Министерство образования и науки Российской Федерации

Калужский филиал

федерального государственного бюджетного образовательного учреждения высшего образования

**«Московский государственный технический университет имени Н.Э. Баумана**

(национальный исследовательский университет)»

(КФ МГТУ им. Н. Э. Баумана)

**О.Ю. Жарова**

**Разработка арифметико-логического устройства под управлением микропрограммного автомата**

Методические указания к выполнению курсового проекта

по дисциплине «Аппаратные средства вычислительной техники»

Калуга 2018

УДК

Методические указания составлены в соответствии с учебным планом КФ МГТУ им. Н,Э. Баумана по специальности «Информационная безопасность автоматизированных систем».

Методические указания рассмотрены и одобрены:

- кафедрой «Информационная безопасность автоматизированных систем» (ЭИУ6-КФ)

протокол № \_\_\_ от \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

Зав. кафедрой ЭИУ6-КФ \_\_\_\_\_\_\_\_\_\_\_\_\_\_­­­­­\_\_д.т.н., доц. Мазин А.В.

- методической комиссией факультета ЭИУК

протокол № \_\_\_ от \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

Председатель метод. комиссии\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_к.т.н., доц. Адкин М.Ю.

- методической комиссией КФ МГТУ им. Н.Э. Баумана

протокол № \_\_\_ от \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

Председатель метод. комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_ д.э.н., профессор Перерва О.Л.

Рецензент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ доцент, к.т.н. декан ЭИУК Адкин М.Ю.

Автор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ассистент кафедры ЭИУ6-КФ Жарова О. Ю.

Аннотация

В методических указаниях изложены практические приемы проектирования узлов вычислительной техники. Предназначено для студентов специальности «Информационная безопасность автоматизированных систем» и может быть рекомендовано к применению при выполнении курсового проекта по курсу «Аппаратные средства вычислительной техники».

© Калужский филиал МГТУ им. Н.Э. Баумана, 2018 г.

© Жарова О.Ю., 2018г.

СОДЕРЖАНИЕ

[Введение 3](#_Toc507175128)

[Выполнение курсового проекта 3](#_Toc507175129)

[Этапы работы 3](#_Toc507175130)

[Оформление пояснительной записки: 3](#_Toc507175131)

[Содержание пояснительной записки: 3](#_Toc507175132)

[Теоретическая часть 3](#_Toc507175133)

[Структура АЛУ 3](#_Toc507175134)

[Реализация операционных устройств вычислительной машины 3](#_Toc507175135)

[Реализация базовых логических операций 3](#_Toc507175136)

[Практическая часть 3](#_Toc507175137)

[Регистр хранения информации 3](#_Toc507175138)

[Сумматор 3](#_Toc507175139)

[Компаратор 3](#_Toc507175140)

[Инвертор 3](#_Toc507175141)

[Дешифратор 3](#_Toc507175142)

[Триггер 3](#_Toc507175143)

[Логический элемент И 3](#_Toc507175144)

[Выполнение математических операций 3](#_Toc507175145)

[Приемы реализации АЛУ 3](#_Toc507175146)

[Работа с шиной в Proteus 3](#_Toc507175147)

[Индивидуальные задания 3](#_Toc507175148)

[Литература 3](#_Toc507175149)

[Основная литература 3](#_Toc507175150)

[Дополнительная литература 3](#_Toc507175151)

[Электронные ресурсы 3](#_Toc507175152)

[Приложение 1. Оформление курсового проекта 3](#_Toc507175153)

[Приложение 2. Пример оформления графической части. 3](#_Toc507175154)

# Введение

**Цель работы**: формирование практических навыков выполнения проектно-конструкторских расчетов и оформления документации при разработке компонентов защищенных автоматизированных систем.

**Задачи**: Разработать АЛУ для заданного набора команд, управление которым должно осуществляться микропрограммным автоматом, на вход которого подается код операции и адрес операндов, после чего происходит, и выполнение заданной операции с занесением результатов в регистр результата.

**Требования к отчету**:

Пояснительная записка должна быть объемом не менее 40 страниц.

Графическая часть выполняется в виде приложения в соответствии с ЕСКД.

# Выполнение курсового проекта

Необходимо разработать АЛУ реализующее заданные команды. Для управления АЛУ в разрабатываемой схеме должен содержаться МПА и СК. Выборка операндов должна производиться из регистровой памяти. Способ адресации - регистровый. результат должен помещаться обратно в регистровую память, либо формировать признак в регистре признаков. Общая схема должна представлять собой упрощенную функциональную схему **фон-неймановской вычислительной машины (**Рисунок 1**).** Для проверки правильности работы схемы необходимо написать программу на ассемблере с использованием заданных команд.

## Этапы работы

Этап 1: отдельные модели на каждую команду в среде моделирования. Программы на языке ассемблер, для каждой команды. Исследовательская часть записки.

Этап 2: Общая модель АЛУ с управляющей частью. Конструкторская часть записки.

Этап 3: Оформление графической части. Технологическая часть записки.

## Оформление пояснительной записки:

Пояснительная записка должна быть оформлена в соответствии с ГОСТ 2.105-95. Пример содержания приведен в приложении 1.

## Содержание пояснительной записки:

**Оглавление:** страницы должны в обязательном порядке совпадать.

**Введение:** общие теоретические сведения по теме курсовой работы цель, задача и задание курсового проекта.

**Исследовательскую часть:**

1.1. Подробное описание работы заданных команд, обязательно приведены примеры работы каждой команды в двоичном виде.

1.2. Выбор элементной базы. Выбор каждого элемента должен производиться на основе анализа двух аналогов. Аналог это микросхема, которая выполняет по меньшей мере те же действия, что и требуемый элемент. Ее функционал может включать дополнительные возможности, но не может быть заведомо ниже требуемого.

**Конструкторскую часть:**

2.1. Описание принципа функционирования схемы при выполнении каждой из заданных команд. Должны быть приведены программы на языке ассемблер реализующие команды с теми же операндами.

2.2. Описание схемы управления АЛУ.

**Технологическую часть:**

3. Подробное описание принципов взаимодействия с моделью. Инструкции использования модели для каждой команды, а именно, каким образом и в какой генератор шаблонов можно ввести код операции и адреса операндов, в каком регистре формируется результат, и как его интерпретировать, если результат не конкретное число а признак.

**Список литературы.**

**Примеры оформления приведены в Приложении 1 (**Приложение 1. Оформление курсового проекта**).**

**В графическом приложении должны содержаться**

* Схема функциональная (отдельно рассмотрен блок управления, блок регистров памяти и отдельно АЛУ) - 1 лист формата А3 (подпись на штампе КР.NNNYY-01 90; где NNN- номер зачетки, YY - год выполнения КР);
* Моделирование работы схемы ( общая схема из среды моделирования Proteus) - 1 лист формата А2 (подпись на штампе КР.NNNYY-01 98 01-М);
* Схема электрическая принципиальная (ГОСТ 2.743-91)- 1 лист формата А1-А0 (подпись на штампе КР.NNNYY-01 98 01-Э);
* Алгоритмы работы команд (ГОСТ 19.701-90) - 1 лист формата А2 (подпись на штампе КР.NNNYY-01 81);
* Перечень элементов - А4 (подпись на штампе КР.NNNYY-01 98 01-П).

**Примеры оформления приведены в Приложении 2 (**Приложение 2. Пример оформления графической части.**).**

# Теоретическая часть

Арифметико-логическое устройство (АЛУ) предназначено для арифметической и логической обработки данных.

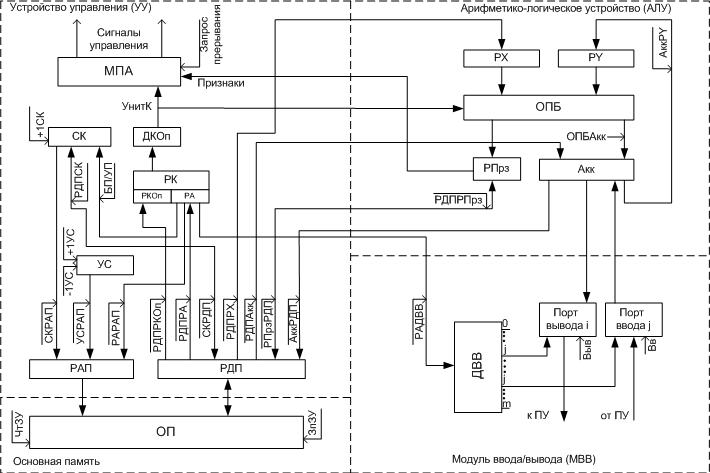


Рисунок 1 - Типовая функциональная структура ВМ.

## Структура АЛУ

**Операционный блок (ОПБ)** непосредственно выполняет арифметические и логические операции над данными. Выбор конкретной операции из возможного списка определяется КОп команды (поступает из ДКОп). В действительности КОп сначала преобразуется в МПА в иную форму, затем из МПА поступает в АЛУ. ОПБ современных АЛУ представляют собой комбинационные схемы, то есть не обладают внутренней памятью и до момента сохранения результата операнды должны присутствовать на входе ОПБ.

**Регистры операндов (PX, PY)** обеспечивают сохранение операндов на входе ОПБ до получения результата операции и его записи.

**Регистр признаков (РПрз)** предназначен для фиксации и хранения признаков (флагов) результата последней выполненной операции. Формирование флагов осуществляется блоком формирования состояний РПрз, который может либо входить в состав ОПБ, либо реализуется в виде внешней схемы, расположенной между ОПБ и РПрз.

**Аккумулятор (Акк)** содержит один из операндов, участвующих в операции. В него же помещается результат выполнения операции.

**Основная память (ОП).** Вне зависимости от типа используемых СБИС ячейка ОП представляет собой массив запоминающих элементов (ЗЭ), способных хранить единицу информации, как правило, в виде одного байта.

Все ячейки ОП являются адресуемыми и организованы в виде матрицы. Выбор определенной ячейки осуществляется подачей разрешающих сигналов на соответствующую строку и столбец этой матрицы. Это обеспечивается дешифратором адреса памяти, преобразующим адрес ячейки из РАП в разрешающие сигналы, подаваемые в горизонтальную и вертикальную линии матрицы, на пересечении которых и находится ячейка.

ОП реализуется на базе нескольких СБИС. Процесс обращения к ячейке состоит из выбора нужной микросхемы (на основании старших разрядов) и выбора ячейки внутри этой микросхемы (на основании младших разрядов).

## Реализация операционных устройств вычислительной машины

Тип ОПБ магистрального ОПУ определяется способом обработки данных.

Различают ОПБ последовательного и параллельного типа. В первом случае операции осуществляются последовательно над всеми разрядами, побитно поступающими на входы ОПБ. Бит переноса, возникающий при этом, учитывается при обработке следующего разряда операнда. Результат побитно заносится в выходной регистр, предыдущее значение которого сдвигается на один разряд. За n циклов в выходном регистре формируется слово результата, где каждый разряд занимает предназначенную для него позицию. Во втором случае операции над всеми разрядами операнда осуществляются побитно параллельно. На практике часто используют альтернативный параллельно-последовательный вариант, при котором слово разбивается на группы (по 2, 4 или 8) разрядов, обработка этих разрядов внутри группы осуществляется параллельно, а сами группы обрабатываются последовательно.

Для большинства современных ВМ общепринятым является формат данных с ФЗ. По этой причине, соответствующие ОПУ называют целочисленными, обеспечивающие сложение/вычитание, умножение и деление. При выполнении операций сложения возможна ситуация, когда при сложении двух n разрядных чисел получается число разрядностью n+1 (переполнение). ОПУ должно выявлять факт переполнения и сигнализировать о нем. Обычно это организуется выставлением определенного флага в статусном регистре.

Чтобы облегчить обнаружение переполнения применяется модифицированный дополнительный код, когда для хранения знака отводится два разряда, причем оба участвуют в арифметической операции наравне с цифровыми разрядами. Когда переполнение отсутствует, оба знаковых разряда содержат одинаковое значение (оба 0 или 1). Различие же в знаковых разрядах говорит о возникновении переполнения.

Умножение является более сложной операцией. В современных ОПУ умножение реализуется по разному. В частности, имеют место алгоритмы со сдвигом. В общем случае, алгоритм сводится к формированию частичных произведений (ЧП), по одному на каждую цифру множителя, с последующим суммированием полученных ЧП. Перед суммированием каждое ЧП должно быть сдвинуто на один разряд относительно предыдущего. Перемножение двух n-разрядных чисел приводит к получению 2n-битного результата. Согласно этому алгоритму в устройстве умножения предполагается наличие множимого, множителя и суммы ЧП, а также сумматора и, возможно, схем сдвига. Средиосновных, различают два варианта умножения:

со сдвигом суммы ЧП вправо при неподвижном множимом, начиная умножение с младших разрядов множителя (алгоритм сдвига вправо (Рисунок 2)),

со сдвигом суммы ЧП влево при неподвижном множимом, начиная умножение со старших разрядов множителя (алгоритм сдвига влево).

Суть алгоритма умножения со сдвигом вправо в следующем:

1. полное значение суммы ЧП полагается равным нулю,

2. анализируется очередная цифра множителя (анализ начинается с младшей цифры).

Если она равна единице, то к сумме частичных произведений прибавляется множимое, в противном случае прибавление не производится,

3. выполняется сдвиг суммы ЧП вправо на один разряд,

4. пункты 2 и 3 выполняются последовательно для всех цифровых разрядов множителя.

Схема аппаратной реализации приведена на Рисунок 2. Первоначально множимое и множитель заносятся в соответствующие регистры РМн. и РМт. Все разряды суммы ЧП устанавливаются в нуль. Умножение проходит за n-шагов, где n − количество разрядов множителя. На каждом шаге в зависимости от состояния младшего разряда множителя, который управляет мультиплексором, на один извыходом n-разрядного сумматора подается либо множимое, либо 0. На второй вход поступает содержимое регистра, используемого для хранения суммы ЧП.

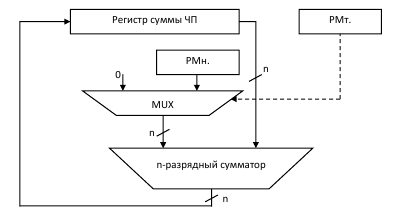


Рисунок 2 - Алгоритм умножения с правым сдвигом

Несколько сложнее обстоит дело с умножением чисел со знаком. Проблема в том, ОПУ выполняют операции над знаковыми разрядами также как и над числовыми. Если же знаковые разряды выделить специальным образом, то определение знака результата возможно при помощи логических функций.

Для ускорения умножения могут быть использованы логические и аппаратные методы. Логические методы позволяют уменьшить количество сложений в ходе умножения, а также обеспечить обработку нескольких разрядов за один шаг. Аппаратные способы ускорения умножения основаны на распараллеливании отдельных операций. Это приводит к существенному усложнению аппаратуры и как следствие возрастанию ее стоимости.

Рассмотрим вопрос о реализации базовых операций для данных с ПЗ. Они существенным образом отличаются от операций с ФЗ. Поэтому их как правило реализуют на базе отдельного ОПУ. Как минимум, это устройство должно реализовывать выполнение четырех арифметических действий: сложения, вычитания, умножения и деления.

Представление данных в формате с ПЗ полагает трехкомпонентное представление числа. Во-первых, речь идет о его знаке, во-вторых, о нормализованной мантиссе и, в третьих, о порядке числа. В силу этого, ОПУ состоят из трех соответствующих компонент: блока обработки знаков (БОЗ), блока обработки порядков (БОП) и блоков обработки мантисс (БОМ). Для хранения операндов и результата в ОПУ предусмотрены отдельные регистры. На подготовительном этапе операций с ПЗ осуществляется его разбиение на три составляющие. На этом этапе выполняется проверка на равенство нулю отдельных операндов. Это позволяет в дальнейшем исключить лишние операции. Действия на заключительном этапе выполнения операций с ПЗ идентичны и сводятся к проверке нулевого значения мантиссы, ее нормализации.

Сложение (вычитание) чисел с ПЗ состоит из следующих этапов:

* подготовительный этап,
* определение операнда, имеющего меньший порядок, и сдвиг его мантмссы вправо на число разрядов, равное разности порядков,
* приравнивание порядка результата большему из порядков операндов,
* сложение или вычитание мантисс и определение знака результата,
* проверка на переполнение,
* заключительный этап.

При выполнении умножения (или деления) на предварительном этапе осуществляется проверка на нуль одного из сомножителей, что позволяет сразу выдать результат или сообщение об ошибке. Далее выполняется сложение (или вычитание) порядков. Если порядок выходит за допустимые границы (переполнение порядка или потеря значимости), то выдается соответсвующее сообщение об ошибке. Операции над порядками осуществляются также как и операции в целочисленной арифметике. Затем идут операции над мантиссами (перемножение / деление) как с числами с ФЗ, которые завершаются операциями округления, нормализации и компоновки результата.

Реализация базовых логических операций

Помимо рассмотренных основных арифметических действий ОПУ в любой ВМ реализуются операции логические («И», «ИЛИ», «НЕ», «исключающее ИЛИ») и сдвига. Чаще всего, эти операции реализуются дополнительными схемами, входящими в состав целочисленных ОПУ.

На практике логические операции в ОПУ выполняются сразу над совокупностью логических переменных, объединенных в один байт или слово. Причем над всеми битами выполняется одна логическая операция. Возможная структура подобного ОПБ представлена на Рисунок 3. Выбор нужной операции (одной из четырех) осуществляется бинарным управляющим кодом на входах дешифратора.

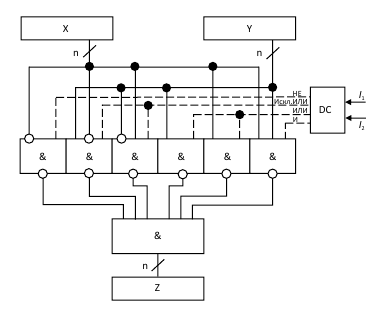


Рисунок 3 - Структура ОПБ для выполнения логических операций.

# Практическая часть

Для построения схемы АЛУ потребуется знание стандарта [ГОСТ 2.743-91 «ОБОЗНАЧЕНИЯ УСЛОВНЫЕ ГРАФИЧЕСКИЕ В СХЕМАХ. ЭЛЕМЕНТЫ ЦИФРОВОЙ ТЕХНИКИ», введенного 01.01.93](http://docs.cntd.ru/document/1200010863).

Настоящий стандарт устанавливает общие правила построения условных графических обозначений (УГО) элементов цифровой техники в схемах, выполняемых вручную или с помощью печатающих и графических устройств вывода ЭВМ во всех отраслях промышленности. Ниже приведены основные положения и примеры УГО элементов, которые потребуются для построения АЛУ.

Возможные варианты элементов для реализации АЛУ

Использование вышеупомянутого ГОСТ и следующего описания элементов поможет реализовать схему АЛУ.

## Пример элементной базы

### Регистр хранения информации

В качестве регистра хранения информации можно использовать восьмиразрядный регистр 74HC373.

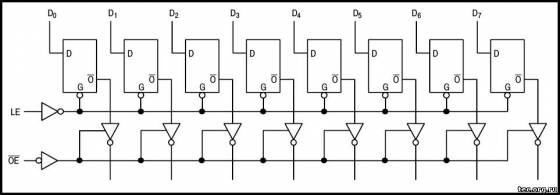
[](http://tec.org.ru/_bd/4/8)

Рисунок 4 - Функциональная схема регистра хранения информации

Таблица 1 - Таблица состояний микросхемы 74HC373

|  |  |  |  |
| --- | --- | --- | --- |
| Входы | | | Выходы |
| OE | LE | Dn | On |
| 1 | X | X | Z |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 0 | X | Oo |

* 0 - низкий уровень
* 1 - высокий уровень
* Z - высокоимпедансное состояние
* X - любое состояние
* Oo - предыдущее состояние выхода

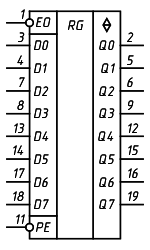


Рисунок 5 – Условное графическое обозначение регистра 74HC373

### 

### Сумматор

В качестве сумматора можно использовать микросхему 74HC283, который представляет собой четырехзначный двоичный сумматор с ускоренным переносом

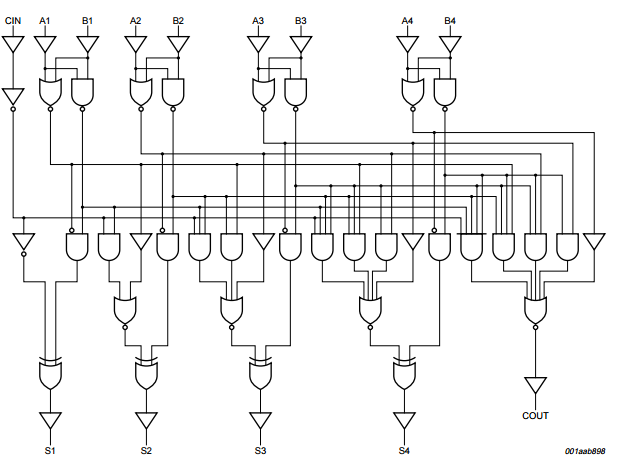


Рисунок 6 – Функциональная схема микросхемы 74HC283

Таблица 2 - Таблица состояний микросхемы 74HC283

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Входы | | | Выходы | |
| Ai | Bi | C0 | C1 | Si |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

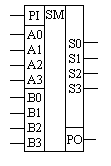


Рисунок 7 – Условное графическое обозначение микросхемы 74HC283

### Компаратор

Для сравнения чисел в качестве компаратора можно использовать микросхему 7485 (4-хразрядный компаратор).

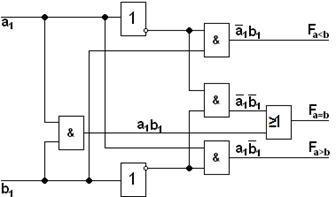
[](http://vk.com/photo93429965_343)

Рисунок 8- Функциональная схема регистра 7485

Таблица 3 - Таблица состояний компаратора 7485

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Входы | | Выходы | | |
| x | y | FA=B | FA>B | FA<B |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |

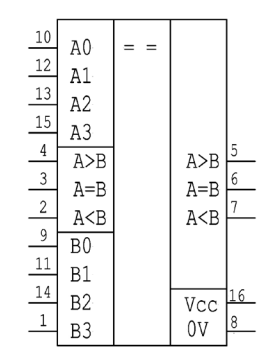


Рисунок 9 – Условное графическое обозначение микросхемы 7485

### 

### 

### Инвертор

В качестве инвертора можно использовать микросхему 74ALS которая осуществляет преобразование логической единицы в логический ноль и наоборот.



Рисунок 10- УГО инвертора.

Таблица 4 - Таблица состояний логического элемента 74ALS

|  |  |
| --- | --- |
| Вход | Выход |
| x | y |
| 0 | 1 |
| 1 | 0 |

### Дешифратор

В качестве дешифратора в управленческой части можно использовать дешифратор 74HC4028.

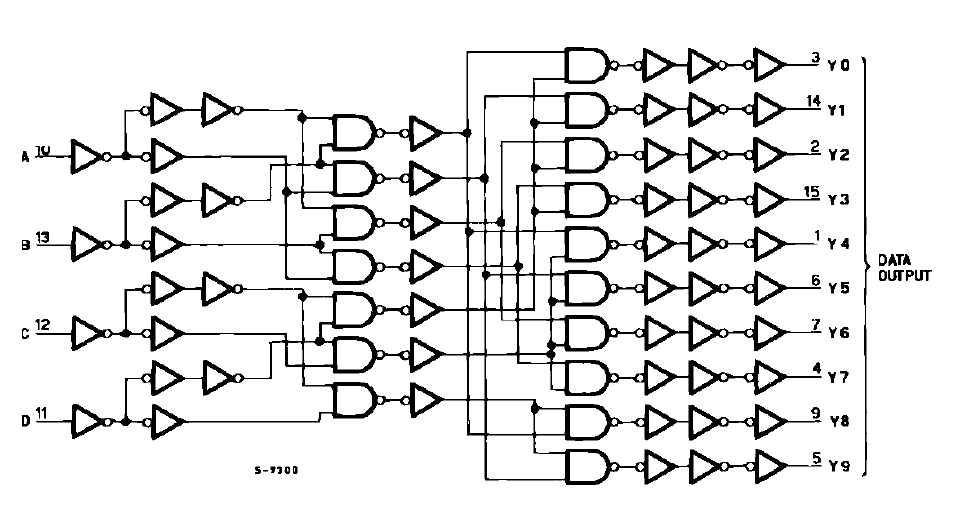


Рисунок 11 – Функциональная схема микросхемы 74HC4028

Таблица 5 - Таблица состояний микросхемы 74HC4028

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **D** | **C** | **B** | **A** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



Рисунок 12 – Условное графическое обозначение микросхемы 74HC138

### Триггер

В качестве триггера можно использовать микросхему CD4013B, которая является двухступенчатым D – триггером flip-flop.

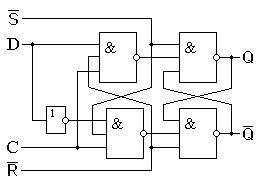


Рисунок 13 – Функциональная схема микросхемы CD4013B

Таблица 6 - Таблица состояний D– триггера

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Режим работы | Входы | | | | Выходы | |
| Асинхронный | | Синхронный | |
| S | R | C | D | Q | !Q |
| Асинхронная установка 1 | 0 | 1 | x | x | 1 | 0 |
| Асинхронная установка 0 | 1 | 0 | x | x | 0 | 1 |
| Запрещенное состояние | 0 | 0 | x | x | 1 | 1 |
| Установка 1 | 1 | 1 | ↑ | 1 | 1 | 0 |
| Установка 0 | 1 | 1 | ↑ | 0 | 0 | 1 |

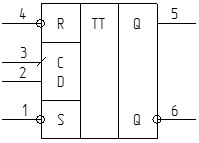


Рисунок 14 – Условное графическое обозначение микросхемы CD4013B

### Логический элемент И

В качестве этого элемента можно использовать микросхему 74HC08. При подаче напряжения низкого уровня на один или оба входа каждого элемента на выходе микросхемы 74HC08 устанавливается напряжение низкого уровня. Если на оба входа подается напряжение высокого уровня, то на выходе формируется напряжение высокого уровня

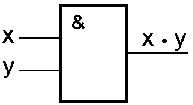


Рисунок 15 – Условное графического обозначение логического элемента И

.

Таблица 7 - Таблица истинности микросхемы 74HC08

|  |  |  |
| --- | --- | --- |
| Входы | | Выходы |
| A | B | Y |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

## Выполнение математических операций

В каждом варианте необходимо реализовать 4 различные операции. Рассмотрим простейший пример выполнения операции сложения с помощью АЛУ.

**Описание операции:**

На вход поступают 2 числа: x и y. Результатом выполнения операции является переменная rez. Переменная x записывается в регистр W. При использовании операции ADDWF над числами x и y в переменную y записывается значение, равное сумме x и y. Далее новое значение переменной y присваивается переменной rez.

**Пример работы операции:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | y | W | Выполнение операции | rez |
| 00000101 | 00000001 | 00000101 |  | 00000110 |

## Приемы реализации АЛУ

Построение схемы АЛУ в данной курсовой работе выполняется в среде разработки Proteus.

Изначальное задание операндов в регистровой памяти, задание кода операции и адресов операндов проще всего реализовать с использованием генераторов шаблонов PatternGenerator, при преобразовании модели в электрическую принципиальную схему, информационные входы подсоединяются к некоторому условному разъему, как и информационные выходы результирующего регистра. К этому же разъему подсоединяются управляющие входы, по которым задаются код операции и адреса операндов в регистровой памяти. Генераторы шаблонов в электрической принципиальной схеме не отображаются. Для оптимизации схемы необходимо использовать шину.

### Работа с шиной в Proteus

Для того чтобы пользоваться шиной, необходимо в боковом меню выбрать BusesMode, как показано на рисунке:

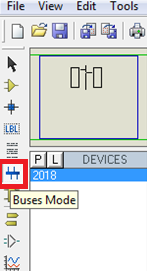


Рисунок 16 – Работа с шиной

В поле создания схемы нужно установить курсор и выполнить однократное нажатие левой кнопкой мыши, после чего можно провести линию (шину) необходимой длины. Так же однократным кликом левой кнопки мыши указываются повороты шины, а двойным – конец шины. Далее можно поставить 2 элемента, входы/выходы которых будут соединяться.

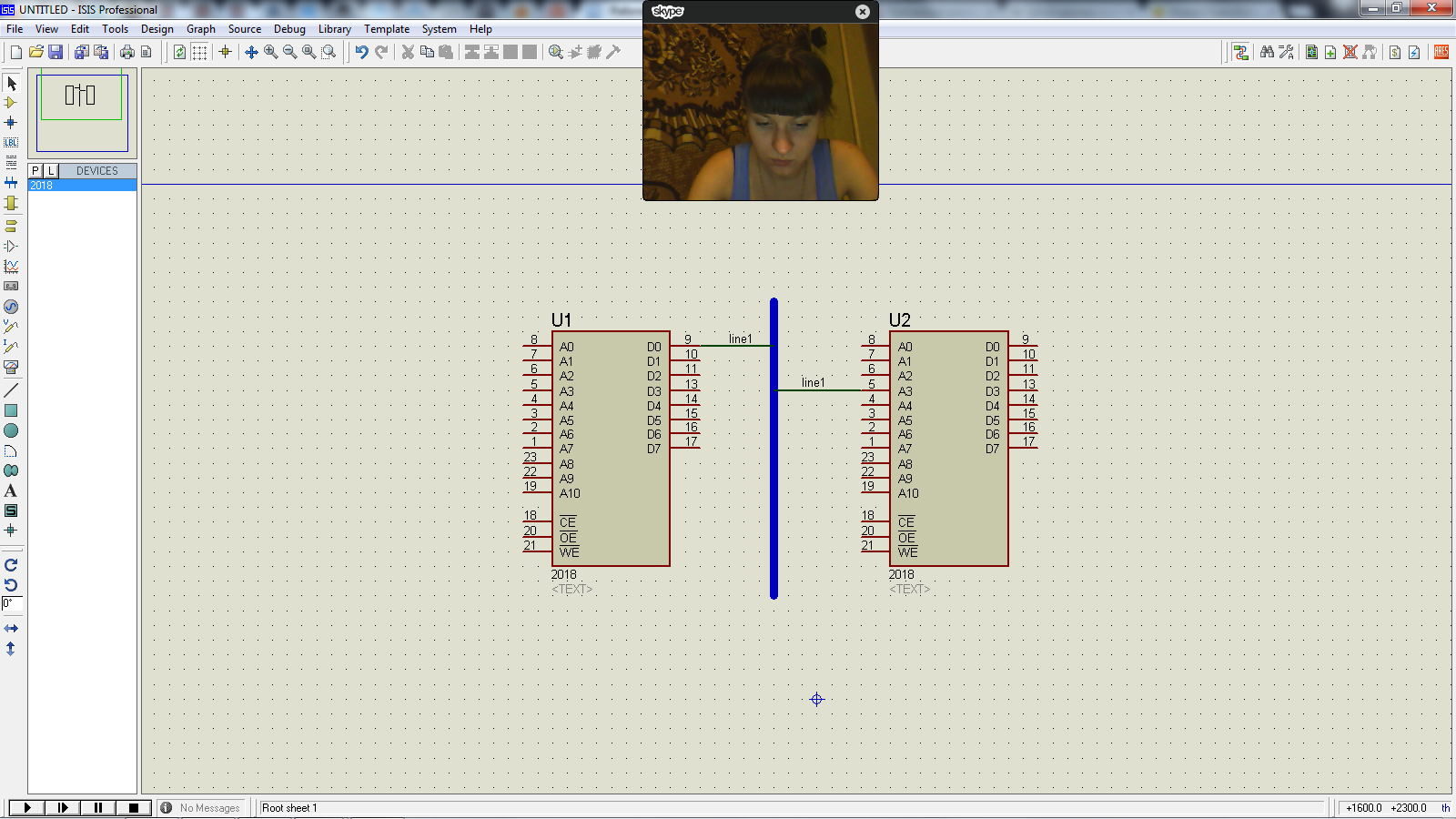
****

Рисунок 17 – Обозначение контактов

Для того чтобы подать сигнал с выхода «9» регистра U1 на вход «5» регистра U2, следует провести провода от «9» и «5» к шине, после чего добавить для них одинаковый Label. Label создаётся при помощи клика правой кнопки мыши по проводу и выбора в появившемся меню пункта «PlaceWireLabel»

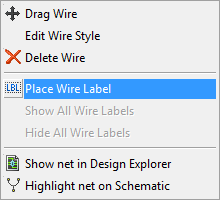


Рисунок 18 – Задание меток контактов

Использование шины делает схему менее нагруженной, упрощает процесс ее создания, а так же облегчает понимание схемы.

# Индивидуальные задания

Выполняются на основе системы команд PIC18FXX 2 (см [стр.204-205 datasheet](http://www.microchip.ru/files/d-sheets-rus/PIC18FXX2_manual.pdf)), и процессоров Intel, так же каждому отдельно выдается определенный вид команды деления.

Таблица - 1

|  |  |
| --- | --- |
| Вар. | Команды |
| 1. | Add, Shl, Xor |
| 2. | Shr, Neg, adc |
| 3. | Sal, [XCHG](http://asmworld.ru/spravochnik-komand/xchg/), sub |
| 4. | Sar, OR, TEST |
| 5. | Ror, NOT, SBB |
| 6. | Rol, And, BTC |
| 7. | IMUL, dec, BTR |
| 8. | Or, inc, Add |
| 9. | [XCHG](http://asmworld.ru/spravochnik-komand/xchg/), mul, TEst |
| 10. | SBB, CMP, sHRD |
| 11. | ADD, CBW, Sal |
| 12. | XOR, IMUL, Ror |

Таблица -2

|  |  |
| --- | --- |
| Вар. | Команды PIC |
|  | ADDWF, RRCF, IORLW |
|  | NEGF, SUBLW, RLNCF |
|  | RLCF, MULLW, BTFSS |
|  | MULWF, INFSNZ, BTG, |
|  | ADDLW, SETF, IORWF |
|  | SUBWF, CLRF, INCF |
|  | SUBFWB, BCF, ANDWF |
|  | COMF, BTFSS, IORLW |
|  | XORWF, RRNCF, SUBLW |
|  | COMF, ANDLW, SETF |
|  | BSF, RLNCF, ADDWFC |
|  | ADDWF, IORWF, DECF |

# Литература

## Основная литература

1. Громов, Ю. Ю. Архитектура ЭВМ и систем: учебное пособие [Электронный ресурс]:/ Ю.Ю. Громов, О. Г. Иванова, М. Ю. Серегин, М. А. Ивановский, В. Е. Дидрих, — Электрон. дан. — Тамбов: Издательство ФГБОУ ВПО «ТГТУ», 2012 - 200 с. - Режим доступа:

[http://biblioclub.ru./index.php?page=book\_red&id=277352&sr=1#](http://biblioclub.ru./index.php?page=book_red&id=277352&sr=1)

2. Бройдо В.Л., Ильина О.П. Архитектура ЭВМ и систем. СПб.: Питер, 2009. - 720с.

## Дополнительная литература

3. Усачев, Ю.Е. Вычислительные машины, сети и системы телекоммуникаций [Электронный ресурс] : учебное пособие / Ю.Е. Усачев, И.В. Чигирёва. — Электрон. дан. — Пенза : ПензГТУ (Пензенский государственный технологический университет), 2014. — 313 с. — Режим доступа: <http://e.lanbook.com/books/element.php?pl1_id=62577>

## Электронные ресурсы

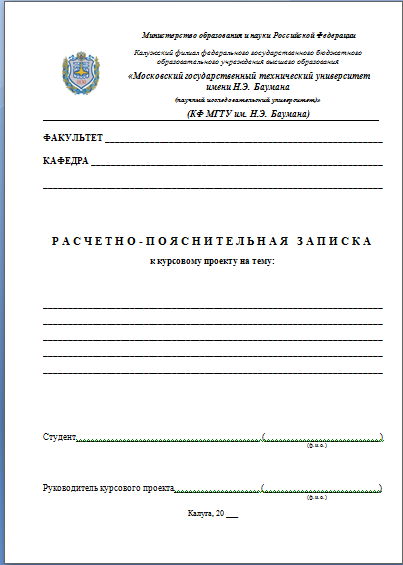
4. Научная электронная библиотека: [http://eLIBRARY.RU](http://elibrary.ru/).

5. Издательство «Лань». Электронно-библиотечная система. <http://e.lanbook.com>.

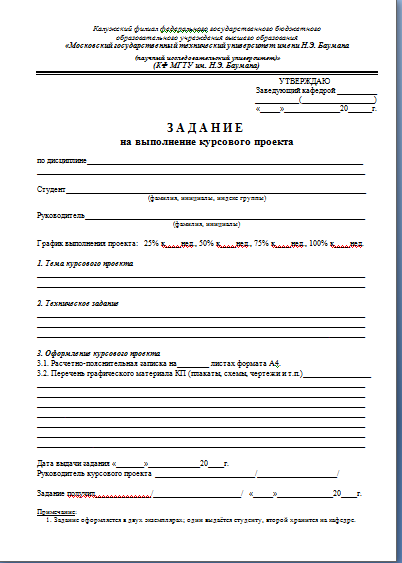
6. Научная электронная библиотека: <http://biblioclub.ru>

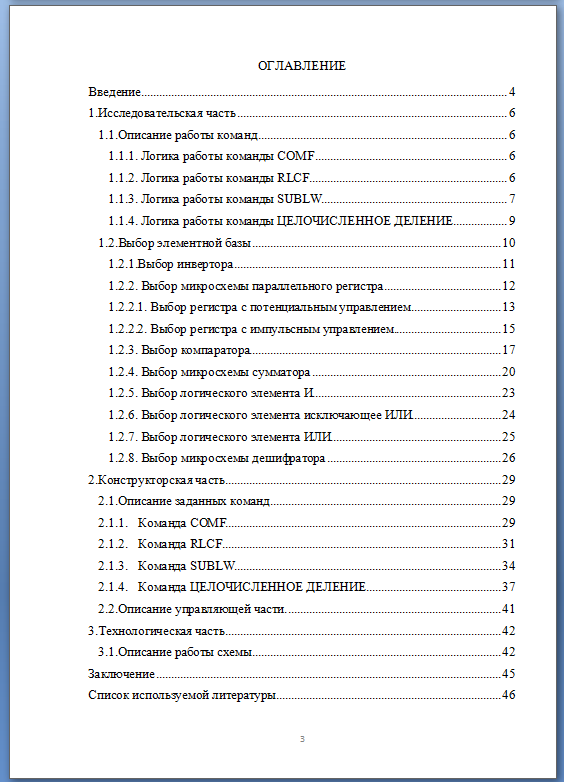
# Приложение 1. Оформление курсового проекта

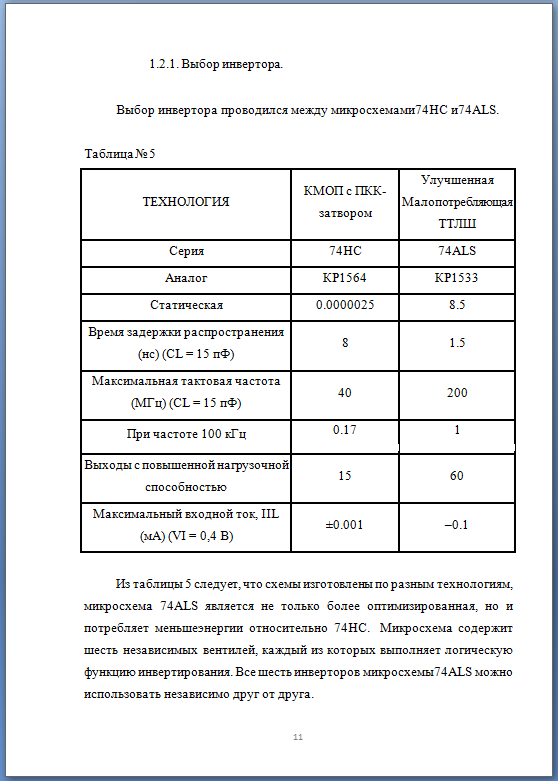
## Пример титульного листа

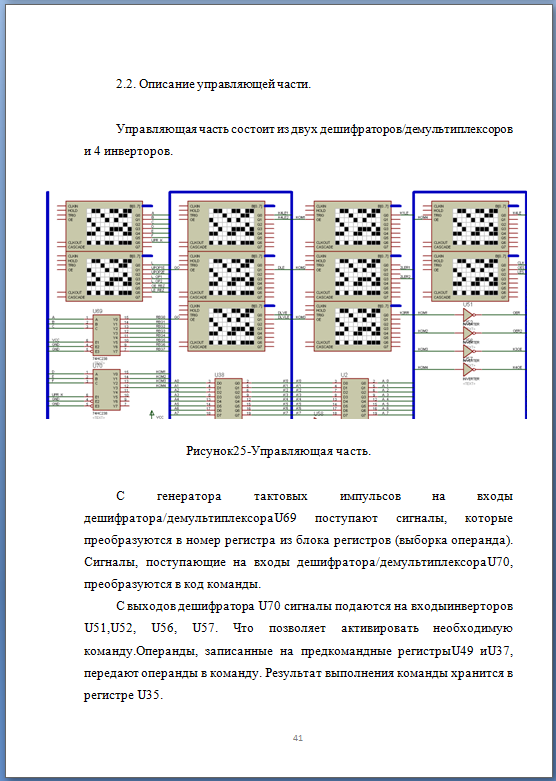
****

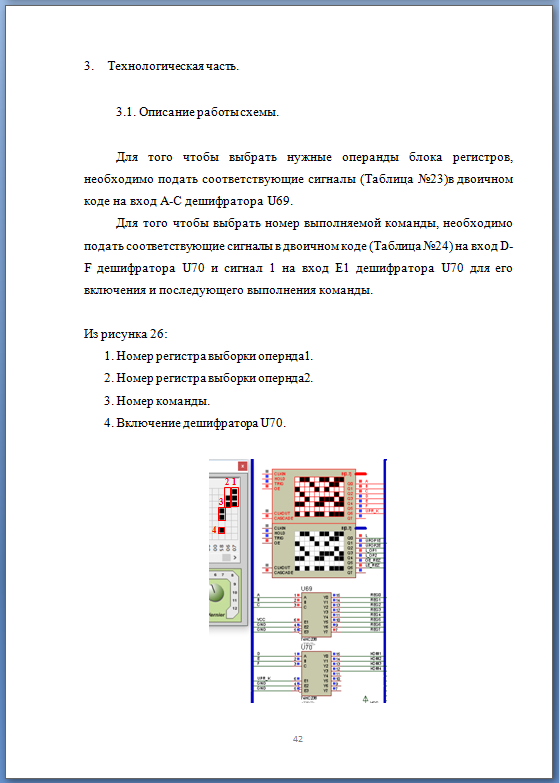
## Пример оформления технического задания на курсовой проект

****

****

****

****

****

# Приложение 2. Пример оформления графической части.http://cs610031.vk.me/u123884010/docs/fc2ce5921ce4/Skhema_funktsionalnaya.jpg?extra=Hr6631Kn81viIgR4dicndCYzPoAc6Oox-h47NK6ObrR0Zy3XsogfWb7hF40ucvbWYjBmGjBOwBcThksEprlXTw3mj-jN

