

**МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра космического приборостроения и средств связи

УТВЕРЖДАЮ: Западный  
государственный  
университет»  
Проректор по учебной работе  
(ЮЗГУ)

О.Б. Локтионова  
« 31 » 04 2019 г.

**СХЕМОТЕХНИКА ТЕЛЕКОММУНИКАЦИОННЫХ  
УСТРОЙСТВ**

Методические указания к лабораторным занятиям

Курск – 2019

УДК 681.3

Составитель О. Г. Бондарь, Е.О. Брежнева

Доктор технических наук, профессор кафедры вычислительной техники *И. Е. Чернецкая*

**Схемотехника телекоммуникационных устройств :**  
Методические указания к лабораторным занятиям / Юго-Зап. гос. ун-т; сост.: О. Г. Бондарь, Е. О. Брежнева. – Курск, 2019. – 73 с.: ил. 38. табл.4: Библиогр.: с. 73.

Приводятся сведения по применению средств исследования электронных устройств и варианты заданий на исследование характеристик электронных устройств.

Методические указания соответствуют требованиям программы, утвержденной учебно-методическим объединением по специальностям автоматике и электроники (УМО АЭ).

Предназначены для обучающихся по направлению подготовки 11.03.02 Инфокоммуникационные технологии и системы связи всех форм обучения.

Текст печатается в авторской редакции

Подписано в печать 31.09.19. Формат 60×84 1/16.  
Усл. печ. л. 4,24. Уч.-изд. л. 3,84. Тираж 100 экз. Заказ 49.  
Бесплатно.

Юго-Западный государственный университет.  
305040, г. Курск, ул. 50 лет Октября, 94

## Оглавление

ВВЕДЕНИЕ.....	8
СРЕДА MULTISIM.....	8
Интерфейс программы .....	8
Обзор компонентов.....	10
Виртуальные приборы.....	11
Мультиметр .....	11
Генератор сигналов.....	12
Осциллограф.....	12
Построитель частотных характеристик (Bode Plotter).....	13
Спектральный анализатор.....	14
Ваттметр.....	14
Токовый пробник .....	15
Измерительный пробник.....	15
Генератор слов (Word Generator) .....	15
Логический анализатор (Logic Analyzer) .....	17
Создание схем .....	20
Анализ электронных устройств.....	21
Postprocessor и Grapher .....	23
Общие правила моделирования .....	24
Топология схем .....	25
1    ПРОЕКТИРОВАНИЕ И ИССЛЕДОВАНИЕ КОМБИНАЦИОННОГО УСТРОЙСТВА.....	26
Цель работы.....	26
Подготовка к лабораторной работе .....	26
Программа исследований.....	27
Методические указания.....	28
Контрольные вопросы .....	29

Содержание отчета .....	29
Приложение .....	30
<b>2 МИНИМИЗАЦИЯ И ИССЛЕДОВАНИЕ КОМБИНАЦИОННОГО УСТРОЙСТВА.....</b>	<b>31</b>
Цель работы.....	31
Подготовка к лабораторной работе .....	31
Программа исследований.....	32
Контрольные вопросы .....	33
Содержание отчета .....	33
Приложение. Варианты заданий .....	34
<b>3 ПРОЕКТИРОВАНИЕ И ИССЛЕДОВАНИЕ СЕМИСЕГМЕНТНОГО ДЕШИФРАТОРА.....</b>	<b>35</b>
Цель работы.....	35
Подготовка к лабораторной работе .....	35
Программа исследований.....	36
Контрольные вопросы .....	36
Содержание отчета .....	37
Приложение. Варианты заданий .....	37
<b>4 ПРОЕКТИРОВАНИЕ И ИССЛЕДОВАНИЕ RS-ТРИГГЕРА .....</b>	<b>38</b>
Цель работы.....	38
Подготовка к лабораторной работе .....	38
Программа исследований.....	39
Контрольные вопросы .....	40
Содержание отчета .....	40
<b>5 ИССЛЕДОВАНИЕ СДВИГОВОГО РЕГИСТРА .....</b>	<b>41</b>
Цель работы.....	41
Основные сведения.....	41
Оборудование .....	43

Выполнение лабораторной работы .....	45
Исследование регистра сдвига в статическом режиме .....	46
Режим сдвига вправо .....	46
Режим сдвига влево .....	47
Режим параллельной загрузки.....	48
Режим хранения .....	49
Исследование регистра сдвига в динамическом режиме ....	50
Содержание отчёта .....	51
Контрольные вопросы .....	51
<b>6 ИССЛЕДОВАНИЕ ДВОИЧНОГО СЧЁТЧИКА .....</b>	<b>52</b>
Цель работы.....	52
Основные сведения.....	52
Оборудование.....	55
Выполнение лабораторной работы.....	56
Исследование двоичного счетчика в статическом режиме.....	57
Исследование счётчика в динамическом режиме .....	59
Содержание отчёта .....	60
Контрольные вопросы .....	60
<b>7 СРЕДСТВА РАЗРАБОТКИ МИКРОСИСТЕМ НА МИКРОКОНТРОЛЛЕРАХ AVR .....</b>	<b>61</b>
1 Цель работы.....	61
2 Краткие сведения о микроконтроллерах AVR .....	61
2.1 Общие особенности микроконтроллеров AVR.....	61
2.2 Организация адресного пространства микроконтроллера AT90S2313 .....	69
2.2 Загружаемая память программ.....	70
2.3 EEPROM память данных .....	70
2.4 Статическое ОЗУ данных .....	71
2.5 Файл регистров общего назначения .....	71

2.7	Пространство ввода/вывода.....	73
3	Инструменты разработки и отладки программного обеспечения.....	75
3.1	Общая характеристика аппаратно-программных средств разработки программного обеспечения.....	76
3.2	Создание/загрузка проекта в ASTUDIO4.....	82
3.3	Ассемблер AVR .....	83
3.4	Отладка программы .....	86
3.5	Программирование микроконтроллера в составе прототипной системы .....	88
4	Подготовка к лабораторной работе .....	90
5	Программа исследований и порядок работы.....	92
6	Контрольные вопросы .....	93
8	ОРГАНИЗАЦИЯ ЦИФРОВОГО ВВОДА/ВЫВОДА В СИСТЕМАХ НА МИКРОКОНТРОЛЛЕРАХ AVR.....	95
1	Цель работы.....	95
2	Порты ввода/вывода микроконтроллеров AVR .....	95
2.1	Организация портов ввода/вывода .....	95
2.1	Адресация портов ввода/вывода микроконтроллера AT90S2313.....	96
2.2	Работа с портами ввода/вывода.....	97
3	Подготовка к лабораторной работе .....	98
4	Программа исследований и порядок работы.....	100
5	Методические указания.....	101
6	Контрольные вопросы .....	102
7	Содержание отчёта .....	102
3	ПОДПРОГРАММЫ И СТЕК.....	103
1	Цель работы.....	103
2	Организация стека в микроконтроллерах AVR.....	103

3 Подготовка к лабораторной работе .....	105
4 Программа исследований и порядок работы .....	108
5 Контрольные вопросы .....	109
6 Содержание отчёта .....	109
Литература .....	111

## **ВВЕДЕНИЕ**

Лабораторный цикл нацелен на получение начальных навыков проектирования и исследования цифровых устройств, а также знакомит с архитектурой микроконтроллеров на ядре AVR фирмы Microchip (ATMEL), средой разработки программного обеспечения AStudio, основами программирования на языке ассемблера.

Поскольку проектирование и исследование цифровых устройств осуществляется в программной среде, то методические указания содержат введение в программную среду проектирования электронных устройств Multisim фирмы National Instruments.

## **СРЕДА MULTISIM**

NI Multisim позволяет объединить процессы разработки электронных устройств и тестирования на основе технологии виртуальных приборов для учебных и производственных целей. Именно технология виртуальных приборов является основной отличительной особенностью данного продукта. Multisim обеспечивает возможность интерактивного SPICE-моделирования и анализа электрических цепей, используемых в схемотехнике аналоговых, цифровых и смешанных устройств. Этот инструмент позволяет, с одной стороны, сделать очень наглядным изучение дисциплины, а с другой стороны, подготовиться к работе в реальной лаборатории, изучив методику планирования и проведения экспериментов.

### **Интерфейс программы**

Multisim является программой с многооконным графическим интерфейсом, позволяющим строить и редактировать схемы, модели и изображения компонентов, а также представлять результаты расчетов в удобном графическом виде. Пользовательский интерфейс программы показан на рисунке 1 и состоит из следующих элементов:

1 – *строка меню* (главное меню) позволяет выбирать команды для всех функций;



- 2 – панель разработки позволяет управлять различными элементами схемы;
- 3 – панель инструментов состоит из кнопок для быстрого доступа к командам и элементам меню;
- 4 – окно редактирования (рабочая область);
- 5 – приборная панель состоит из набора кнопок для доступа к моделям контрольно-измерительных приборов.

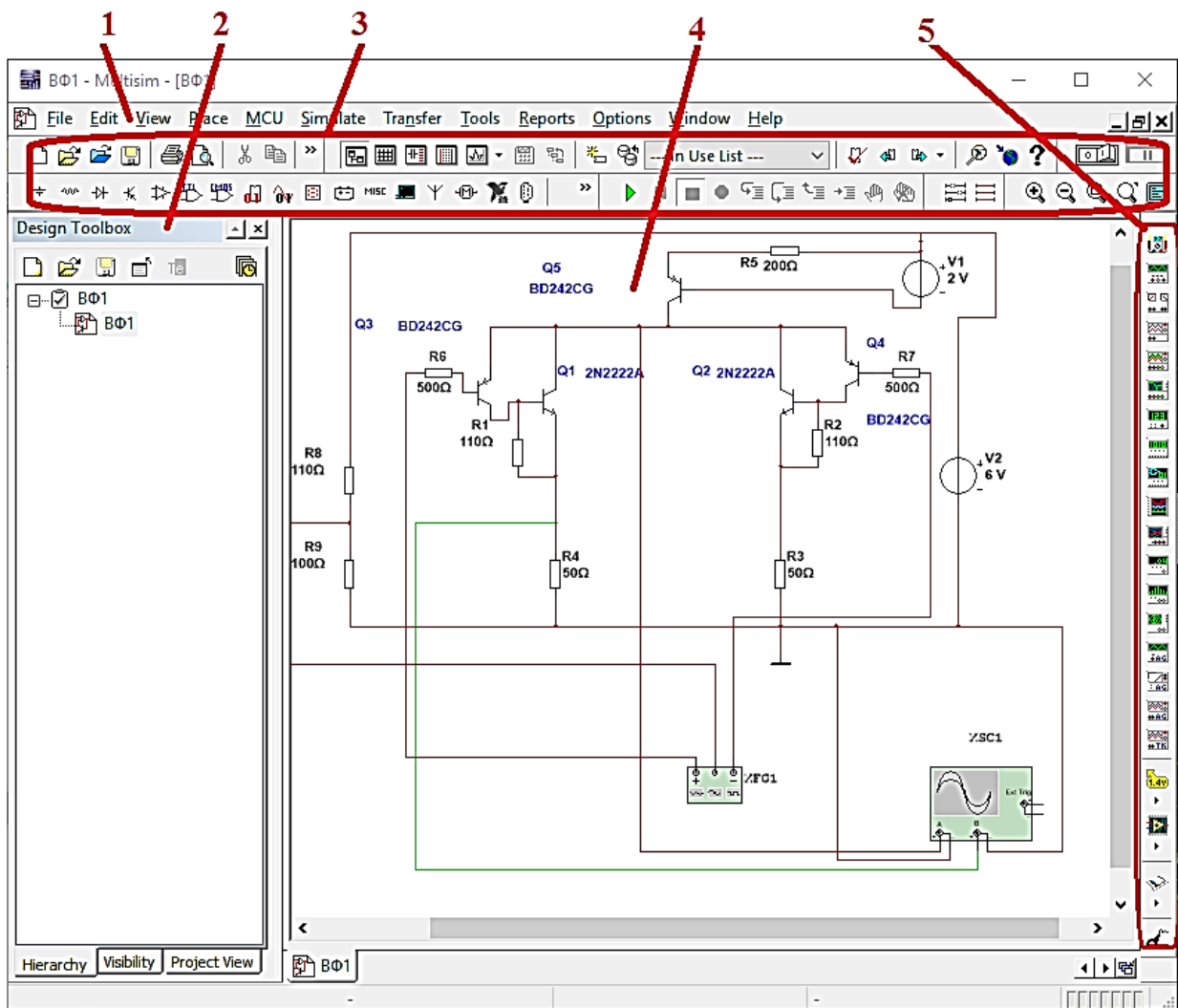


Рисунок 1 – Интерфейс Multisim

## Обзор компонентов

Как известно все электронные устройства состоят из компонентов (резисторы, конденсаторы, катушки индуктивности, диоды, транзисторы и т.п.).

Multisim имеет базы данных трёх уровней:

- Из Главной базы данных ( Master Database) можно только считывать информацию, в ней находятся все компоненты;
- Пользовательская база данных ( User Database) соответствует текущему пользователю компьютера. Она предназначена для хранения компонентов, которые нежелательно предоставлять в общий доступ;
- Корпоративная база данных (Corporate Database). Предназначена для тех компонентов, которые должны быть доступны другим пользователям по сети.

Средства управления базами данных позволяют перемещать компоненты, объединять две базы в одну и редактировать их. Все базы данных разделяются на группы, а они, в свою очередь, на семейства. Когда пользователь выбирает компонент и помещает его в схему, создается новая копия. Все изменения с ней никак не затрагивают информацию, хранящуюся в базе данных.

База данных Master Database разделена на группы.

- **Sources.** Содержит все источники напряжения и тока, заземления. Например, power sources (источники постоянного, переменного напряжения, заземление, беспроводные соединения - VCC, VDD, VSS, VEE), signal voltage sources (источники прямоугольных импульсов, источник сигнала через определенные промежутки времени), signal current sources (постоянные, переменные источники тока, источники прямоугольных импульсов).
- **Basic.** Содержит основные элементы схемотехники: резисторы, индуктивные элементы, емкостные элементы, ключи, трансформаторы, реле, коннекторы и т.д.
- **Diodes.** Содержит различные виды диодов: фотодиоды, диоды

- Шоттки, светодиоды и т.д.
- **Transistors.** Содержит различные виды транзисторов: pnp-, npn-транзисторы, биполярные транзисторы, МОП-транзисторы, КМОП-транзисторы и т.д.
  - **Analog.** Содержит все виды усилителей: операционные, дифференциальные, инвертирующие и т.д.
  - 
  - **TTL.** Содержит элементы транзисторно-транзисторной логики.
  - **CMOS.** Содержит элементы КМОП-логики.
  - **MCU Module** – управляющий модуль многопунктовой связи (от англ. multipoint control unit).
  - **Advanced\_Peripherals.** Содержит подключаемые внешние устройства (дисплеи, терминалы, клавишные поля).
  - **Misc Digital.** Содержит различные цифровые устройства.
  - **Mixed.** Содержит комбинированные компоненты
  - **Indicators.** Содержит измерительные приборы (вольтметры, амперметры), лампы и т.д.

## Виртуальные приборы

Все приборы расположены на панели инструментов. Рассмотрим основные.

### Мультиметр

Мультиметр (рисунок 2) предназначен для измерения переменного или постоянного тока, или напряжения, сопротивления или затухания между двумя узлами схемы.

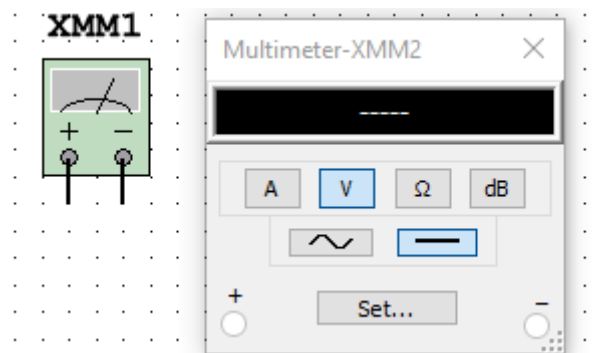


Рисунок 2 - Мультиметр

Диапазон измерений мультиметра подбирается автоматически. Его внутреннее сопротивление и ток близки к идеальным значениям, но их можно изменить.

### Генератор сигналов

Генератор сигналов (function generator) – это источник напряжения, который может генерировать синусоидальные, пилообразные и прямоугольные импульсы (рисунок 3). Можно изменить форму сигнала, его частоту, амплитуду, коэффициент заполнения и постоянный сдвиг. Диапазон генератора достаточен, чтобы воспроизвести сигналы с частотами от нескольких герц до аудио и радиочастотных.

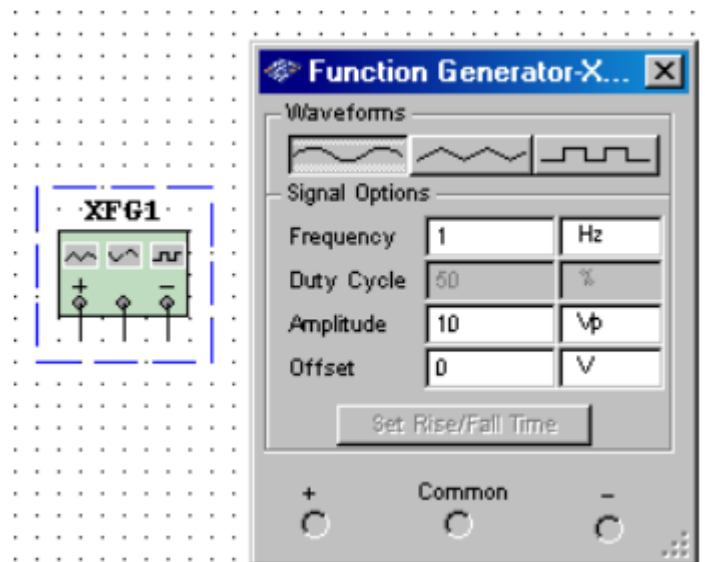


Рисунок 3 – Генератор сигналов

### Осциллограф

В Multisim есть несколько модификаций осциллографов, которыми можно управлять как настоящими. Они позволяют устанавливать параметры временной развертки и напряжения, выбирать тип и уровень запуска измерений. Данные осциллографов можно посмотреть после эмуляции с помощью самописца (Grapher) из меню Вид\Плоттер (View/Grapher).

В Multisim есть следующие осциллографы:

- 2-х канальный (рисунок 4)

- 4-х канальный
- осциллограф смешанных сигналов Agilent 54622D
- 4-х канальный цифровой осциллограф с записью Tektronix TDS 2024.

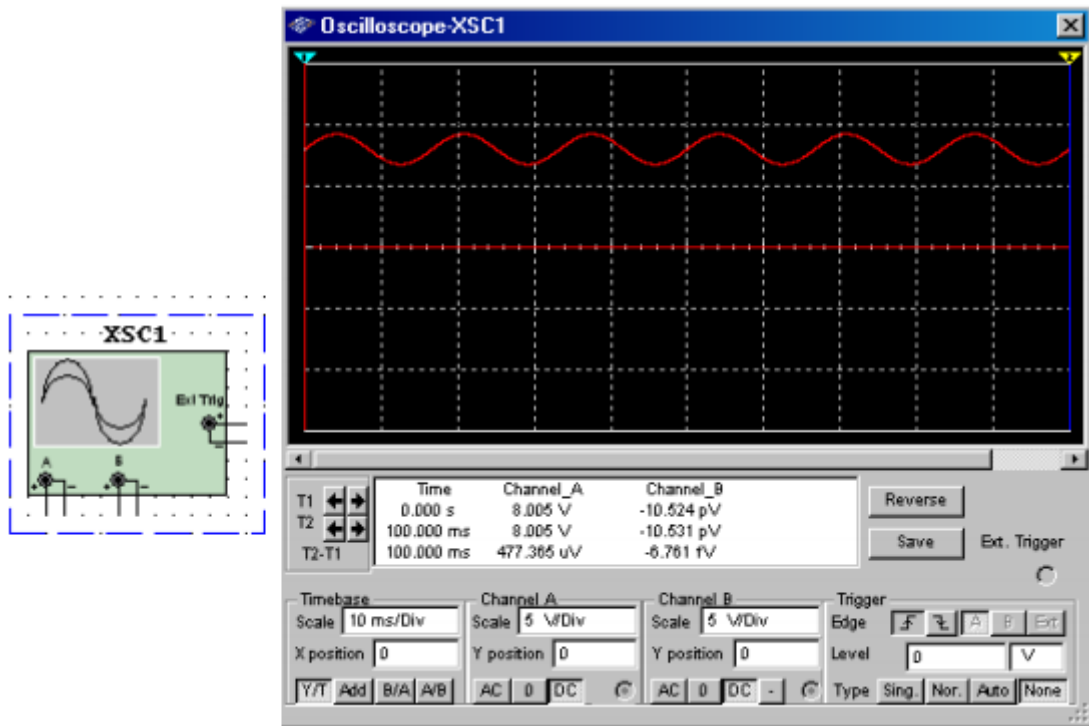


Рисунок 4 – Двухканальный осциллограф

### Построитель частотных характеристик (Bode Plotter)

Отображает относительный фазовый или амплитудный отклик входного и выходного сигналов (рисунок 5). Это особенно удобно при анализе свойств полосовых фильтров.



Рисунок 5 – Построитель частотных характеристик

## Спектральный анализатор

Спектральный анализатор (spectrum analyzer) служит для измерения амплитуды гармоник с заданной частотой (рисунок 6). Также он может измерить мощность сигнала и частотных компонент, определить наличие гармоник в сигнале.

Результаты работы спектрального анализатора отображаются в частотной области, а не временной. Обычно сигнал - это функция времени, для её измерения используется осциллограф. Иногда ожидается синусоидальный сигнал, но он может содержать дополнительные гармоники, в результате, невозможно измерить уровень сигнала. Если же сигнал измеряется спектральным анализатором, получается частотный состав сигнала, то есть определяется амплитуда основной и дополнительных гармоник.

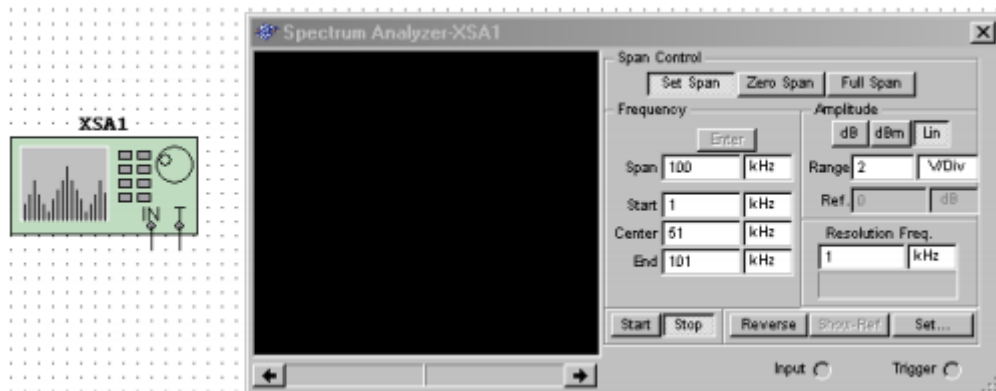


Рисунок 6 – Спектральный анализатор

## Ваттметр

Прибор предназначен для измерения мощности и коэффициента мощности (рисунок 7).

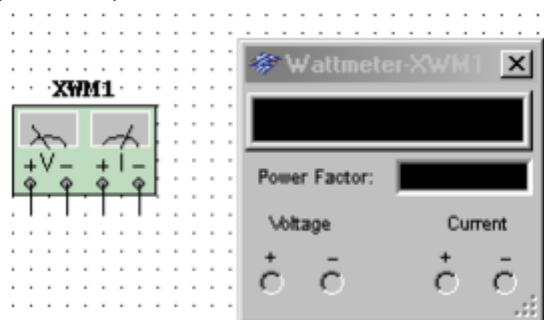


Рисунок 7 - Ваттметр

### Токовый пробник

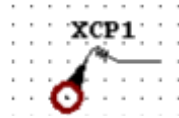


Рисунок 8 – Токовый пробник

### Измерительный пробник

Показывает постоянные и переменные напряжения и токи на участке цепи, а также частоту сигнала (рисунок 9).

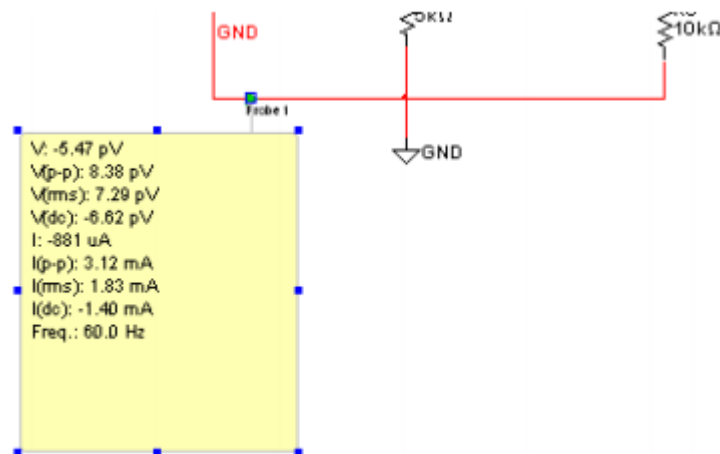


Рисунок 9 – Измерительный пробник

### Генератор слов (Word Generator)

Генератор слов предназначен для генерации 32-разрядных двоичных слов (рисунок 10). Ввод генерируемых слов производится в буфере ввода. Формат отображения кодовых слов выбирается с помощью группы кнопок Display (Hex – шестнадцатеричный, Dec – десятичный, Binary – двоичный, ASCII – символьный). Частота генерации кодовых слов задается в окне Frequency (Частота) и лежит в диапазоне от 1 Гц до 1000 МГц. В процессе работы на каждом выводе генератора появляется логический уровень согласно разряду двоичного кодового слова, при этом генератор работает в трех режимах:

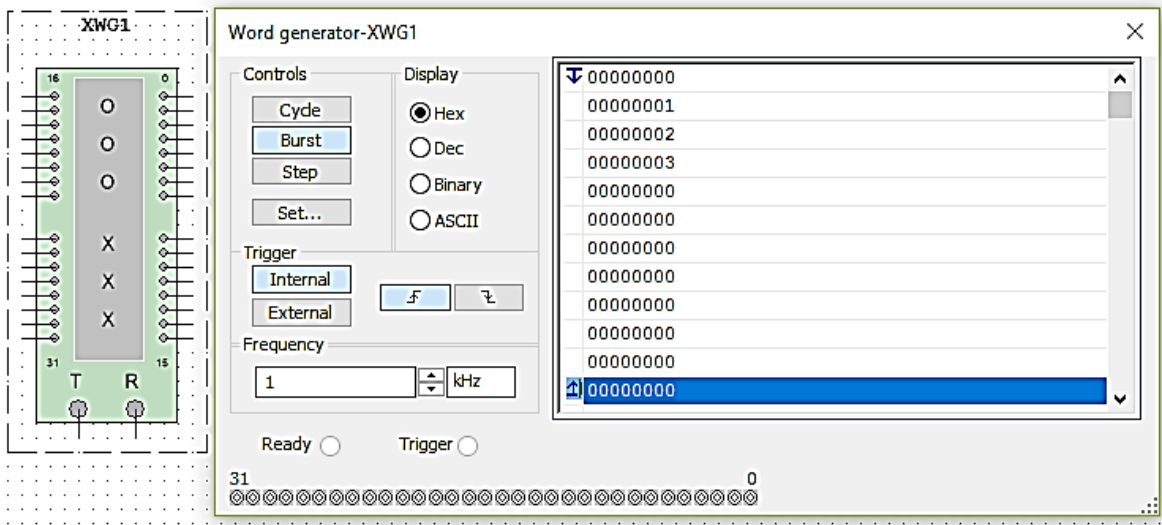


Рисунок 10 – Генератор слов

- Step (Пошаговый) – каждый раз при подаче очередного слова на выход моделирование останавливается;
- Burst (Пакетный) – генерируется последовательность кодовых слов, начиная с начальной позиции и заканчивая конечной позицией, моделирование останавливается при достижении конечной позиции;
- Cycle (Циклический) – на выводах генератора последовательно появляются логические уровни согласно комбинации слов, генерирование осуществляется до тех пор, пока не будет остановлено моделирование или достигнута точка прерывания (Breakpoint).

Во время моделирования курсор в окне «буфер вывода» указывает на текущее генерируемое слово. Остановив моделирование, можно изменить положение курсора, начальную позицию, конечную позицию, а также точку прерывания. Выходные контакты, пронумерованные как 31,32...1,0 соответствуют битам набранного двоичного кода (0 – младший разряд).

При нажатии кнопки «Set» открывается диалоговое окно свойств буфера (рисунок 11, а):

- No Change – оставить без изменений;
- Load – загрузить кодовые слова из файла (с расширением .dp);
- Save – сохранить кодовые слова в файл;
- Clear buffer – обнулить содержимое буфера;



Up Counter – заполнить буфер кодовыми словами, начиная с кода, указанного в поле Initial Pattern (по умолчанию 0x0000), с последующим увеличением на 1 в каждой следующей строке;

Down Counter – заполнить буфер кодовыми словами, начиная с кода, указанного в поле Initial Pattern (по умолчанию 0x0400), с последующим уменьшением на 1 в каждой следующей строке;

Shift Right – заполнить буфер кодовыми словами, начиная с кода, указанного в поле Initial Pattern (по умолчанию 0x80000000), с последующим двоичным сдвигом вправо на 1 разряд в каждой следующей строке (рисунок 11, б);

Shift Left – заполнить буфер кодовыми словами, начиная с кода, указанного в поле Initial Pattern (по умолчанию 0x0001), с последующим двоичным сдвигом влево на 1 разряд в каждой следующей строке (рисунок 11, в);

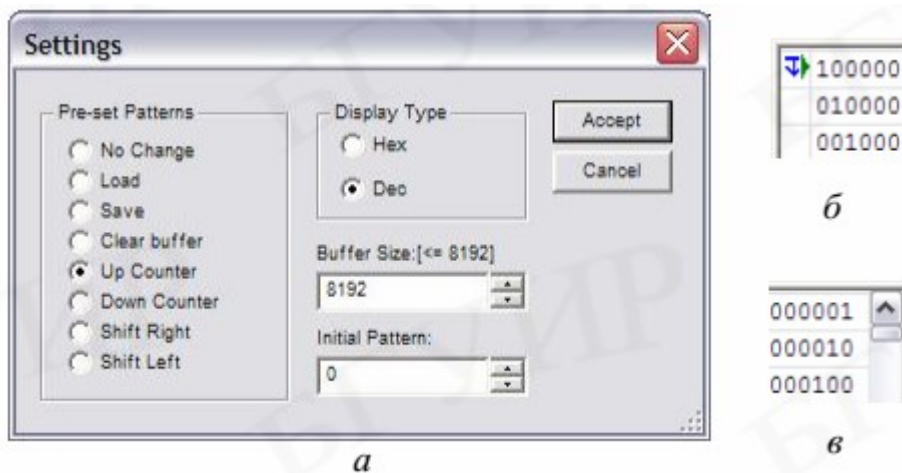


Рисунок 11 – Окно свойств буфера (а) и его заполнение (б, в)

Запуск генератора может синхронизироваться как внутренним (Internal), так и внешним (External) сигналом синхронизации. На вывод Ready подается сигнал готовности.

### Логический анализатор (Logic Analyzer)

Логический анализатор (ЛА) – устройство, предназначенное для диагностики цифровых схем. ЛА позволяет отслеживать и записывать состояния логических элементов цифровых электронных

устройств, анализировать и визуализировать их. Внешний вид и лицевая панель ЛА показаны на рисунке 12.

ЛА имеет 16 каналов для съёма сигналов, а также несколько входов запуска. Кроме этого прибор снабжен двумя курсорами, позволяющими проводить измерения во временной области. Если вход 1 считать младшим разрядом, а вход 16 – старшим, то состояние всех входов может быть представлено 16-разрядным двоичным кодом. Код, соответствующий позиции курсора, отображается в поле «входной код» (рисунок 13).

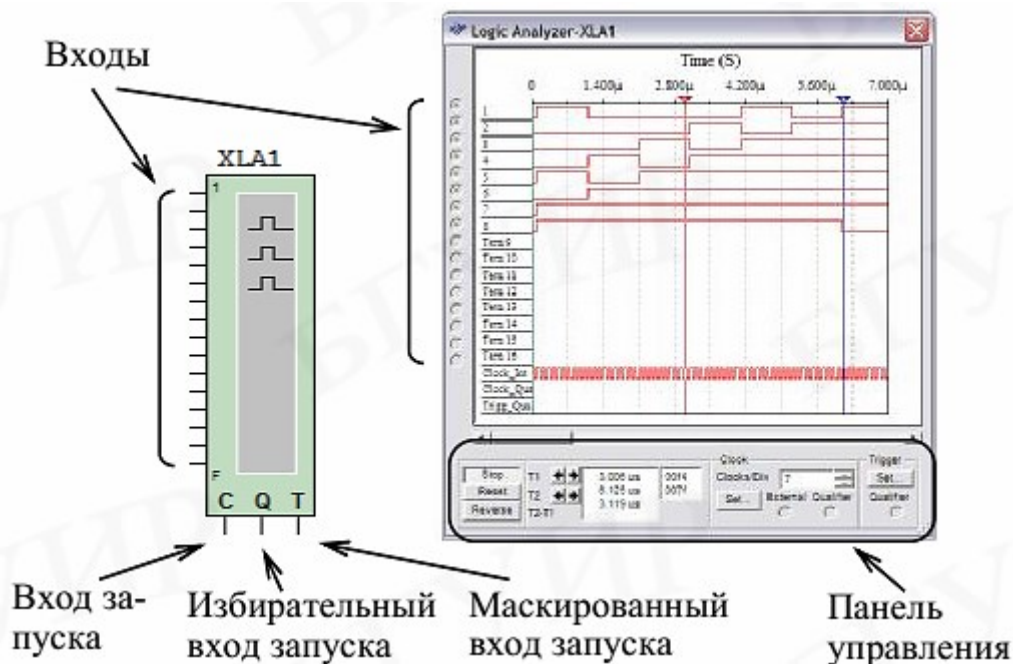


Рисунок 12 – Логический анализатор



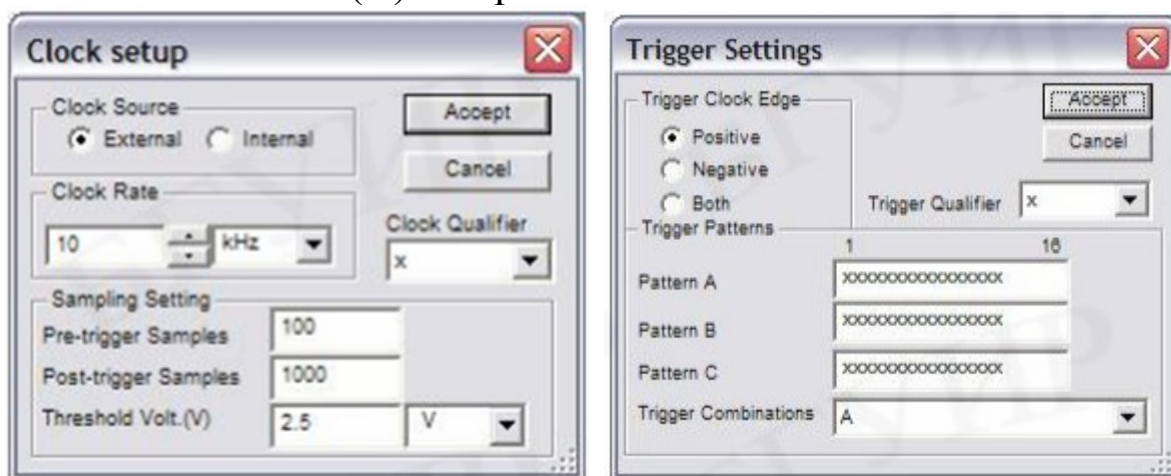
Рисунок 13 – Панель управления логического анализатора

При нажатии кнопки «Set» в группе Clock (тактовый генератор) открывается диалоговое окно настройки параметров тактирования входных сигналов (рисунок 14, а).

Тактирование сигналов осуществляется с использованием внешнего (External) или внутреннего (Internal) источника. В поле Clock Qualifier устанавливается активный уровень сигнала синхронизации. В поле Clock Rate устанавливается частота выборки анализатора.

В группе Sampling Setting задаются параметры выборки сигналов:

- Pre-trigger Samples – сбор данных производится до поступления импульса запуска;
- Post-trigger Samples – сбор данных начинается после поступления импульса запуска и продолжается до тех пор, пока не будет набрано заданное количество отсчетов;
- Threshold Volt (V) – пороговая величина.



а

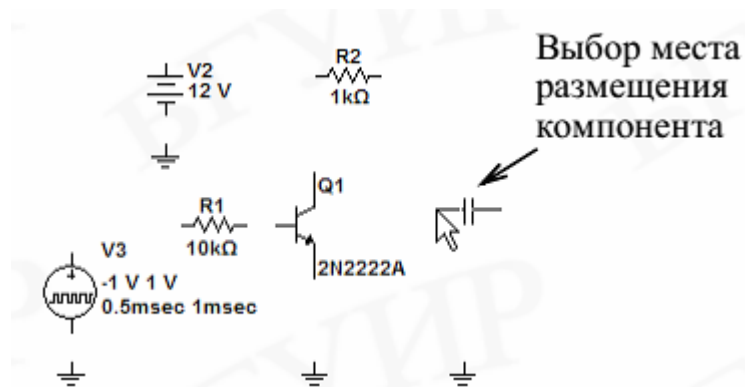
б

Рисунок 14 – Настройка параметров синхронизации

Дополнительные условия запуска анализатора осуществляются с помощью диалогового окна Trigger Settings (рисунок 14, б). В данном окне настраивается маска, по которой осуществляется фильтрация логических уровней и синхронизация входных каналов.

## Создание схем

Рассмотрим процесс создания схемы на примере электронного транзисторного ключа. Процесс создания схемы начинается с выбора компонентов схемы. Выбранный компонент автоматически прикрепляется к курсору мыши. После этого компонент раз-



мещается в любом месте рабочего окна (рисунок 15).

Рисунок 15 – Размещение компонентов схемы

Далее устанавливается ориентация компонентов (если это необходимо) с помощью команд всплывающего меню (нажатие правой кнопки мыши на выбранном компоненте, рисунок 16, *а*) и изменяются параметры элементов питания, рисунок 16, *б*.

После размещения компонентов схемы производится соединение их выводов проводниками. При этом необходимо учитывать, что к выводу подключается один проводник. Для выполнения подключения курсор мыши подводится к выводу компонента и после превращения курсора в жирную точку с перекрестием устанавливается соединение щелчком по главной кнопке мыши. Далее курсор перемещается в к другому выводу и при изменении его вида также выполняется щелчок.

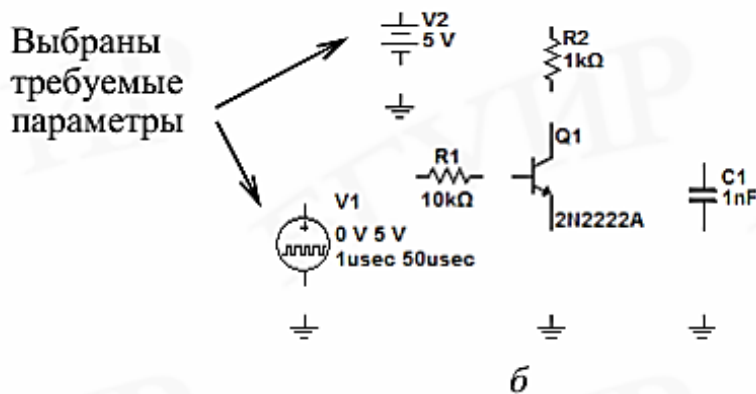
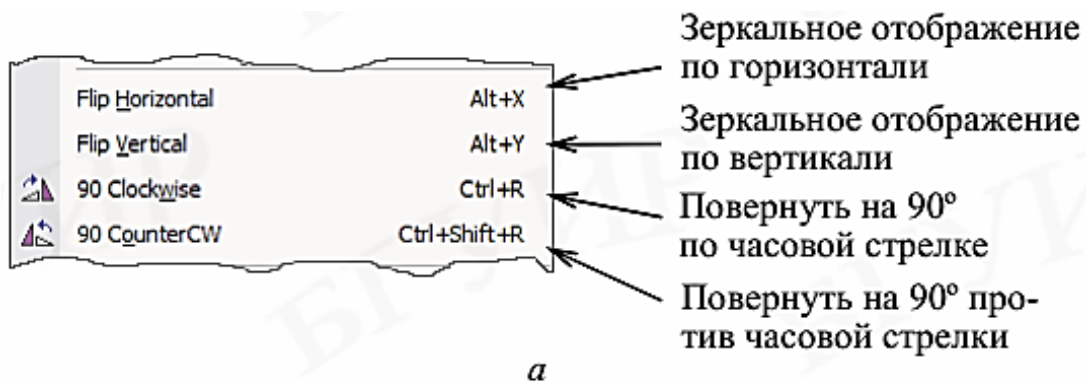


Рисунок 16 – Ориентация элементов схемы (*a*), при их добавлении (*б*)

## Анализ электронных устройств

В Multisim предусмотрено множество режимов анализа данных эмуляции, от простых до самых сложных, в том числе и вложенных. Основные виды анализа:

1. DC – анализ цепи на постоянном токе.

Анализ цепей на постоянном токе осуществляется для резистивных схем. Это правило следует напрямую из теории электрических цепей; при анализе на постоянном токе конденсаторы заменяют разрывом, катушки индуктивности – коротким замыканием, нелинейные компоненты, такие как диоды и транзисторы, заменяют их сопротивлением постоянному току в рабочей точке. Анализ цепи на постоянном токе выявляет узловые потенциалы исследуемой схемы.

2. AC – анализ цепи на переменном токе.

Анализ цепей на переменном токе заключается в построении частотных характеристик.

### 3. Transient – анализ переходных процессов.

Анализ переходных процессов в цепях позволяет определить форму выходного сигнала, то есть построить график сигнала как функции времени.

Чтобы начать анализ, следует выбрать пункт меню Simulate\Analyses и далее требуемый режим.

Список всех функций Multisim приведен на рисунке 17:

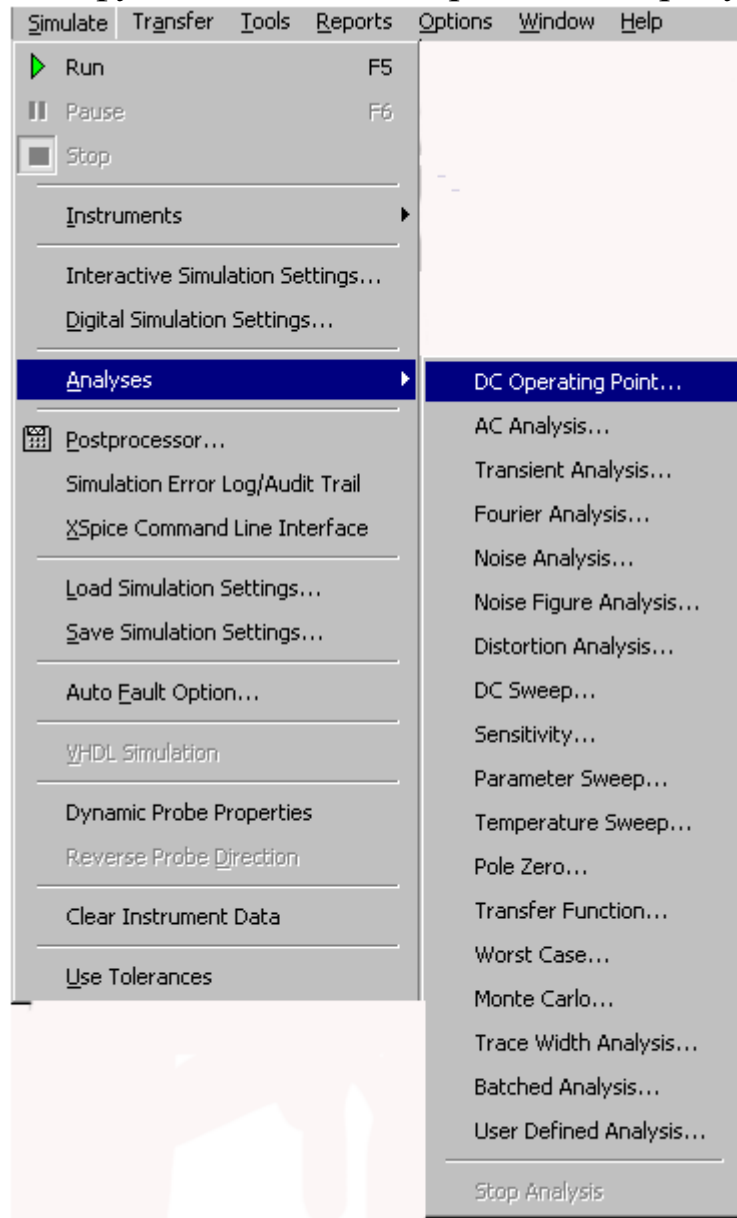


Рисунок 17 – Функции анализа Multisim

Кроме встроенных функций анализа есть возможность определить свою функцию с помощью команд SPICE.



При подготовке к анализу необходимо настроить его параметры, например, диапазон частот для анализатора переменного тока (AC analysis). Необходимо также выбрать выходные каналы (traces).

Плоттер (Grapher) – основной инструмент просмотра результатов эмуляции. Он открывается из меню View/Grapher и автоматически при работе эмуляции.

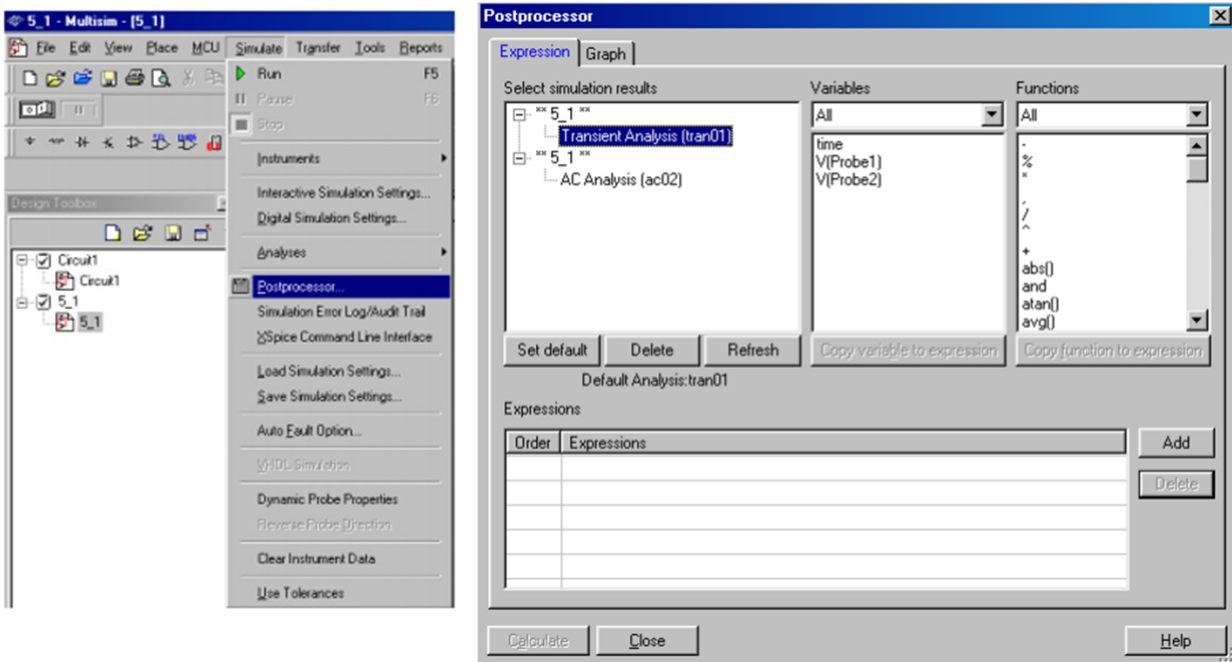
Множество настроек плоттера находятся в окне свойств. Например, можно изменять масштабы, диапазоны, заголовки, стили линий осей (рисунок 18).



Рисунок 18 – Панель настройки плоттера

### Postprocessor и Grapher

Postprocessor и Grapher — это программы пакета Multisim, которые позволяют отобразить результаты моделирования в графическом виде. Данная функция позволяет строить необходимые графики после проведенного анализа. Для работы с функцией Postprocessor (рисунок 19, а) необходимо знать названия узлов. Только те параметры (входные и выходные переменные), которые указываются при выполнении любого вида анализа (AC Sweep, DC Sweep, Transient Analysis и т.д.) отображаются на графиках функции Postprocessor и Grapher. С помощью данной функции можно создать несколько графиков, изменять параметры графика, удалять объекты, производить логические и алгебраические операции над графиками (сложение, умножение, возведение в квадрат и т.д.).



а

б

Рисунок 19 – Постпроцессор

### Создание графика:

-внесение данных, необходимых для построения (рисунок 19 б):  
 Select simulation results-добавление данных проведенного анализа.  
 Variables-переменные, необходимые для построения графика.  
 Functions-алгебраические действия над графиками.  
 В окне Expressions available выбираем необходимые графики для построения.

Для типовых задач формирование заданий проще осуществлять в визуальной форме, используя виртуальные приборы, рассмотренные выше.

### Общие правила моделирования

При моделировании схем необходимо соблюдать следующие общие правила:

1. Любая схема должна обязательно содержать хотя бы один символ заземления.
2. Любые два конца проводника либо контакта устройства, встречающихся в точке, всегда считаются соединенными.
3. При соединении трех концов (Т-соединение) необходимо



использовать символ соединения (узел). Те же правила применяются при соединении четырех и более контактов.

4. В схемах должны присутствовать источники сигнала (тока или напряжения), обеспечивающие входной сигнал, и не менее одной контрольной точки (за исключением анализа схем постоянного тока).

#### **Топология схем**

1. В схеме не должны присутствовать контуры из катушек индуктивности и источников напряжения.
2. Источники тока не должны соединяться последовательно.
3. Не должно присутствовать короткозамкнутых катушек.
4. Источник напряжения должен соединяться с катушкой индуктивности и трансформатором через последовательно включенный резистор. К конденсатору, подключенному к источнику тока, обязательно должен быть параллельно присоединен резистор.

## **1 ПРОЕКТИРОВАНИЕ И ИССЛЕДОВАНИЕ КОМБИНАЦИОННОГО УСТРОЙСТВА**

### **Цель работы**

Получение навыков построения функциональных схем цифровых устройств по логическим выражениям.

Изучение приемов моделирования комбинационных устройств.

### **Подготовка к лабораторной работе**

Повторить правила построения минтермов и макстермов функции  $n$  – переменных. Повторить правила построения СДНФ и СКНФ таблично заданной функции.

В соответствии с номером рабочей станции выбрать логическую функцию из таблицы, приведенной в приложении. Представить таблицу для выбранной функции.

Записать выражения для логической функции в совершенных дизъюнктивной и конъюнктивной нормальных формах.

Начертить функциональную электрическую схему комбинационного устройства для любого из полученных логических выражений. При этом необходимо, пользуясь справочником по цифровым интегральным схемам, выбрать ИС с соответствующим числом входов. ИС с необходимым числом входов можно синтезировать из ИС с меньшим числом входов.

Для самоконтроля ответить на приведенные ниже вопросы.

1. Каковы правила формирования индексов минтермов и макстермов?
2. Как выбирается значение первичного терма при записи  $i$ -го минтерма?
3. Как выбирается значение первичного терма при записи  $i$ -го макстерма?
4. Какие минтермы и макстермы входят в логическое выражение функции заданной таблично?
5. Сколько логических элементов образуют максимально длинную цепочку функциональной схемы комбинационного

устройства при отсутствии ограничений на количество входов?

6. Как построить элемент 3ИЛИ из элементов 2ИЛИ?

### Программа исследований

1. Загрузить программу «Multisim». Создать новый файл под именем Comb1. Выбрать ТТЛ модель интегральных схем. Построить схему, разработанную при домашней подготовке. Подключить ко входу схемы генератор цифровых кодов **Word Generator**. Для наблюдения сигналов подключить логический анализатор **Logic Analyzer**.
  - На входы каналов 1-4 подать соответственно входные сигналы X0-X2 и выходной сигнал Y.
  - На входы каналов 5,6,... - подать сигналы с выходов конъюнкторов.
2. Задать последовательность входных цифровых сигналов из 8 слов от 0 до 7. Установить пакетный режим (Burst). Частоту генератора установить  $F_{\text{такт}} = 1$  КГц.
3. Настроить цифровой анализатор в соответствии с п. 7-8 методических указаний.
4. Задать режим идеальной - быстрой цифровой симуляции (Simulate/Digital Simulation Setting/Ideal). Запустить симуляцию (Simulate/Run или через кнопку молнии на панели инструментов). Отрегулировать установки симуляции исходя из удобства наблюдения диаграмм.
5. Сохранить временные диаграммы, наблюдаемые на экране логического анализатора. Сопоставить значения исходной функции и полученные при моделировании для одних и тех же наборов входных переменных. При несоответствии провести анализ схемы и устранить ошибки. После устранения ошибок повторно провести исследование разработанной схемы.
6. Задать тип используемых ИС - КМОП (с напряжением питания 5В). Установить частоту генератора кодовых последовательностей  $F_{\text{такт}} = 10$  МГц. сохранить полученные временные диаграммы. Сравнить с исходными. Объяснить различия в диаграммах.

7. Методом деления интервала с нижней рабочей границей и нижней нерабочей границей пополам определить верхнюю рабочую частоту спроектированной схемы с погрешностью не свыше 10%.

### **Методические указания**

1. После запуска программы Multisim в разделе Options/Global Preferences/Components выбрать стандарт DIN (западноевропейский).
2. При построении функциональных схем источники питания к интегральным схемам не подключаются.
3. Выходы генератора слов нумеруются сверху – вниз, слева – направо. Целесообразно зеркально развернуть генератор по горизонтали. Для этого вызвать меню вспомогательной кнопкой мыши при курсоре, позиционированном на генераторе и выбрать пункт Flip Horizontal.
4. Настройка генератора слов начинается с вызова панели настройки двойным щелчком на изображении генератора. Для организации коротких циклических последовательностей следует при инициализации генератора кодов задать адрес первого и последнего генерируемого кода (Set Initial Position, Set Final Position) и выбрать циклический режим. Это достигается выделением соответствующей позиции в окне генерируемых кодов генератора слов (Word Generator) и вызовом меню вспомогательной кнопкой мыши. Текущий код устанавливается записью в соответствующей строке генератора слов нужной кодовой комбинации.
5. Нумерация разрядов кодовых слов осуществляется справа – налево (традиционно).
6. Способ представления кодовых слов предварительно следует задать в позиции Display. Наибольшую наглядность имеет двоичный способ представления благодаря однозначному соответствию между позицией бита в кодовом слове и номером контакта. Однако для удобства ввода установите десятичную кодировку слов, а после набора

смените кодировку на двоичную. Поинтересуйтесь в какой из позиций двоичного кода значения изменяются чаще. Это позволит идентифицировать переменные на временной диаграмме.

7. Настройка логического анализатора начинается с вызова панели настройки двойным щелчком главной кнопки мыши на изображении анализатора.
8. Основные настройки – частота тактирования ( $\text{Clock Rate} = (10-100) \times F_{\text{такт}}$ ) количество выборок до начала отображения ( $\text{Pre-trigger} = 1$ ) и после его завершения ( $\text{Post-trigger} = 80-800$ ), масштаб, выраженный в количестве тактов на деление ( $\text{Clock/Div} = 10-100$ ). Целесообразно поэкспериментировать с настройками, что бы «почувствовать» их влияние на временные диаграммы.
9. Следует помнить, что  $\text{Clock Rate}$  определяет частоту дискретизации цифровых сигналов, т.е. временное разрешение. Ошибка измерения любых временных интервалов определяется этой установкой.

### **Контрольные вопросы**

1. Как измерить величину задержки выходного сигнала?
2. Какова абсолютная погрешность при измерении задержки?
3. Как уменьшить величину погрешности при измерении задержки?
4. Какова задержка сигнала в спроектированной комбинационной схеме?
5. Объяснить разницу в величинах задержек выходного сигнала для разных его фаз.

### **Содержание отчета**

Отчет должен включать:

1. Номер варианта и исходную функцию, заданную в табличной форме.
2. Аналитическое выражение логической функции с комментариями в СДНФ и СКНФ.
3. Перечень выбранных логических элементов для построения

схемы, исходную и конечную функциональные электрические схемы комбинационного устройства. Исходная схема получена в результате синтеза, а конечная после устранения всех ошибок.

4. Все промежуточные (при необходимости) и окончательные временные диаграммы (скриншоты с экрана логического анализатора) для частоты исследования 1 кГц, 10 МГц и предельной рабочей частоты.
5. Преобразованные в табличную форму значения функции полученные из временных диаграмм. Реальные значения функции вносятся в дополнительные столбцы таблицы в графы экспериментальных значений функции.
6. Письменные ответы на контрольные вопросы.

### Приложение

Таблица 1. Варианты заданий к лабораторной работе для первой подгруппы.

№ п/п	X2	X1	X0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10
0	0	0	0	1				1				1	
1	0	0	1		1				1				1
2	0	1	0		1	1	1			1			1
3	0	1	1				1	1		1	1		
4	1	0	0	1		1		1			1	1	
5	1	0	1	1	1				1				
6	1	1	0			1			1	1			
7	1	1	1				1				1	1	1

Таблица 2. Варианты заданий к лабораторной работе для второй подгруппы.

№ п/п	X2	X1	X0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10
0	0	0	0								1	1	
1	0	0	1	1	1						1		
2	0	1	0				1	1	1		1	1	
3	0	1	1	1	1			1					1
4	1	0	0		1	1		1					
5	1	0	1	1		1			1	1			1
6	1	1	0			1	1		1	1			
7	1	1	1				1			1		1	1

## 2 МИНИМИЗАЦИЯ И ИССЛЕДОВАНИЕ КОМБИНАЦИОННОГО УСТРОЙСТВА

### Цель работы

- Получение навыков минимизации логических функций и приведения их к заданному базису.
- Анализ статических и динамических рисков в схемах цифровых устройств.

### Подготовка к лабораторной работе

Повторить правила построения СДНФ и СКНФ таблично заданных функций и минимизации их графическим методом.

Повторить правила приведения функций к различным базисам.

Изучить раздел 5 лекционного курса «Задержки в цифровых цепях».

В соответствии с номером бригады выбрать логическую функцию из таблицы, приведенной в приложении. Представить таблицу для выбранной функции.

Провести минимизацию функции в форме, наиболее удобной для последующего приведения к заданному базису.

Записать выражение для логической функции в заданном базисе.

Начертить функциональную электрическую схему комбинационного устройства. При этом необходимо, пользуясь справочником по цифровым интегральным схемам, выбрать ИС с соответствующим числом входов. ИС с необходимым числом входов можно синтезировать из ИС с меньшим числом входов.

Для самоконтроля ответить на приведенные ниже вопросы.

1. В какой форме следует записать логическую функцию для последующего представления в базисе И-НЕ?
2. В какой форме следует записать логическую функцию для последующего представления в базисе ИЛИ-НЕ?
3. В какой форме следует записать логическую функцию для последующего представления в базисе И-ИЛИ-НЕ?

4. Сформулируйте правило записи дизтерма, соответствующего прямоугольнику, покрывающему карту Карно.
5. Сформулируйте правило записи контерма, соответствующего прямоугольнику, покрывающему карту Карно.
6. Как выявить наборы входных переменных, приводящие к статическим и динамическим рискам?

### **Программа исследований**

1. Загрузить программу «Multisim». Создать новый файл под именем Comb2. Построить схему, разработанную при домашней подготовке. Подключить к входу схемы генератор цифровых кодов. Для наблюдения сигналов подключить логический анализатор. На входы каналов 1-5 подать соответственно входные сигналы X0-X3 и выходной сигнал Y.
2. Задать последовательность из 16 возбуждающих сигналов от 0 до 15. Задать циклический режим. Частоту генератора оставить по умолчанию (1 КГц). Задать тип используемых ИС - КМОП.
3. Зарисовать временные диаграммы, наблюдаемые на экране логического анализатора. Проверить правильность функционирования предварительным преобразованием диаграмм к табличной форме. При несоответствии функциональности собранной схемы заданию провести анализ схемы и устранить ошибки. После устранения ошибок повторно провести исследование разработанной схемы.
4. Задать поочередно три пары комбинаций входных переменных, приводящих к статическим и динамическим рискам. Установить частоту генератора кодовых последовательностей 10 МГц. Зарисовать полученные временные диаграммы. Провести анализ временных диаграмм и привести объяснения.
5. Определить максимальную частоту функционирования разработанного устройства.
6. Опираясь на параметры используемых ИС (исходные данные взять из даташитов, размещённых в Интернет) определить теоретически предельную рабочую частоту устройства.



### **Контрольные вопросы**

1. К каким последствиям приведет использование логических схем КМОП типа при пониженном напряжении питания (многие серии ИС этого типа допускают работу при напряжениях 3-15 В).
2. Какой фактор будет определять вид выходного сигнала для последовательностей вызывающих состязания? Пояснить с помощью временных диаграмм для одной из последовательностей, использованных в лабораторной работе.

### **Содержание отчета**

Отчет должен включать:

1. Исходную функцию, заданную в табличной форме.
2. Аналитическое выражение минимизированной логической функции.
3. Перечень выбранных логических элементов для построения схемы и функциональную электрическую схему комбинационного устройства.
4. Все промежуточные (при необходимости) и окончательные временные диаграммы, снятые с экрана логического анализатора.
5. Выводы, результаты измерений и теоретической оценки предельной рабочей частоты устройства.
6. Письменные ответы на контрольные вопросы.

## Приложение. Варианты заданий

Таблица 1. Варианты заданий к лабораторной работе для первой подгруппы.

№ п/п	X3	X2	X1	X0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10
0	0	0	0	0	1				1				1	
1	0	0	0	1		1	1			1	1			1
2	0	0	1	0		1	1	1			1			1
3	0	0	1	1				1	1		1	1		
4	0	1	0	0	1		1		1			1	1	
5	0	1	0	1	1	1	1			1				
6	0	1	1	0	1		1			1	1		1	
7	0	1	1	1				1				1	1	1
8	1	0	0	0	1				1				1	1
9	1	0	0	1		1	1			1				1
10	1	0	1	0		1	1	1			1	1		1
11	1	0	1	1		1		1	1		1	1		
12	1	1	0	0	1		1		1			1	1	
13	1	1	0	1	1	1			1	1				
14	1	1	1	0			1	1		1	1			1
15	1	1	1	1				1		1	1	1	1	1
Базис					/	↑	/	↑	/	↑	/	↑	/	↑

Таблица 2. Варианты заданий к лабораторной работе для второй подгруппы.

№ п/п	X3	X2	X1	X0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10
0	0	0	0	0	1				1				1	
1	0	0	0	1		1	1			1	1			
2	0	0	1	0		1	1	1			1			1
3	0	0	1	1							1	1		
4	0	1	0	0	1		1		1			1	1	
5	0	1	0	1			1			1				
6	0	1	1	0	1		1			1	1			
7	0	1	1	1				1					1	1
8	1	0	0	0	1				1				1	1
9	1	0	0	1		1	1							1
10	1	0	1	0		1		1			1	1		1
11	1	0	1	1		1		1	1			1		
12	1	1	0	0	1		1		1			1	1	
13	1	1	0	1	1	1			1	1				
14	1	1	1	0			1	1		1	1			1
15	1	1	1	1				1		1	1	1	1	1
Базис					↑	/	↑	/	↑	/	↑	/	↑	/

### **3 ПРОЕКТИРОВАНИЕ И ИССЛЕДОВАНИЕ СЕМИСЕГМЕНТНОГО ДЕШИФРАТОРА**

#### **Цель работы**

- Изучение принципа работы семисегментного дешифратора.
- Изучение методов поиска неисправностей в комбинационных устройствах.

#### **Подготовка к лабораторной работе**

Изучить характеристики, параметры и схемы включения семисегментных светодиодных индикаторов [3,4,5].

Выбрать по [3] матрицы, удовлетворяющие требованиям задания.

Провести проектирование преобразователя двоично-десятичного кода в семисегментный код. Вариант задания выбрать в соответствии с приложением. Составить таблицу истинности для дешифратора, преобразующего двоично-десятичный код в семисегментный код.

- Выбрать состав базовых элементов с учетом библиотеки элементов Multisim и заданного базиса.
- Провести минимизацию функции, заданной таблично с учетом выбранного базиса.
- Привести функцию к выбранному базису.
- Начертить функциональную электрическую схему дешифратора.
- По величину тока сегмента светодиодного индикатора (ориентируясь на справочные значения), рассчитать величину резисторов, включённых последовательно с сегментами светодиодной матрицы.
- Проверить условия работоспособности логических элементов выходной ступени дешифратора с семисегментным индикатором.
- Принять меры для удовлетворения условий работоспособности (при необходимости).

Для самоконтроля ответить на приведенные ниже вопросы (письменно).

1. Каковы максимальные выходные токи логических интегральных схем, выбранной вами серии элементов?
2. Как зависит ток светодиода от величины напряжения на нём? Привести типичную ВАХ сегмента светодиодной матрицы.
3. Что произойдёт при несоответствии токов сегмента светодиодной матрицы и выходного тока логического элемента?
4. Как воспользоваться тем обстоятельством, что логическая функция является не полностью определенной.

### **Программа исследований**

1. Загрузить программу «Multisim». Создать новый файл под именем Comb3. Построить схему, разработанную при домашней подготовке. Подключить ко входу схемы генератор цифровых кодов. К выходам дешифратора подключить семисегментный индикатор.
2. Задать последовательность из 10 возбуждающих сигналов от 0 до 9.
3. В пошаговом режиме проверить правильность функционирования. После устранения найденных ошибок повторно провести тестирование разработанной схемы.
4. Провести измерение величины тока сегмента и полного тока индикатора для наихудшего случая.

### **Контрольные вопросы**

1. Как влияет тип светодиодной матрицы (общий анод/катод) на способ подключения его к выводам цифровых схем?
2. Как отразятся на работе светодиодного индикатора возникающие в устройстве риски сбоя?
3. Дешифратор реализует 7 логических функций от одного и того же набора входных переменных. Как воспользоваться этим при минимизации устройства?

4. Изобразите эквивалентную схему цепи питания одного сегмента светодиодного индикатора. Проанализируйте по ней возможные причины разной яркости сегментов и их значимость.

## Содержание отчета

### Отчет должен включать:

1. Исходную функцию, заданную в табличной форме.
2. Аналитическое выражение минимизированных логических функций для каждого сегмента.
3. Перечень выбранных логических элементов для построения схемы и функциональную электрическую схему дешифратора.
4. Все промежуточные (при необходимости) и окончательные временные диаграммы, снятые с экрана логического анализатора.
5. Скриншоты работающей схемы для каждого состояния индикатора при пошаговом режиме работы.
6. Письменные ответы на все вопросы.

## Приложение. Варианты заданий

Номер варианта определяется по алфавитному списку группы.

Вариант	Тип матрицы	Базис	Ток сегмента, мА
1	С разобшёнными катодами	И-НЕ	5-10
2	С разобшёнными анодами	ИЛИ-НЕ	5-10
3	С разобшёнными катодами	И-НЕ	15-20
4	С разобшёнными анодами	ИЛИ-НЕ	15-20
5	С разобшёнными катодами	И-НЕ	5
6	С разобшёнными анодами	ИЛИ-НЕ	5
7	С разобшёнными катодами	И-НЕ	2-4
8	С разобшёнными анодами	ИЛИ-НЕ	2-4
9	С разобшёнными катодами	И-НЕ	10
10	С разобшёнными анодами	ИЛИ-НЕ	10
11	С разобшёнными катодами	И-НЕ	20
12	С разобшёнными анодами	ИЛИ-НЕ	20

## **4 ПРОЕКТИРОВАНИЕ И ИССЛЕДОВАНИЕ RS-ТРИГГЕРА**

### **Цель работы**

- Изучение устройства RS-триггеров.
- Изучение особенностей функционирования асинхронного и синхронизируемого уровнем RS-триггера.

### **Подготовка к лабораторной работе**

Изучить устройство и функционирование асинхронных и синхронизируемых уровнем RS-триггеров [2].

Построить схему асинхронного RS-триггера в базисе И-НЕ.

Построить схему асинхронного RS-триггера в базисе ИЛИ-НЕ.

Построить схему синхронизируемого уровнем RS-триггера в базисе И-НЕ.

Привести таблицы функционирования этих RS-триггеров.

Привести характеристические уравнения триггеров и диаграммы состояний.

Разработать диаграммы испытательных сигналов для полного тестирования синхронизируемого RS-триггера. Диаграммы должны демонстрировать все особенности его функционирования при переходе в режим хранения после сброса, установки, запрещённой комбинации.

**Для самоконтроля ответить на приведенные ниже вопросы.**

1. Какая комбинация сигналов является запрещённой для RS-триггера в базисе И-НЕ?
2. Какая комбинация сигналов является запрещённой для RS-триггера в базисе ИЛИ-НЕ?
3. Что представляет собой режим хранения.
4. Описать отличия в работе асинхронного и синхронизируемого уровнем RS-триггеров.

## Программа исследований

1. Загрузить программу «Multisim». Создать новый файл под именем RS1A. Построить схему, асинхронного RS-триггера в базисе И-НЕ. Подключить ко входу схемы генератор слов. Подключить многоканальный анализатор к выходам генератора кодов и выходам исследуемого триггера.
2. Задать циклическую последовательность состояний генератора кодов 01-11-10-11-01-11, и установить частоту равной 1 КГц.
3. Получить и сохранить временные диаграммы. Сравнить их с таблицей функционирования асинхронного RS-триггера. Объяснить различия в функционировании на разных отрезках временных диаграмм.
4. Создать новый файл под именем RS1. Построить схему, синхронизируемого уровнем RS-триггера в базисе И-НЕ. Подключить ко входу схемы генератор слов. Подключить многоканальный анализатор к выходам генератора кодов и выходам исследуемого триггера.
5. Задать кодовую последовательность входных сигналов (двух информационных и синхронизации) разработанную при домашней подготовке и установить частоту смены кодовых комбинаций генератора слов равной 1 КГц.
6. Запустить симуляцию и сохранить полученные временные диаграммы.
7. Сопоставить полученные диаграммы с таблицей функционирования триггера, для чего в таблице функционирования ввести дополнительную графу «Эксперимент» и заполнить её. Сформулировать характерные особенности функционирования синхронного RS-триггера.
8. Увеличить частоту следования сигнала до величины, позволяющей надёжно измерить величину задержки триггера.
9. Получить совмещённые временные диаграммы всех входных и выходных сигналов и определить задержки выходного сигнала для всех характерных режимов. Сравнить с задержкой одиночного элемента.
10. Определить минимальную длительность импульса установки

и сброса асинхронного триггера, необходимую для надёжного изменения состояния триггера.

### **Контрольные вопросы**

1. Приведёт ли изменение уровней входных сигналов к изменению выходных сигналов при высоком уровне сигнала синхронизации? Объяснить. Подтвердить снятой временной диаграммой
2. Почему нельзя подавать на вход RS-триггера запрещённую комбинацию сигналов? Что произойдёт, если такое случится?
3. Как теоретически определить величину задержки сигнала триггером?

### **Содержание отчета**

Отчет должен включать:

1. Титульный лист
2. Функциональные электрические схемы всех триггеров.
3. Таблицы состояний всех RS-триггеров по п.2.
4. Результаты исследований и отладки устройства на частоте 1 КГц (при необходимости).
5. Временные диаграммы, полученные при исследованиях.
6. Результаты анализа и сопоставления теоретических и практических результатов.
7. Результаты измерений параметров триггеров и сопоставления их с результатами теоретического анализа.
8. Письменные ответы на контрольные вопросы.



## 5 ИССЛЕДОВАНИЕ СДВИГОВОГО РЕГИСТРА

### Цель работы

Изучение устройства и работы сдвигового регистра.

### Основные сведения

Регистр сдвига – это регистр, содержимое которого при подаче управляющего сигнала на тактовый вход  $C$  может сдвигаться в сторону старших или младших разрядов. Схема регистра сдвига из цепочки J-K-триггеров показана на рисунке 1.

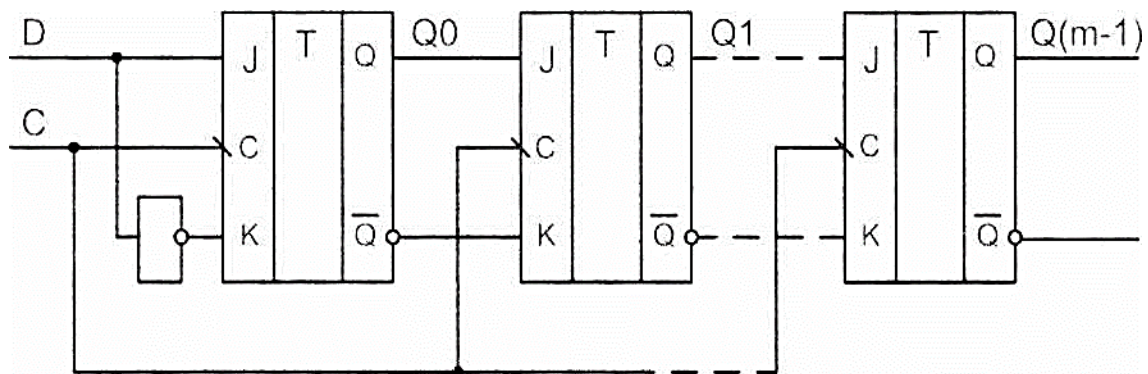


Рисунок 1 – Схема регистра сдвига

Пусть левый по схеме триггер соответствует младшему разряду регистра, а правый триггер – старшему разряду. Тогда вход каждого триггера (кроме левого) подключен к выходу соседнего младшего триггера. Когда на все входы  $C$  триггеров поступает срез входного тактового импульса, выход каждого триггера  $Q_i$  принимает состояние предыдущего каскада и, таким образом, информация, содержащаяся в регистре, сдвигается на один разряд в сторону старших разрядов. Триггер младшего разряда принимает при этом состояние последовательного входа  $D$ . Информация, поступившая на вход  $D$  схемы, появится на ее выходе  $Q(m-1)$  через  $m$  тактов.

Существенным является то, что схема построена на двухступенчатых триггерах. Если использовать триггеры с потенциальным управлением, то при активном уровне сигнала  $C$  все триггеры будут открыты для записи, и сигнал  $D$  успеет пройти столько триггеров, сколько позволит длительность сигнала  $C$ .

Часто требуются более сложные регистры: с параллельной синхронной записью информации, реверсивные, с параллельно-последовательной записью. Такие регистры называются универсальными. Примером универсального регистра служит интегральная микросхема К555ИР11, условное графическое обозначение которой показано на рисунке 2.

Регистр К555ИР11 может работать в следующих режимах (табл.1): сброс, хранение данных, сдвиг влево, сдвиг вправо, и параллельная загрузка. Микросхема имеет входы: тактовый (С), параллельной загрузки (D0 - D3), выбора режима работы (S0 и S1), асинхронного сброса (R). Данные также могут поступать в регистр в последовательном коде на входы DL (при сдвиге влево) и DR (при сдвиге вправо). Все операции кроме сброса выполняются в регистре синхронно по фронту тактовых импульсов. Внутренний код регистра может быть прочитан на выходах Q0 - Q3.

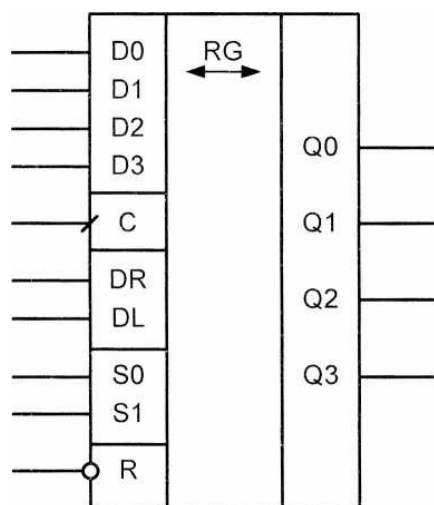


Рисунок 2 – Условное графическое обозначение регистра сдвига.

Области применения сдвиговых регистров весьма разнообразны. В двоичной арифметике сдвиг числа на один разряд влево соответствует умножению его на 2, а сдвиг на один разряд вправо - делению на 2. В аппаратуре передачи данных универсальные регистры преобразуют параллельный код в последовательный и обратно.

Таблица 1. Режимы регистра сдвига

Режим работы	Вход							Выход			
	R	C	S1	S0	DR	DL	Dn	Q0	Q1	Q2	Q3
Сброс	0	x	x	x	x	x	x	0	0	0	0
Хранение	1	x	0	0	x	x	x	q <sub>0</sub>	q <sub>1</sub>	q <sub>2</sub>	q <sub>3</sub>
Сдвиг влево	1	↑	1	0	x	0	x	q <sub>1</sub>	q <sub>2</sub>	q <sub>3</sub>	0
	1	↑	1	0	x	1	x	q <sub>1</sub>	q <sub>2</sub>	q <sub>3</sub>	1
Сдвиг вправо	1	↑	0	1	0	x	x	0	q <sub>0</sub>	q <sub>1</sub>	q <sub>2</sub>
	1	↑	0	1	1	x	x	1	q <sub>0</sub>	q <sub>1</sub>	q <sub>2</sub>
Параллельная загрузка	1	↑	1	1	x	x	d <sub>n</sub>	d <sub>0</sub>	d <sub>1</sub>	d <sub>2</sub>	d <sub>3</sub>

**Примечания:**

- символ x обозначает безразличное состояние входа;
- символ ↑ обозначает фронт тактового сигнала.

Передача данных последовательным кодом по сравнению с параллельной передачей существенно экономит число линий связи, однако при этом увеличивается время обмена.

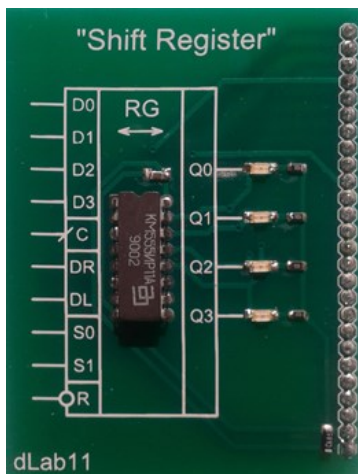
**Оборудование**

Рисунок 3 – Внешний вид модуля регистра сдвига

Исследуемый регистр сдвига размещён на лабораторном модуле в виде печатной платы. Его внешний вид представлен на рисунке 3.

Модуль устанавливается в разъём цифровых устройств макетной платы рабочей станции ELVIS II. Исследуемый модуль устанавливается в разъём только при выключенном питании рабочей станции. Размещение модуля на макетном поле рабочей станции представлено на рисунке 4.

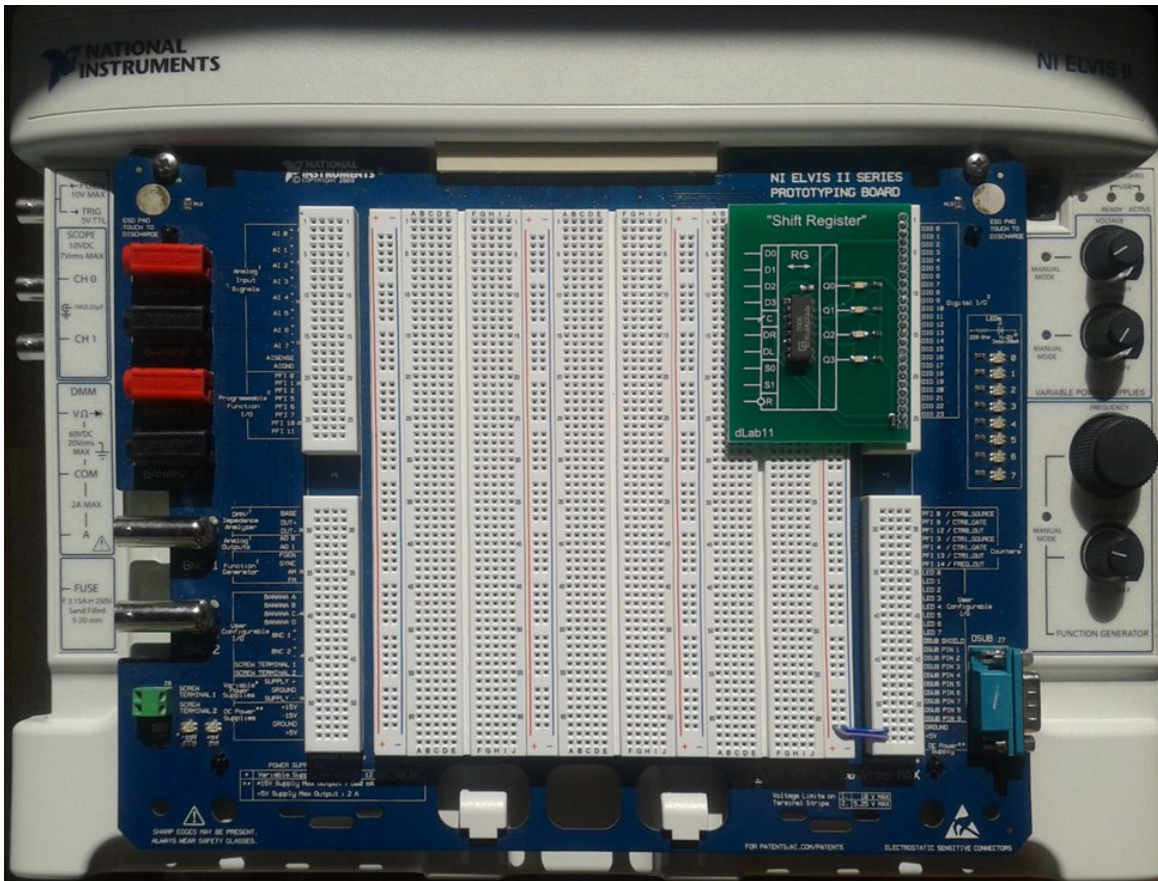


Рисунок 4 – Рабочая станция с исследуемым модулем

Перед установкой исследуемого модуля при необходимости следует установить ограничители посадочного места и проводники питания в соответствии с рисунком 5.

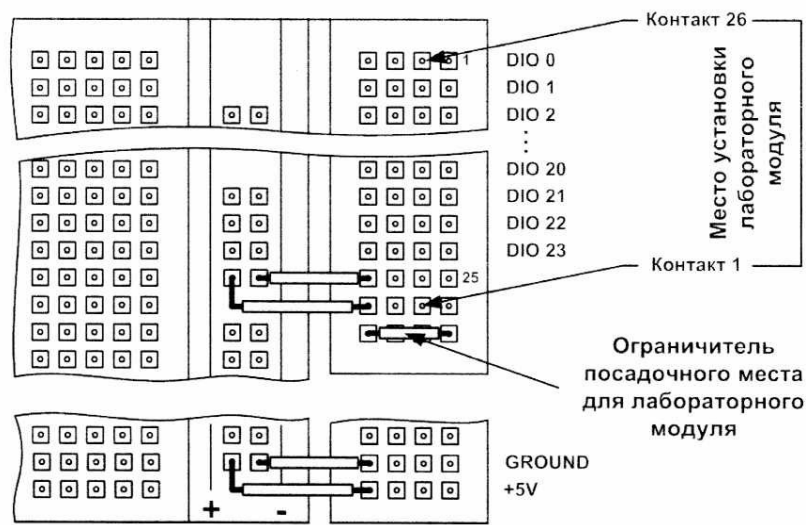


Рисунок 5 – Размещение дополнительных проводников на макетной плате рабочей станции

## Выполнение лабораторной работы

- Выключите рабочую станцию ELVIS II выключателем на задней панели, если ранее она была включена.
- Соедините кабелем USB-2 разъёмы рабочей станции и персонального компьютера (ПК).
- При установленном исследуемом модуле включите рабочую станцию выключателем на задней панели.
- Выключателем в правой верхней части рабочей станции включите питание макетной платы (должен загореться зелёный индикатор).
- Загрузить файл виртуального прибора для исследования регистра сдвига dLab-11.vi. при этом на экране ПК появится изображение панели виртуального прибора (ВП) представленное на рисунке 6.

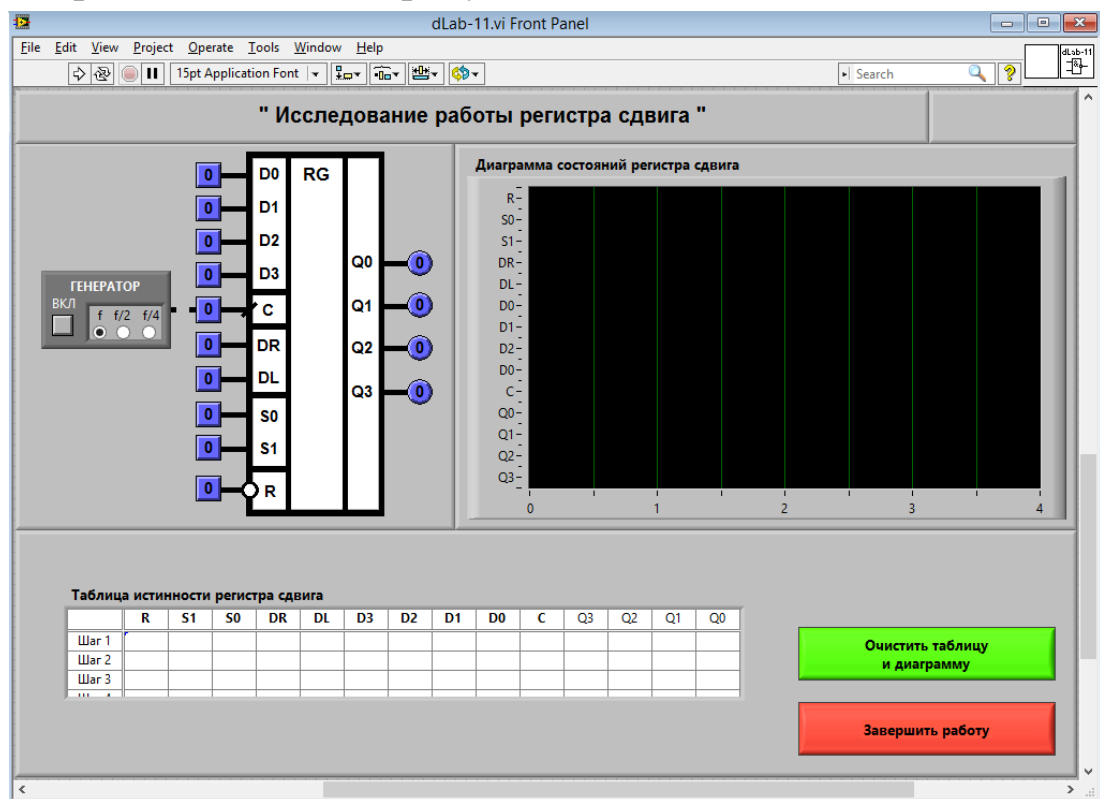


Рисунок 6 – Панель виртуального прибора

Дальнейшее управление исследованием осуществляется исключительно с панели виртуального прибора.

## Исследование регистра сдвига в статическом режиме

Статический режим исследования регистра сдвига реализуется при подаче на его тактовый вход «С» одиночных импульсов в ручном режиме. Для этого генератор импульсов («ГЕНЕРАТОР»), расположенный на лицевой панели ВП, должен быть выключен (кнопка «ВКЛ» отжата). Подача одиночного импульса прямоугольной формы на тактовый вход «С» регистра сдвига производится однократным нажатием с помощью манипулятора мышь на кнопку квадратной формы, расположенную около этого входа.

### Режим сдвига вправо

- 1 Выключите генератор импульсов, если он был включен.
- 2 Нажмите на кнопку «Очистить таблицу и диаграмму».
- 3 Установите на входах выбора режима сигналы:  $S_0=1$ ,  $S_1=0$ ,  $R=1$ . Логический уровень изменяется при однократном нажатии с помощью манипулятора мышь на кнопку квадратной формы, расположенную около соответствующего входа. При этом на кнопке отображается состояние входа («0» - синий цвет или «1» - оранжевый цвет).
- 4 Выполните сброс регистра. Для этого на лицевой панели ВП с помощью манипулятора мышь сначала переведите в отжатое, а затем в нажатое состояние кнопку квадратной формы, расположенную около входа «R». На индикаторах выходных сигналов регистра «Q0», «Q1», «Q2» и «Q3» должны установиться нулевые значения.
- 5 Установите на входе последовательных данных «DR» логический сигнал «1».
- 6 Нажмите и отпустите кнопку, расположенную около входа «С». На индикаторах круглой формы, расположенных около выходов «Q0», «Q1», «Q2» и «Q3» регистра, будет отображено состояние его выходных сигналов. Логические состояния входов и выходов триггера будут автоматически занесены в таблицу истинности и на диаграмму состояний. В



графу «С» таблицы истинности заносится символ «ЛГ», означающий подачу импульса на вход «С».

- 7 Установите на входе последовательных данных «DR» логический сигнал «0».
- 8 Повторите п. 6 три раза подряд.
- 9 Скопируйте таблицу истинности и диаграмму состояний в отчет.

### **Режим сдвига влево**

- 1 Нажмите на кнопку «Очистить таблицу и диаграмму».
- 2 Установите на входах выбора режима сигналы:  $S_0=0$ ,  $S_1=1$ ,  $R=1$ .
- 3 Выполните сброс регистра. Для этого на лицевой панели ВП с помощью манипулятора мышью сначала переведите в отжатое, а затем в нажатое состояние кнопку квадратной формы, расположенную около входа «R». На индикаторах выходных сигналов регистра «Q0», «Q1», «Q2» и «Q3» должны установиться нулевые значения.
- 4 Установите на входе последовательных данных «DL» логический сигнал «1».
- 5 Нажмите и отпустите кнопку, расположенную около входа «С». На индикаторах круглой формы, расположенных около выходов «Q0», «Q1», «Q2» и «Q3» регистра, будет отображено состояние его выходных сигналов. Логические состояния входов и выходов триггера будут автоматически занесены в таблицу истинности и на диаграмму состояний. В графу «С» таблицы истинности заносится символ «ЛГ», означающий подачу импульса на вход «С».
- 6 Установите на входе последовательных данных «DL» логический сигнал «0».
- 7 Повторите п.14 три раза подряд.
- 8 Скопируйте таблицу истинности и диаграмму состояний в отчет.
- 9 По таблице истинности и диаграмме состояний определите, в

каком направлении (от Q0 к Q3 или от Q3 к Q0) смещается логическая единица, записанная в регистр на первом такте. Вывод запишите в отчет.

### Режим параллельной загрузки

- 1 Нажмите на кнопку «Очистить таблицу и диаграмму».
- 2 Установите на входах выбора режима сигналы:  $S_0=1$ ,  $S_1=1$ ,
- 3  $R=1$ .
- 4 Выполните сброс регистра. Для этого на лицевой панели ВП с помощью манипулятора мышь сначала переведите в отжатое, а затем в нажатое состояние кнопку квадратной формы, расположенную около входа «R». На индикаторах выходных сигналов регистра «Q0», «Q1», «Q2» и «Q3» должны установиться нулевые значения.
- 5 Установите на входах параллельной загрузки «D0», «D1», «D2» и «D3» значения сигналов, приведенные в первой строке табл. 2.

Таблица 2

<i>Вход</i>	<i>Вход</i>	<i>Вход</i>	<i>Вход</i>
0	1	0	1
1	1	1	1
0	0	1	0
0	0	1	0

- 6 Нажмите и отпустите кнопку, расположенную около входа «C». На индикаторах круглой формы, расположенных около выходов «Q0», «Q1», «Q2» и «Q3» регистра, будет отображено состояние его выходных сигналов. Логические состояния входов и выходов триггера будут автоматически занесены в таблицу истинности и на диаграмму состояний. В графу «C» таблицы истинности заносится символ «LГ», означающий подачу импульса на вход «C».



- 7 Повторите пп. 23 - 24 для остальных строк таблицы 2.
- 8 Скопируйте таблицу истинности и диаграмму состояний в отчет.
- 9 По таблице истинности и диаграмме состояний проверьте соответствие выходных сигналов регистра Q0, Q1, Q2 и Q3 сигналам на входах параллельной загрузки D0, D1, D2 и D3. Вывод запишите в отчет.

### Режим хранения

- 1 Нажмите на кнопку «Очистить таблицу и диаграмму».
- 2 Установите режим параллельной загрузки и загрузите в регистр сдвига цифровой код 1010. Правильность выполнения операции проконтролируйте по выходным индикаторам на лицевой панели ВП.
- 3 Установите на входах выбора режима сигналы: S0= 0, S1 =0, R=1.
- 4 Установите на входах последовательных данных сигналы DR=1, DSL=1.
- 5 Установите на входах параллельной загрузки «D0», «D1», «D2» и «D3» значения сигналов, приведенные в первой строке табл. 11.3.

Таблица 3.

<i>Вход</i>	<i>Вход</i>	<i>Вход</i>	<i>Вход</i>
0	1	0	1
1	1	1	1
0	0	1	0
0	0	1	0

- 6 Нажмите и отпустите кнопку, расположенную около входа «С». На индикаторах круглой формы, расположенных около выходов «Q0», «Q1», «Q2» и «Q3» регистра, будет отображено состояние его выходных сигналов. Логические состояния входов и выходов триггера будут автоматически занесены в таблицу истинности и на диаграмму состояний. В

графу «С» таблицы истинности заносится символ «ЛГ», означающий подачу импульса на вход «С».

- 7 Повторите пп. 32-33 для остальных строк таблицы 3.
- 8 Скопируйте таблицу истинности и диаграмму состояний в отчет.
- 9 По таблице истинности и диаграмме состояний убедитесь, что при значениях сигналов  $S_0=0$ ,  $S_1=0$  и подаче импульсов на тактовый вход «С» регистр сдвига сохраняет на выходе первоначально занесенный в него цифровой код. Вывод запишите в отчет.
- 10 По результатам исследования в статическом режиме составьте сводную таблицу истинности регистра сдвига.

### **Исследование регистра сдвига в динамическом режиме**

Динамический режим исследования регистра сдвига реализуется при подаче на его тактовый вход «С» последовательности импульсов. Для этого генератор импульсов («ГЕНЕРАТОР»), расположенный на лицевой панели ВП, должен быть включен (кнопка «ВКЛ» нажата). На выходе генератора формируется последовательность прямоугольных импульсов и подается на вход «С» регистра. С помощью кнопок « $f$ », « $f/2$ » и « $f/4$ » можно изменять частоту следования импульсов для выбора удобного режима наблюдения временной диаграммы.

- 1 Нажмите на кнопку «Очистить таблицу и диаграмму».
- 2 Включите генератор импульсов. На графический индикатор выводится временная диаграмма входных и выходных сигналов регистра. В этом режиме таблица истинности не заполняется, а кнопка «Очистить таблицу и диаграмму» заблокирована от нажатия и имеет затененное изображение.
- 3 Изменяя входные сигналы регистра, получите временные диаграммы, отражающие его работу в режимах сдвига вправо, сдвига влево, параллельной загрузки, сброса. Каж-

дый раз, получив требуемое изображение, следует остановить работу регистра, выключив генератор, и скопировать диаграмму в отчет.

- 4 По полученным диаграммам определите, по какому перепаду на тактовом входе «С» регистра сдвига происходят изменения состояния в режимах сдвига вправо, сдвига влево, параллельной загрузки и сброса. Результаты исследований запишите в отчет.
- 5 Выключите ВП, для чего нажмите на панели ВП кнопку «Завершить работу».

### **Содержание отчёта**

- Отчёт должен содержать титульный лист;
- Цель работы;
- Краткие теоретические сведения;
- Таблицы функционирования для каждого раздела исследований;
- Временные диаграммы для каждого раздела исследований;
- Краткое описание алгоритма функционирования; представленного каждой временной диаграммой;
- Ответы на контрольные вопросы;
- Список использованных источников.

### **Контрольные вопросы**

- Что называется регистром сдвига?
- В чем состоит отличие сдвигового регистра от параллельного регистра?
- Предложите схему сдвигающего регистра на D-триггерах и на JK-триггерах.
- Какой регистр называется реверсивным?

## 6 ИССЛЕДОВАНИЕ ДВОИЧНОГО СЧЁТЧИКА

### Цель работы

Изучение устройства и работы двоичного счётчика.

### Основные сведения

Счетчиком называется устройство для подсчета числа входных импульсов. При поступлении каждого импульса на тактовый вход С состояние счетчика изменяется на единицу. Счетчик можно реализовать на нескольких триггерах, при этом состояние счетчика будет определяться состоянием его триггеров. В суммирующих счетчиках каждый входной импульс увеличивает число на его выходе на единицу, в вычитающих счетчиках каждый входной импульс уменьшает это число на единицу. Наиболее простые счетчики - двоичные. Схема суммирующего двоичного счетчика показана на рисунке 1.

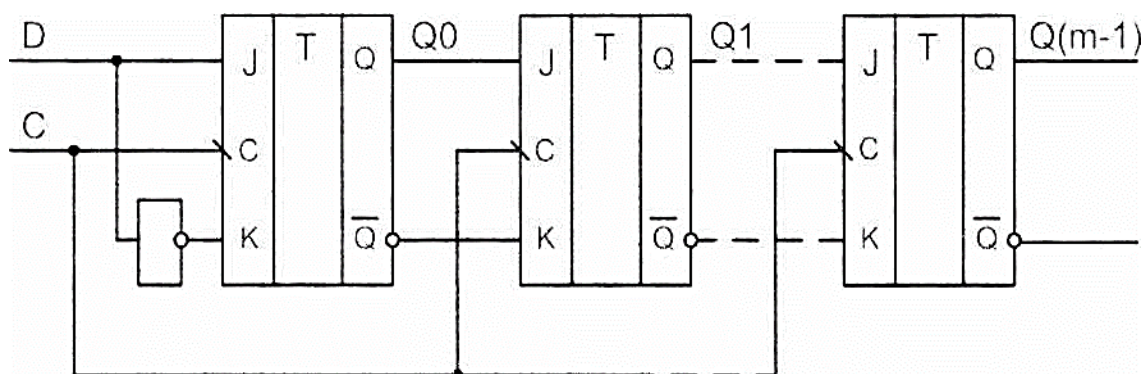


Рисунок 1 – Схема двоичного суммирующего счётчика

При построении счетчика триггеры соединяют последовательно. Выход каждого триггера непосредственно действует на тактовый вход следующего. Для того чтобы реализовать суммирующий счетчик, необходимо счетный вход очередного триггера подключать к инверсному выходу предыдущего. Для того чтобы изменить направление счета (реализовать вычитающий счетчик), можно предложить следующие способы:

- считывание выходных сигналов счетчика не с прямых, а с инверсных выходов триггеров.

- изменение структуры связей в счетчике путем подачи на счетный вход триггера сигнала не с инверсного, а с прямого выхода предыдущего каскада.

Счетчики характеризуются числом состояний в течение одного периода (цикла) счета. Число состояний определяется количеством триггеров  $m$  в структуре счетчика. Так для двоичного счетчика при  $m = 3$  число состояний равно  $2^m = 2^3 = 8$  (выходной код изменяется от 000 до 111).

Число состояний счетчика принято называть коэффициентом пересчета  $K_{сч}$ . Этот коэффициент равен отношению числа импульсов  $N_{вх}$  на входе к числу импульсов  $N_{вых}$  на выходе старшего разряда счетчика за период счета:

$$K_{сч} = N_{вх} / N_{вых} \quad (1)$$

Если на вход счетчика подавать периодическую последовательность импульсов с частотой  $f_{вх}$ , то частота  $f_{вых}$  на выходе старшего разряда счетчика будет меньше в  $K_{сч}$  раз:

$$K_{сч} = f_{вх} / f_{вых} \quad (2)$$

Поэтому счетчики можно использовать в качестве делителей частоты, величина  $K_{сч}$  в этом случае будет называться коэффициентом деления. Для увеличения  $K_{сч}$  приходится увеличивать число триггеров в схеме счетчика. Каждый дополнительный триггер удваивает число состояний счетчика, а, следовательно, и число  $K_{сч}$ . Для уменьшения коэффициента  $K_{сч}$  можно в качестве выхода счетчика рассматривать выходы триггеров промежуточных каскадов. Например, для счетчика на трех триггерах  $K_{сч} = 8$ , если взять выход 2-го триггера, то  $K_{сч} = 4$ . При этом  $K_{сч}$  всегда будет являться целой степенью числа 2, а именно: 2, 4, 8, 16 и т. д.

Интегральная микросхема К555ИЕ5 содержит 4 триггера. Первый триггер работает как делитель на 2. Он имеет тактовый вход С0 и выход Q0. Три остальных триггера образуют делитель на 8. Этот делитель имеет вход С1 и три выхода: Q1, Q2 и Q3. Оба делителя

могут работать независимо друг от друга. Для организации счетчика-делителя на 16 нужно выход  $Q_0$  делителя на 2 соединить с тактовым входом  $C_1$  делителя на 8. На рис. 2 показано условное графическое обозначение двоичного счетчика К555ИЕ5, включенного с коэффициентом пересчета  $K_{сч} = 16$ .

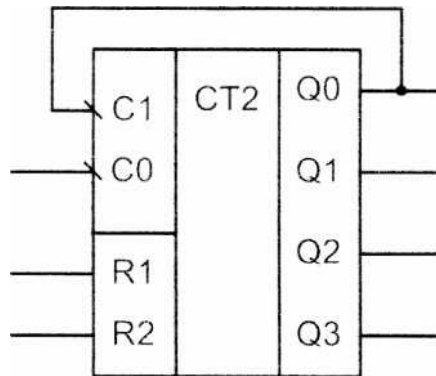


Рисунок 2 – Условное графическое обозначение двоичного счетчика К555ИЕ5

Режимы работы микросхемы К555ИЕ5, включенной с коэффициентом пересчета  $K_{сч} = 16$ , при различных значениях входных сигналов приведены в табл. 1.

Таблица 1. Режимы двоичного суммирующего счётчика

<i>Режим работы</i>	<i>Вход</i>			<i>Выход</i>			
	<i>R1</i>	<i>R2</i>	<i>C0</i>	<i>Q0</i>	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>
Сброс	1	1	х	0	0	0	0
Счёт	0	1	↓	Увеличение кода			
	1	0	↓				
	0	0	↓				

**Примечания:**

- символ х обозначает безразличное состояние входа;
- символ ↓ обозначает срез тактового сигнала.

Микросхема имеет два входа асинхронного сброса R1 и R2, которые объединены логической функцией «И». При одновременной

подаче сигналов логической 1 на входы сброса все триггеры устанавливаются в состояние логического 0. В режиме счета по срезу каждого тактового импульса, поступающего на вход C0, происходит увеличение выходного кода счетчика на единицу.

### Оборудование

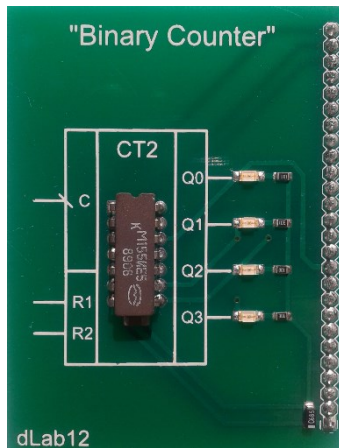


Рисунок 3 – Внешний вид модуля счётчика.

Исследуемый счётчик размещён на лабораторном модуле в виде печатной платы. Его внешний вид представлен на рисунке 3.

Модуль устанавливается в разъём цифровых устройств макетной платы рабочей станции ELVIS II. Исследуемый модуль устанавливается в разъём только при выключенном питании рабочей станции. Размещение модуля на макетном поле рабочей станции представлено на рисунке 4.

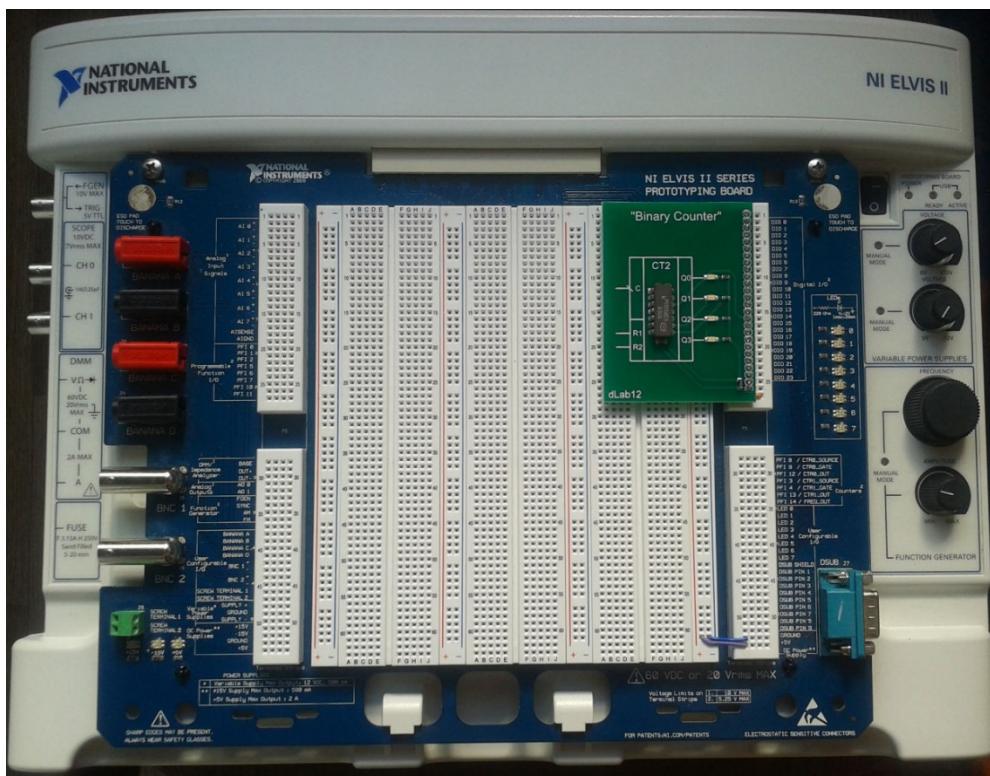


Рисунок 4 – Рабочая станция с исследуемым модулем

Перед установкой исследуемого модуля при необходимости следует установить ограничители посадочного места и проводники питания в соответствии с рисунком 5.

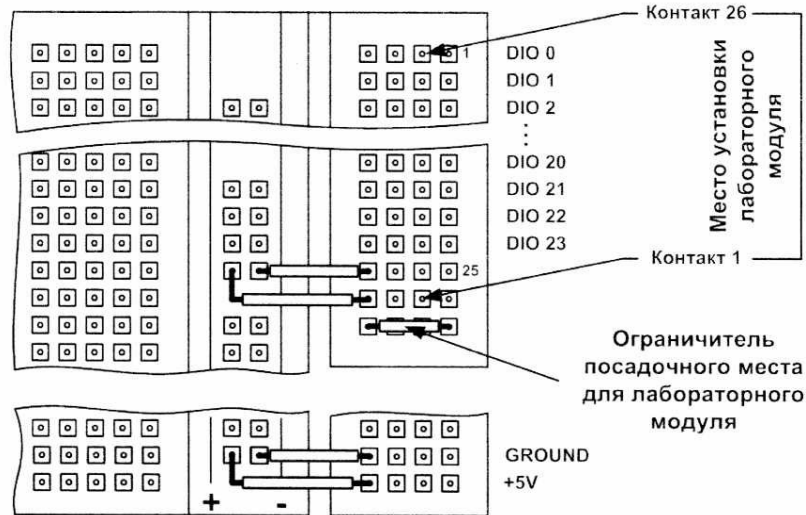


Рисунок 5 – Размещение дополнительных проводников на макетной плате рабочей станции

### Выполнение лабораторной работы

- Выключите рабочую станцию ELVIS II выключателем на задней панели, если ранее она была включена.
- Соедините кабелем USB-2 разъёмы рабочей станции и персонального компьютера (ПК).
- При установленном исследуемом модуле включите рабочую станцию выключателем на задней панели.
- Выключателем в правой верхней части рабочей станции включите питание макетной платы (должен загореться зелёный индикатор).
- Загрузить файл виртуального прибора для исследования регистра сдвига dLab-12.vi. при этом на экране ПК появится изображение панели виртуального прибора (ВП) представленное на рисунке 6.



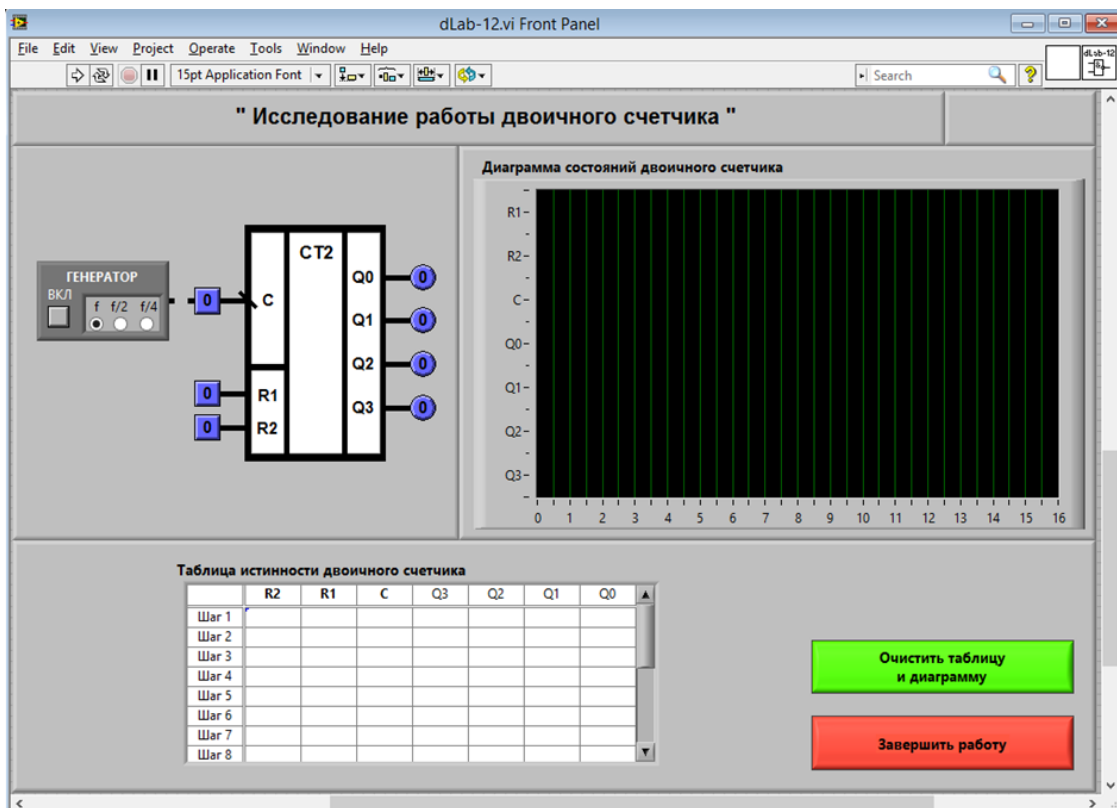


Рисунок 6 – Панель виртуального прибора

Дальнейшее управление исследованием осуществляется исключительно с панели виртуального прибора.

### Исследование двоичного счетчика в статическом режиме

Статический режим исследования счетчика реализуется при подаче на его тактовый вход «С» одиночных импульсов в ручном режиме. Для этого генератор импульсов («ГЕНЕРАТОР»), расположенный на лицевой панели ВП, должен быть выключен (кнопка «ВКЛ» отжата). Подача одиночного импульса прямоугольной формы на тактовый вход «С» счетчика производится однократным нажатием с помощью манипулятора мышь на кнопку квадратной формы, расположенную около этого входа.

- 1 Выключите генератор импульсов, если он был включен.
- 2 Нажмите на кнопку «Очистить таблицу и диаграмму».
- 3 Установите на входах асинхронного сброса счетчика сигналы  $R1=0$  и  $R2=0$ . Логический уровень изменяется при однократном нажатии с помощью манипулятора мышь на кнопку

- квадратной формы, расположенную около соответствующего входа. При этом на кнопке отображается состояние входа («0» - синий цвет или «1» - оранжевый цвет).
- 4 Выполните сброс счетчика. Для этого установите оба входа асинхронного сброса «R1» и «R2» сначала в состояние «1», а затем в состояние «0». На индикаторах выходных сигналов счетчика «Q0», «Q1», «Q2» и «Q3» должны установиться нулевые значения.
  - 5 Нажмите и отпустите кнопку, расположенную около входа «С». На индикаторах круглой формы, расположенных около выходов «Q0», «Q1», «Q2» и «Q3» счетчика, будет отображено состояние его выходных сигналов. Логические состояния входов и выходов триггера будут автоматически занесены в таблицу истинности и на диаграмму состояний. В графу «С» таблицы истинности заносится символ «LГ», означающий подачу импульса на вход «С».
  - 6 Повторите п. 5 пятнадцать раз подряд для получения полного цикла пересчета счетчика.
  - 7 Скопируйте таблицу истинности и диаграмму состояний в отчет. Копирование таблицы следует выполнить за два приема. Сначала сдвиньте полосу вертикальной прокрутки таблицы в верхнее положение и скопируйте первую половину таблицы истинности в буфер обмена, щелкнув правой кнопкой мыши на изображении таблицы и выбрав из контекстного меню команду «Copy Data». Затем перейдите в редактор **MS Word** и вставьте изображение таблицы из буфера обмена на страницу отчета. Для копирования второй половины таблицы сдвиньте полосу вертикальной прокрутки таблицы в нижнее положение и повторите описанные действия. Копирование диаграммы состояний выполняется аналогично.
  - 8 По таблице состояний определите тип исследуемого счетчика: суммирующий или вычитающий, а также его коэффициент пересчета  $K_{сч}$ .

## Исследование счётчика в динамическом режиме

Динамический режим исследования счетчика реализуется при подаче на его тактовый вход «С» последовательности импульсов. Для этого генератор импульсов («ГЕНЕРАТОР»), расположенный на лицевой панели ВП, должен быть включен (кнопка «ВКЛ» нажата). На выходе генератора формируется последовательность прямоугольных импульсов и подается на вход «С» триггера. С помощью кнопок « $f$ », « $f/2$ » и « $f/4$ » можно изменять частоту следования импульсов для выбора удобного режима наблюдения временной диаграммы.

- 1 Нажмите на кнопку «Очистить таблицу и диаграмму».
- 2 Включите генератор импульсов. На графический индикатор выводится временная диаграмма входных и выходных сигналов регистра. В этом режиме таблица истинности не заполняется, а кнопка «Очистить таблицу и диаграмму» заблокирована от нажатия и имеет затененное изображение.
- 3 Изменяя в процессе работы счетчика состояние входов асинхронного сброса «R1» и «R2», определите по временной диаграмме, при каких состояниях этих входов счетчик находится в режиме счета, а при каких - в режиме сброса. Для удобства наблюдения и анализа временных диаграмм можно остановить работу триггера, выключив тактовый генератор. По результатам исследований заполните табл. 2.

Таблица 2.

<i>Вход R2</i>	<i>Вход R1</i>	<i>Режим работы</i>
0	0	
0	1	
1	0	
1	1	

- 4 Изучите работу двоичного счетчика в режиме счета. По временной диаграмме и выходным индикаторам «Q0», «Q1», «Q2» и «Q3» определите, по какому перепаду уровня им-

пульсов на входе «С» («0» → «1» или «1» → «0») происходит переключение счетчика.

- 5 Скопируйте в отчет временные диаграммы, отражающие полный цикл работы счетчика в режимах счета и сброса.
- 6 Выключите ВП, для чего нажмите на панели ВП кнопку «Завершить работу».

### **Содержание отчёта**

- Отчёт должен содержать титульный лист;
- Цель работы;
- Краткие теоретические сведения;
- Таблицы функционирования для каждого раздела исследований;
- Временные диаграммы для каждого раздела исследований;
- Краткое описание алгоритма функционирования; представленного каждой временной диаграммой;
- Ответы на контрольные вопросы;
- Список использованных источников.

### **Контрольные вопросы**

- Что такое счетчик, какие разновидности счётчиков вы знаете?
- Предложите схемы суммирующих и вычитающих счётчиков на D- триггерах и JK-триггерах.
- Как преобразовать суммирующий счетчик в вычитающий?
- Что такое коэффициент пересчета счетчика?

## 7 СРЕДСТВА РАЗРАБОТКИ МИКРОСИСТЕМ НА МИКРОКОНТРОЛЛЕРАХ AVR

### 1 Цель работы

Изучение основ архитектуры микроконтроллеров серии AVR и базовых средств разработки и отладки программного обеспечения.

### 2 Краткие сведения о микроконтроллерах AVR

#### 2.1 Общие особенности микроконтроллеров AVR

Микроконтроллеры серии AVR выпускаются компанией Atmel с 1997 года. Название происходит от Alf-Egil Bogen и Vegard Wollen +RISC. На архитектуре AVR базируются три семейства контроллеров: "tiny", "classic" и "mega". Они отличаются объёмом памяти, составом периферийных устройств на кристалле, количеством портов ввода вывода, количеством выводов корпусов, предельными тактовыми частотами. Обобщённая структурная схема микроконтроллеров представлена на рисунке 1.

Отметим ключевые особенности платформы AVR 8-bit RISC.

- Скоростная **RISC**-архитектура Гарвардского типа с 32 регистрами общего назначения.
- **Flash**-память программ, которая может быть загружена как с помощью параллельного программатора, так и с помощью **SPI**-интерфейса (трёхпроводной последовательный интерфейс), в том числе непосредственно на целевой плате не менее 1000 раз. При этом обеспечена функциональная совместимость AVR с объёмом памяти программ от 1 до 128 кбайт.

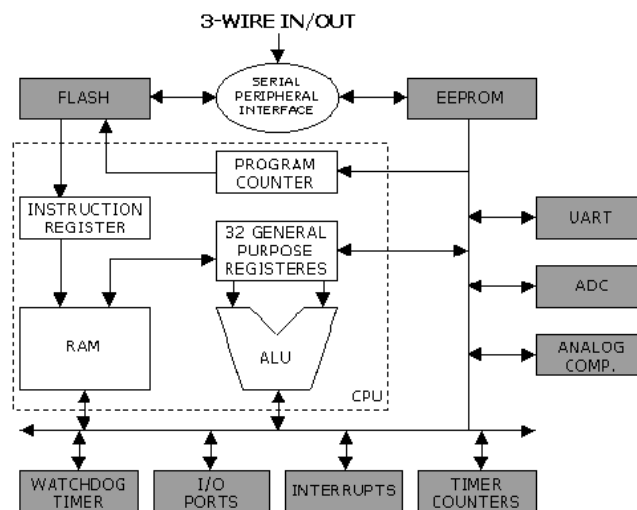


Рисунок 1- Структурная схема микроконтроллеров AVR.

- **EEPROM**-энергонезависимая электрически стираемая память для хранения промежуточных данных, различных констант, таблиц перекодировок, калибровочных коэффициентов с числом циклов перезаписи - не менее 100000.
- Внутренняя оперативная память **RAM** у всех AVR семейств "classic" и "mega" и у некоторых кристаллов семейства "tiny", с возможностью подключения для некоторых микроконтроллеров внешней памяти данных объемом до 64К.
- Важнейшая составляющая микроконтроллеров AVR - активно развивающаяся периферия, отслеживающая мировые требования к современным микроконтроллерам.
- Сторожевой (**WATCHDOG**) таймер предназначен для защиты микроконтроллера от сбоев в процессе работы. Он имеет свой собственный RC-генератор, работающий на частоте 1 МГц. Эта частота является приближенной и зависит прежде всего от величины напряжения питания микроконтроллера и от температуры. WATCHDOG-таймер снабжен своим собственным предделителем входной частоты с программируемым коэффициентом деления, что позволяет подстраивать временной интервал переполнения таймера и сброса микроконтроллера. WATCHDOG-таймер может быть отключен программным образом во время работы

микроконтроллера как в активном режиме, так и в любом из режимов пониженного энергопотребления. В последнем случае это приводит к значительному снижению потребляемого тока.

- Микроконтроллеры AVR имеют в своем составе от 1 до 4 таймеров/счетчиков общего назначения (**Timer Counters**) с разрядностью 8 или 16 бит, которые могут работать и как таймеры от внутреннего источника опорной частоты, и как счетчики внешних событий с внешним тактированием. Все таймеры/счётчики имеют программируемые предделители входной частоты с различными градациями коэффициента деления. Отличительной чертой является возможность работы таймеров/счетчиков на основной тактовой частоте микроконтроллера без предварительного ее понижения, что существенно повышает точность генерации временных интервалов системы. Они могут функционировать независимо от режима работы процессорного ядра микроконтроллера (т.е. они могут быть как считаны, так и загружены новым значением в любое время). Допускается работа от внутреннего источника опорной частоты, или в качестве счетчика событий. Верхний частотный порог определен в этом случае как половина основной тактовой частоты микроконтроллера. Выбор перепада внешнего источника (фронт или срез) программируется пользователем. Важным обстоятельством является наличие различных векторов прерываний для различных событий (переполнение, захват, сравнение).
- **I/O Ports** - порты ввода/вывода имеют от 3 до 53 независимых линий "Вход/Выход". Каждый разряд порта может быть запрограммирован на ввод или на вывод информации.
- **Analog comparator** - аналоговый компаратор входит в состав большинства микроконтроллеров AVR. Типовое напряжение смещения равно 10 мВ, время задержки распространения составляет 500 нс и зависит от напряжения питания микроконтроллера. Так, например, при напряжении питания 2,7 Вольт оно равно 750 нс. Аналоговый компаратор имеет свой собственный вектор прерывания в общей системе прерываний микроконтроллера. При

этом тип перепада, вызывающий запрос на прерывание при срабатывании компаратора, может быть запрограммирован пользователем как фронт, срез или переключение. Логический выход компаратора может быть программным образом подключен ко входу одного из 16-разрядных таймеров/счетчиков, работающего в режиме захвата. Это дает возможность измерять длительность аналоговых сигналов, а также максимально просто реализовать АЦП двухтактного интегрирования.

- **ADC** - аналого-цифровой преобразователь (**АЦП**) построен по классической схеме последовательных приближений с устройством выборки/хранения (**УВХ**). Каждый из аналоговых входов может быть соединен со входом УВХ через аналоговый мультиплексор. Устройство выборки/хранения имеет свой собственный усилитель и гарантирует, что измеряемый аналоговый сигнал будет стабильным в течение всего времени преобразования. Разрядность АЦП составляет 10 бит при нормируемой погрешности  $\pm 2$  единицы младшего разряда. АЦП может работать в двух режимах - однократное преобразование по любому выбранному каналу и последовательный циклический опрос всех каналов. Время преобразования выбирается программно с помощью установки коэффициента деления частоты специального предделителя, входящего в состав блока АЦП. Оно составляет 70...280 мкс для ATmega103 и 65...260 мкс для всех остальных микроконтроллеров, имеющих в своем составе АЦП. Важной особенностью аналого-цифрового преобразователя является функция подавления шума при преобразовании. Пользователь имеет возможность, выполнив короткий ряд программных операций, запустить АЦП в то время, когда центральный процессор находится в одном из режимов пониженного энергопотребления. При этом на точность преобразования не будут оказывать влияние помехи, возникающие при работе процессорного ядра.
- **AVR** - микроконтроллеры могут быть переведены программным путем в один из **шести режимов пониженного энергопо-**



**требления.** Для разных семейств AVR и разных микроконтроллеров в пределах каждого семейства изменяются количество и сочетание доступных режимов пониженного энергопотребления. Подробную информацию можно найти в оригинальной технической документации Atmel Corporation (Microship).

Режим холостого хода (**IDLE**), в котором прекращает работу только процессор и фиксируется содержимое памяти данных, а внутренний генератор синхросигналов, таймеры, система прерываний и WATCHDOG-таймер продолжают функционировать.

Режим микропотребления (**Power Down**), в котором сохраняется содержимое регистрового файла, но останавливается внутренний генератор синхросигналов. Выход из Power Down возможен либо по общему сбросу микроконтроллера, либо по сигналу (уровень) от внешнего источника прерывания. При включенном WATCHDOG-таймере ток потребления в этом режиме составляет около 60...80 мкА, а при выключенном - менее 1 мкА для всех типов AVR. Вышеприведенные значения справедливы для величины питающего напряжения 5 В.

Режим сохранения энергии (**Power Save**), который реализован только у тех микроконтроллеров, которые имеют в своем составе систему реального времени. В основном, режим Power Save идентичен Power Down, но здесь допускается независимая работа дополнительного таймера/счетчика **режима реального времени (RTC)**. Выход из режима Power Save возможен по прерыванию, вызванному или переполнением таймера/счетчика RTC, или срабатыванием блока сравнения этого счетчика. Ток потребления в этом режиме составляет 6...10 мкА при напряжении питания 5 В на частоте 32,768 КГц.

Режим подавления шума при работе аналого-цифрового преобразователя (**ADC Noise Reduction**). Как уже отмечалось, в этом режиме останавливается процессорное ядро, но разрешена работа АЦП, двухпроводного интерфейса **I2C** и сторожевого таймера.

Основной режим ожидания (**Standby**). Идентичен режиму Power Down, но работа тактового генератора не прекращается. Это гарантирует быстрый выход микроконтроллера из режима ожидания всего за 6 тактов генератора.

Дополнительный режим ожидания (**Extended Standby**). Идентичен режиму Power Save, но работа тактового генератора тоже не прекращается. Это гарантирует быстрый выход микроконтроллера из режима ожидания всего за 6 тактов генератора.

– Микроконтроллеры AVR функционируют в широком диапазоне питающих напряжений от 1,8 до 6,0 Вольт. Энергопотребление в активном режиме зависит от величины напряжения питания, от рабочей частоты и от конкретного типа микроконтроллера.

С точки зрения программиста AVR представляет собой 8-разрядный RISC микроконтроллер, имеющий быстрый Гарвардский процессор, память программ, память данных, порты ввода/вывода и различные интерфейсные схемы. Гарвардская архитектура AVR реализует полное логическое и физическое разделение не только адресных пространств, но и информационных шин для обращения к памяти программ и к памяти данных, причем способы адресации и доступа к этим массивам памяти также различны. Подобное построение уже ближе к структуре цифровых сигнальных процессоров и обеспечивает существенное повышение производительности. Центральный процессор работает одновременно как с памятью программ, так и с памятью данных; разрядность шины памяти программ расширена до 16 бит.

В микроконтроллерах AVR используется одноуровневый конвейер при обращении к памяти программ и короткая команда в общем потоке выполняется за один машинный цикл. Цикл у AVR составляет всего один период тактовой частоты. При этом пиковая производительность соответствует одному MIPS/MHz (миллион инструкций в секунду на один мегагерц тактовой частоты).

Следующая отличительная черта архитектуры микроконтроллеров AVR - регистровый файл быстрого доступа, структурная схема которого показана на рисунке 2.

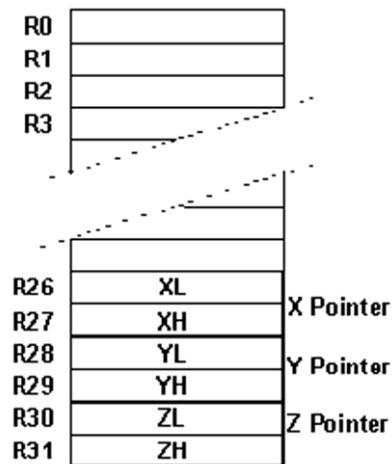


Рисунок 2- Регистровый файл AVR.

Каждый из 32-х регистров общего назначения длиной один байт непосредственно связан с арифметико-логическим устройством (ALU) процессора. Другими словами, в AVR существует 32 регистра - аккумулятора. Это обстоятельство позволяет в сочетании с конвейерной обработкой выполнять одну операцию в арифметико-логическом устройстве (ALU) за один машинный цикл. Так, два операнда извлекаются из регистрового файла, выполняется команда и результат записывается обратно в регистровый файл в течение только одного машинного цикла.

Шесть из 32-х регистров файла могут использоваться как три 16-разрядных указателя адреса при косвенной адресации данных. Один из этих указателей (**Z Pointer**) применяется также для доступа к данным, записанным в памяти программ микроконтроллера. Использование трех 16-битных указателей (X, Y и Z Pointers) существенно повышает скорость пересылки данных при работе прикладной программы.

Регистровый файл занимает младшие 32 байта в общем адресном пространстве SRAM AVR. Такое решение позволяет получать доступ к быстрой "регистровой" оперативной памяти микроконтроллера двумя путями - непосредственной адресацией в коде команды к любой ячейке и другими способами адресации ячеек SRAM. В технической документации фирмы Atmel это полезное свойство носит название "быстрое контекстное переключение". Оно

является еще одной отличительной особенностью архитектуры AVR, повышающей эффективность работы микроконтроллера и его производительность (следует отметить, что принцип контекстного переключения трактуется несколько вольно, а пример его корректной реализации можно увидеть у микроконтроллеров фирмы Zilog семейств Z86 и Z8Encore). Особенно заметно данное преимущество при реализации процедур целочисленной 16-битной арифметики, когда исключаются многократные пересылки между различными ячейками памяти данных при обработке арифметических операндов в ALU.

Система команд AVR весьма развита и насчитывает до 133 различных инструкций. Почти все команды имеют фиксированную длину в одно слово (16 бит), что позволяет в большинстве случаев объединять в одной команде и код операции, и операнд(ы). Лишь немногие команды имеют размер в 2 слова (32 бит) и относятся к группе команд вызова процедуры CALL, длинных переходов в пределах всего адресного пространства JMP, возврата из подпрограмм RET и команд работы с памятью программ LPM.

Различают пять групп команд AVR: условного ветвления, безусловного ветвления, арифметические и логические операции, команды пересылки данных, команды работы с битами. В последних версиях кристаллов AVR семейства "mega" реализована функция аппаратного умножения.

По разнообразию и количеству реализованных инструкций AVR больше похожи на CISC, чем на RISC процессоры.

АЛУ поддерживает арифметические и логические операции с регистрами, с константами и регистрами. Операции над отдельными регистрами также выполняются в АЛУ.

Кроме регистровых операций, для работы с регистровым файлом могут использоваться доступные режимы адресации, поскольку регистровый файл занимает адреса \$00-\$1F в области данных, обращаться к ним можно как к ячейкам памяти.

Пространство ввода состоит из 64 адресов для периферийных функций процессора, таких как управляющие регистры, таймеры/счетчики и другие. Доступ к пространству ввода/вывода может осуществляться непосредственно, как к ячейкам памяти, расположенным после регистрового файла (\$20-\$5F).

При обработке прерываний и вызове подпрограмм адрес возврата запоминается в стеке. Стек у микроконтроллеров, имеющих ОЗУ размещается в памяти данных общего назначения, соответственно размер стека ограничен только размером доступной памяти данных и ее использованием в программе. Все программы пользователя должны инициализировать указатель стека (SP) в программе, выполняемой после сброса (до того, как вызываются подпрограммы и разрешаются прерывания). 8-разрядный указатель стека доступен для чтения/записи в области ввода/вывода.

Доступ к 128 байтам статического ОЗУ, регистровому файлу и регистрам ввода/вывода осуществляется при помощи пяти доступных режимов адресации, поддерживаемых архитектурой AVR. Все пространство памяти AVR является линейным и непрерывным.

Гибкий модуль прерываний имеет собственный управляющий регистр в пространстве ввода/вывода, и флаг глобального разрешения прерываний в регистре состояния. Каждому прерыванию назначен свой вектор в начальной области памяти программ. Различные прерывания имеют приоритет в соответствии с расположением их векторов. По младшим адресам расположены векторы с большим приоритетом.

Детали архитектуры AVR зависят от типа микроконтроллера. Далее они рассматриваются на примере микроконтроллера AT90S2313.

## **2.2 Организация адресного пространства микроконтроллера AT90S2313**

Адресное пространство микроконтроллера включает в себя 3 независимых подпространства: память программ, перепрограммируемая память данных и статическое ОЗУ данных.

## **2.2 Загружаемая память программ**

АТ90S2313 содержит 2кБ загружаемой флэш-памяти для хранения программ. Поскольку все команды занимают одно 16-разрядное слово, флэш-память организована как 1К 16-разрядных слов. Флэш-память выдерживает не менее 1000 циклов перезаписи. Программный счетчик имеет ширину 10 бит и таким образом адресуется к 1024 словам программной флэш-памяти. Таблицы констант могут располагаться в диапазоне адресов 0-2К.

## **2.3 EEPROM память данных**

АТ90S2313 содержит 128 байт электрически стираемой энергонезависимой памяти (EEPROM). EEPROM организована как отдельная область данных, каждый байт которой может быть прочитан и перезаписан. EEPROM выдерживает не менее 100000 циклов записи/стирания. Доступ к энергонезависимой памяти данных задается регистром адреса, регистром данных и управляющим регистром.

## 2.4 Статическое ОЗУ данных

На рисунке 3 показана организация памяти данных в AT90S2313.

<b>Регистровый файл</b>	<b>Область адресов данных</b>
R0	\$00
R1	\$01
...	...
R30	\$1E
R31	\$1F

<b>Регистры вв./выв</b>	<b>Встроенное ОЗУ</b>
\$00	\$20
\$00	\$21
...	...
\$3E	\$5E
\$3F	\$5F

Рисунок 3- Организация памяти данных

224 ячейки памяти включают в себя регистровый файл, память ввода/вывода и статическое ОЗУ данных. Первые 96 адресов используются для регистрового файла и памяти ввода/вывода, следующие 128 - для ОЗУ данных.

## 2.5 Файл регистров общего назначения

Все команды, оперирующие регистрами прямо адресуются к любому из регистров за один машинный цикл. Исключением являются пять команд, оперирующих с константами SBCI, SUBI, CPI,

ANDI, ORI и команда LDI, загружающая регистр константой. Эти команды работают только со второй половиной регистрового файла - R16-R31. Команды SBC, SUB, CP, AND и OR, также, как и все остальные, применимы ко всему регистровому файлу.

Каждому регистру присвоен адрес в пространстве данных, они отображаются на первые 32 ячейки ОЗУ. Хотя регистровый файл физически размещен вне ОЗУ, подобная организация памяти дает гибкий доступ к регистрам. Регистры X, Y и Z могут использоваться для индексации любого регистра.

Кроме обычных функций, регистры R26-R31 имеют дополнительные функции, эти регистры можно использовать как адресные указатели в области памяти данных. Эти регистры обозначаются как X, Y, Z и определены следующим образом:

	15	0		
Регистр X	7	0	7	0
	R27(\$1B)		R26(\$1A)	
	15	0		
Регистр Y	7	0	7	0
	R29(\$1D)		R28(\$1C)	
	15	0		
Регистр Z	7	0	7	0
	R31(\$1F)		R30(\$1E)	

При различных режимах адресации эти регистры могут использоваться как фиксированный адрес, для адресации с автоинкрементом или с автодекрементом.

При обращении к памяти используются пять различных режимов адресации: прямой, косвенный со смещением, косвенный, косвенный с предварительным декрементом и косвенный с постинкрементом. Регистры R26-R31 регистрового файла используются как указатели для косвенной адресации.



Косвенная адресация со смещением используется для доступа к 63 ячейкам, базовый адрес которых задается содержимым регистров Y или Z. Для косвенной адресации с инкрементом и декрементом адреса используются адресные регистры X, Y и Z.

При помощи любого из этих режимов производится доступ ко всем 32 регистрам общего назначения, 64 регистрам ввода/вывода и 128 ячейкам ОЗУ.

## 2.7 Пространство ввода/вывода

Ниже приведено описание пространства ввода/вывода для процессоров AT90S2313.

Все устройства ввода/вывода и периферийные устройства AT90S2313 располагаются в пространстве ввода/вывода. Различные ячейки этого пространства доступны через команды IN и OUT, пересылающие данные между одним из 32-х регистров общего назначения и пространством ввода/вывода. К регистрам \$00..\$1F можно осуществлять побитовый доступ командами SBI и CBI. Значение отдельного бита этих регистров можно проверить командами SBIC и SBIS. Дополнительную информацию по этому вопросу можно найти в описании системы команд.

При использовании специальных команд **IN**, **OUT**, **SBIS** и **SBIC**, должны использоваться адреса \$00-\$3F. При доступе к регистру ввода/вывода как к ячейке ОЗУ, к его адресу необходимо добавить \$20. В приведенной ниже таблице 1 адреса регистров в памяти данных приведены в скобках.

**Таблица 1. Пространство ввода/вывода AT90S2313**

\$3F(\$5F)	SREG	Status REGister	Регистр Состояния
\$3D(\$5D)	SPL	Stack pointer low	Указатель стека, мл. байт
\$3B(\$5B)	GIMSK	General Interrupt MaSK register	Общий регистр маски прерываний
\$3A(\$5A)	GIFR	General Interrupt Flag register	Общий регистр флагов прерываний
\$39(\$59)	TIMSK	Timer/counter Interrupt mask register	Регистр маски прерываний от таймера/счетчика

\$38(\$58)	TIFR	Timer/counter Interrupt Flag register	Регистр флага прерывания таймера/счетчика
\$35(\$55)	MCUCR	MCU general Control Register	Общий регистр управления микроконтроллером
\$33(\$53)	TCCR0	Timer/Counter 0 Control Register	Регистр управления таймером счетчиком 0
\$32(\$52)	TCNT0	Timer/Counter 0 (8-бит)	Таймер/счетчик 0 (8 бит)
\$2F(\$4F)	TCCR1A	Timer/Counter 1 Control Register A	Регистр А управления таймером счетчиком 1
\$2E(\$4E)	TCCR1B	Timer/Counter 1 Control Register B	Регистр В управления таймером счетчиком 1
\$2D(\$4D)	TCNT1H	Timer/Counter 1 High byte	Таймер/счетчик 1 старший байт
\$2C(\$4C)	TCNT1L	Timer/Counter 1 Low byte	Таймер/счетчик 1 младший байт
\$2B(\$4B)	OCR1H	Output Compare Register 1 high byte	Выход регистра совпадения 1 старший байт
\$2A(\$4A)	ICR1L	Output Compare Register 1 low byte	Выход регистра совпадения 1 младший байт
\$25(\$45)	ICR1H	T/C 1 Input Capture Register High Byte	Регистр захвата Т\С 1 старший байт
\$24(\$44)	ICR1L	T/C 1 Input Capture Register Low Byte	Регистр захвата Т\С 1 младший байт
\$21(\$41)	WDTCR	Watchdog Timer Control Register	Регистр управления сторожевым таймером
\$1E(\$3E)	EEAR	EEPROM Address Register	Регистр адреса энергонезависимой памяти
\$1D(\$3D)	EEDR	EEPROM Data Register	Регистр данных энергонезависимой памяти
\$1C(\$3C)	EECR	EEPROM Control Register	Регистр управления энергонезависимой памятью
\$18(\$38)	PORTB	Data Register, Port B	Регистр данных порта В
\$17(\$37)	DDRB	Data Direction Register Port B	Регистр направления данных порта В
\$16(\$36)	PINB	Input pins, Port B	Выводы порта В
\$12(\$32)	PORTD	Data Register, Port D	Регистр данных порта D

\$11(\$31)	DDRD	Data Direction Register Port D	Регистр направления данных порта D
\$10(\$30)	PIND	Input pins, Port D	Выводы порта D
\$0C(\$2C)	UDR	UART Data Register	Регистр данных последовательного порта
\$0B(\$2B)	USR	UART Status Register	Регистр состояния последовательного порта
\$0A(\$2A)	UCR	UART Control Register	Регистр управления последовательного порта
\$09(\$29)	UBRR	UART Baud Rate Register	Регистр скорости последовательного порта
\$08(\$28)	ACSR	Analog Comparator Control and Status Register	Регистр управления и состояния аналогового компаратора

*Примечание: зарезервированные и неиспользуемые ячейки не показаны.*

Форматы и назначение полей регистров пространства ввода/вывода детально описаны в [6].

### **3 Инструменты разработки и отладки программного обеспечения**

Разработка программного обеспечения (ПО) нередко происходит параллельно с созданием аппаратных средств системы, что создаёт определённые трудности. Для микросистем такая ситуация скорее правило, чем исключение. На практике это означает, что ответы на некоторые вопросы, появляющиеся в процессе разработки программного обеспечения, приходится получать, прибегая к моделированию функций в программной среде чужой системы, или на прототипной системе, не полностью совпадающей с целевой системой.

Разработка ПО многоэтапная процедура. Она начинается с выбора методов решения основных задач и составления алгоритма функционирования системы. Далее осуществляется разбиение программы на отдельные модули и их написание. Важнейшим этапом,

занимающим до 60-70% времени, является отладка ПО. В микросистемах на заключительном этапе **исполняемый или двоичный** код программы должен быть загружен в память программ микроконтроллера с помощью специализированных аппаратно-программных средств – программаторов.

Отладка может осуществляться в среде программного симулятора, с помощью аппаратных эмуляторов, но на последнем этапе - на целевой системе.

### **3.1 Общая характеристика аппаратно-программных средств разработки программного обеспечения**

**Стенд**, используемый в лабораторном цикле, состоит из: платы микроконтроллера с разъёмом программирования по внутреннему стандарту фирмы ATMEL ISP10; подключаемых периферийных устройств; программатора; адаптера питания. Периферийные устройства подключаются к стенду стандартным 40-жильным интерфейсным кабелем IDE.

**Плата микроконтроллера** (рисунки 4, 5) содержит ИС микроконтроллера AT90S2313 с 2Кбайт перепрограммируемой памяти программ; кварцевый резонатор, определяющий тактовую частоту процессора (4 или 10 МГц в зависимости от экземпляра платы); цепь принудительного рестарта микроконтроллера с кнопкой сброса; стабилизатор напряжения питания с выходным напряжением +5 В.

В состав периферийных устройств стенда входят: устройство статической индикации, устройство динамической индикации и адаптер канала RS232.

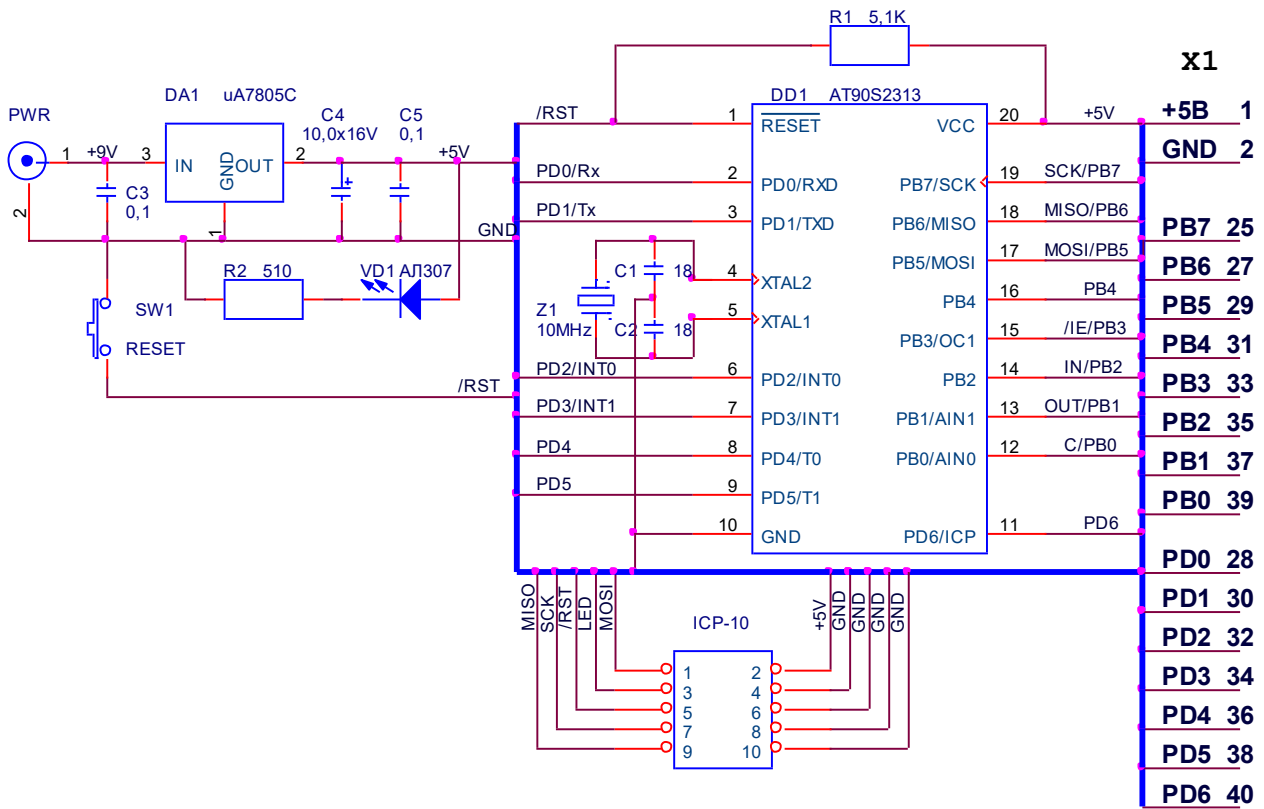


Рисунок 4 - Схема платы микроконтроллера.

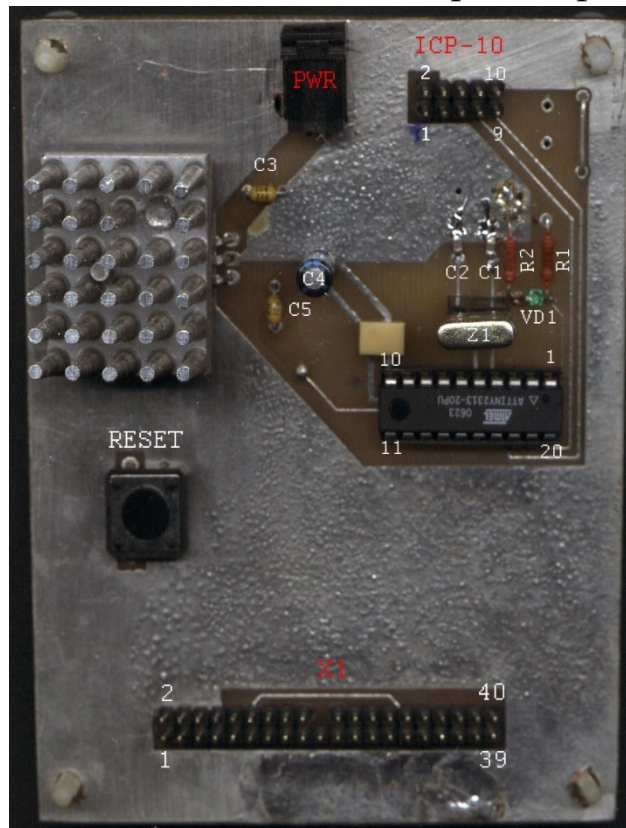


Рисунок 5 - Плата микроконтроллера.

**Устройство статической индикации** содержит 8 светодиодов и два ключа. Светодиоды подключены к порту **В** микроконтроллера. Ключи подключены к входам внешних прерываний микроконтроллера Int0, Int1 (PortD, разряды 2 и 3). Схема и внешний вид устройства представлены на рисунках 6 и 7.

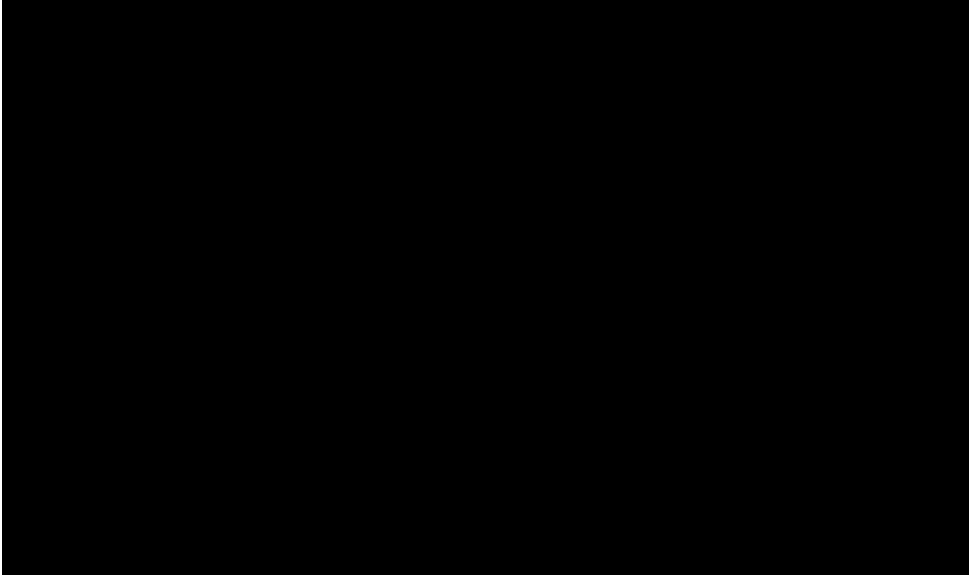


Рисунок 6 - Схема устройства статической индикации.

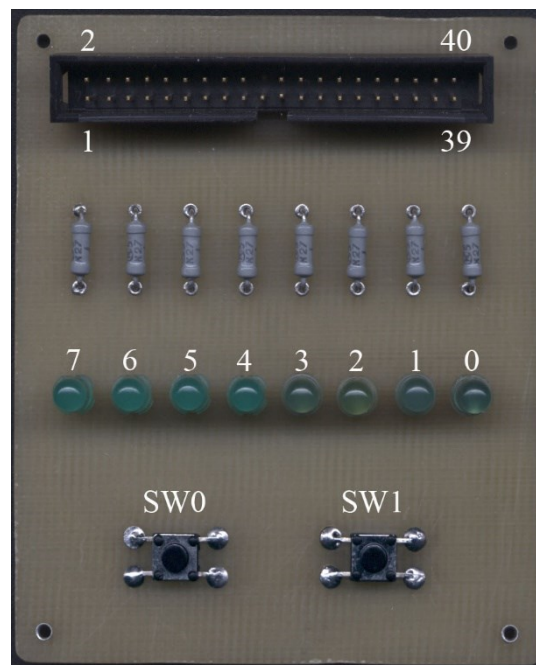


Рисунок 7 - Устройство статической индикации.

Устройство динамической индикации состоит: из 16- битового регистра с последовательной загрузкой на ИС DD6-DD7 (1533ИР24), пяти силовых ключей для управления разрядами индикатора на транзисторах VT1-VT5, пятиразрядного индикаторного устройства на семисегментных светодиодных индикаторах, пяти ключей S1-S5. Светодиодные индикаторы прикрыты красным светофильтром. Схема и внешний вид устройства представлены на рисунках 8 и 9.

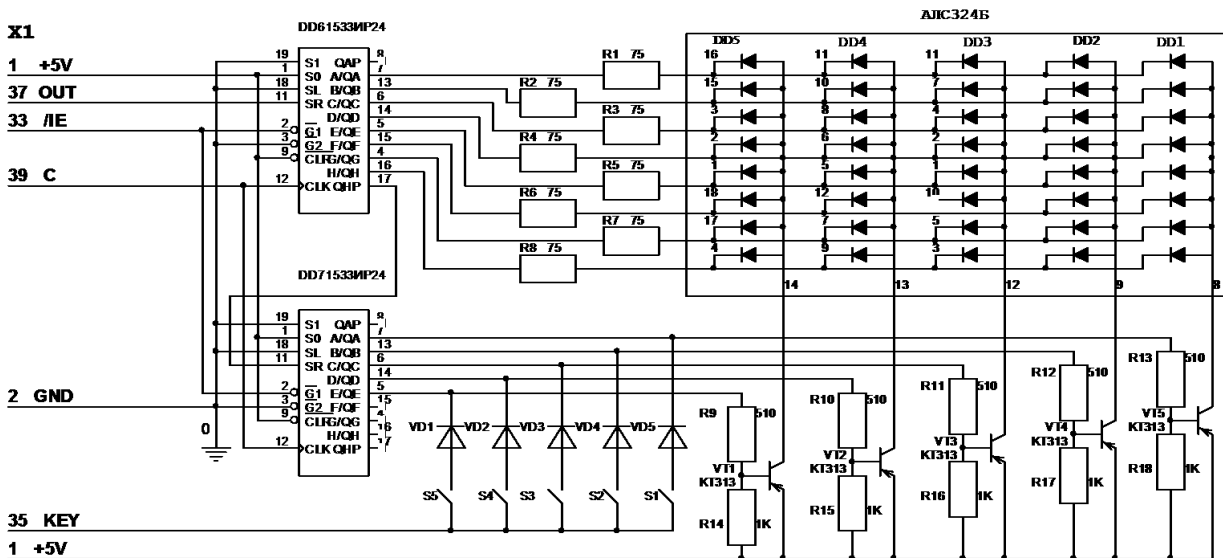


Рисунок 8-Схема устройства динамической индикации.

Третье периферийное устройство - адаптер последовательного канала. Он поддерживает связь микроконтроллера с персональным компьютером. Адаптер представляет собой гальванически развязанный преобразователь уровня. При этом ТТЛ уровни микроконтроллера приводятся к уровням интерфейса RS232 персонального компьютера. Гальваническая развязка на транзисторных оптопары DD4-DD5 типа 4N35 совместно с встроенным изолированным преобразователем постоянного тока допускает режим горячего подключения адаптера к персональному компьютеру.

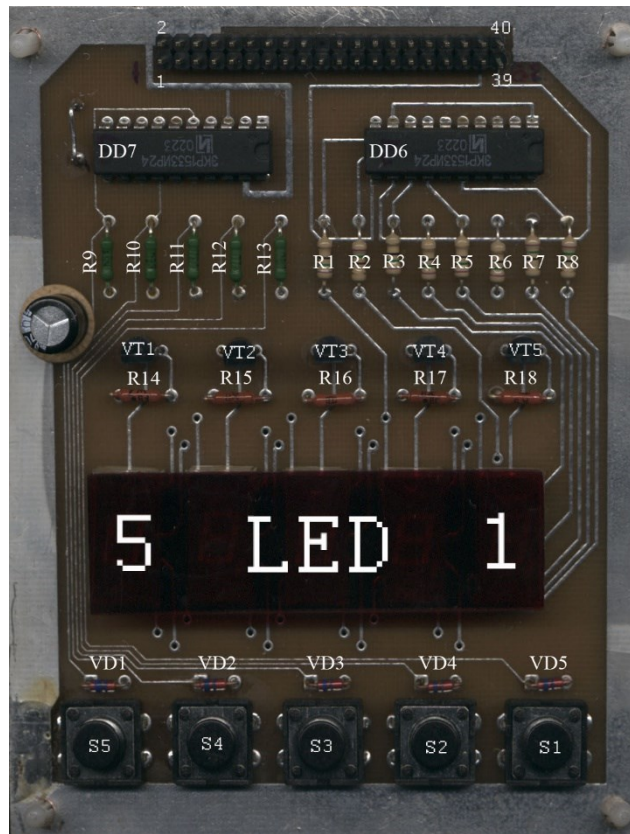


Рисунок 9 - Устройство динамической индикации.

**Программатор** USBasp V2.0 (рисунок 10) имеет несколько исполнений. Подключение программатора к компьютеру осуществляется кабелем или непосредственно, в зависимости от конструкции. К плате микроконтроллера программатор подключается десятипроводным шлейфом. Программатор управляется программой “Khazama”. Программа имеет простой интуитивно понятный интерфейс (рисунок 10).



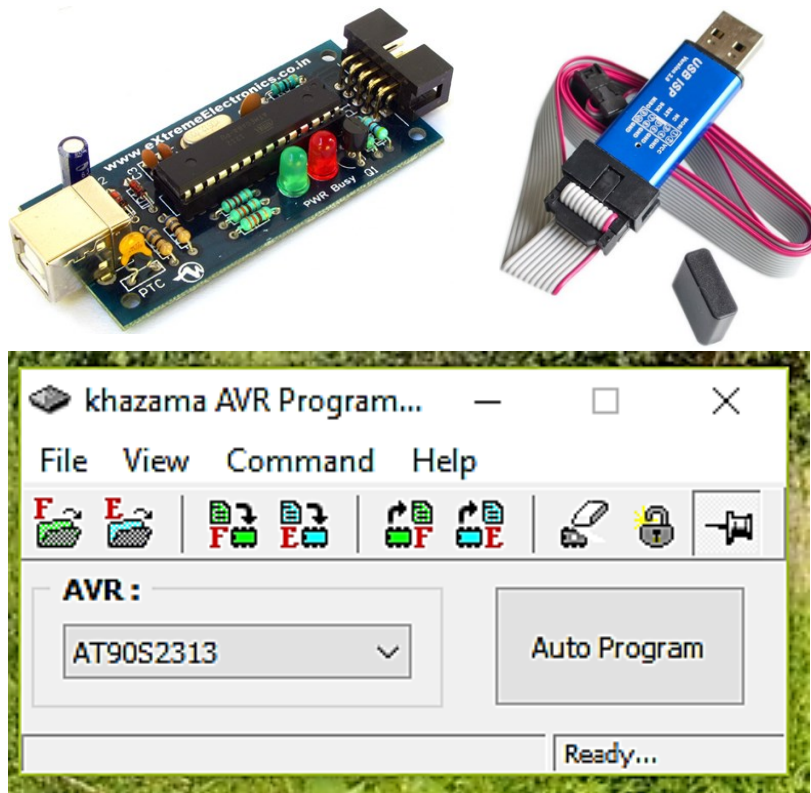


Рисунок 10 - Программатор.

Программное обеспечение состоит из интегрированной среды разработки ASTUDIO4 и программы Khazama, реализующей процедуру программирования микроконтроллера. Интегрированная среда поддерживает программирование на языке **C** и языке **ассемблера**. В лабораторном цикле используется только язык ассемблера.

Интегрированная среда поддерживает все стадии разработки: написание программы, её компиляцию, отладку в режиме симуляции и программирование конечной системы при применении фирменных программаторов. В лабораторном цикле для последней стадии используется специализированный программатор со своим программным обеспечением.

Детальная информация по ассемблеру AVR и полный перечень команд микроконтроллера приведен в фирменных документах [3,4].

### 3.2 Создание/загрузка проекта в ASTUDIO4

Запуск интегрированной среды разработки может быть осуществлён традиционными для Windows способами:

- через ярлык на рабочем столе,
- из меню программ,
- через файл описания существующего проекта с расширением .aps.

В первых двух случаях после запуска программы потребуется создать или загрузить готовый проект. Для загрузки готового проекта следует выбрать функцию открытия проекта и выбрать путь к нему, как показано на рисунке 11.

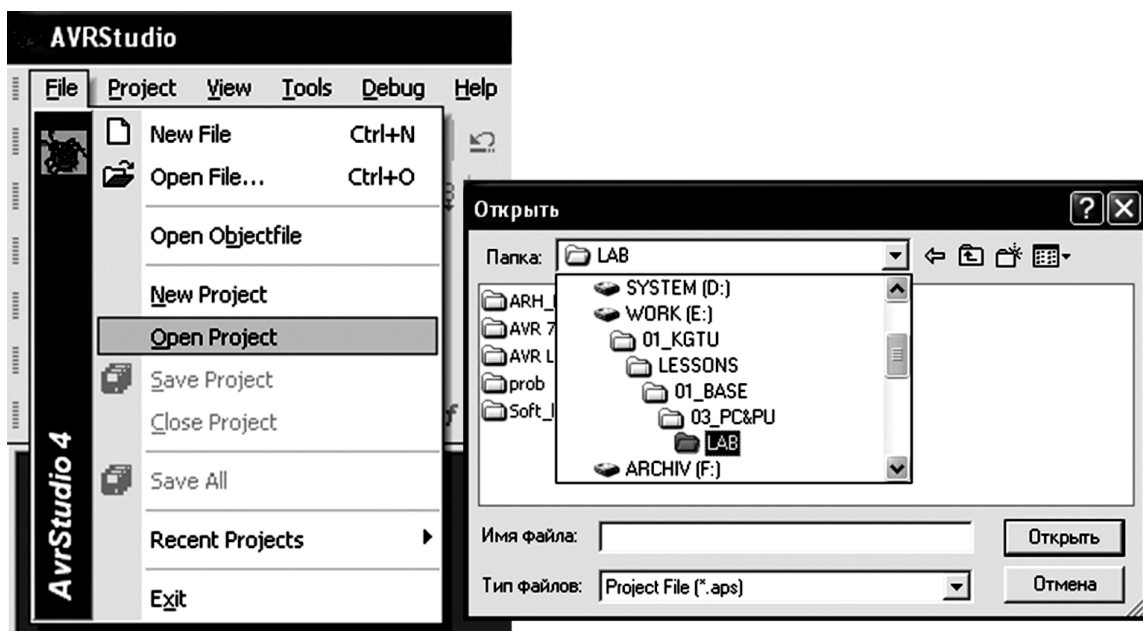


Рисунок 11- Последовательность открытия существующего проекта.

Создание нового проекта осуществляется через пункт меню **New Project** (рисунок11). При этом в всплывающем окне (рисунок12) надо выбрать имя для нового проекта и, указав путь для его размещения, перейти к следующему окну (**Next**).

В следующем окне следует выбрать аппаратные средства для отладки или программный симулятор и завершить создание заготовки нового проекта (**Finish**).

После этого в основном поле программы появится текстовое окно будущей программы.

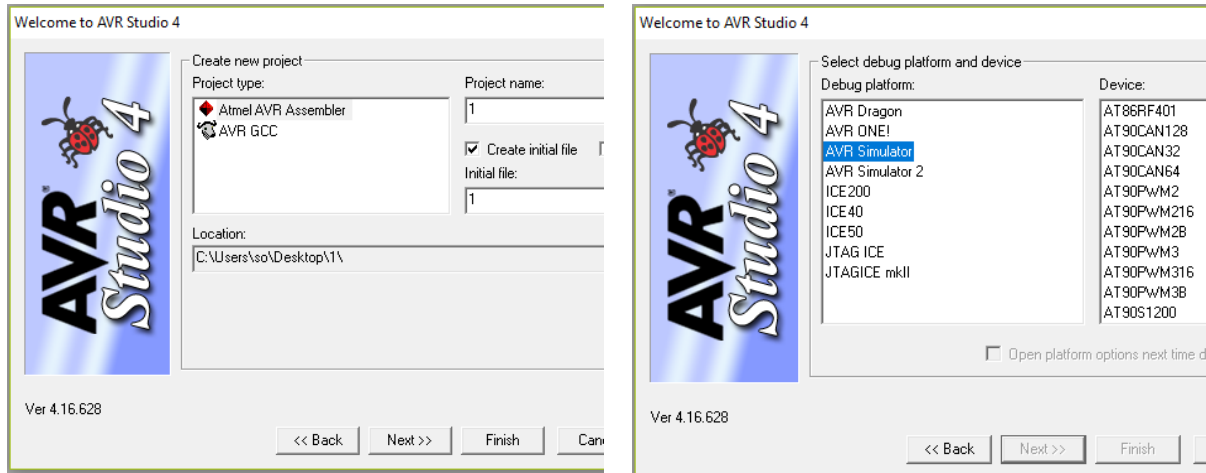


Рисунок 12- Создание нового проекта.

### 3.3 Ассемблер AVR

Средствами встроенного текстового редактора в нём набирается исходный текст программы в соответствии с соглашениями ассемблера. Рекомендуется типовые части текстов заимствовать из готовых проектов и далее редактировать в соответствии с особенностями проекта. Это существенно экономит время при разработке программного обеспечения. Лабораторный цикл целесообразно выполнять на основе шаблонов программ, размещаемых в методических указаниях. Текст программы может быть непосредственно скопирован в редактор среды разработки.

Ниже в качестве примера приведена небольшая программа на языке ассемблера, иллюстрирующая основные соглашения и синтаксис. Она является шаблоном для первой лабораторной работы. Программа осуществляет визуализацию содержимого временного регистра, которое циклически наращивается на единицу на каждом шаге (инкремент).

```

; ***** 01_AVR_L.asm
; визуализация содержимого циклически
; инкрементируемого регистра

.include "2313def.inc"
.device AT90S2313

.def Temp      =r16      ; рабочий регистр
.def Delay_1   =r17 ; регистры хранения кода
.def Delay_2   =r18 ; трёхбайтовой
.def Delay_3   =r19 ; задержки

; Инициализация (настройка оборудования)

INIT:
    ser Temp      ; установка всех бит регистра в единицу
    out DDRB,Temp ; PORTB ориентирован на вывод
;=====
; вывод на индикацию буфера отображения Temp

loop:
    out PORTB,Temp ; вывод данных в PORTB
    inc Temp      ; увеличение на 1 отображаемого кода

;=====
; Задержка на время визуализации.

    ldi Delay_1,0 ;
    ldi Delay_2,0 ; задать величину задержки
    ldi Delay_3,1 ; трёхбайтовым числом

;=====
; задержки для визуализации результата
; реализуется декрементом переменной до заёма

```

- ; время задержки до следующего инкрементирования
- ; определяется количеством повторов 4-х команд цикла
- ; количество повторов представлено двоичным
- ; трёхбайтовым числом «Delay\_3 Delay\_2 Delay\_1»
- ; в данном примере  $1\ 00000000\ 00000000_2 = 100_{16} = 65536_{10}$
- ; однократное повторение цикла занимает 5 тактов МК
- ; при этом общее время задержки примерно 327 680 тактов

DLY:

```

subi    Delay_1,1 ;уменьшить на единицу
sbc    Delay_2,0 ;24-х битовый счётчик
sbc    Delay_3,0 ;задержки и если он не обнулился
brcc   DLY       ;повторять до обнуления, а затем
;=====
rjmp   loop ;повторный цикл визуализации по завершении задержки

```

### Примечание:

1. Задержка может реализовываться вложенными циклами
2. В некоторые программы лабораторного цикла включены преднамеренные ошибки.

Строка текста может иметь одну из четырёх форм:

[метка:]	директива	[операнды]	[Комментарий]
[метка:]	инструкция	[операнды]	[Комментарий]
	Комментарий		
	Пустая строка		

Комментарий имеет следующую форму:

```
; [Текст]
```

Позиции в квадратных скобках необязательны. Текст после точки с запятой (;) и до конца строки игнорируется компилятором. Первый комментарий – имя программы.

Директивы ассемблера начинаются с точки. Первая из них подключает специальный файл, в котором описываются особенности конкретного микроконтроллера. Там же определяются типовые имена для регистров управления, имена отдельных битов и т.п. Вторая директива напрямую определяет тип устройства. Следует обратить внимание, что ASTUDIO выдаст сообщение об ошибке при компиляции проекта, если в подключаемом файле уже имеется директива определения типа устройства. В этом случае вторую директиву следует исключить из текста программы. Полезно ознакомиться с содержанием подключаемых файлов, для чего следует открыть их с помощью любого текстового редактора.

Последующие три директивы определяют имена и адреса регистров, используемых в программе (необязательно, но полезно). Использование символических имён упрощает восприятие программы и облегчает её написание и сопровождение.

После завершения написания теста программы следует выполнить компиляцию проекта через подпункт **Build** пункта основного меню **Project**. При сообщениях об ошибках изучить соответствующие строки исходной программы и исправить ошибки.

### 3.4 Отладка программы

При корректной компиляции переходят к отладке программы. В лабораторном цикле отладка выполняется в режиме симуляции (режим выбирается при создании проекта). Для этого выбирают подпункт меню “**Start Debugging**” пункта “**Debug**” основного меню. Далее в пункте основного меню программы “**View**” выбирают необходимое количество окон для отображения информации о состоянии микроконтроллера. Любое окно может быть настроено на отображение памяти данных, констант, памяти, регистров или регистров ввода вывода, как показано на рисунке 13.

Помимо этого, можно дополнительно включить отображение рабочей области (**Workspace**), информация в которой представляется побитово. Этот режим нагляден при работе с внешними устройствами (рисунок 14.).

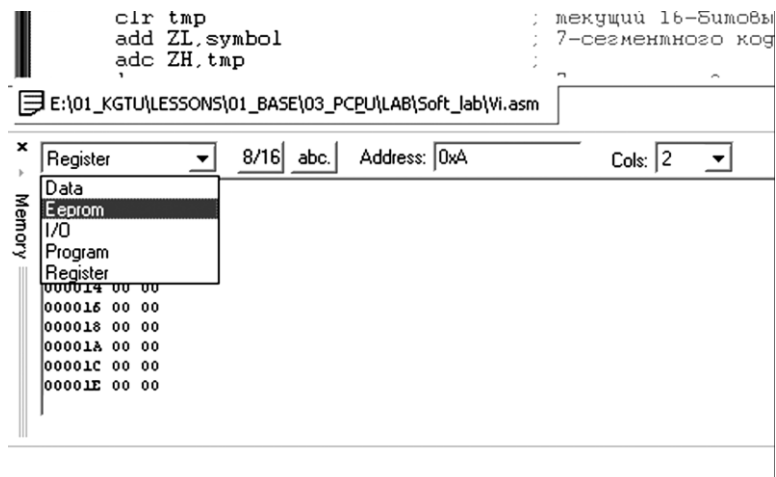


Рисунок 13 – Выбор отображаемых объектов в окне отладчика.

С помощью инструментов отладки на панели инструментов или через подпункты меню “**Debug**” можно провести пошаговую отладку программы, выполнить фрагмент программы, включить автоматический пошаговый режим, установить или снять точки останова.

Выбор адекватных методов отладки и разработка тестовых примеров являются самостоятельной задачей. Кроме того, отладка на симуляторе позволяет проконтролировать и отладить логику программы, но не исключает отладку в режиме реального времени на прототипе системы.

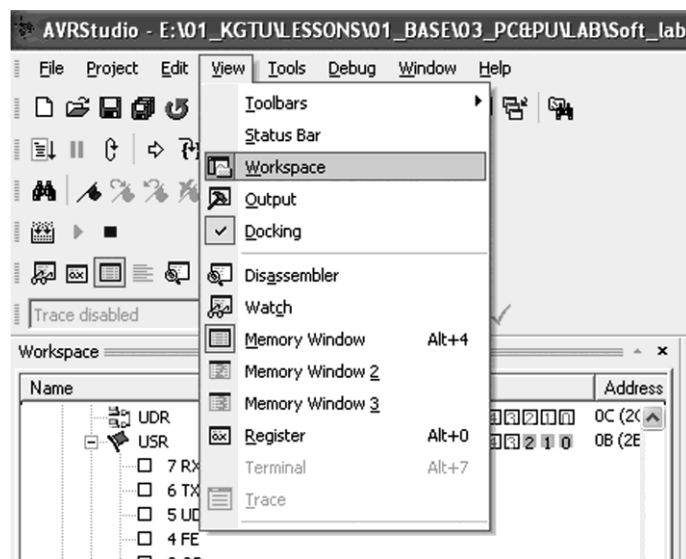


Рисунок 14 – Включение отображения рабочей области.

В лабораторном цикле рекомендуется проводить отладку по приведенной методике.

1. Определить оценку сверху максимального количества тестов необходимых для полной проверки программы. В качестве этой оценки (избыточной) можно взять количество различных путей от точки входа до выхода из программы.
2. Формально описать все тесты и осуществить поочерёдный прогон тестов с контролем результатов до обнаружения первой ошибки.
3. Локализовать ошибку, сочетая метод точек останова и пошаговый режим.
4. Определить тип ошибки и исправить её.
5. Продолжить тестирование, начиная с первого теста.

Программа считается отлаженной в режиме симуляции, если все тесты завершены безошибочно.

Отладчик среды разработки включает все варианты пошаговых процедур.

**Step Into** – остановка осуществляется на вызываемой процедуре, команде без её выполнения.

**Step Over** – остановка осуществляется после выполнения вызываемой процедуры, функции или команды.

**Step Out** – предназначена для выхода из функции в вызывающую функцию. Выполняется функция и осуществляется остановка на первой строке после выхода из неё.

**Auto Step** – автоматический пошаговый режим с визуализацией изменений. При его использовании рекомендуется изменить в требуемую сторону начальные значения переменных или констант, определяющих скорость выполнения программы с целью облегчения наблюдения за состоянием программы.

### **3.5 Программирование микроконтроллера в составе прототипной системы**

В лабораторной работе программирование осуществляется с помощью программатора, подключаемого к USB порту компьютера



и программного продукта Khazama. Перед программированием следует подключить кабель программирования к USB порту компьютера и разъёму программирования стенда. Далее необходимо включить компьютер, затем стенд и загрузить программное обеспечение. Выбрать тип устройства AT90S2313, затем загрузить файл программы в hex формате, нажав на панели инструментов иконку с изображением папки и символом **F**, и/или файл устройства с расширением. e2p, если надо запрограммировать не только память программ, но и память констант, воспользовавшись иконкой с изображением папки и символом **E** (рисунок 15). Выбрать действие можно и в меню команд программатора.

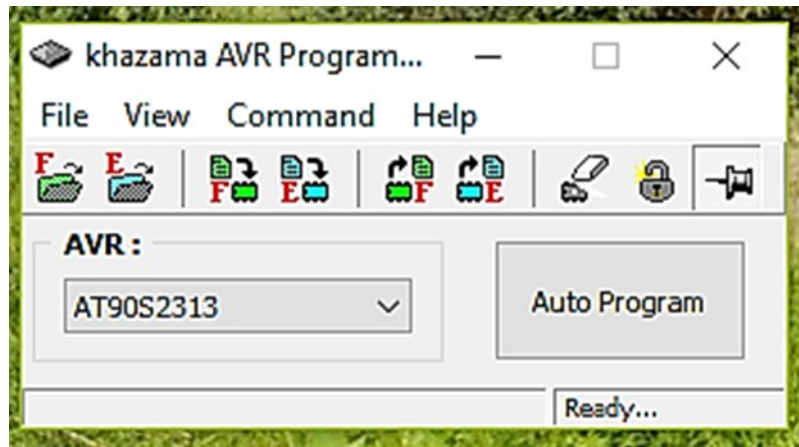


Рисунок 15 - Панель программирования.

С помощью панели инструментов программирования или пункта меню “**Command**” выполнить запись программы, и/или памяти констант. Индикатор хода программирования покажет ход выполнения процедуры. В конце её будет выдано сообщение об ошибке или нормальном завершении процесса программирования. Возможные проблемы могут быть связаны с неправильным подключением кабеля программатора, переходного кабеля к разъёму программирования на плате микроконтроллера (ключ должен быть у первого контакта разъёма), шлейфа соединяющего платы микро-

контроллера и устройства ввода вывода. Если все соединения выполнены правильно, следует в диспетчере устройств убедиться в наличии драйвера устройства программирования.

#### 4 Подготовка к лабораторной работе

При внеаудиторной подготовке к лабораторной работе следует придерживаться изложенного ниже плана.

1. Ознакомиться со структурой микроконтроллера AT2313, организацией портов ввода вывода. Привести адреса регистров настройки и собственно портов ввода/вывода микроконтроллера.
2. Изучить электрические функциональные модуля микроконтроллера и модуля индикации на 8-и светодиодах. Изобразить схему подключения индикатора к модулю микроконтроллера.
3. Изучить текст программы, представленной в разделе 3.3.
4. Изменить исходный текст программы в соответствии с вариантом задания и детально комментировать изменения.

Варианты заданий отличаются длительностями задержки при переходе к следующей комбинации светящихся и погашенных светодиодов и самими комбинациями.

– Величина задержки определяется в соответствии с выражением

$$T_z = 5 \times (30 \times N_1 + 15 \times N_2 + N_3) \text{ мс,}$$

где  $N_1$  – номер группы (1 или 2),  $N_2$  – номер подгруппы (1 или 2),  $N_3$  – порядковый номер в алфавитном списке группы.

– Отображаемые на дисплее комбинации выбираются в соответствии с таблицей 2. Неуказанные разряды должны быть всегда погашены.

5. Для каждой команды привести краткое описание выполняемых действий, формат и состояние флагов. Указать пределы изменения операндов и их конкретные значения.
6. Опираясь на комментарии в тексте программы и описание действия команд, представить программу в виде функциональных

модулей и составить граф-схему алгоритма, выполненную по ГОСТ 19.701-90.

7. Изучить порядок создания проекта средствами ASTUDIO4.
9. Продумать последовательность и способ отладки каждого модуля программы. Привести план отладки.

Таблица 2. Варианты заданий

<b>№ вар.</b>	<b>Вариант индикации</b>
1	Циклическое перемещение одиночного светящегося пятна слева направо
2	Циклическое перемещение одиночного светящегося пятна справа налево
3	Циклическое перемещение светящейся пары справа налево
4	Циклическое перемещение светящейся пары слева направо
5	Циклическое зажигание светодиодов слева направо от одного до восьми одновременно
6	Циклическое зажигание светодиодов справа налево от одного до восьми одновременно
7	Циклический инкремент разрядов 3-7
8	Циклический декремент разрядов 3-7
9	Циклический инкремент разрядов 2-7
10	Циклический декремент разрядов 1-6
11	Циклический инкремент разрядов 2-5
12	Циклический инкремент разрядов 3-7
13	Циклический инкремент разрядов 1-5
14	Циклический инкремент разрядов 5-7
15	Циклический декремент разрядов 4-7

Для самоконтроля письменно ответить на приведенные далее вопросы.

1. Как организовать длинную программную задержку?
2. Как задать режим работы конкретной линии порта ввода/вывода (ввод или вывод)?

3. Какой командой можно последовательно смещать единицу в регистре?
4. Какая команда позволяет принудительно установить некоторые биты регистра в 0?
5. Какие логические уровни соответствуют светящемуся/погашенному состояниям диода?

## 5 Программа исследований и порядок работы

Запустить программу AVR Studio, создать проект и ввести текст программы управления светодиодным индикатором модифицированный при самостоятельной подготовке к лабораторной работе.

1. Компилировать проект и протоколировать сообщения. Если имеются ошибки, привести ошибочные строки в отчёте, сделать анализ ошибок и ввести исправления.
2. Выполнить пошаговую отладку модулей программы в соответствии с методикой, разработанной при самостоятельной подготовке. Протоколировать содержимое модифицируемых регистров при пошаговой процедуре, сопоставляя их с предсказанными значениями.
3. Опираясь на показания счётчика циклов “**Cycle Counter**” в разделе “**PROCESSOR**” окна “**WORKSPACE**” определить время исполнения в тактах одного внутреннего и внешнего цикла задержки. Сопоставить полученные экспериментальные данные с результатами домашних расчетов.
4. По завершении отладки внести в отчёт и изучить листинг программы, находящийся в папке проекта. Сравнить транслированные адреса управления портом и регистра вывода порта с адресами, установленными при самостоятельной подготовке и результат сравнения фиксировать в отчёте.
5. Подключить стенд с помощью программирующего кабеля к параллельному порту компьютера (см. раздел 3.5 Программирование микроконтроллера в составе прототипной системы), затем

включить питание стенда, запустить программу «PonyProg» и настроить программатор.

6. Загрузить HEX-файл откомпилированной программы, скопировать дампы памяти программ из окна программатора и внести его в отчёт. Провести сопоставление с листингом программы. Фиксировать и объяснить расхождения.
7. Выполнить программирование микроконтроллера. По завершении программирования загруженная программа начнёт исполняться автоматически.
8. Описать функционирование стенда, выделив характерные особенности поведения индикатора. Сравнить соответствие исполняемых функций варианту задания и результат внести в отчёт.
9. С помощью секундомера измерить и внести в отчёт длительность цикла индикации. Сделать выводы о соответствии времени исполнения расчётному значению или о причинах расхождения. При необходимости внести коррективы в программу.

### **6 Контрольные вопросы**

1. Напишите команды, позволяющие выборочно установить некоторые линии порта В в режим ввода, а другие – вывода?
2. Почему при инициализации управляющего регистра порта ввода вывода используется временный регистр?
3. Какие команды позволяют изменить состояние линий порта вывода микроконтроллера?
4. В чём смысл представления области памяти данных в двойной интерпретации: в виде отдельных сегментов и общего массива?

## 7 Содержание отчёта

Отчёт должен содержать:

1. титульный лист;
2. схему электрическую функциональную устройства и её краткое описание;
3. текст программы по варианту задания;
4. краткое описание всех команд разработанной программы в соответствии с п.4.3 методических указаний;
5. описание модулей программы и их функций;
6. граф-схему алгоритма;
7. план отладки программы;
8. ответы на вопросы для самопроверки;
9. протокол отладки с анализом ошибок (при наличии);
10. листинг программы с выделением адреса регистра управления вводом/выводом и адреса регистра вывода;
11. дампы памяти программ, скопированный из окна программатора и результаты сравнения его с листингом программы;
12. результаты оценки длительности задержки в среде симулятора (в тактах микроконтроллера);
13. результаты экспериментальной проверки программы на прототипной системе;
14. ответы на контрольные вопросы.

## 8 ОРГАНИЗАЦИЯ ЦИФРОВОГО ВВОДА/ВЫВОДА В СИСТЕМАХ НА МИКРОКОНТРОЛЛЕРАХ AVR

### 1 Цель работы

Изучение организации портов ввода/вывода микроконтроллеров серии AVR и приёмов работы с ними.

### 2 Порты ввода/вывода микроконтроллеров AVR

#### 2.1 Организация портов ввода/вывода

Порты ввода/вывода имеют от 3 до 53 независимых линий "Вход/Выход". Каждый разряд порта может быть запрограммирован на ввод или на вывод информации. Мощные выходные драйверы обеспечивают токовую нагрузочную способность 20 мА на линию порта (втекающий ток) при максимальном значении 40 мА, что позволяет, например, непосредственно подключать к микроконтроллеру светодиоды и биполярные транзисторы. Общая токовая нагрузка на все линии одного порта не должна превышать 80 мА (все значения приведены для напряжения питания 5 В). Для каждого физического вывода существует 3 бита контроля/управления. Упрощенная структурная схема элемента ввода/вывода AVR - микроконтроллера приведена на рисунке 1. Здесь **DDR<sub>x</sub>** - бит контроля направления передачи данных и привязки вывода к шине питания ( $V_{CC}$ ), **PORT<sub>x</sub>** - бит привязки вывода к  $V_{CC}$  и бит выходных данных, **PIN<sub>x</sub>** - бит для отображения логического уровня сигнала на физическом выводе микросхемы.

Архитектура построения портов ввода/вывода AVR с тремя битами контроля/управления позволяет разработчику полностью контролировать процесс ввода/вывода. Если необходимо получить реальное значение сигнала на физическом выводе микроконтроллера - читайте содержимое бита по адресу PIN<sub>x</sub>. Если требуется обновить выходы - прочитайте PORT<sub>x</sub> защелку и потом модифицируйте данные.

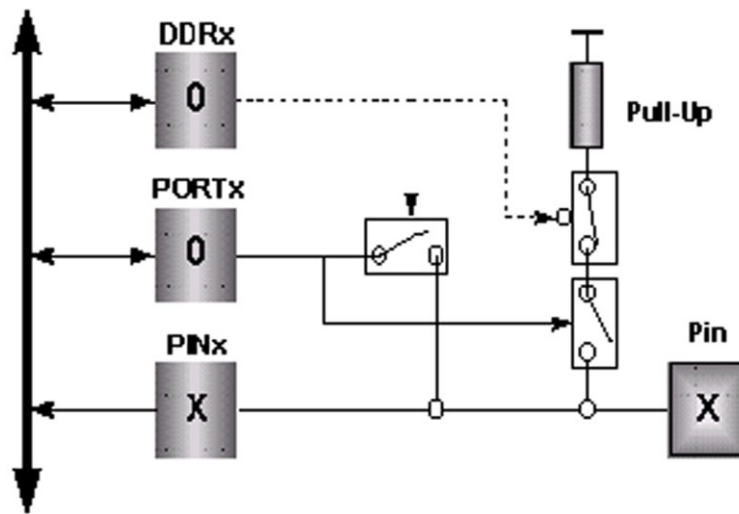


Рисунок 1 – Структура элемента ввода/вывода.

Это позволяет избежать необходимости иметь копию содержимого порта в памяти для безопасности и повышает скорость работы микроконтроллера при работе с внешними устройствами. Особую значимость приобретает данная возможность AVR для реализации систем, работающих в условиях внешних электрических помех.

## 2.1 Адресация портов ввода/вывода микроконтроллера AT90S2313

Порты ввода/вывода и регистры управления ими располагаются в пространстве ввода/вывода. Они доступны через команды IN и OUT, пересылающие данные между одним из 32-х регистров общего назначения и пространством ввода/вывода.

При использовании команд IN, OUT, SBIS и SBIC, должны использоваться адреса из диапазона \$00-\$3F. При доступе к регистру ввода/вывода как к ячейке ОЗУ, к его адресу необходимо добавить \$20. В таблице 1 приведены адреса портов и регистров управления ими [1].



**Таблица 1. Порты ввода/вывода и регистры управления ими.**

\$18(\$38)	PORTB	Data Register, Port B	Регистр данных порта B
\$17(\$37)	DDRB	Data Direction Register Port B	Регистр направления данных порта B
\$16(\$36)	PINB	Input pins, Port B	Выводы порта B
\$12(\$32)	PORTD	Data Register, Port D	Регистр данных порта D
\$11(\$31)	DDRD	Data Direction Register Port D	Регистр направления данных порта D
\$10(\$30)	PIND	Input pins, Port D	Выводы порта D

## 2.2 Работа с портами ввода/вывода

Порт B 8-разрядный двунаправленный порт.

Для обслуживания порта отведено три регистра: регистр данных PORTB, регистр направления данных - DDRB и выводы порта PINB. Адрес выводов порта B предназначен только для чтения, в то время как регистр данных и регистр направления данных - для чтения/записи. Все выводы порта имеют отдельно подключаемые подтягивающие резисторы. Помимо этого, порт может выполнить дополнительные функции.

PINB не является регистром, по этому адресу осуществляется доступ к физическим значениям каждого из выводов порта B. При чтении PORTB, читаются данные из регистра-защелки, при чтении PINB читаются логические значения, присутствующие на выводах порта. После сброса начальные значения соответствуют следующим:

$PORTB = 0x00$ ,

$DDRB = 0x00$ ,

PINB = высокоимпедансное состояние.

Все 8 бит порта B при использовании для ввода/вывода равноправны. Бит DDB<sub>n</sub> регистра DDRB выбирает направление передачи

данных. Если бит установлен (1), вывод сконфигурирован как выход. Если бит сброшен (0) - вывод сконфигурирован как вход. Если  $PORTB_n$  установлен и вывод сконфигурирован как вход, включается КМОП подтягивающий резистор. Для отключения резистора,  $PORTB_n$  должен быть сброшен (0) или вывод должен быть сконфигурирован как выход (таблица 2).

Таблица 2. Влияние  $DDRB_n$  на выводы порта В

<b>DDRB<sub>n</sub></b>	<b>PORTB<sub>n</sub></b>	<b>Вх/Вых</b>	<b>Подт. резист.</b>	<b>Комментарий</b>
0	0	Вход	Нет	Третье состояние (Hi-Z)
0	1	Вход	Да	$PB_n$ источник тока $I_{OL}$ , если извне соединен с землей
1	0	Выход	Нет	Выход установлен в 0
1	1	Выход	Нет	Выход установлен в 1

$n = 7, 6...0$  - номер вывода

### 3 Подготовка к лабораторной работе

При внеаудиторной подготовке к лабораторной работе придерживаться изложенного ниже плана.

1. Изучить организацию портов ввода/вывода и порядок их настройки.
2. Изучить шаблон программы, выполняющей следующие функции:
  - Если ключ “SW0” нажат, наращивать с заданным темпом счётчик и отображать на светодиодном индикаторе;
  - Если ключ “SW1” нажат, уменьшать с заданным темпом счётчик и отображать на светодиодном индикаторе;
  - Если не нажат ни один из ключей выключить светодиодный индикатор.

```

;***** Цифровой ввод/вывод
.include "2313def.inc"
.device AT90S2313

.def Temp    =r16      ;temporary register
.def Test =r19        ;temporary test register
.def Delay   =r17      ;Delay Variable
.def Delay2=r18       ;Delay
.def Delay3=r19       ;Delay

;***** Инициализация
RESET:
    ser    Temp          ;Temp = $FF
    <AVR Instruction> ;PORTB = все на вывод
    <AVR Instruction> ;включить светодиоды
    <AVR Instruction> ;включить подтягивающие
                        ;резисторы порта D
;**** Контроль ключей SW0 и SW1
loop:    sbis PIND,0     ;SW0 нажат?
        dec    Temp      ;декрементировать Temp
    <AVR Instruction> ;SW1 нажат?
    <AVR Instruction> ;инкрементировать Temp
    <AVR Instruction> ;Читать входы порта D
    <AVR Instruction> ;Контроль ключей
    <AVR Instruction> ;SW0 или SW1 нажат?
    ser    Temp          ;SW0 и SW1 разомкнуты –
                        ;светодиоды выключить

outled:
    out    PORTB,Temp    ;отобразить на индикаторе
**** задержка для визуализации результата
DLY:
<AVR Instruction> ;команды задержки
    rjmp  loop          ;продолжить после задержки

```

3. Используя шаблон программы вставить необходимые инструкции микроконтроллера AVR (описание инструкций приведено в [2]).
4. Представить программу в виде функциональных модулей с кратким описанием функции каждого модуля в текстовой форме.
5. Опираясь на представление программы в виде функциональных модулей и текст программы составить граф-схему алгоритма.
6. Продумать последовательность и способ отладки каждого модуля программы. Привести план отладки.
7. Изобразить схему электрическую функциональную аппаратных средств необходимых для выполнения данной лабораторной работы. Ключи должны быть присоединены к разрядам D2 и D3 порта D. Светодиодные индикаторы подключаются к порту В.

Для самоконтроля письменно ответить на приведенные далее вопросы.

1. В каком режиме должны работать разряды D2 и D3 порта D?
2. С какой целью для этих разрядов подключаются внутренние подтягивающие резисторы?
3. Как программно отличить состояние ключей?
4. Какие уровни соответствуют каждому из состояний?
5. Чем определяются уровни напряжений, попадающие на входы порта D0, D1?
6. С какой целью последовательно со светодиодами включают резисторы?
7. Какой логический уровень в соответствии со схемой подключения светодиодов обеспечивает их свечение.

#### **4 Программа исследований и порядок работы**

1. Создать проект и ввести текст программы, созданной при самостоятельной подготовке к лабораторной работе.

2. Компилировать проект и протоколировать сообщения. Если имеются ошибки, привести ошибочные строки в отчёте, сделать анализ ошибок и ввести исправления.
3. Выполнить отладку программы в **среде симулятора** в соответствии с методикой разработанной при самостоятельной подготовке. В процессе отладки протоколировать содержимое модифицируемых регистров, сопоставляя с предсказанными значениями.
4. Подключить стенд с помощью программирующего кабеля к параллельному порту компьютера и выполнить программирование микроконтроллера.
5. По завершении программирования загруженная программа начнёт исполняться автоматически. Нажимая поочерёдно клавиши, запротоколировать реакцию прототипного устройства.

## 5 Методические указания

При симуляции взаимодействия ключей клавиатуры (любых устройств, подключаемых к портам ввода) и микроконтроллера программист должен самостоятельно устанавливать значение логической переменной на соответствующем контакте порта D (PIND). Для этого необходимо выполнить последовательность действий, представленную ниже.

- Запустить отладчик.
- Открыть окно рабочей области отладчика, выделив пункт View/Workspace главного меню.
- Развернуть в окне **Workspace** пункт **I/O AT90S2313**.
- Перед выполнением в пошаговом режиме команды in Test,PIND

следует установить значения актуальных битов щелчком на их изображении в окне. Выделенная чёрным цветом прямоугольная область соответствует логической единице, невыделенная – нулю.

Симуляция внешних устройств не поддерживается. Это означает, что подключение подтягивающих резисторов не приводит к автоматической установке всех входов в состояние логической единицы.

## **6 Контрольные вопросы**

1. К каким последствиям в работе программы приведёт отсутствие модуля инициализации?
2. Что изменится в работе программы, если при выборе режима ввода для порта D программно не подключить подтягивающие резисторы?
3. Почему в программе не инициализирован режим ввода порта D?
4. Как обнаружить ошибку, связанную с неподключением подтягивающих резисторов порта D?

## **7 Содержание отчёта**

Отчёт должен содержать:

1. титульный лист;
2. наименование работы и цель исследований;
3. схему электрическую функциональную устройства и её краткое описание;
4. исходный текст программы;
5. описание модулей программы и их функций;
6. граф-схему алгоритма;
7. план отладки программы;
8. ответы на вопросы для самопроверки;
9. текст программы в соответствии с вариантом задания;
10. протокол отладки с анализом ошибок (при наличии);
11. листинг программы;
12. дампы программной памяти микроконтроллера;
13. результаты экспериментальной проверки программы на прототипной системе;
14. ответы на контрольные вопросы.

### **3 ПОДПРОГРАММЫ И СТЕК**

#### **1 Цель работы**

Изучение особенностей организации и использования стека и механизма подпрограмм в семействе микроконтроллеров AVR.

#### **2 Организация стека в микроконтроллерах AVR**

Как известно, стековый механизм организации работы с памятью существенно повышает скорость обмена данными между памятью и регистрами за счёт применения метода косвенно-регистровой адресации и автоинкрементного, автодекрементного способа модификации адреса. Память стека может быть изолированной (предназначенной исключительно для сохранения данных и адресов возврата в основную программу) или представлять собой часть общего поля памяти данных. Кроме того, стек может размещаться во внешней или внутренней памяти микроконтроллеров.

Важнейшей составной частью стекового механизма памяти является регистр косвенной адресации, именуемый указателем стека. В общем случае для микроконтроллеров AVR указатель представляет собой 16-битовый регистр. Он состоит из двух 8-битовых регистров – младшего SPL и старшего SPH. При этом объём адресуемой памяти может составлять до 64 Кбайт. В младших моделях микроконтроллеров с объёмом памяти данных не превышающим 256 байт присутствует лишь однобайтовый регистр SPL.

Указатель стека процессора AT90S2313 8-разрядный регистр с адресом \$3D (\$5D). 8-ми разрядов достаточно для адресации ОЗУ в пределах \$60-\$DF. Его содержимое доступно для чтения и записи. Указатель стека указывает на область памяти (на конкретную ячейку, именуемую вершиной стека), в которой расположен стек вызова подпрограмм и прерываний. Область стека в ОЗУ должна быть задана до того, как произойдет любой вызов подпрограммы или будут разрешены прерывания.

Указатель стека уменьшается на 1 при записи данных в стек командой PUSH и уменьшается на 2 при вызове подпрограммы командой CALL или обработке прерывания. Указатель стека увеличивается на 1 при выборе данных из стека командой POP и увеличивается на 2 при выполнении команд возврата из подпрограммы или обработчика прерывания (RET или RETI). Такой подход к управлению содержимым указателя стека принят в большинстве микроконтроллеров.

Основной проблемой при использовании стека является перекрытие области данных (программ) и стека. При этом могут искажаться и данные и содержимое стека. Следствием являются ошибки при обработке данных или полная неработоспособность системы. Последняя возникает при затирании адресов возврата из подпрограмм или обработчиков прерываний. Для корректной работы стека нужно выполнить несколько условий.

1. При инициализации указателя стека вершину стека задают так, чтобы заполняющийся и освобождающийся стек ни при каких обстоятельствах не перекрывался с областью памяти содержащей данные (или программу в микроконтроллерах принстонского класса). Одним из разумных подходов к этой проблеме состоит в заполнении памяти данными с младших адресов и организации стека в старших адресах. В этом случае при инициализации в указатель стека заносится адрес последней ячейки оперативной памяти.
2. Размер стека не должен превышать объема свободной оперативной памяти. Размер определяется количеством сохраняемых в стеке данных при работе программ и так называемой **вложенностью** подпрограмм и прерываний. Вложенность – это использование подпрограмм внутри подпрограмм или обработка прерывания во время работы другого обработчика прерывания. По возможности следует запрещать вложенные прерывания или ограничиваться двумя уровнями вложенности, разрешая прерывания



внутри прерываний лишь для одного – критического события. Проблема вложенности может стоять очень остро для систем с событийным механизмом обслуживания (по прерываниям) на основе микроконтроллеров с малым объёмом оперативной памяти.

3. Причиной безудержного роста стека могут быть ошибка программиста, состоящая в нарушении правила **парности стековых операций**. В соответствии с ним каждому вызову подпрограммы должен соответствовать возврат из неё, каждому прерыванию – возврат из прерывания, каждому оператору записи в стек – оператор чтения стека.

### 3 Подготовка к лабораторной работе

При внеаудиторной подготовке к лабораторной работе придерживаться изложенного ниже плана.

1. Изучить и описать формат и действия, выполняемые командами CALL, RET, PUSH, POP.
2. Выяснить и записать разрядность и адреса указателя стека микроконтроллера AT90S2313 в шестнадцатеричной системе счисления.
3. Уточнить и записать объём, адрес первой и последней ячейки оперативной памяти микроконтроллера в шестнадцатеричной системе счисления.
4. Изучить шаблон программы для устройства, изученного в лабораторной работе №2, выполняющей следующие функции:
  - Если ключ “SW0” нажат, отображать на светодиодном индикаторе пятно, бегущее слева – направо;
  - Если ключ “SW1” нажат, отображать на светодиодном индикаторе пятно бегущее справа – налево;
  - Если не нажат ни один из ключей, не изменять изображение.

```

;***** Применение подпрограмм
;по нажатию SW1 (PortD3)- бегущий огонь влево, SW0(D2)-вправо
.include "2313def.inc"
.device AT90S2313; Fтакт = 10МГц
.def Temp =r16 ;временный регистр

```

```

.def  Test  =r20      ;временный регистр контроля
.def  Delay=r17      ;младший регистр задержки
.def  Delay2  =r18   ;второй регистр задержки
.def  Delay3  =r19   ;старший регистр задержки

;***** Инициализация
RESET:
<AVR Instruction>    ;установить вершину
<AVR Instruction>    ;стека

ser   Temp          ;Temp = $FF
out   DDRB,Temp     ;PORTB = все на вывод
out   PORTB,Temp    ;погасить светодиоды
out   PORTD,Temp    ;подтягив. резисторы PORTD
                        ;в регистре бегущих огней
ldi   Temp,1        ;подготовит LED0 к зажиганию

;**** чтение порта клавиатуры и формирование кода ключа
in_kbd:
in    Test,PIND     ;Читать состояние клавиатуры
andi  Test,12       ;наложение маски <0000 1100>

;**** анализ кодов ключей и подготовка к индикации
L_Test:
cpi   Test,4        ;ключ SW1 нажат?
brne  L2_Test       ;нет - проверять дальше

lsl   Temp          ;бегущий огонь влево
brne  LE_Test       ;если обнулится,
ldi   Temp,1        ;переместить 1 в младший разряд
rjmp  LE_Test       ;и завершить подготовку

L2_Test:
cpi   Test,8        ;ключ SW0 нажат?)
brne  LE_Test       ;нет - завершить подготовку

<AVR Instruction>    ;бегущий огонь вправо
<AVR Instruction>    ;не нуль завершить, иначе
<AVR Instruction>    ;переместить 1 в старший разряд

```

LE\_Test: ;подготовка к индикации завершена

;\*\*\*\* отобразить бегущий огонь (с учётом 1-горит,0-погашен)

outled:

*<AVR Instruction>* ;Отобразить на индик. регистр Temp

ldi Delay,255 ; задать задержку

ldi Delay2,255 ;

ldi Delay3,2 ;

rcall DLY ;вызвать подпрограмму

rjmp in\_kbd ;повторять бесконечно

;\*\*\*\* подпрограммы

;\*\*\*\* задержка для визуализации результата

DLY:

subi Delay,1 ;уменьшить на единицу

sbc\_i Delay2,0 ;24-х битовый счётчик

sbc\_i Delay3,0 ;задержки и если он не обнулится

brcc DLY ;повторять до обнуления, а затем

ret ;завершить подпрограмму задержки

5. Используя шаблон программы вставить необходимые инструкции микроконтроллера AVR (описание инструкций приведено в [2]).
6. Представить программу в виде функциональных модулей с кратким описанием функции каждого модуля в текстовой форме.
7. Изучить используемый в программе модуль формирования задержки и детально (покомандно) описать его.
8. Опираясь на представление программы в виде функциональных модулей и текст программы составить граф-схему алгоритма.
9. Продумать последовательность и способ отладки каждого модуля программы. Привести план отладки.

Для самоконтроля письменно ответить на приведенные далее вопросы.

1. В каком режиме работает порт D – ввода или вывода?
2. Как можно загрузить указатель стека? Написать команды загрузки.
3. Каков адрес последней ячейки оперативной памяти контроллера AT90S2313?
4. В чём смысл наложения маски на код, считанный с порта в модуле `scan_cod: (andi Test,12)`?
5. Насколько, и в какую сторону изменится указатель стека при выполнении команды `rcall`?
6. Как рассчитать величину задержки в подпрограмме задержки?

#### **4 Программа исследований и порядок работы**

1. Рассчитать шестнадцатеричный код задержки для величины задержки  $T = 300 + 50 \times n$  мс, где  $n$  – номер компьютера.
2. Создать проект и ввести текст программы, созданной при самостоятельной подготовке к лабораторной работе.
3. Содержимое регистров задержки установить в соответствии с расчётами.
4. Компилировать проект и протоколировать сообщения. Если имеются ошибки, привести ошибочные строки в отчёте, сделать анализ ошибок и ввести исправления.
5. Выполнить отладку модулей программы в соответствии с методикой разработанной при самостоятельной подготовке. В процессе отладки протоколировать содержимое модифицируемых регистров, сопоставляя с предсказанными значениями.
6. После сброса в пошаговом режиме наблюдать и протоколировать изменение содержимого стековой области и указателя стека. Для сокращения количества шагов задержку установить равной 1.

7. Дополнить подпрограмму задержки двумя командами сохранения содержимого произвольных регистров в стеке. Предпринять попытку выполнения нескольких циклов вывода на индикацию. Протоколировать содержимое указателя стека, стековой области и последовательности выполняемых операторов. Объяснить результаты наблюдения (задержка должна быть равна 1).
8. Восстановить программу и установить величину задержки в соответствии с вариантом. Провести оценку реальной задержки.
9. Подключить стенд с помощью программирующего кабеля к параллельному порту компьютера и выполнить программирование микроконтроллера.
10. По завершении программирования загруженная программа начнёт исполняться автоматически. Нажимая поочерёдно ключи, запротоколировать реакцию прототипного устройства.

## **5 Контрольные вопросы**

1. К каким последствиям в работе программы приведёт отсутствие команд инициализации стека?
2. Если вершина стека указывает на младшую ячейку ОЗУ, что произойдёт при двух последующих загрузках стека?

## **6 Содержание отчёта**

Отчёт должен содержать:

1. титульный лист;
2. наименование работы и цель исследований;
3. исходный текст программы;
4. описание модулей программы и их функций;
5. граф-схему алгоритма;
6. план отладки программы;
7. ответы на вопросы для самопроверки;
8. текст программы в соответствии с вариантом задания;

9. протокол отладки с анализом ошибок (при наличии);
10. результаты исследований и анализа работы стека;
11. дампы памяти программ;
12. результаты экспериментальной проверки программы на прототипной системе;
13. ответы на контрольные вопросы.

## Литература

1. Батоврин, В.К. LabVIEW: Практикум по цифровым элементам вычислительной и информационно-измерительной техники: Лабораторный практикум / В.К. Батоврин, А.С. Бессонов, В.В. Мошкин. - М.: Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования “Московский государственный технический университет радиотехники, электроники и автоматики”, 2011. - 118 с.
2. Угрюмов, Е. П. Цифровая схемотехника [Текст] : учебное пособие / Е. П. Угрюмов. – 2-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2005. - 800 с.
3. Бондарь, О.Г. Учебно-методический комплекс «Схемотехника». [Электронный документ]: Справочники: Справочник по светоизлучающим полупроводниковым приборам. – Курск. ЮЗГУ, Сервер кафедры КП и СС //main/sensey/b/start\_00.
4. Опадчий, Ю.Ф. Аналоговая и цифровая электроника [Текст]: учебник для вузов / Ю.Ф. Опадчий, О.П. Глудкин, А.И. Гу-ров; под ред. О. П. Глудкина. - М.: Горячая Линия - Телеком, 2000. - 768 с.
5. Голубцов, М.С. Микроконтроллеры AVR: от простого к сложному [Текст] /М. С. Голубцов. – М.: СОЛОН-Пресс, 2003. - 288 с.
6. Евстифеев, А.В. Микроконтроллеры AVR-семейств Tiny и Mega фирмы “ATMEL” [Текст] / А.В. Евстифеев. -М.: Издательский дом “Додэка-XXI”, 2004. - 560 с.
7. 0839I–AVR–06/02. [Электронный ресурс] : AT90S2313.pdf, Atmel Corporation, 2002.
8. AVR Instruction Set. [Электронный ресурс] : Doc0856.pdf, Atmel Corporation, 1999.