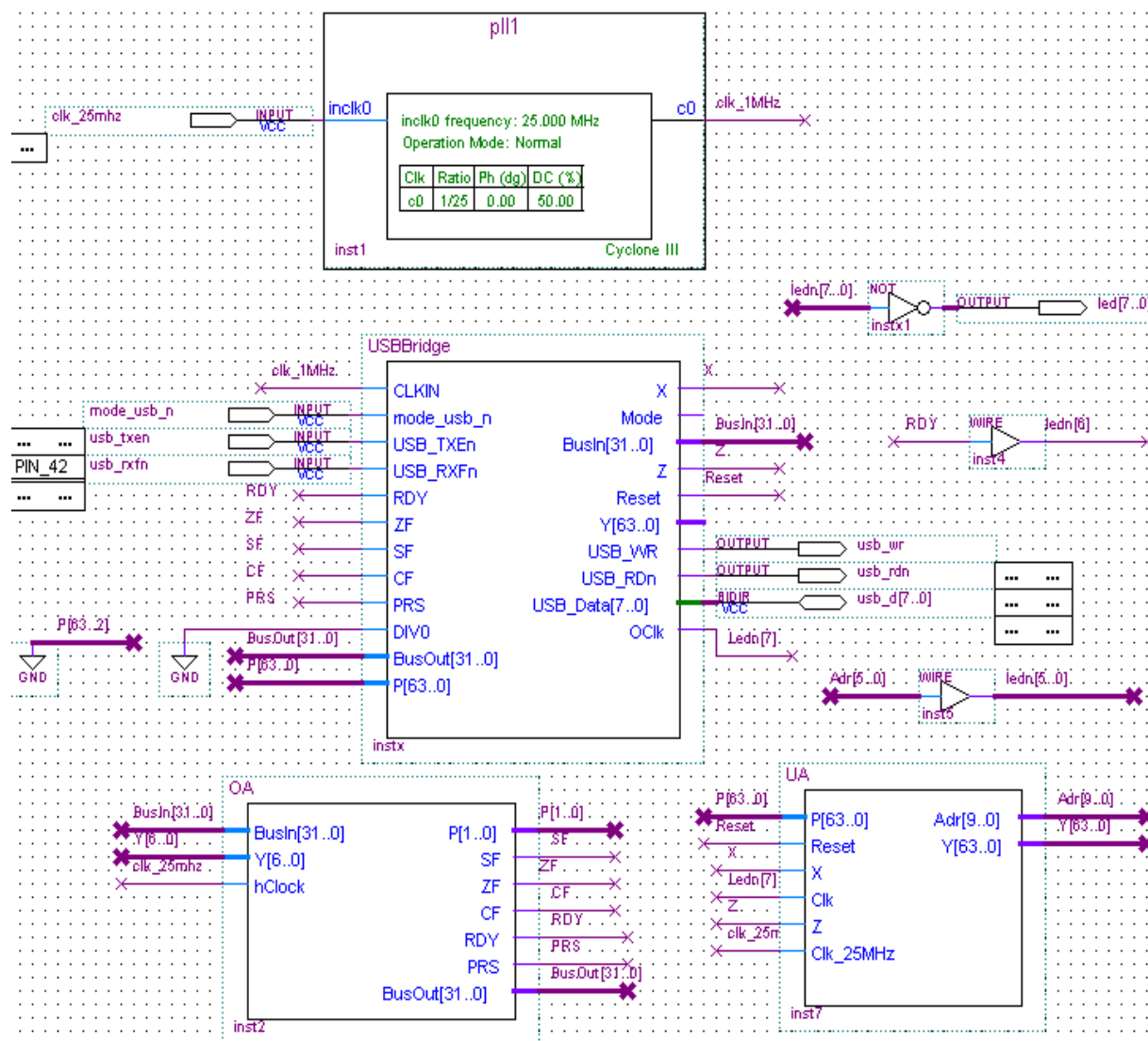


Рисунок 39 – Схема АЛУ с УУ

33. Скомпилируйте проект, переставьте джамперы, запрограммируйте ПЛИС.

34. Переставьте джамперы и запустите программу SchemKP.exe. Выполните проверку проекта:

- подайте и снимите сигнал Reset. Убедитесь, что все светодиоды погасли;
- подайте тактовый импульс (загорится светодиод led0 – состояние 1);
- подайте несколько тактовых сигналов и убедитесь, что состояние не меняется;

- введите число -0,75 в ПК (не забудьте нажать клавишу [Enter]), установите флажок X; дайте несколько тактовых импульсов и посмотрите, как меняются состояния УА (2 → 3) (флажок X сбросится);
- введите число 0,5 в ПК, установите флажок X; подавайте тактовые импульсы и смотрите, как меняются состояния УА: 3 → 4 → 5 → 7 → 8 → (состояния 5 → 7 → 8 повторяются 29 раз – поскольку младшие 29 разрядов множителя нули, производятся только сдвиги) → 5 → 6 → 7 → 8 → 5 → 6 → 7 → 8 → 9 (дважды повторяющаяся последовательность 5 → 6 → 7 → 8 – суммирование и сдвиги – соответствует двум единицам множителя); обратите внимание на то, что включается светодиод led6 – индикация сигнала RDY. Проверьте состояние RDY в программе, нажав кнопку [Обновить флаги и ШиВых];
- установите флажок Z и подайте два тактовых импульса. Проверьте результат.

35. Проверьте работу АЛУ в режиме «Генератор»:

- установите режим «Генератор» (светодиод led7 загорается вполнакала);
- введите число -0,2 в ПК, установите флажок X;
- введите число -0,3 в ПК, установите флажок X;
- убедитесь в том, что светодиод led7 горит и установите флажок Z. Проверьте результат.

36. Проверьте работу АЛУ еще на 5 вариантах исходных данных (с одинаковыми/разными знаками, с нулем, с очень маленькими числами и т.д.). Запишите результаты в таблицу.

Лабораторные работы №6, 7

«Разработка АЛУ для выполнения отдельных операций на базе ПЛИС»

«Разработка АЛУ для выполнения набора операций на базе ПЛИС»

В процессе выполнения шестой лабораторной работы для каждой операции из задания на курсовой проект разрабатывается и тестируется свое АЛУ. При этом за основу берется проект, созданный в процессе выполнения лабораторной работы №5. Фактически, для каждой операции необходимо поменять только схему операционного устройства и управляющую микропрограмму. Исследование каждого отдельного АЛУ завершается тестированием на примерах (не менее 6 примеров для арифметических операций, и не менее 3 примеров для логических операций).

Лабораторная работа №7 предполагает разработку АЛУ, способного выполнить любую из заданных операций по выбору пользователя. Выбор кода операции можно осуществлять неиспользуемыми сигналами Y (например, $Y_{61}-Y_{63}$). В тех случаях, когда требуется постоянно подавать код операции в операционное устройство, можно в схеме на верхнем уровне иерархии проекта (в пятой работе это была схема lab5.bdf) добавить регистр-защелку (рисунок 40).

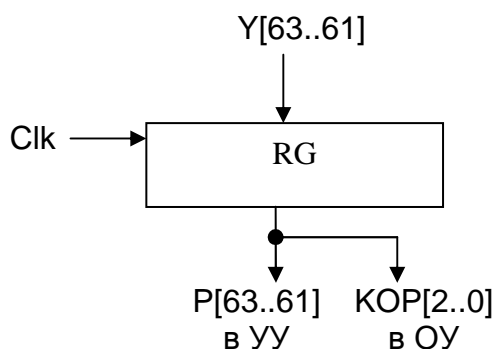


Рисунок 40 – Формирование КОП

Необходимо также обратить внимание на сложные условия, которые не могут непосредственно обрабатываться УУ, например, условие $P1 \vee \sim P2$ или условие $f1 \vee f2 \vee f5$. В этом случае придется либо писать более сложные микропрограммы, содержащие несколько подряд идущих микрокоманд, анализирующих фрагменты сложного условия, либо сформировать дополнительные осведомительные сигналы: $P20 = P1 \vee \sim P2$, $P21 = DC1(KOP) \vee DC2(KOP) \vee DC5(KOP)$, для чего добавить в схему несколько элементов (элементы «И», «ИЛИ», «НЕ», дешифраторы).

Схемы и алгоритмы, разработанные для курсового проекта, должны соответствовать схемам и алгоритмам, реализуемым в лабораторных работах 6 и 7, так как эти работы предназначены для проверки и отладки функциональных схем и алгоритмов курсового проекта.

8 Задания на лабораторные исследования

Лабораторная работа №1 – «Исследование логических элементов»

Цель работы: изучение работы заданного логического элемента, синтез на его основе других логических элементов, измерение некоторых параметров ЛЭ.

Задание на лабораторные исследования:

а) построить на основе заданной микросхемы следующие элементы: «НЕ», «2И», «2ИЛИ», «2И-НЕ», «2ИЛИ-НЕ». Проверить правильность работы синтезированных ЛЭ. Для каждого построить таблицу истинности;

б) используя созданные элементы, проверить истинность следующих логических выражений;

$$\sim(\sim X) = X$$

$$X \cdot 0 = 0$$

$$X \cdot 1 = X$$

$$X \cdot \sim X = 0$$

$$X \vee 0 = X$$

$$X \vee 1 = 1$$

$$X \vee \sim X = 1$$

в) для схемы, собранной для проверки первого логического выражения, провести измерения параметров $U_{\text{ВЫХ}}^1$, $U_{\text{ВЫХ}}^0$, $I_{\text{ВХ}}^1$, $I_{\text{ВХ}}^0$;

г) (дополнительное) построить на основе заданной микросхемы элемент «исключающее ИЛИ» и построить для него таблицу истинности.

Отчет должен содержать титульный лист, цель работы, задание. Для каждого задания нужно привести принципиальную схему и таблицу истинности, а также заключение о том, что созданный элемент работает верно (или что исследуемое логическое выражение верно). По результатам измерений электрических параметров должна быть построена таблица, в которой должны отображаться измеренные значения параметров, а также значения этих параметров из справочника по используемой микросхеме. По таблице необходимо сделать заключение о соответствии или несоответствии измеренных параметров справочным. В случае несоответствия необходимо привести объяснение отклонения параметров.

Отчет завершается выводами, обобщающими результаты работы.

Лабораторная работа №2 – «Исследование триггеров»

Цель работы: изучение работы заданного триггера и синтез на его основе других типов триггеров.

Задание на лабораторные исследования:

а) подключить заданный триггер, проверить правильность его работы и составить таблицу переходов триггера (например, см. таблицу X);

б) синтезировать на основе заданного триггера следующие типы триггеров: RS-триггер, D-триггер, Т-триггер, JK-триггер.

в) (дополнительное) на основе заданной микросхемы синтезировать 2-разрядный счетчик.

Отчет должен содержать титульный лист, цель работы, задание. Для каждого задания нужно привести УГО реализуемого триггера, принципиальную схему, таблицу переходов и временную диаграмму (см. рисунок X), а также заключение о том, что созданный триггер работает верно. Отчет завершается выводами, обобщающими результаты работы.

Таблица X – Пример таблицы переходов для D-триггера с асинхронным сбросом

D	C	R	Q(t)	Q(t+1)
*	0, 1 или ↓	0	0	0
*	0, 1 или ↓	0	1	1
*	*	1	*	0
0	↑	0	*	0
1	↑	0	*	1

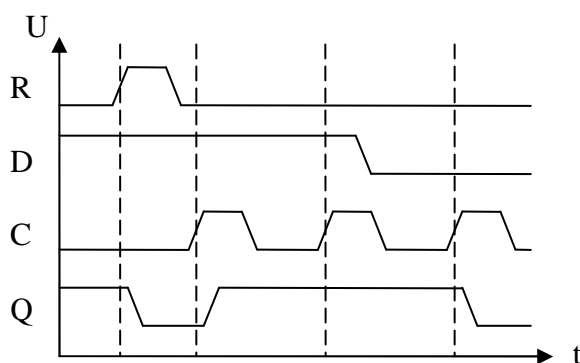


Рисунок X – Временная диаграмма работы D-триггера с асинхронным сбросом

Лабораторная работа №3 – «Исследование регистров»

Цель работы: изучение работы заданного регистра и синтез на его основе устройств сдвига.

Задание на лабораторные исследования:

а) подключить заданный регистр (K155ИР1 или аналогичный), проверить правильность его работы и составить таблицу переходов регистра (например, см. таблицу X). Выходы регистра должны быть подключены к индикатору в правильной последовательности: слева старший значащий разряд, справа – младший (например, выходы Q3-Q0 подключаются к индикаторам стенда L15-L12);

б) реализовать режим сдвига вправо (в сторону младших разрядов) с последовательным занесением информации. Занести несколько значений, например, следующие: 0110, 1101, 0001;

в) добавить к предыдущей схеме функцию циклического сдвига влево (в сторону старших разрядов). Проверить правильность работы схемы в обоих режимах, занеся последовательно при сдвиге вправо какое-либо значение (например, 1011: 0000→1000→1100→0110→1011), а затем 4 раза сдвинув его влево (например: 1011→0111→1110→1101→1011);

г) на основе микросхемы ТМ8 или аналогичной (набор D-триггеров или несдвиговой регистр с динамической синхронизацией) реализовать 4-разрядный сдвиговой (влево) регистр с последовательным занесением). Проверить правильность его работы;

д) соединить оба регистра таким образом, чтобы информация во второй регистр заносилась при сдвиге первого влево. При выполнении четырех сдвигов влево информация, занесенная в первый регистр при сдвигах вправо, переносится во второй регистр. Построить временную диаграмму работы регистров, содержащую занесение данных и копирование информации из первого регистра во второй (8 тактов) (например, см. рисунок X). Вариант исходных данных для построения диаграммы взять в приложении А;

е) (дополнительное) используя мультиплексор (например, КП2 или построенный на основе простой логики), подключить ко входам тактового сигнала, используемого для сдвига влево обоих регистров, выхода ГОИ1 и генератора Г1. Проверить работу схемы, занеся в первый регистр значение 0110, переключив схему в режим сдвига влево и подключив (то есть, подав соответствующий сигнал адреса на входы мультиплексора) к тактовым входам генератор.

Отчет должен содержать титульный лист, цель работы, задание. Для каждого задания нужно привести реализуемую функциональную схему. Привести таблицу переходов (задание а) и временную диаграмму работы (задание д), Необходимо составить две принципиальные схемы:

первая для задания а) и вторая – для итоговой схемы. Отчет должен завершаться выводами, обобщающими результаты работы.

Таблица X – Пример таблицы переходов для сдвигового регистра

D	DSR	MO	C	R	Q(t)	Q(t+1)
*	*	*	0, 1 или ↓	0	Q(t)	Q(t)
*	*	*	*	1	*	0000
D3D2D1D0	*	0	↑	0	*	D3D2D1D0
*	DSR	1	↑	0	Q3Q2Q1Q0	DSRQ3Q2Q1

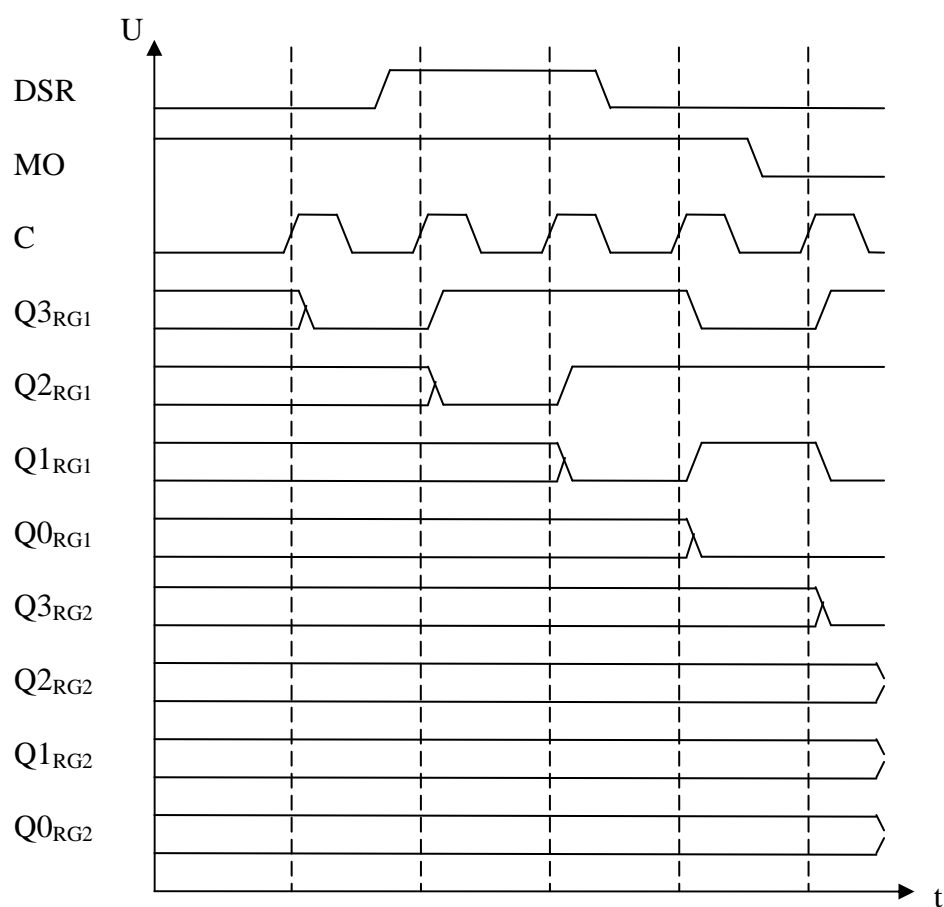


Рисунок X – Фрагмент временной диаграммы работы схемы на основе сдвиговых регистров

Лабораторная работа №4 – «Исследование счетчиков»

Цель работы: изучение работы заданного счетчика и синтез на его основе счетчика с заданным коэффициентом пересчета.

Задание на лабораторные исследования:

а) подключить заданный счетчик (K155ИЕ7 или другой, заданный преподавателем), проверить правильность его работы и составить таблицу переходов счетчика (например, см. таблицу X). Выходы счетчика должны быть подключены к индикатору в правильной последовательности: слева старший значащий разряд, справа – младший ко входам L3-L0 так, чтобы текущее значение счета отображалось в шестнадцатеричном виде на соответствующем индикаторе стенда;

б) расширить разрядность счетчика до 8, при этом старшие разряды счетчика подключить ко входам L7-L4. Таким образом, должен быть получен 8-разрядный реверсивный двоичный счетчик, текущее состояние которого должно отображаться на индикаторе стенда в шестнадцатеричном виде. Проверить правильность счета, подавая импульсы счета вручную и подключая на счетные входы выход генератора G1;

в) на основе полученной схемы реализовать счетчик с заданным коэффициентом пересчета (см. приложение А). Коэффициент пересчета должен быть один и тот же как при счете вверх, так и при счете вниз. Например, для счетчика с коэффициентом пересчета 137 при счете вверх, а затем вниз из начального значения счета 134 значения должны сменяться так: $134 \uparrow 135 \uparrow 136 \uparrow 0 \uparrow 1 \uparrow 2 \downarrow 1 \downarrow 0 \downarrow 136 \downarrow 135 \dots$ (соответствующие шестнадцатеричные значения: $86 \uparrow 87 \uparrow 88 \uparrow 00 \uparrow 01 \uparrow 02 \downarrow 01 \downarrow 00 \downarrow 88 \downarrow 87 \dots$).

г) нарисовать временную диаграмму работы счетчика, на которой отобразить несколько счетов вверх с переходом через ноль и затем несколько счетов вниз (например, как показано в п. в).

Таблица X – Пример таблицы переходов для счетчика

D	PE	C-1	C+1	R	Q(t)	Q(t+1)
*	1	0, 1 или ↓	0, 1 или ↓	0	Q(t)	Q(t)
*	*	*	*	1	*	0000
D3D2D1D0	0	*	*	0	*	D3D2D1D0
*	1	1	↑	0	Q(t)	Q(t)+1
*	1	↑	1	0	Q(t)	Q(t)-1

Лабораторная работа №5 –
«Разработка АЛУ для выполнения операции умножения на базе ПЛИС»

Цель работы: изучение основ проектирования арифметико-логических устройств ЭВМ, разработка и тестирование АЛУ для выполнения операции умножения.

Задание на лабораторные исследования:

- а) разработка схемы для тестирования лабораторной установки; тестирование лабораторной установки;
- б) разработка схемы операционного устройства; тестирования операционного устройства вручную на одном варианте исходных данных;
- в) по заданной граф-схеме алгоритма разработать управляющую микропрограмму;
- г) добавление в схему АЛУ устройства микропрограммного управления; тестирования АЛУ на нескольких (не менее 5) вариантах исходных данных.

Отчет должен содержать титульный лист, цель работы, задание. Для каждого задания приводится схема (копируется из САПР Quartus II) (для третьего задания вместо схемы в табличном виде приводится управляющая микропрограмма). Кроме того, для второго задания приводится исходная функциональная схема, для третьего задания – исходная ГСА. По результатам выполнения четвертого задания составляется таблица, содержащая для каждого примера оба операнда, результат, полученный в УУ, результат, полученный расчетным путем, абсолютную и относительную погрешности. Все результаты должны быть пояснены текстом.

Отчет завершается выводами, обобщающими (и объясняющими) результаты работы.

Лабораторная работа №6 –
«Разработка АЛУ для выполнения отдельных операций на базе ПЛИС»

Цель работы: разработка и тестирование АЛУ для каждой из заданного набора операций.

Задание на лабораторные исследования: для каждой из заданных операций выполнить следующие пункты:

- а) на основе функциональной схемы, разработанной в ходе курсового проектирования, синтезировать схему операционного устройства АЛУ в САПР Quartus II;

б) на основе схемы алгоритма, разработанной в ходе курсового проектирования, создать управляющую микропрограмму для устройства управления;

в) протестировать работу каждого АЛУ не менее чем на 6 вариантах исходных данных для арифметических операций, и не менее чем на 3 вариантах исходных данных для логических операций.

Отчет должен содержать титульный лист, цель работы, задание. Для каждой операции приводятся исходная функциональная схема, схема из САПР, исходная ГСА, разработанная микропрограмма и результаты тестирования в табличном виде. Таблица должны содержать для каждого примера операнды, результат, полученный в УУ, результат, полученный расчетным путем, абсолютную и относительную погрешности. Все результаты должны быть пояснены текстом.

Отчет завершается выводами, обобщающими результаты работы.

Лабораторная работа №7 – «Разработка АЛУ для выполнения набора операций на базе ПЛИС»

Цель работы: разработка и тестирование АЛУ для заданного набора операций.

Задание на лабораторные исследования:

а) на основе функциональной схемы, разработанной в ходе курсового проектирования, синтезировать схему операционного устройства АЛУ в САПР Quartus II;

б) на основе схемы алгоритма, разработанной в ходе курсового проектирования, создать управляющую микропрограмму для устройства управления;

в) протестировать работу АЛУ для каждой операции не менее чем на 6 вариантах исходных данных для арифметических операций, и не менее чем на 3 вариантах исходных данных для логических операций.

Отчет должен содержать титульный лист, цель работы, задание, исходную функциональную схему, схему из САПР, исходную ГСА, разработанную микропрограмму и результаты тестирования в табличном виде. Таблица должны содержать для каждого примера операнды, результат, полученный в УУ, результат, полученный расчетным путем, абсолютную и относительную погрешности. Все результаты должны быть пояснены текстом.

Отчет завершается выводами, обобщающими результаты работы.

Приложение А
Примеры вариантов заданий на лабораторные исследования

Лабораторная работа №1 – «Исследование логических элементов»

Вариант	Микросхема
1	K155ЛА3
2	K555ЛА3
3	K155ЛА4
4	K155ЛЕ1
5	KP1531ЛА3
6	K155ЛА6
7	K155ЛА8
8	K155ЛА13

Лабораторная работа №2 – «Исследование триггеров»

Вариант	Микросхема
1, 3, 5, 7	K155ТМ2
2, 4, 6, 8	K155ТВ1

Лабораторная работа №3 – «Исследование регистров»

Вариант	Исходные данные
1	0110
2	1001
3	0011
4	1100
5	0101
6	1010
7	1101
8	0100

Приложение Б
Таблица подключений пользовательских выводов ПЛИС

Наименование сигнала	Вывод ПЛИС	Описание
pba	34	Кнопка РВА (при нажатии подается ноль)
pbb	51	Кнопка РВА (при нажатии подается ноль)
sw0	24	Переключатели SW0-SW7. В зависимости от положения переключателя на вывод подается ноль или единица
sw1	25	
sw2	46	
sw3	49	
sw4	91	
sw5	90	
sw6	89	
sw7	88	
led0	64	Светодиоды LED0-LED7. Светодиод загорается, если с соответствующего вывода ПЛИС выдан ноль
led1	60	
led2	59	
led3	58	
led4	55	
led5	54	
led6	53	
led7	52	
clk_25MHz	22	Тактовый сигнал с частотой 25МГц
RAM_A0	141	Шина адреса микросхемы памяти – 18 разрядов – 256к адресов. На эту шину должен быть подан адрес 16-разрядной ячейки.
RAM_A1	138	
RAM_A2	137	
RAM_A3	136	
RAM_A4	135	
RAM_A5	119	
RAM_A6	115	
RAM_A7	114	
RAM_A8	113	
RAM_A9	112	
RAM_A10	111	
RAM_A11	110	
RAM_A12	106	
RAM_A13	105	
RAM_A14	104	
RAM_A15	144	
RAM_A16	143	
RAM_A17	142	
RAM_D0	132	

Наименование сигнала	Вывод ПЛИС	Описание
RAM_D1	129	Шина данных микросхемы памяти – 16 разрядов. При записи на эту шину подаются данные. При чтении данные снимаются с шины.
RAM_D2	128	
RAM_D3	127	
RAM_D4	126	
RAM_D5	125	
RAM_D6	124	
RAM_D7	121	
RAM_D8	103	
RAM_D9	101	
RAM_D10	100	
RAM_D11	99	
RAM_D12	11	
RAM_D13	10	
RAM_D14	7	
RAM_D15	4	
RAM_WE_N	120	Сигнал записи в память (активный низкий уровень). Перед записью должны быть поданы адрес и записываемые данные
RAM_OE_N	1	Сигнал разрешения выдачи данных. Используется при чтении. Активный низкий уровень.
RAM_BLE_N	3	Сигналы разрешения работы со старшим байтом 16-разрядного слова (BHE) и с младшим байтом (BLE). Активный уровень низкий. Для работы с 16-разрядными словами на оба входа нужно подать ноль.
RAM_BHE_N	2	
RAM_CE_N	133	Сигнал разрешения работы микросхемы памяти (активный низкий уровень). При высоком уровне память не выполняет никаких действий, а выходы находятся в третьем состоянии.
USB_D0	38	Шина данных интерфейса USB На нее подаются данные с ПЛИС для записи в очередь FIFO при передаче данных в ЭВМ. При чтении данных, отправленных с ЭВМ на плату, данные снимаются с этой шины.
USB_D1	32	
USB_D2	33	
USB_D3	44	
USB_D4	50	
USB_D5	30	
USB_D6	28	
USB_D7	31	
USB_RDN	43	Сигнал чтения данных из очереди FIFO (активный уровень низкий). При переходе из

Наименование сигнала	Вывод ПЛИС	Описание
		единицы в ноль считывается очередной байт из очереди FIFO чтения. Байт выдается на шину данных, пока сигнал удерживается в низком уровне.
USB_RXFN	39	Сигнал осведомляет о наличии данных готовых для чтения (низкий уровень) или об отсутствии таковых (высокий уровень).
USB_WR	98	Записывает данные, поданные на шину, в буфер FIFO передачи. Активный уровень высокий. Запись происходит по переходу из единицы в ноль. Предварительно должны быть поданы данные.
USB_TXEN	42	Сигнал разрешает запись в буфер FIFO передачи (низкий уровень) или запрещает ее (высокий уровень).
MODE_USB_N	23	Индикация режима USB. Если ноль, интерфейс USB используется для программирования (использовать его для обмена данными с ЭВМ нельзя; выводы ПЛИС, соединенные с шиной данных USB должны быть в третьем состоянии).
AA1	87	Вывод 1 разъема PR_A
AA2	86	Вывод 3 разъема PR_A
AA3	85	Вывод 5 разъема PR_A
AA4	84	Вывод 7 разъема PR_A
AA5	83	Вывод 13 разъема PR_A
AA6	80	Вывод 15 разъема PR_A
AB1	79	Вывод 2 разъема PR_A
AB2	77	Вывод 4 разъема PR_A
AB3	76	Вывод 6 разъема PR_A
AB4	75	Вывод 8 разъема PR_A
AB5	74	Вывод 14 разъема PR_A
AB6	73	Вывод 16 разъема PR_A
BA1	72	Вывод 1 разъема PR_B
BA2	71	Вывод 3 разъема PR_B
BA3	70	Вывод 5 разъема PR_B
BA4	69	Вывод 7 разъема PR_B
BA5	68	Вывод 13 разъема PR_B
BA6	67	Вывод 15 разъема PR_B
BB1	66	Вывод 2 разъема PR_B
BB2	65	Вывод 4 разъема PR_B

Приложение В
Пример оформления титульного листа отчета

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Кафедра электронных вычислительных машин

ИССЛЕДОВАНИЕ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ

отчет
по лабораторной работе №1
по дисциплине «Схемотехника ЭВМ»

Выполнили: студенты гр. ВМ-21	_____	Иванов И.И.
09.02.2011	_____	Петров П.П.
	_____	Сидоров С.С.
Проверил: доцент кафедры ЭВМ	_____	Томчук М.Н.

Киров 2011

Библиографический список

1. Хоровиц, Пауль. Искусство схемотехники / Хоровиц, Пауль, Хилл, Уинфилд. - 6-е изд., перераб. - М.: Мир, 2001, 2003. - 704с.: ил. (в библиотеке - 3 экз.)
2. Угрюмов, Е. П. Цифровая схемотехника: Учеб. пособие / Угрюмов, Е. П. - СПб.: БХВ-Петербург, 2001. - 528с.: ил. - Библиогр.: с. 511. (в библиотеке - 2 экз.)
3. Райхлин, Вадим Абрамович. Основы цифровой схемотехники: Учеб. пос. для вузов / Райхлин, Вадим Абрамович; КГТУ. - Казань: Изд-во КГТУ, 2000. - 352с.: ил. (в библиотеке - 6 экз.)
4. Алексенко, Андрей Геннадьевич. Основы микросхемотехники / Алексенко, Андрей Геннадьевич. - 3-е изд., перераб. и доп. - М.: Лаборатория Базовых Знаний: ФИЗМАТЛИТ: ЮНИМЕДИАСТАЙЛ, 2002. - 448с.: ил. - Библиогр.: с. 438. (в библиотеке - 5 экз.)
5. Разевиг, Всеволод Данилович. Схемотехническое моделирование с помощью Micro-CAP 7 / Разевиг, Всеволод Данилович. - М.: Горячая линия-Телеком, 2003. - 368с.: ил. (в библиотеке - 2 экз.)
6. Схемотехника ЭВМ : учеб. / под ред. Г. Н. Соловьева. - М. : Высш. шк., 1985. - 391 с. : ил.. - Библиогр.: с. 386 (в библиотеке - 47 экз.)
7. Схемотехника электронных систем. Цифровые устройства. - СПб. : БХВ-Петербург, 2004. - 512 с. : ил.. - Библиогр.: с. 493-497 (в библиотеке - 60 экз.)
8. Cyclone III Device Handbook [Электронный ресурс]. – Режим доступа: http://www.altera.com/literature/hb/cyc3/cyclone3_handbook.pdf
9. Комолов, Д.А. Системы автоматизированного проектирования фирмы Altera MAX+plus II и Quartus II. Краткое описание и само-учитель / Д.А. Комолов, Р. А. Мяльк, А. А. Зобенко, А. С. Филиппов – М.: ИП РадиоСофт, 2002 - 352 с.: ил.
10. Quartus II Development Software Handbook [Электронный ресурс]. – Режим доступа: http://www.altera.com/literature/hb/qts/archives/quartusii_handbook_9.1.2.pdf
11. FT245R [Электронный ресурс]. – Режим доступа: http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT245R.pdf