

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Емельянов Сергей Геннадьевич

Должность: ректор

Дата подписания: 16.12.2021 21:00:49

Уникальный программный ключ:

9ba7d3e34c012eba476ffd2d064cf2781953be730df2374d16f3c0ee536f0fc6

МИНОБРНАУКИ РОССИИ

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

Кафедра биомедицинской инженерии

УТВЕРЖДАЮ

Проректор по учебной работе

О.А. Тактиконова

« 1 » _____ (ЮЗГУ) 2018 г.



АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ РАСЧЕТА И ПРОЕКТИРОВАНИЯ ЭЛЕМЕНТОВ И УЗЛОВ БИОТЕХНИЧЕСКИХ СИСТЕМ

Методические рекомендации по организации и выполнению
практических занятий для обучающихся по программам высшего
образования по направлениям 12.04.04, 12.06.01 и 09.06.01

УДК 004.93:61

Составители: А.А.Кузьмин, С.А.Филист.

Рецензент

Доктор технических наук, профессор Р.А. Томакова

Автоматизированные системы расчета и проектирования элементов и узлов биотехнических систем: методические рекомендации по организации и выполнению практических занятий / Юго-Зап. гос. ун-т; сост.: А.А. Кузьмин , С.А.Филист. - Курск, 2018. - 44 с.

Методические указания соответствуют требованиям программ дисциплин «Автоматизированные системы расчета и проектирования элементов и узлов биотехнических систем».

Предназначены для обучающихся по программам высшего образования по направлениям 12.04.04 «Биотехнические системы и технологии», 12.06.01 «Фотоника, приборостроение, оптические и биотехнические системы и технологии (Приборы, системы и изделия медицинского назначения)» и 09.06.01 «Информатика и вычислительная техника (Системный анализ, управление и обработка информации (технические и медицинские системы))».

Текст печатается в авторской редакции

Подписано в печать 1.03.18. Формат 60x84 1/16.

Усл.печ.л. 214. Уч.-изд.л. 212 Тираж 100 экз. Заказ: 1458. Бесплатно.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

Практическая работа №1

Автоматизированное проектирование приборов и систем с использованием САПР PROTEUS

1 Краткие теоретические сведения

Система Proteus предназначена для моделирования аналоговых и цифровых электронных схем, в том числе широкой номенклатуры микроконтроллеров. Среди прочих САПР, которые тоже позволяют моделировать электронные схемы, Proteus выгодно отличаются мощные возможности отладки программ для микроконтроллеров, а также интерактивного симулирования схем в реальном времени с вводом-выводом информации на реальные физические порты компьютера (COM, USB).

Proteus состоит из двух основных частей:

ISIS – средство для построения принципиальных схем, схемного моделирования и отладки программ микроконтроллеров.

ARES (Advanced Routing and Editing Software) – модуль для проектирования печатных плат.

Рассмотрим основные приемы работы с модулем ISIS. Запустите ISIS. Вид ISIS версии 7 показан на рисунке 1.

Подробнее опишем элементы интерфейса (рисунок 1):

- 1 - создается новый проект с установками по умолчанию;
- 2 – загрузка и запись проекта;
- 3, 4 – загрузка из файла и запись в файл выделенного блока (секции). Операция по действию аналогична кнопке 18;
- 5 – вывод проекта на печатающее устройство (принтер);
- 6 – выделяет область, выводимую на принтер;
- 7 – перерисовка экрана;
- 8 – включает/выключает сетку;
- 9 – установка относительного центра координат. Позволяет установить новый центр координат, после чего в панели 36 будут выводиться новые относительные смещения курсора;
- 10 – установка центра дисплея над позицией курсора;
- 11 – кнопки изменения масштаба. Аналогичная функция присвоена колесу мыши.
- 12 – масштаб изображения – весь лист проекта;
- 13 - масштаб изображения – выделенная область;
- 14 – кнопки отмены/возврата изменений;

- 15 – вырезать выделенные элементы в буфер обмена;
- 16 – скопировать выделенные элементы в буфер обмена;
- 17 – вставить элементы из буфера обмена;
- 18 – скопировать выделенный блок;
- 19 - перенести выделенный блок (доступно также из контекстного меню при выделении объекта – команда Drag Object);

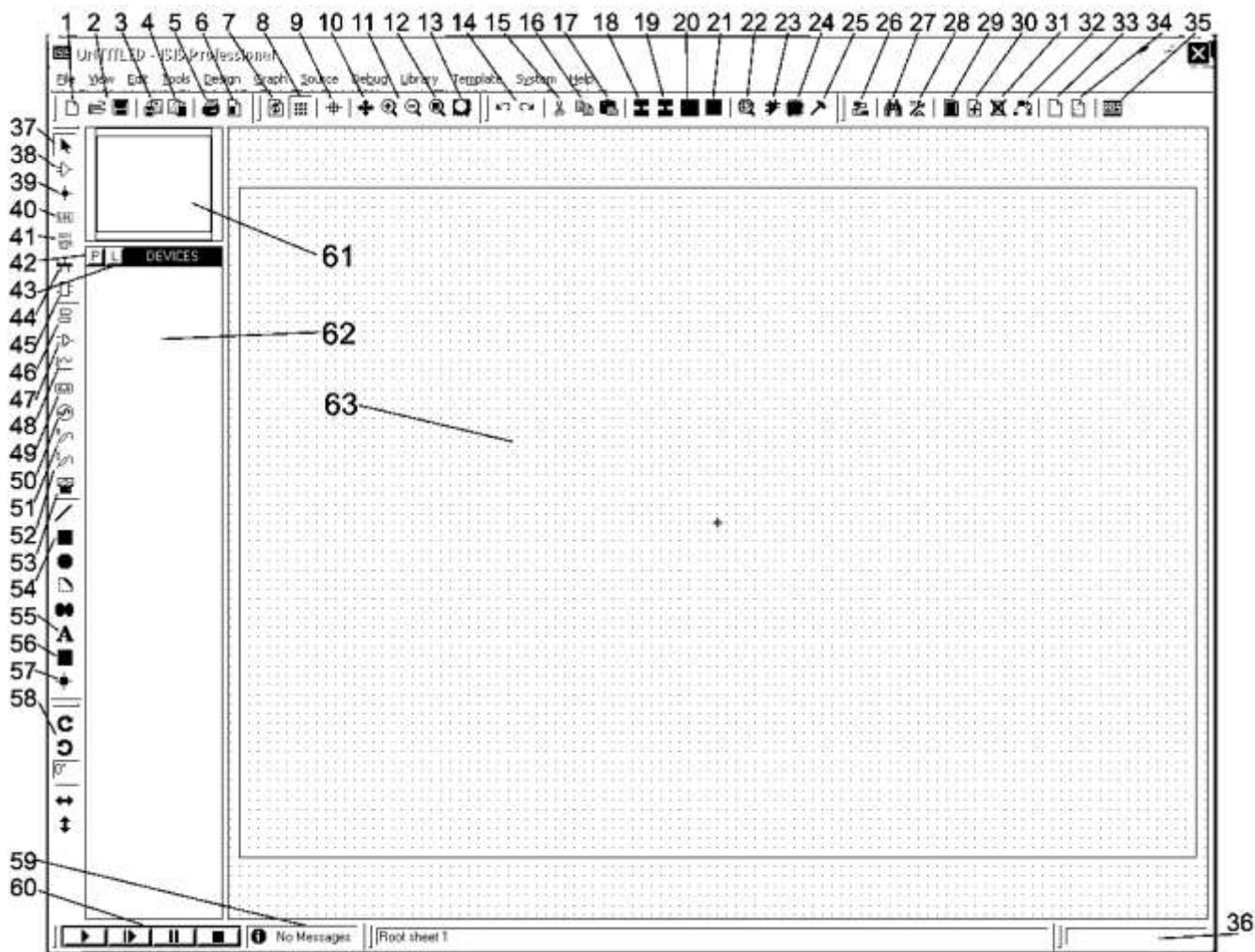


Рисунок 1 - Вид окна ISIS при запуске

- 20 – повернуть выделенный блок (некоторые варианты поворота доступны также из контекстного меню при выделении объекта);
- 21 - удалить выделенный блок (доступно также из контекстного меню при выделении объекта – команда Delete Object, или повторное нажатие правой клавиши мышки над выделенным объектом);
- 22 – выбор схематического элемента, компонента, устройства. Аналог кнопке 42;
- 23 – вызывает диалог (мастер) объединения выделенной схемы в отдельный компонент и сохранения компонента в библиотеку.

24 - вызывает диалог установки корпуса компонента;
25 – разбивает выделенный компонент на примитивы;
26 – включает/выключает автороутер для проводников (wire autorouter). При включенном автороутере проводники прокладываются автоматически под прямыми углами. Для кратковременного отключения автороутера при прокладывании проводника (например, для диагонального соединения двух точек) нажмите клавишу CTRL;

27 – поиск компонентов, свойства которых удовлетворяют заданному критерию;

28 – вызов диалога присваивания значений свойствам компонентов;

29 – представление проекта в табличном виде для исследования списка элементов, соединений и навигации по ним.

30 – создание нового листа проекта. Перемещение между листами осуществляется клавишами PageUp и PageDown.

31 - удаление листа проекта;

32 – возврат к родительскому листу проекта;

33 – создание отчета-списка используемых материалов (Bill Of Materials);

34 – проверка правил электрических соединений;

35 – создание списка соединений и запуск ARES для дальнейшего проектирования печатной платы;

36 – панель вывода относительных координат;

37 – режим выбора объектов;

38 – режим размещения компонентов и проводников;

39 – установка соединений для х-образных пересечений проводов. По умолчанию соединяющимися считаются только т-образные пересечения проводов.

40 – установка метки на провод. Провода с одинаковыми метками считаются соединенными.

41 – ввод текстовых скриптов;

42 – выбор схемотехнического элемента, компонента, устройства. Аналог кнопке 22;

43 – менеджер подключаемых библиотек;

44 – ввод шин;

45 – размещение линий ввода-вывода составных схем (subcircuit);

46 – размещение внутрисхемных соединителей (terminals), в том числе и таких как земля, питание и др.

47 – размещение выводов компонента;

- 48 – размещение аналитических графиков (переходных процессов, частотного анализа, цифровых диаграмм и т.п.);
- 49 – установка «магнитофона» (tape recorder);
- 50 – размещение генераторов сигналов (постоянного тока, переменного, импульсного, из звукового файла и т.п.);
- 51 – установка пробника напряжения;
- 52 – установка пробника тока;
- 53 – установка интерактивных инструментов (осциллографа, генератора сигналов, логического анализатора, виртуального терминала и т.п.);
- 54 – размещение двумерных графических примитивов (линии, прямоугольника, окружности, дуги, многоугольника);
- 55 – размещение произвольного текста;
- 56 – размещение графического символа из библиотеки;
- 57 – установка различных маркеров при проектировании элементов;
- 58 – элементы вращения и отражения объекта, подлежащего размещению;
- 59 – информационная панель предупреждений и ошибок;
- 60 – панель кнопок запуска/отладки/остановки интерактивной симуляции.
- 61 – предварительный просмотр макета страницы или компонента;
- 62 – список используемых компонентов;
- 63 – рабочая область.

Пусть в системе Proteus требуется смоделировать гирлянду из восьми светодиодов.

Для управления светодиодами будем применять микроконтроллер PIC16F877. Это достаточно функциональный микроконтроллер среднего семейства фирмы Microchip. Все последующие устройства и примеры программ, а также лабораторные макеты будут использовать именно этот микроконтроллер.

Составим необходимый список элементов. Для этого щелкаем на кнопку 42. Получаем диалог поиска элемента (рисунок 2).

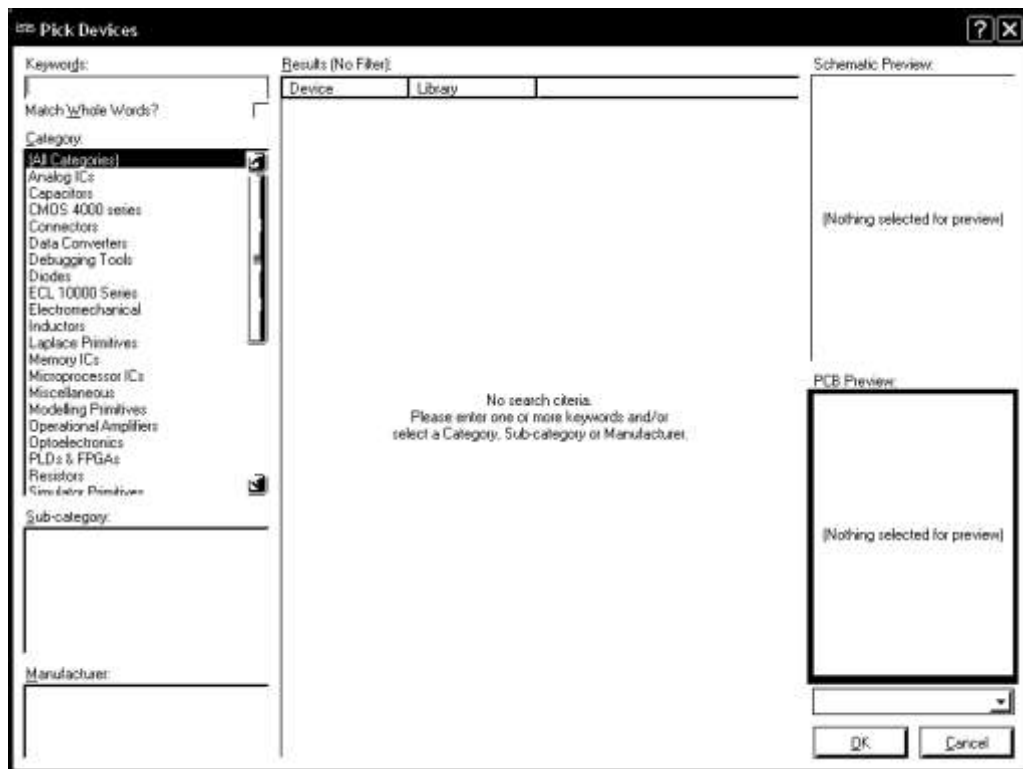


Рисунок 2 – Диалог поиска элемента

В этом диалоге поле «Keywords» необходимо для ввода ключа поиска, список Category позволяет выбрать определенную группу (категорию) компонентов, Manufacturer – производителя. В списке Results отображаются результаты поиска, в картинке Schematic Preview – условное графическое отображение компонента, а также информация о модели компонента или об ее отсутствии, PCB Preview – вид корпуса и контактных площадок.

В поле Keywords вводим ключевые слова для поиска компонента. Для микроконтроллера ключом будет текст «PIC16F877», для светодиода «led-red», для ограничительных резисторов «res». Двойным щелчком по необходимому элементу списка Results добавляем компоненты в список используемых элементов, который отображается в поле 62.

Закрываем диалог поиска элемента и, выбирая необходимые компоненты из сформированного списка используемых элементов, перемещаем их на рабочую область, примерно как показано на рисунке 3. При этом желательно включить режим автоматического назначения позиционных обозначений (пункт меню Tools\Real Time Annotation должен быть включен), иначе позиционные обозначения придется расставлять вручную. Символ схемной земли (GROUND) берется из списка Terminals (кнопка 46).

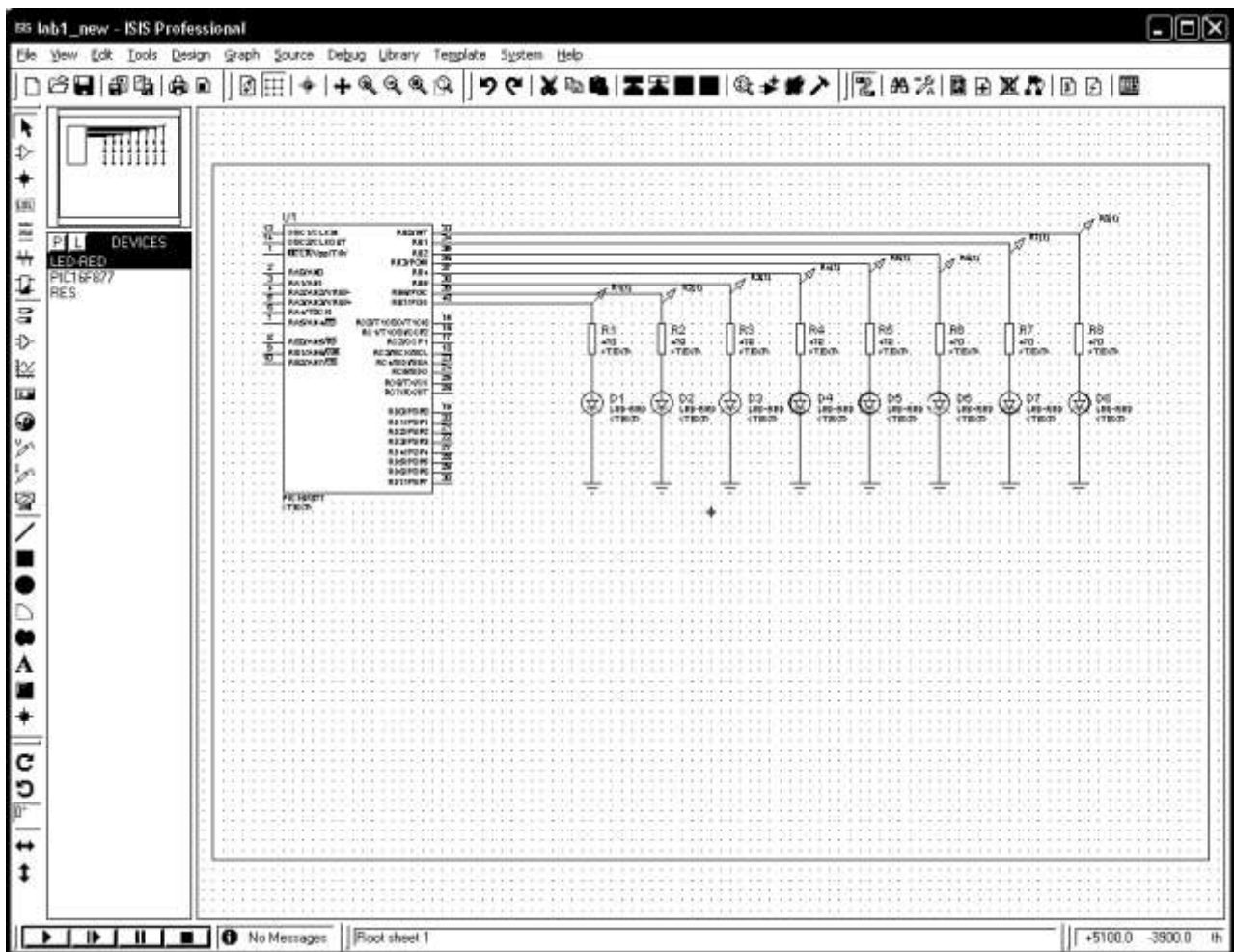


Рисунок 3 – Схема гирлянды в Proteus

Соединяем между собой элементы проводниками (режим проведения проводника включается по умолчанию – курсор в форме карандаша, а при наведении курсора мыши на вывод элемента карандаш окрашивается в зеленый цвет). При этом если имеются два вывода элементов, то между ними обязательно нужно проложить проводник, чтобы они считались соединенными. То есть нельзя соединить, допустим, светодиод и резистор простым перемещением одного из элементов и визуальным совмещением выводов этих элементов. Необходимо отдельно в рабочую область поместить светодиод, отдельно резистор и соединить их выводы проводом.

Затем можно заменить номиналы по умолчанию всех резисторов (10к) на значение 470, или даже еще меньше, иначе яркость свечения светодиодов будет неудовлетворительной. Для этого двойным щелчком по надписи «10к» рядом с резистором открываем диалог редактирования свойства и заменяем номинал резистора на 470 (Ом). Указание единиц без суффиксов устанавливает значение параметра, выраженное в единицах С, т.е. в вольтах, амперах, омах, фарадах. Если

необходимо установить значение в кило- или мега- единицах, то после цифрового значения указывают суффикс «к» или “М”. Если необходимо установить значение в мили-, или микро-, или нано-, или пико-единицах, то после цифрового значения указывают соответственно суффикс «m», «u», «n», «p». Обратите внимание на различие в значении строчного или заглавного написания суффикса в мили- и мега-единицах!

Отредактировать все свойства элемента можно путем выбора из контекстного меню пункта Edit Properties. Или выделить элемент щелчком мыши и нажать Ctrl+E.

Далее необходимо задать программу микроконтроллеру. Обычно исходный текст программы для микроконтроллера пишется или на языке ассемблера, или на языке C. Затем необходимый компилятор транслирует программу в машинные коды и записывает результат в файл с расширением hex.

Создадим в текстовом файле Lab1.asm следующую программу на языке ассемблера:

```
;Программа лабораторной работы №1
;Операции ввода-вывода микроконтроллеров
PIC16F877
;Назначение - включение/выключение светодиодов
;Схема включения - светодиоды на PORTB
;Автор программы =Кузьмин А.А.

list p=16f877
#include p16f877.inc
;-----
---
;Определение переменных
temp EQU 0x21 ;Регистр быстрой задержки
cnt1 EQU 0x22 ;Регистр1 длинной задержки
cnt2 EQU 0x23 ;Регистр2 длинной задержки
cnt3 EQU 0x24 ;Регистр3 длинной задержки
;-----
---
org 0x00
nop ;Первая операция - nop
```

```

; для внутрисхемных отладчиков
goto start ;"Стандартное" начало
; (для программ без прерываний
org 0x8 ;можно и пропустить этот оператор)

start
;-----Инициализация-----
---
bsf STATUS,RP0 ;Регистр TRISB не в
;нулевой странице!
movlw 0x00 ;все выходы PORTB как выходы
movwf TRISB
bcf STATUS,RP0 ;Возвращаемся к нулевой странице

;-----Главный цикл-----
-

loop
movlw 0x55 ;01010101 в PORTB
movwf PORTB
call delay ;задержка

movlw 0xaa ;10101010 в PORTB
movwf PORTB
call delay ;задержка

goto loop
;-----
---
;-----Подпрограммы-----
---
;-----
--

;Подпрограмма задержки
;Организуется программная задержка
;в один цикл loordelay2
;Используемые регистры: temp

delay2
movlw 0xff

```

```

    movwf temp
loopdelay2
    decfsz temp,f
    goto loopdelay2
    return
;-----
---
;Подпрограмма задержки. Версия 2
;Задержка для этой подпрограммы задается
;перед обращением
; к ней как загрузка числа в регистр cnt1.
;Общее число циклов определяется по формуле:
; 256*256*cnt1*7.
;Для примера, приблизительно 1 секунда
;при частоте 20 МГц задается загрузкой 11 в
cnt1.

;Название: delay
;Входные данные: число, пропорциональное
задержке
;в регистре cnt1.
;Выходные: задержка.
;Используемые регистры: cnt1, cnt2, cnt3.

; cnt1 = Cycles/256/256/7 =
;(20000000/4)/256/256/7 = 10.899 = 11

delay          movlw .11
               movwf cnt1
               clrf cnt2
               clrf cnt3
dloop         decfsz cnt3,f
               goto $+2
               decfsz cnt2,f
               goto $+2
               decfsz cnt1,f
               goto dloop
               return

End

```

Несколько заметок к этому листингу. Строки, начиная от символа «;» и до конца строки являются комментариями. На работоспособность контроллера они не влияют, но комментарии очень рекомендуются для описания ключевых элементов программ, таких как название, назначение, авторство и т.д., как целой программы, так и отдельных подпрограмм или модулей программы.

Необходимо внимательно вводить пробелы в командах, например в директиве «list p=16f877» между символами t и p пробел обязателен. Также необходимо отличать буквы «O» от символа нуля «0», например, в обозначении бита «RP0» в конце стоит именно символ нуля.

Сохраним проект (кнопка 2 Save Design). В пути проекта не рекомендуется использовать нелатинские символы и пробелы. Не рекомендуется сохранять проект на рабочий стол-это может повлиять на работоспособность симулятора проекта. Поместим файл Lab1.asm в тот же каталог, куда мы сохранили проект. Далее для формирования hex-файла кликнем на пункт меню Source\Add Remove Source files. Вызывается диалог редактирования свойств исходных программных файлов (рисунок 4).

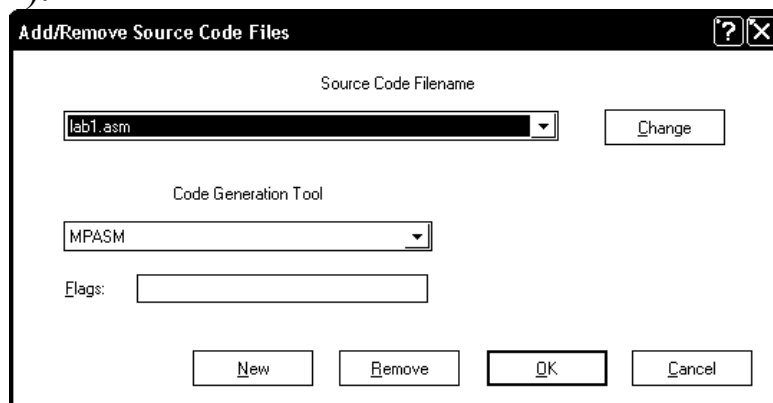


Рисунок 4 - Диалог редактирования свойств исходных программных файлов

В поле Code Generation Tool выбираем ассемблер фирмы Microchip – MPASMWIN, нажимаем кнопку New. В открывшемся диалоге выбора файлов выбираем файл Lab1.asm. Теперь при каждом запуске симуляции исходная программа на ассемблере будет компилироваться выбранным компилятором MPASM в файл Lab1.hex.

Полученный hex файл можно записывать непосредственно в микросхему микроконтроллера с помощью программатора или внутрисхемного отладчика. Для прошивки микроконтроллера нашего проекта необходимо в свойствах компонента PIC16F877 (диалог редактирования свойств, как отмечалось выше, вызывается выбором

мышкой компонента и нажатием на Ctrl+E) в поле Program File задать Lab1.hex (рисунок 5). Тут же задаем частоту работы микроконтроллера Processor Clock Frequency в 20 МГц (именно на такую частоту рассчитана секундная задержка в подпрограмме delay). К микроконтроллеру можно присоединить частотоподающую цепи, такие как кварц, RC-цепочку, генераторы и т.д., однако модель по соображениям эффективности работы будет ориентироваться только на частоту, введенную в этом диалоге.

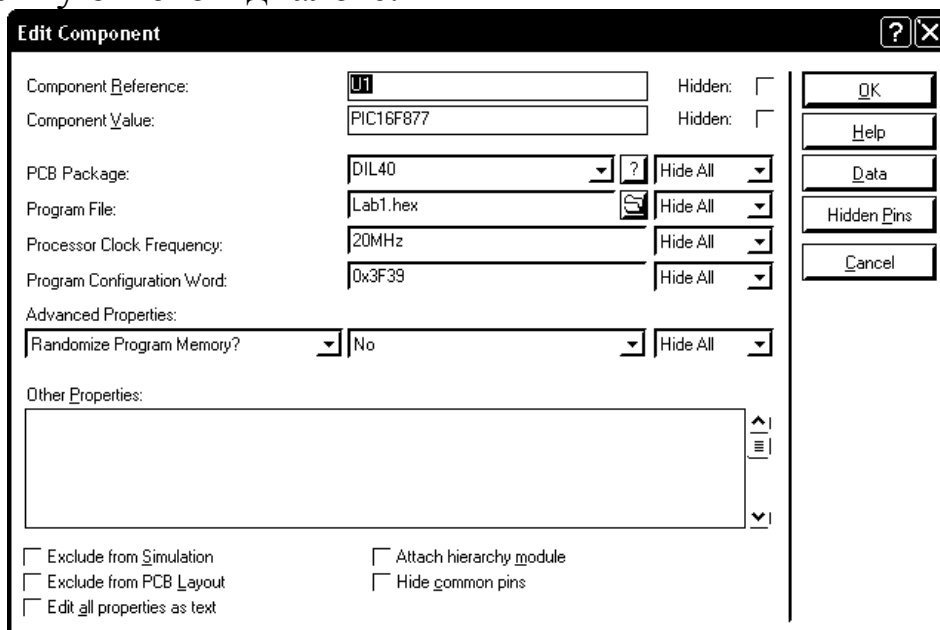


Рисунок 5 – Окно редактирования свойств микроконтроллера

Теперь можно запускать симуляцию кнопкой Play на панели 60. Если все сделано правильно, и нет никаких ошибок, то гирлянда с частотой примерно раз в секунду будет переключать группы зажженных светодиодов.

Изменим эти группы. Для этого в исходном файле с помощью любого текстового редактора заменим в строках

```
..... . .
movlw 0x55      ; 01010101 в PORTB
.....
movlw 0xaa      ; 10101010 в PORTB
.....
```

числа на другие значения (например, 0x0 и 0xff). В Proteus тоже есть редактор кода, однако пользоваться им очень неудобно, так как он неправильно отражает русскую кодировку.

Остановим симуляцию (кнопка Stop the simulation панели 60). Сохраним программу. Запускаем симуляцию. Теперь группы

светодиодов, которые зажигаются в тот или иной момент времени, поменялись!

При симуляции логические уровни на проводниках отмечаются цветными маркерами в виде квадратов. Если маркер красного цвета, то в данной цепи уровень напряжения, соответствующий логической единице, если маркер синего цвета – то логическому нулю, если маркер серого цвета – то вывод находится в высокоимпедансном третьем (Z) состоянии, если маркер желтого цвета – то на данном проводнике наблюдается конфликт, т.е. одновременно выводится и уровень логической единицы, и уровень логического нуля.

Одним из важнейших режимов симуляции является отладка программ. Управление отладкой программ производится в группе меню Debug. Выберем пункт меню Debug\Start Restart Debugging или нажмем Ctrl+F12. Должно появиться окно исходного кода программы PIC CPU Source Code (рисунок 6). Если исходная программа на ассемблере задана, а окно не появляется автоматически, то необходимо щелкнуть по пункту меню Debug\ PIC CPU Source Code.

В окне исходного кода программы следующие элементы управления:

1 – текущая команда, которую выполняет микроконтроллер – отмечается красным треугольником, и выделенная команда – отмечается синей полосой;

2 – поле для установки точек прерывания (точек останова, breakpoints). Точки прерывания устанавливаются нажатием на кнопку 9 или двойным щелчком мыши на этом поле и отмечаются красными кружками;

3 – файл исходной программы со служебной информацией;

4 – кнопка запуска симуляции (до ближайшей точки останова);

5 – кнопка пошагового выполнения программы без захода в процедуры (процедуры вызываются командой call);

6 - кнопка пошагового выполнения программы с заходом в процедуры;

7 – кнопка выполнения программы до выхода из текущей процедуры;

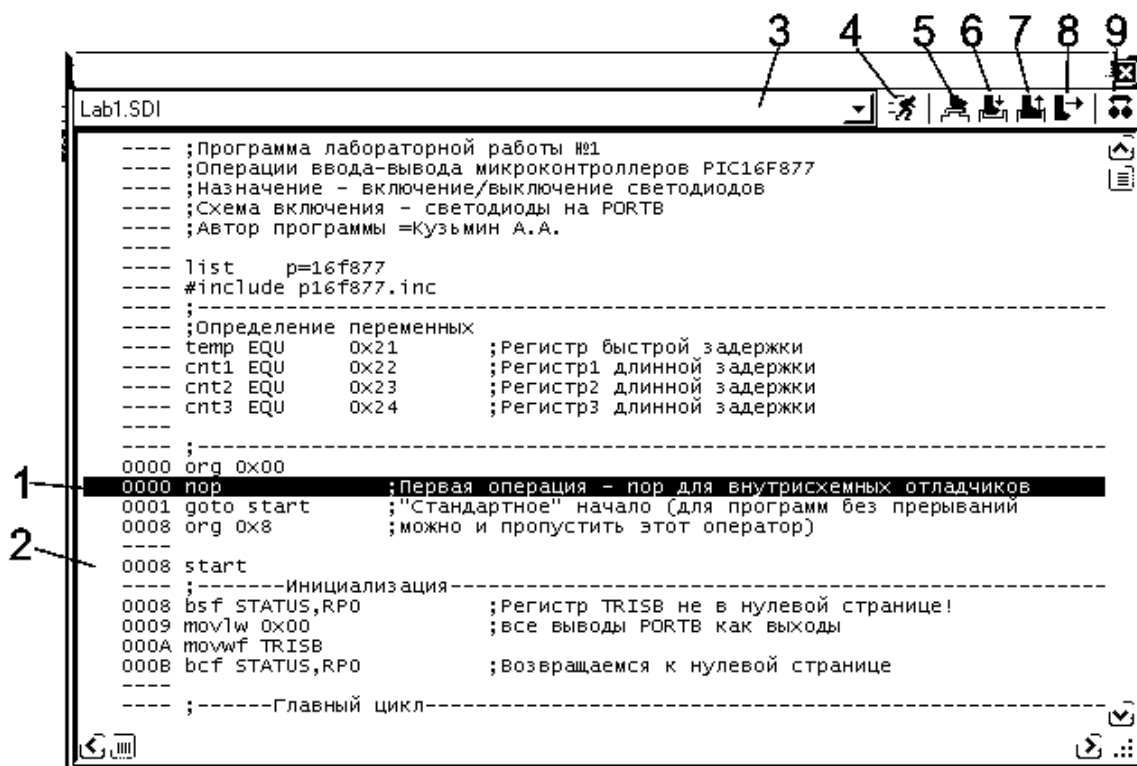


Рисунок 6 - Окно исходного кода программы PIC CPU Source Code

8 – кнопка выполнения программы до выделенной синей полосой команды;

9 – установка/снятие точки прерывания.

Очень важная информация для отладки программ также содержится в окнах PIC CPU Registers – в этом окне содержится информация о наиболее важных регистрах микроконтроллера, PIC CPU Data Memory – в этом окне содержится информация о всех регистрах данных микроконтроллера, PIC CPU EPROM Memory - в этом окне содержится информация о состоянии EEPROM памяти микроконтроллера, PIC CPU Program Memory - в этом окне содержится информация о состоянии памяти программ микроконтроллера, PIC CPU Stack – информация о стеке. Все эти окна можно открыть из группы меню Debug.

Для удобства наблюдения за состоянием определенного регистра существует окно Watch Window. Добавим, например, в окно Watch Window ссылку на регистр1 длинной задержки из нашей программы, который определяется в программе как

```
cnt1 EQU    0x22      ;Регистр1 длинной задержки.
```

Для этого открываем окно из пункта меню Debug\ Watch Window. В контекстном меню этого окна выбираем пункт Add Items (By Adress). Получаем окно ввода регистра по адресу (рисунок 7).

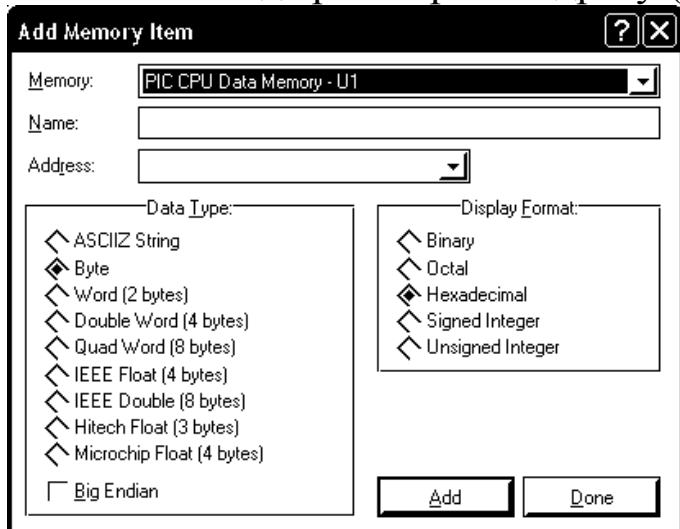


Рисунок 7 - Окно ввода регистра по адресу

В этом диалоге можем выбрать формат данных, формат представления, вид памяти, откуда будет выбираться значение, символическое имя переменной и адрес переменной. Оставим настройки вывода и вид памяти по умолчанию, в имени переменной введем cnt1, а в адресе 0x22. Нажимаем кнопку Add и Done. В результате в окне Watch Window появится информация о состоянии этой переменной.

Важнейшим инструментом отладки является условная точка остановки, т.е. остановка при выполнении определенного условия. Условную точку остановки при, например, равенстве нашего регистра cnt1 допустим двойке, можно поставить в окне Watch Window путем выбора пункта контекстного меню Watchpoint Condition (рисунок 8).

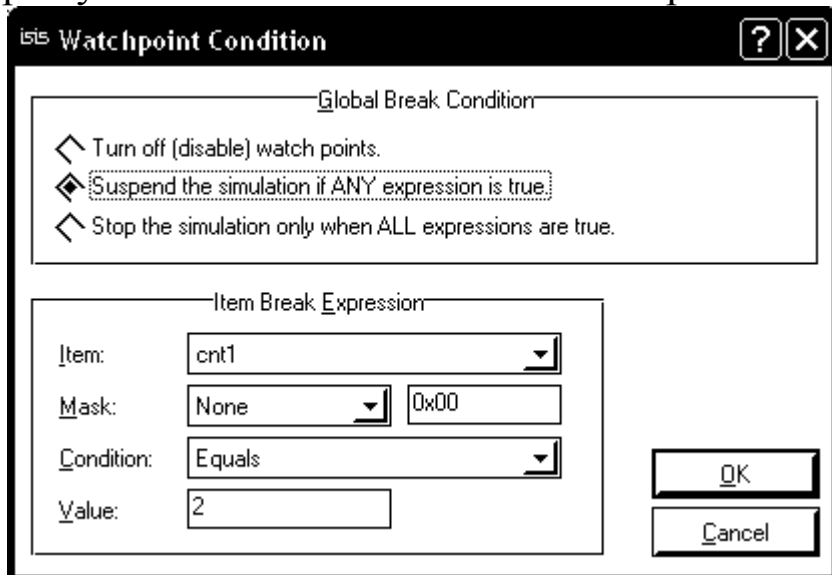


Рисунок 8 – Установка условной точки остановки

Выбираем в этом диалоге следующие настройки: приостановить симуляцию, если условие истинно (Suspend the simulation if ANY expression is true), проверяемый регистр (Item) – cnt1, условие остановки (Condition) – равенство (Equals), значение остановки (Value) – 2.

Запускаем симуляцию. При равенстве содержимого регистра cnt1 двум, программа остановится.

Продемонстрируем, как с помощью Proteus можно исследовать переходные процессы в схемах и цифровые диаграммы. Для демонстрации исследования цифровых процессов вполне хватит созданной нами гирлянды. Добавим на выводы микроконтроллера, которые соединены с резисторами, пробники (кнопка 51). Если пробники помещать ближе к резисторам, то их названия автоматически присваиваются следующим значениям: R1(1), R2(1)... R8(1). Если пробники помещать ближе к выводам микроконтроллера, то названия у них будут другие: U1(RB0/INT), U1(RB1) и т.д., но смысл пробника останется тем же – он будет показывать напряжение на этом проводе во время симуляции. Кроме того, на цепь, помеченную пробником, легче ссылаться во время анализа переходных процессов. Поместим в рабочую область цифровую диаграмму путем выбора графика DIGITAL из набора графиков для анализа (кнопка 48). Распахнем на весь экран график (пункт Maximize контекстного меню). Получим диалог расчета цифровых диаграмм DIGITAL ANALYSIS (рисунок 9).

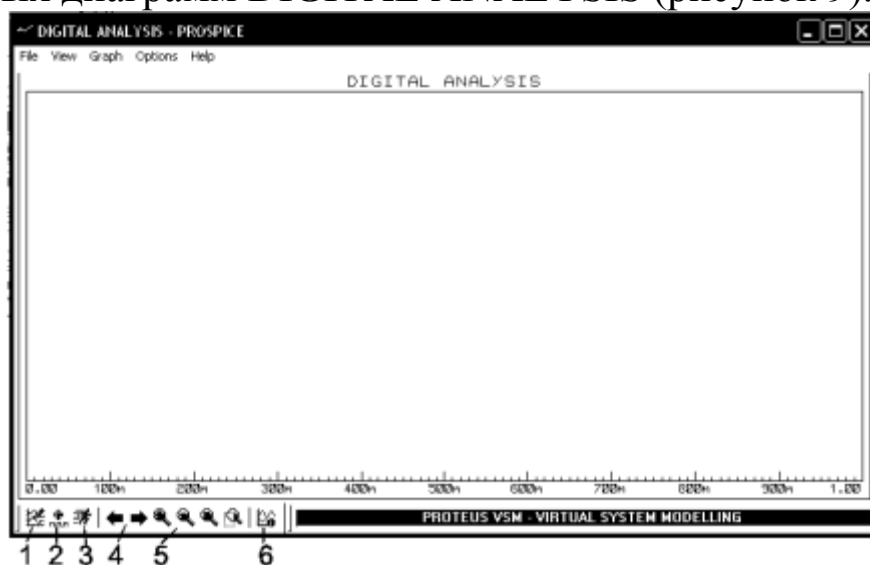


Рисунок 9 – Диалог расчета цифровых диаграмм DIGITAL ANALYSIS

Основные элементы управления этого диалога:

1 – Редактирование общих свойств графика, таких как заголовок, названия осей, начального и конечного времени, а также свойств программы-вычислителя диаграмм Spice. По умолчанию расчет ведется до одной секунды. Так как у нас задержка в переключении диодов рассчитана тоже примерно на одну секунду, то для наблюдения переходного процесса поставим значение Stop Time в 2 секунды.

2 – Добавляет пробники для расчета – в диалоге выбирается напряжение на каких пробниках будет добавляться в рассчитанные цифровые диаграммы. Выберем здесь созданные ранее нами пробники R1(1), R2(1)... R8(1) (или U1(RB0/INT), U1(RB1) и т.д.).

3 – Запуск симуляции. Рассчитываются цифровые диаграммы выбранных пробников. Результат нашей симуляции показан на рисунке 10.

4 – Кнопки навигации по графикам по оси времени.

5 – Кнопки выбора масштаба изображения.

6 – Просмотр системных сообщений, выдаваемых во время расчета графиков.

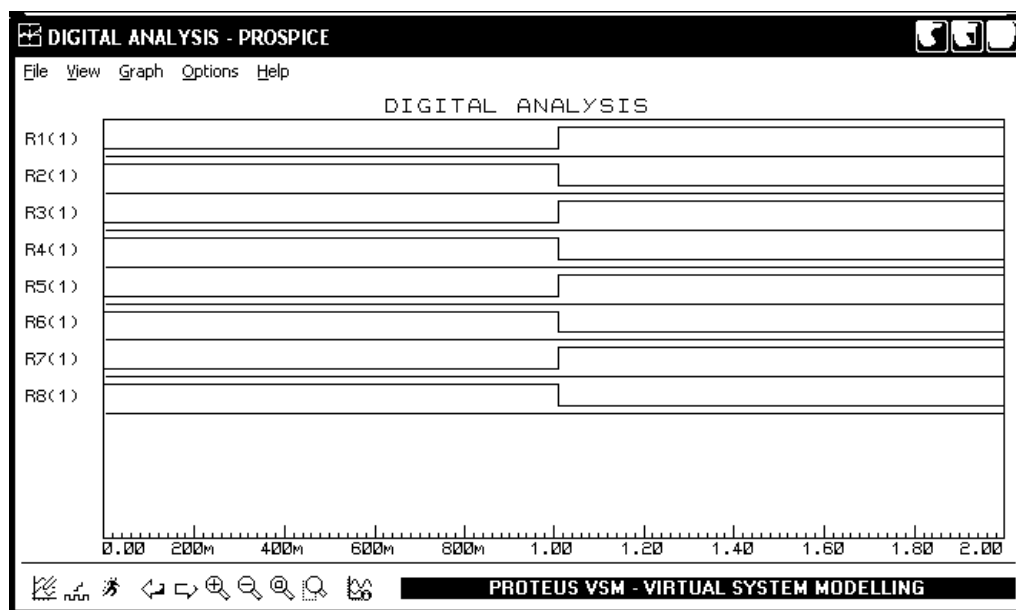


Рисунок 10 - Результат расчета цифровых диаграмм гирлянды

Аналогично происходит и расчет аналоговых переходных процессов (график ANALOGUE).

2. Цель работы

Целью работы является приобретение базовых навыков в моделировании электронных схем с использованием программы Proteus.

3. Порядок выполнения работы

3.1 Собрать схему, согласно выданному варианту.

3.2 Задать соответствующую программу микроконтроллеру.

3.3 Промоделировать собранную схему с использованием отладчика, точек останова, условных точек останова.

3.4 Снять осциллограммы с контрольных точек.

4. Содержание отчета:

Титульный лист с названием и номером работы, а также с фамилиями исполнителей;

Цель работы;

Задание на лабораторную работу;

Снимок экрана (скриншот) системы Proteus с программой, остановленной на точке останова;

Осциллограммы сигналов в контрольных точках;

Выводы.

5 Контрольные вопросы

5.1 Основные элементы интерфейса оболочки ISIS.

5.2 Приемы поиска необходимых элементов в оболочке ISIS.

5.3 Какие электронные компоненты Proteus вы знаете?

5.4 Как поместить на разрабатываемую схему символ земли GROUND?

5.5 Как вводятся соединяющиеся и пересекающиеся проводники?

5.6 Как изменяются номиналы простейших аналоговых компонентов?

5.7 Какие применяются суффиксы для модификации цифровых значений параметров компонентов?

5.8 Почему при изменении номинала ограничивающего резистора изменяется яркость свечения светодиода?

5.9 Как меняются свойства компонентов в Proteus?

5.10 Как микроконтроллерам задаются программы, по которым они работают?

5.11 Как задается частота, на которой работает микроконтроллер?

5.12 Какие ошибки могут возникнуть при запуске симуляции схемы?

5.13 Что обозначают цветные квадраты рядом с проводниками во время симуляции?

5.14 Какими элементами интерфейса управляется процесс отладки программ?

5.15 Как при пошаговой отладке отрабатывается выполнение процедур?

5.16 Как установить и снять точку останова?

5.17 Какие дополнительные отладочные окна поддерживает система Proteus?

5.18 Что такое условная точка остановки и как ее установить в Proteus?

5.19 Зачем нужны пробники напряжения в Proteus?

5.20 Как происходит расчет графиков переходных процессов (цифровых диаграмм) ?

5.21 Какое различие между цифровыми и аналоговыми графиками переходных процессов?

Практическая работа №2

Автоматизированное проектирование приборов и систем с использованием САПР SPICE

Целью данной лабораторной работы является изучение особенностей построения моделей элементов в системе Proteus, имитационная часть которых основана на SPICE-моделях.

1 Краткие сведения о содержании работы

Пусть необходимо построить модель микросхемы AD633 фирмы Analog Devices. Из документации, которую можно скачать с сайта производителя (www.analog.com), узнаем, что AD633 – это законченный четырехквadrанный аналоговый умножитель (рисунок 2.1).

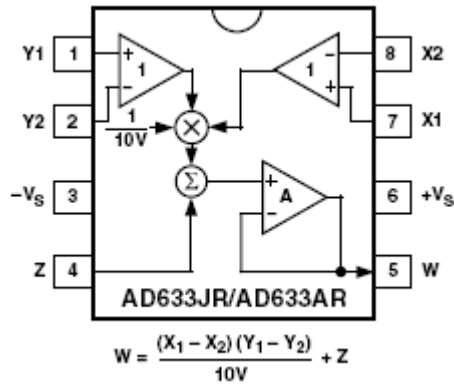


Рисунок 2.1 – Структура умножителя AD633

На сайте производителя также можно найти SPICE модель микросхемы «AD633.cir». SPICE-модель представляет собой текстовый файл, в котором на языке SPICE описаны параметры и соединения лежащих в основе модели цепей. Например, модель AD633 определяется следующим образом (приведен только фрагмент модели):

```

* Copyright 1993 by Analog Devices, Inc.
*
* Refer to "README.DOC" file for License
Statement. Use of this model
* indicates your acceptance with the terms and
provisions in the License Statement.
*
* Node assignments
*
*           X1
*           | X2
*           | | Y1
*           | | | Y2
*           | | | | VNEG
*           | | | | | Z
*           | | | | | | W
*           | | | | | | | VPOS
*           | | | | | | | |
.SUBCKT AD633 1 2 3 4 5 6 7 8
*
EREF 100 0 POLY(2) 8 0 5 0 (0,0.5,0.5)
*
* X-INPUT STAGE & POLE AT 15 MHz
*

```

```
IBX1 1 0 DC 8E-7
IBX2 2 0 DC 8E-7
EOSX 10 1 POLY(1) (16,100) (5E-3,1)
RX1A 10 11 5E6
RX1B 11 2 5E6
```

```
.....
GO2 5 72 69 5 11.623E-3
```

```
RO2 72 5 86
```

```
LO 72 7 1E-7
```

```
*
```

```
* MODELS USED
```

```
*
```

```
.MODEL QX NPN(BF=1E4)
```

```
.MODEL DX D(IS=1E-15)
```

```
.ENDS AD633
```

В этом текстовом файле в строке «.SUBCKT AD633 1 2 3 4 5 6 7 8» указывается после ключевого слова SUBCKT (от «Subcircuit» - вложенная цепь) наименование модели и порядок расположения выводов. Интерпретация выводов, как правило, указывается в комментарии. Например, первый вывод интерпретируется как вход X1, второй вывод – как вход X2 и т.д. Порядок расположения выводов очень важен для построения будущей модели Proteus.

Далее как в лабораторной работе 1 строится условное графическое отображение модели, причем выводы назовем следующим образом: четыре основных входа – как X1, X2, Y1, Y2, вход коррекции выходного уровня нуля – Z, выход – W, напряжение питания – V+ и V-. Компоновка модели происходит аналогично тому, как это было сделано в лабораторной работе 1. Однако при компоновке необходимо дополнительно определить три свойства модели (рисунок 2.2).

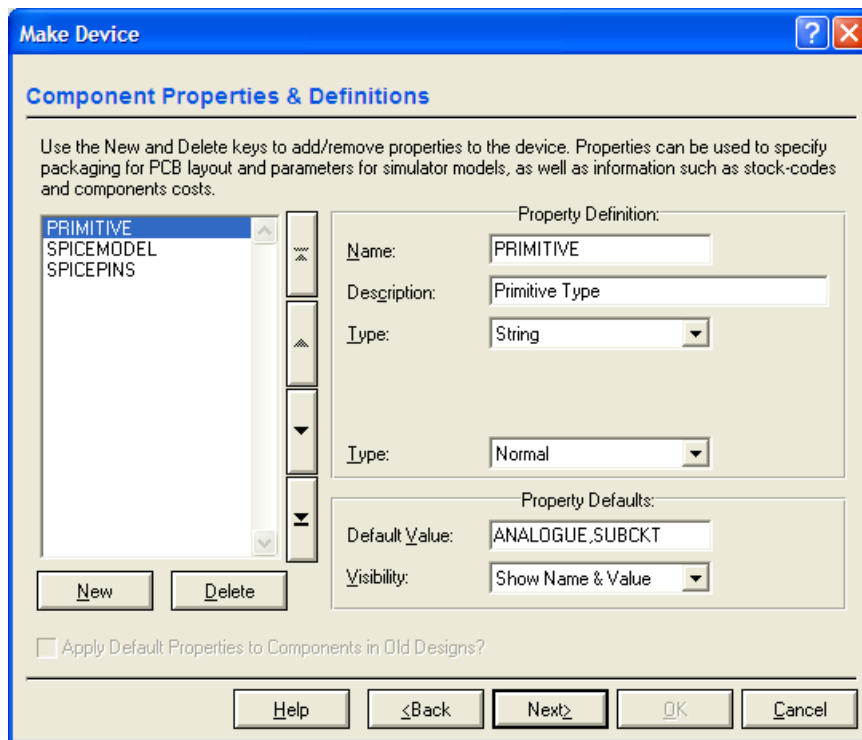


Рисунок 2.2 – Определение свойства PRIMITIVE модели

В диалоге задания свойств по умолчанию модели, кнопка New создает новое свойство; в поле Name определяется имя свойства; в поле Description задается краткое описание свойства; в поле Type задается тип свойства (в нашем случае это строка – String); во втором поле Type задаются разрешения на изменение свойства (нормальное или, например, только для чтения и т.п.); в поле Default Value указывается значение по умолчанию свойства; в поле Visibility определяются параметры вывода на экран свойства.

Первое свойство «PRIMITIVE» должно указывать Proteus, что модель будет аналоговой и задаваться вложенной цепью. Для этой цели значение по умолчанию свойства (Default Value) должно быть равным «ANALOGUE,SUBCKT». Второе свойство «SPICEMODEL» должно указывать Proteus на имя файла модели, и на наименование модели внутри этого файла (например, «AD633,AD633.CIR»). Третье свойство «SPICEPINS» ставит в соответствие список выводов модели Proteus списку выводов SPICE-модели. Список выводов SPICE-модели AD633 приведен выше – это «X1, X2, Y1, Y2, VNEG, Z, W, VPOS». В наименованиях наших выводов этот список выглядит следующим образом: «x1,x2,y1,y2,V-,z,w,V+». Именно эта строка должна содержаться в значении свойства «SPICEPINS» (рисунок 3).

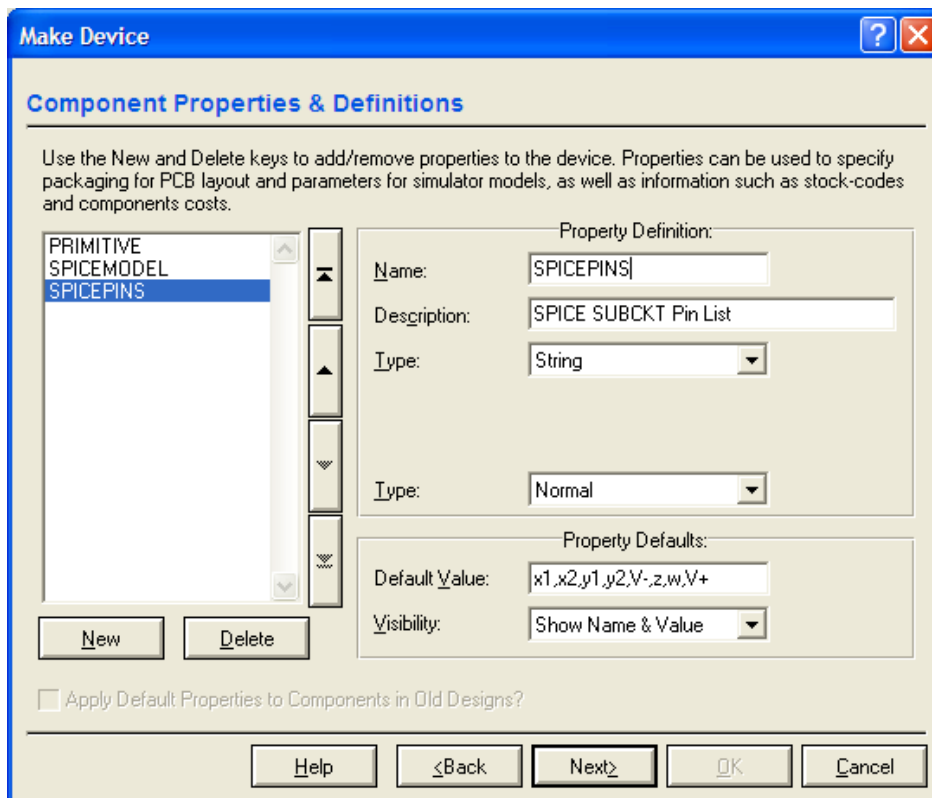


Рисунок 2.3 – Определение свойства SPICEPINS модели

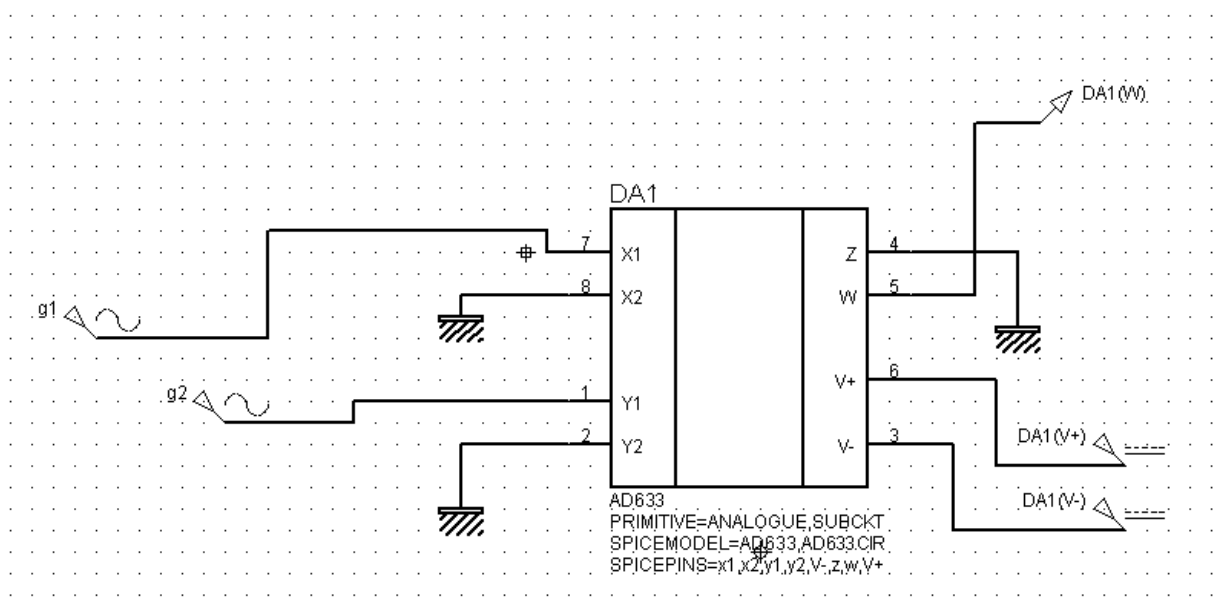


Рисунок 2.4 – Тестовая схема для умножителя AD633

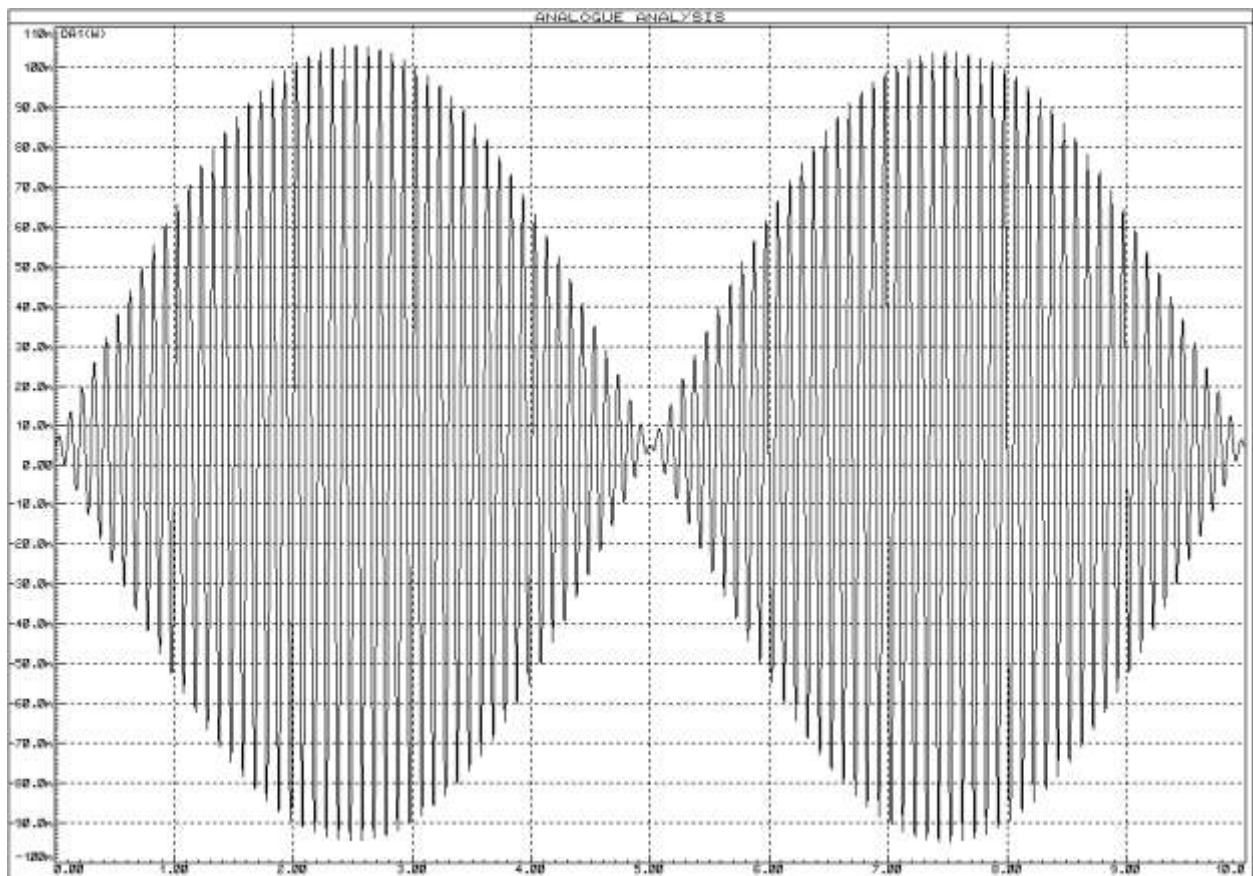


Рисунок 2.5 – Результат перемножения двух синусоид

После окончания процесса конструирования модели, ее необходимо испытать. Для этого подадим на выводы питания соответствующие документации питающие напряжения, входы X2, X3, Z соединим с землей, а на входы X1 и Y1 подадим напряжения с двух синусоидальных генераторов с разной частотой (рисунок 2.4). На вход X1 подадим синусоиду с амплитудой 1 В и частотой 0.1 Гц, а на вход Y1 подадим синусоиду с амплитудой 1 В и частотой 10 Гц. Полученный результат умножения показан на рисунке 2.5. Нетрудно убедиться, что результат соответствует формуле умножителя, приведенной на рисунке 2.1. Следовательно, новый элемент работает правильно.

2. Порядок выполнения работы

2.1 Найти документацию, SPICE-модель микросхемы, согласно выданному варианту.

2.2 Сконструировать новую модель микросхемы в системе Proteus (условное графическое изображение микросхемы выполнять согласно ЕСКД, имитационная часть модели должна обеспечиваться SPICE-моделью).

2.3 Разработать в Proteus испытательный стенд микросхемы.

2.4 Провести испытания работоспособности новой модели микросхемы.

3. Содержание отчета:

Титульный лист с названием и номером работы, а также с фамилией исполнителя;

Цель работы;

Задание на лабораторную работу;

Структура, схема включения микросхемы согласно документации;

Схема испытательного стенда микросхемы;

Результаты испытания работоспособности новой модели микросхемы.

Выводы.

Практическая работа №3

Аналого-цифровое и цифро-аналоговое преобразование, создание цифровых приборов и систем на базе современных микроконтроллеров и микропроцессоров

1 Краткие теоретические сведения

Модуль аналого-цифрового преобразования (АЦП) в микроконтроллере PIC16F877 имеет восемь входных каналов. Входной аналоговый сигнал через коммутатор каналов заряжает внутренний конденсатор АЦП C_{HOLD} . Модуль АЦП преобразует напряжение, удерживаемое на конденсаторе C_{HOLD} , в соответствующий 10-разрядный цифровой код методом последовательного приближения. Источник верхнего и нижнего опорного напряжения может быть программно - выбран с выводов VDD, VSS, AN3/VREF+ или AN2/VREF-.

Допускается работа модуля АЦП в SLEEP режиме микроконтроллера, при этом в качестве источника тактовых импульсов для АЦП должен быть выбран RC генератор.

Для управления АЦП в микроконтроллере используется 4 регистра:

- Регистр результата ADRESH (старший байт);
- Регистр результата ADRESL (младший байт);
- Регистр управления ADCON0;
- Регистр управления ADCON1.

Регистр ADCON0 используется для настройки работы модуля АЦП, а с помощью регистра ADCON1 устанавливается, какие входы микроконтроллера будут использоваться модулем АЦП и в каком режиме (аналоговый вход или цифровой порт ввода/вывода). При сбросе микроконтроллера все выходы, мультиплицированные с модулем АЦП (ANx), настраиваются как аналоговые входы.

Структурная схема модуля АЦП показана на рисунке 1. Структура регистра ADCON0 приведена на рисунке 2. Структура регистра ADCON1 приведена на рисунке 3.

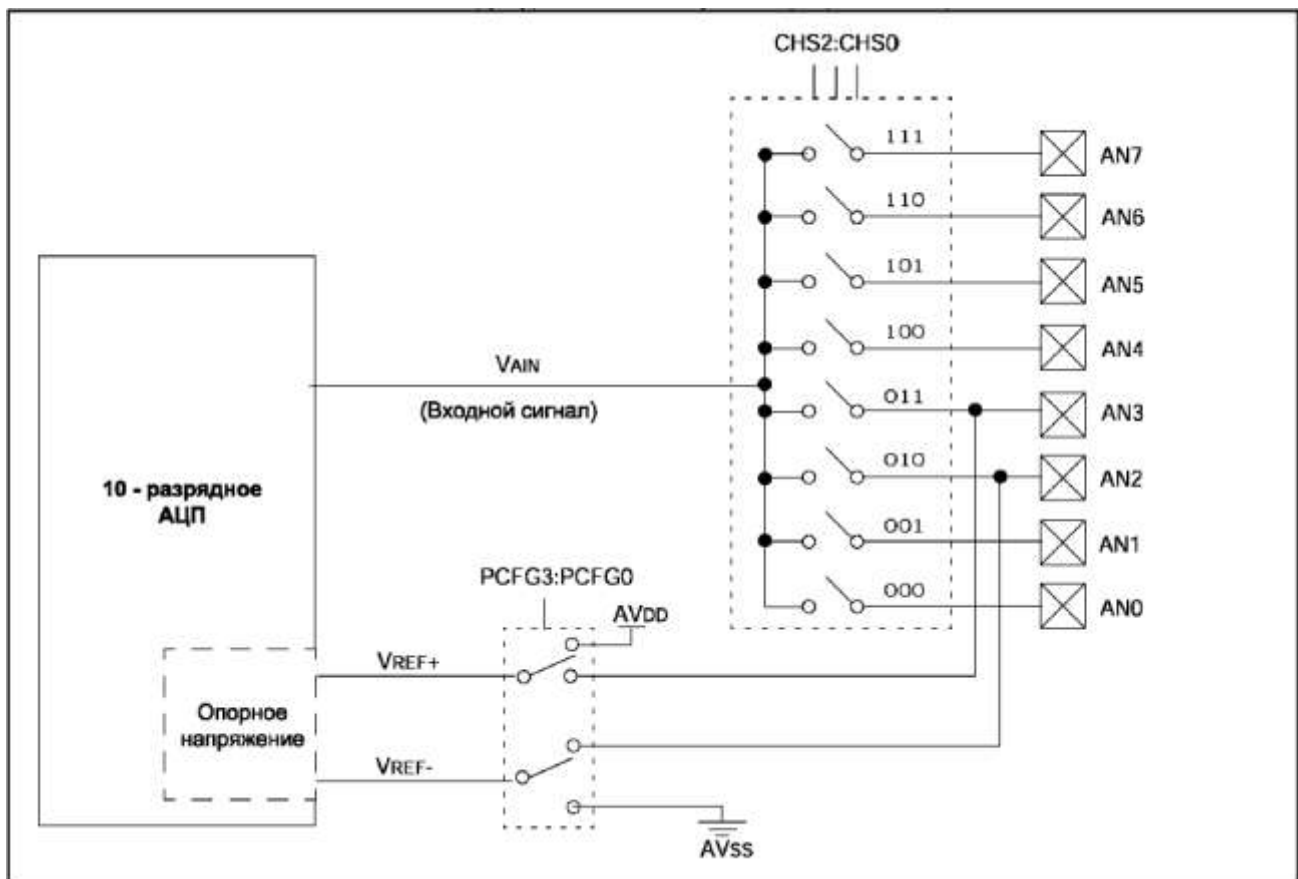


Рисунок 1 - Структурная схема модуля 10 - разрядного АЦП

В регистрах ADRESH:ADRESL сохраняется 10 - разрядный результат аналого-цифрового преобразования. Когда преобразование завершено, результат преобразования записывается в регистры ADRESH:ADRESL, после чего сбрасывается бит GO/-DONE (ADCON0<2>) и устанавливается флаг прерывания ADIF.

После включения и настройки АЦП необходимо выбрать рабочий аналоговый канал. Соответствующие биты TRIS аналоговых каналов должны настраивать канал порта ввода/вывода на вход. Перед началом преобразования необходимо выдержать временную паузу для обеспечения необходимой точности преобразования (конденсатор C_{HOLD} должен успевать полностью заряжаться до уровня входного напряжения). Точные формулы расчета временной паузы приведены в документации. Отметим, что при сопротивлении источника сигнала 10 кОм эта пауза составляет примерно 20 мкс, а при сопротивлении источника сигнала 50 Ом эта пауза составляет примерно 11 мкс.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	-	ADON
Бит 7							Бит 0

R – чтение бита
W – запись бита
U – не реализовано, читается как '0'
-n – значение после POR
-x – неизвестное значение после POR

биты 7-6: **ADCS1:ADCS0**: Выбор источника тактового сигнала
00 = $F_{osc}/2$
01 = $F_{osc}/8$
10 = $F_{osc}/32$
11 = F_{RC} (внутренний RC генератор модуля АЦП)

биты 5-3: **CHS2:CHS0**: Выбор аналогового канала
000 = канал 0, (AN0)
001 = канал 1, (AN1)
010 = канал 2, (AN2)
011 = канал 3, (AN3)
100 = канал 4, (AN4)
101 = канал 5, (AN5)
110 = канал 6, (AN6)
111 = канал 7, (AN7)

Примечание. Для микроконтроллеров, в которых не реализованы все 8 каналов АЦП. Модуль АЦП аппаратно позволяет выполнять выборку нереализованных каналов.

бит 2: **GO/DONE**: Бит статуса модуля АЦП
Если ADON=1
1 = модуль АЦП выполняет преобразование (установка бита вызывает начало преобразования)
0 = состояние ожидания (аппаратно сбрасывается по завершению преобразования)

бит 1: **Не используется**: читается как '0'

бит 0: **ADON**: Бит включения модуля АЦП
1 = модуль АЦП включен
0 = модуль АЦП выключен и не потребляет тока

Рисунок 2 – Структура регистра ADCON0

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0				
ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0				
Бит 7											Бит 0

R – чтение бита
W – запись бита
U – не реализовано, читается как 0
-n – значение после POR
-x – неизвестное значение после POR

бит 7: **ADFM**: Формат сохранения 10-разрядного результата (см. рисунок 23-6)
1 = правое выравнивание, 6 старших бит ADRESH читаются как '0'
0 = левое выравнивание, 6 младших бит ADRESL читаются как '0'

биты 6-4: Не используются: читаются как '0'

биты 3-0: **PCFG3:PCFG0**: Управляющие биты настройки каналов АЦП

PCFG3: PCFG0	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	Кан./ VREF ⁽¹⁾
0000	A	A	A	A	A	A	A	A	AVDD	AVSS	8/0
0001	A	A	A	A	VREF+	A	A	A	AN3	AVSS	7/1
0010	D	D	D	A	A	A	A	A	AVDD	AVSS	5/0
0011	D	D	D	A	VREF+	A	A	A	AN3	AVSS	4/1
0100	D	D	D	D	A	D	A	A	AVDD	AVSS	3/0
0101	D	D	D	D	VREF+	D	A	A	AN3	AVSS	2/1
011x	D	D	D	D	D	D	D	D	AVDD	AVSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6/2
1001	D	D	A	A	A	A	A	A	AVDD	AVSS	6/0
1010	D	D	A	A	VREF+	A	A	A	AN3	AVSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2/2
1110	D	D	D	D	D	D	D	A	AVDD	AVSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1/2

A = аналоговый вход D = цифровой канал ввода/вывода

Примечание 1. В этом столбце указывается число аналоговых каналов, доступных для выполнения преобразования, и число входов источника опорного напряжения.

Рисунок 3 – Структура регистра ADCON1

Рекомендованная последовательность действий для работы с АЦП:

1. Настроить модуль АЦП:

- Настроить выходы как аналоговые входы, входы VREF или цифровые каналы ввода/вывода (ADCON1);
- Выбрать входной канал АЦП (ADCON0);
- Выбрать источник тактовых импульсов для АЦП (ADCON0);
- Включить модуль АЦП (ADCON0).

2. Настроить прерывание от модуля АЦП (если необходимо):

- Сбросить бит ADIF в '0';
- Установить бит ADIE в '1';
- Установить бит PEIE в '1';
- Установить бит GIE в '1'.

3. Выдержать паузу, необходимую для зарядки конденсатора

C_{HOLD} .

4. Начать аналого-цифровое преобразование:
 - Установить GO/-DONE бит в '1' (ADCON0).
5. Ожидать, окончания преобразования:
 - Ждать, пока бит GO/-DONE не будет сброшен в '0'; ИЛИ
 - Ожидать прерывание по окончанию преобразования.
6. Считать результат преобразования из регистров ADRESH:ADRESL, сбросить бит ADIF в '0', если это необходимо.
7. Для следующего преобразования необходимо выполнить шаги, начиная с пункта 1 или 2. Время преобразования одного бита определяется как время T_{AD} . Минимальное время ожидания перед следующим преобразованием должно составлять $2T_{AD}$.

Время получения одного бита результата равно T_{AD} . Для 10-разрядного результата требуется как минимум $11.5 T_{AD}$. Параметры тактового сигнала для АЦП определяются программно, T_{AD} может принимать следующие значения:

- $2T_{Osc}$;
- $8T_{Osc}$;
- $32T_{Osc}$;
- Внутренний RC генератор модуля АЦП (2-6 мкс).

Для получения корректного результата преобразования необходимо выбрать источник тактового сигнала АЦП, обеспечивающий время T_{AD} не менее 1.6 мкс. Таблицы выбора источника тактового сигнала приведены в документации. Отметим, что для частоты 20 МГц необходимо выбирать источником T_{AD} или RC-генератор, или $32T_{OSC}=1.6$ мкс (значения конфигурационных битов $ADCS1:ADCS0 = 10$). Когда тактовая частота микроконтроллера больше 1МГц, рекомендуется использовать RC генератор АЦП только для работы в SLEEP режиме.

В следующем примере показана последовательность действий для работы с АЦП. Выводы настроены как аналоговые входы. Источник опорного напряжения – AV_{DD} , AV_{SS} . Разрешены прерывания от модуля АЦП. Источником импульсов преобразования является RC генератор АЦП. Аналоговое цифровое преобразование выполняется с вывода AN0:

```
BSF STATUS, RP0    ; Выбрать банк 1
CLRf ADCON1       ; Настроить входы АЦП
BSF PIE1, ADIE    ; Разрешить прерывания от АЦП
```

```

    BCF STATUS, RP0 ; Выбрать банк 0
    MOVLW 0xC1      ; Тактовые импульсы от RC
генератора АЦП,
    MOVWF ADCON0   ; включить АЦП, выбрать канал 0
    BCF PIR1, ADIF ; Сбросить флаг прерываний от
АЦП
    BSF INTCON, PEIE ; Разрешить периферийные
прерывания
    BSF INTCON, GIE ; Разрешить прерывания в
системе
    ;
    ; Выдержать паузу, необходимую для заряда
внутреннего конденсатора CHOLD.
    ; Затем начинать преобразование АЦП.
    ;
    BSF ADCON0, GO ; Старт преобразования
    : ; Ожидать установку флага ADIF или сброс
    : ; бита GO/-DONE по завершению преобразования

```

10-разрядный результат преобразования сохраняется в спаренном 16-разрядном регистре ADRESH:ADRESL. Запись результата преобразования может выполняться с правым или левым выравниванием, в зависимости от значения бита ADFM (см. рисунок 4). Не задействованные биты регистра ADRESH:ADRESL читаются как '0'. Если модуль АЦП выключен, то 8-разрядные регистры ADRESH и ADREL могут использоваться как регистры общего назначения.

Модуль АЦП может работать в SLEEP режиме микроконтроллера при условии, что источником импульсов преобразования АЦП будет внутренний RC генератор (ADCS1:ADCS0=11). При выборе RC генератора импульсов модуль АЦП сделает задержку в один машинный цикл перед началом преобразования. Это позволяет программе пользователя выполнить команду SLEEP, тем самым уменьшить “цифровой шум” во время преобразования. После завершения преобразования аппаратно сбрасывается бит GO/-DONE в '0', результат преобразования сохраняется в регистрах ADRESH:ADRESL. Если разрешено прерывание от АЦП, то микроконтроллер выйдет из режима SLEEP. Если же прерывание было запрещено, то после преобразования модуль АЦП будет выключен, хотя бит ADON останется установленным.

Если был выбран другой источник тактовых импульсов АЦП (не внутренний RC генератор), то выполнение программой инструкции SLEEP прервет процесс преобразования и выключит модуль АЦП, оставив установленным бит ADON. Выключение модуля АЦП уменьшит ток потребления микроконтроллера. Инструкция SLEEP должна быть выполнена сразу после команды, устанавливающей бит GO/-DONE в '1'.

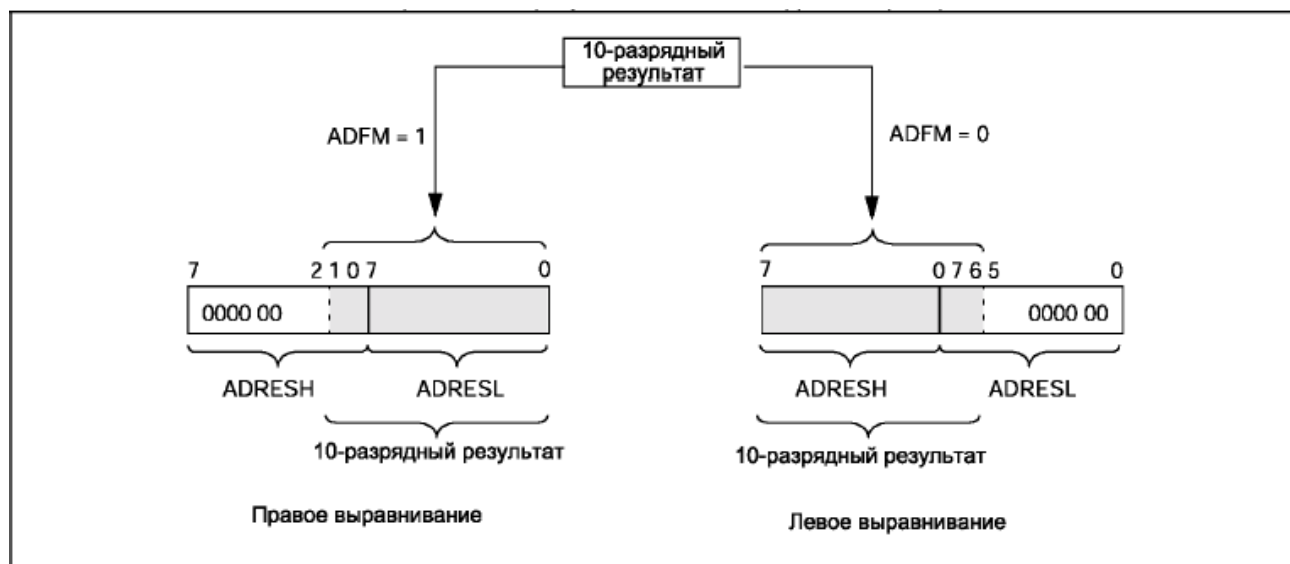


Рисунок 4 - Выравнивание результата аналого-цифрового преобразования

Абсолютная точность АЦП определяется суммарной ошибкой, исходя из ошибки дискретизации, интегральной ошибки, ошибки шкалы, ошибки смещения и монотонности. Суммарная ошибка определяется как максимальный разброс между текущим и идеальным результатом для любого значения. Абсолютная ошибка АЦП меньше ± 1 значащего бита при $VDD=VREF$, но она возрастает при отклонении $VREF$ от VDD .

В некотором диапазоне напряжений на аналоговом входе цифровой результат будет один и тот же. Это возникает из-за дискретизации, которая неизбежна при преобразовании аналоговой величины в цифровую форму. Ошибка дискретизации составляет ± 1 значащего бита, и единственный способ уменьшить ее - увеличить разрядность АЦП.

Ошибку смещения составляет разность между результатом первого преобразования и идеальным значением. Эта ошибка сдвигает всю передаточную функцию, и может быть учтена при помощи

калибровки. Ошибка вносится в результате наложения токов утечки и выходного сопротивления источника сигнала. Ошибка усиления измеряется как максимальное отклонение результата, скорректированного с учетом ошибки смещения. Эта ошибка проявляется в виде изменения наклона передаточной функции. Ошибка усиления может быть откалибрована и учтена. Ошибка линейности определяется как разница в приращении входного напряжения для получения одинакового приращения выходного кода и не поддается калибровке. Интегральная ошибка вычисляется как отклонение результата, скорректированного с учетом ошибки усиления. Дифференциальная ошибка вычисляется как отклонение максимальной длины кода результата от идеальной длины кода без учета других ошибок.

№ пар.	Обоз.	Описание	Мин.	Тип**	Макс.	Ед.	Примечание	
A01	N _R	Разрядность	-	-	10	бит	V _{REF} = V _{DD} = 5.12В, V _{SS} ≤ V _{AIN} ≤ V _{REF}	
A03	E _L	Интегральная погрешность	-	-	< ± 1	LSb	V _{REF} = V _{DD} = 5.12В, V _{SS} ≤ V _{AIN} ≤ V _{REF}	
A04	E _{DL}	Дифференциальная погрешность	-	-	< ± 1	LSb	V _{REF} = V _{DD} = 5.12В, V _{SS} ≤ V _{AIN} ≤ V _{REF}	
A06	E _{OFF}	Ошибка смещения	-	-	< ± 2	LSb	V _{REF} = V _{DD} = 5.12В, V _{SS} ≤ V _{AIN} ≤ V _{REF}	
A07	E _{GN}	Ошибка усиления	-	-	< ± 1	LSb	V _{REF} = V _{DD} = 5.12В, V _{SS} ≤ V _{AIN} ≤ V _{REF}	
A10	-	Монотонность ⁽³⁾	Гарантируется			-	V _{SS} ≤ V _{AIN} ≤ V _{REF}	
A20	V _{REF}	Опорное напряжение (V _{REF+} -V _{REF-})	2.0	-	V _{DD} + 0.3	В	Минимальное значение для 10-разрядного АЦП	
A21	V _{REF+}	Положительное опорное напр.	AV _{DD} - 2.5	-	AV _{DD} + 0.3	В		
A22	V _{REF-}	Отрицательное опорное напр.	AV _{SS} - 0.3	-	V _{REF-} - 2.0	В		
A25	V _{AIN}	Аналоговый вход	V _{SS} - 0.3	-	V _{REF} + 0.3	В		
A30	Z _{AIN}	Сопротивление источника сигн.	-	-	10.0	кОм		
A40	I _{AD}	Потребляемый ток АЦП	F	-	220	-	мкА	Среднее потребление при включенном АЦП ⁽¹⁾
			LF	-	90	-	мкА	
A50	I _{REF}	Потребляемый ток от источника опорного напряжения ⁽²⁾	-	-	1000	-	мкА	Во время выборки V _{AIN} . Основано на дифференц. значении заряда C _{HOLD} до V _{AIN} . Во время преобразования.
			-	-	10	-	мкА	

** - В столбце "Тип." приведены параметры при V_{DD}=5.0В, 25°C, если не указано иное. Эти параметры являются ориентировочными, используются при разработке устройств и не измеряются.

Примечания:

1. Выключенный модуль АЦП не потребляет тока, кроме токов утечки.
2. Ток со входа RA3 или V_{DD} в зависимости от выбранного источника опорного напряжения.
3. Результат АЦП никогда не уменьшается с увеличением напряжения на входе и не имеет кодов отсутствия напряжения.

2. Цель работы: уметь использовать модуль АЦП в микроконтроллерах Microchip.

3. Порядок выполнения работы

3.1 Решить практическую задачу согласно своему варианту с использованием модуля АЦП.

3.2 Проверить работоспособность разработанной программы в системе Proteus.

4. Содержание отчета:

Титульный лист с названием и номером работы, а также с фамилией исполнителя

Цель работы

Задание на лабораторную работу

Схема спроектированной системы

Листинг программы.

Выводы

Практическая работа №4

Автоматизированное проектирование узлов приборов и систем обработки сигнала во временном домене

1 Краткие теоретические сведения

Микросхемы ЦАП позволяют микроконтроллерам производить генерацию аналоговых сигналов. Интерфейсы микросхем ЦАП делятся на параллельные и последовательные интерфейсы. Параллельные интерфейсы занимают больше линий ввода-вывода у микроконтроллера, но они и более быстродействующие. Последовательные интерфейсы ЦАП занимают значительно меньше линий ввода-вывода, так как данные в таких интерфейсах, как правило, передаются по двум основным линиям – линии данных и линии синхронизации. Существуют микросхемы, в которые информация передается даже по одному проводу – шине «One Wire». Однако такая передача данных требует, как правило, больше тактов работы микроконтроллера. Разберем пример подключения ЦАП к микроконтроллеру на примере микросхемы MAX503.

Микросхема MAX503 (см. документацию на www.maxim-ic.com) – это экономичный, 10-битный преобразователь цифрового кода в напряжение, который использует или однополярное напряжение питания +5В, или двухполярное напряжение питания плюс минус 5В. Величина потребляемого тока (250 мкА) от напряжения питания +5В позволяет использовать эту микросхему для производства портативной медицинской аппаратуры.

На рисунке 1 приведена функциональная схема и обозначение выводов микросхемы MAX503.

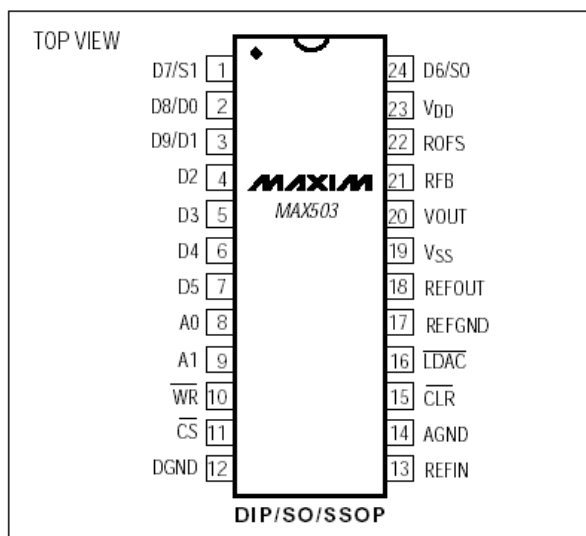
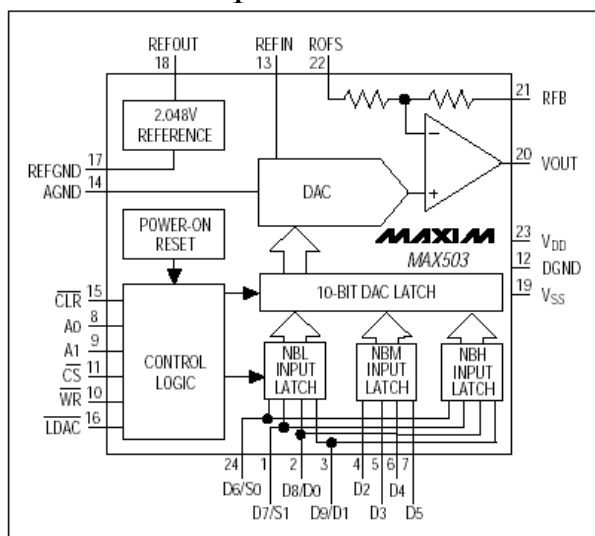


Рисунок 1 - Функциональная схема и обозначение выводов микросхемы MAX503

Назначение выводов микросхемы MAX503 приведено в таблице 1. На рисунке 2 приведена временная диаграмма циклов записи в MAX503.

MAX503 предоставляет несколько путей записи информации в свои внутренние регистры. Используем способ, при котором изменение цифровой комбинации во входном регистре сразу производит изменение аналогового напряжения на выходе, т.е. режим без дополнительной отдельной пересылки 10 бит данных из входного регистра в регистр ЦАП, при таком способе не используется управляющий вывод LDAC (LDAC=0). Диаграмма обмена информацией при таком способе показана на рисунке 3.

На рисунке 4 приведена схема включения MAX503 в лабораторном макете.

С учетом всей вышеприведенной информации (схемы включения, режима обмена, цикла записи и т.д.) напишем программу обмена информацией между микроконтроллером и ЦАП MAX503 (листинг 1).

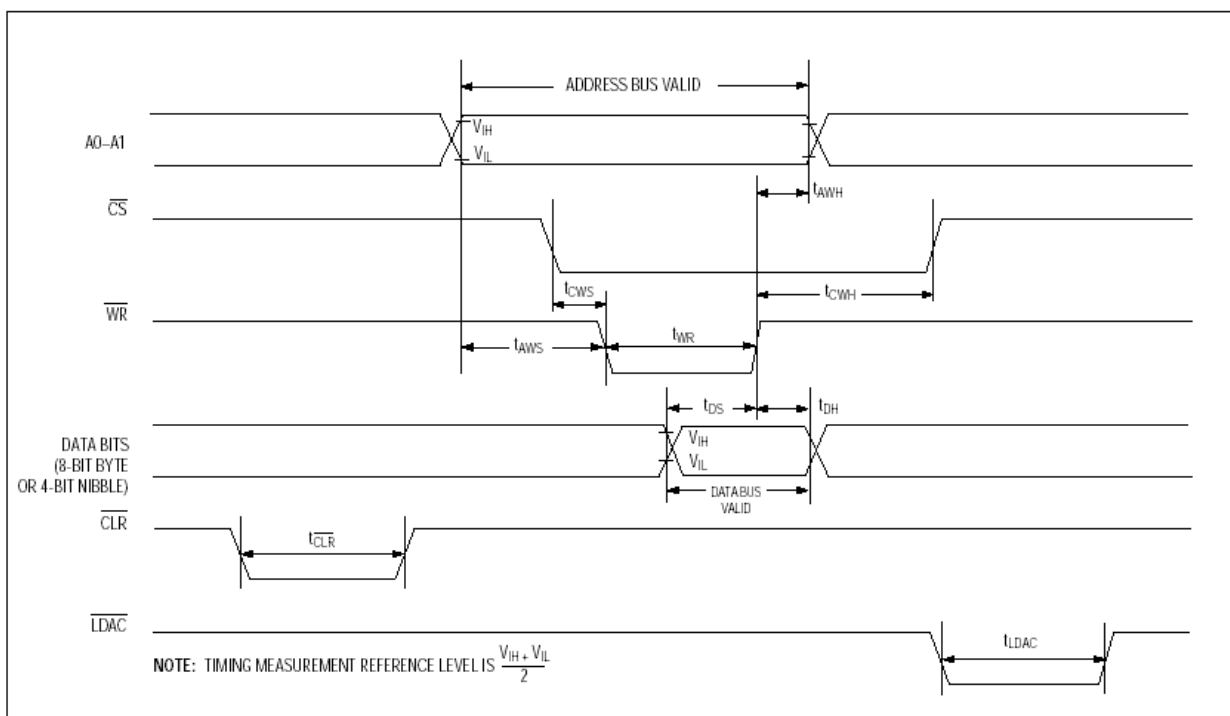


Рисунок 2 - Временная диаграмма циклов записи в MAX503

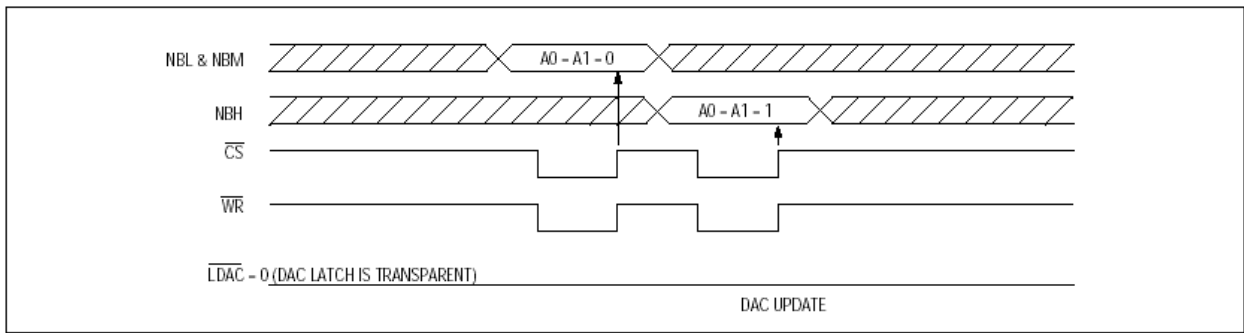


Рисунок 3 – Диаграмма обмена информацией при «прозрачных» защелках регистра ЦАП ($LDAC=0$). При режиме записи NBL и NBM во входной регистр записываются 8 младших бит информации, при режиме записи NBH записываются два старших бита.

Таблица 1 - Назначение выводов микросхемы MAX503

Номер вывода	Имя вывода	Назначение вывода
1	D7/ S1	Вход D7 когда A0 = A1 = 1 или вход S1 когда A0 = 0 и A1 = 1.
2	D8/ D0	Вход D8 когда A0 = A1 = 1 или вход D0 когда A0 = 0 и A1 = 1.
3	D9/ D1	Вход D9 когда A0 = A1 = 1 или вход D1 когда A0 = 0 и A1 = 1.
4	D2	Вход D2.
5	D3	Вход D3.
6	D4	Вход D4.
7	D5	Вход D5.
8	A0	Адресная линия A0.
9	A1	Адресная линия A1.
10	WR	Вход записи (активный уровень – низкий). Используется совместно с CS для записи данных во входные защелки, которые выбраны адресными линиями A0 и A1.
11	CS	Выбор кристалла (активный уровень – низкий). Позволяет включить запись в микросхему данных из общих шин
12	DGND	Цифровая земля.
13	REFIN	Вход источников опорного напряжения (ИОН). Подключите внешний источник опорного напряжения к этому выводу или подключите вывод REFOUT (вывод 18) для использования внутреннего источника на 2.048 В.
14	AGND	Аналоговая земля.
15	CLR	Очистка (активный уровень низкий). Очищает защелки ЦАП (регистр ЦАП) в нули.
16	LDAC	Загрузка защелок в ЦАП (активный уровень низкий). Перемещает содержимое входных защелок в регистр ЦАП и изменяет выходное напряжение.
17	REFGND	Земля ИОН. Должна быть присоединена к AGND когда используется внутренний ИОН. Подсоединяется к VDD для отключения внутреннего ИОН и экономии энергии.

18	REFOUT	Выход внутреннего ИОН. Внутренний ИОН на 2.048 В.
19	VSS	Отрицательный вывод напряжения питания. Часто подключается к земле при однополярном питании или к -5В при двухполярном питании.
20	VOUT	Выход ЦАП(напряжение).
21	RFB	Вывод резистора обратной связи выходного усилителя. Рекомендуется подключать напрямую к VOUT.
22	ROFS	Вывод резистора смещения выходного усилителя. Подключается к VOUT для коэффициента усиления = 1, к AGND для коэффициента усиления = 2 или к REFIN для биполярного выхода.
23	VDD	Положительный вывод напряжения питания. (+5V)
24	D6/S0	Вход D6 когда A0 = A1 = 1 или вход S0 когда A0 = 0 и A1 = 1.

Листинг 1 – Программа обмена информацией между микроконтроллером и ЦАП МАХ503

```
; При инициализации:
clrfs PORTC      ;сигнал ldac должен быть равен 0 !!!
(rc2)
;.....
;-----
;Функции работы с цап МАХ503СWГ
;-----
;8 бит информации - в регистре dacreg
;2 старших бита - в регистре dacreg2
;сигнал ldac должен быть равен 0 !!!
sendtodac

    bcf PORTC,0    ;a0,a1=low
    bcf PORTC,1
    bcf PORTE,2    ;cs=0
    call delay2    ;выдержка tcws
    bcf PORTE,1    ;wr=0
                ;сначала передаются 6 бит d0-d5, а d6=0,
d7=0
    movf dacreg,w
    andlw 0x3f
    movwf PORTD
    call delay2
    bsf PORTE,1    ;wr=1
    call delay2
    bsf PORTC,0    ;a0,a1=high
    bsf PORTC,1
    call delay2    ;выдержка tcws
    bcf PORTE,1    ;wr=0
                ;потом передается d6,d7, а в d0 и d1
передаются d8 и d9
    movf dacreg,w
    andlw 0xc0
    iorwf dacreg2,w
    movwf PORTD
    call delay2
    bsf PORTE,1    ;wr=1
```

```
call delay2
bsf PORTE,2    ;cs=1
return
```

2 Цель работы: уметь находить и пользоваться документацией на микросхемы; реализовывать аналоговый вывод из микроконтроллерных устройств.

3. Порядок выполнения работы

3.1 Найти документацию на заданную согласно своему варианту микросхему ЦАП.

3.2 Разобраться в режимах работы микросхемы ЦАП, ее протоколах информационного обмена.

3.3 Решить практическую задачу по сопряжению микросхемы ЦАП и микроконтроллера. Разработать подпрограмму информационного обмена микроконтроллера с микросхемой ЦАП.

3.4 Проверить работоспособность разработанной системы в Proteus.

4. Содержание отчета:

Титульный лист с названием и номером работы, а также с фамилией исполнителя

Цель работы

Задание на лабораторную работу

Схема спроектированной системы

Листинг программы.

Основные выдержки из фирменной документации на микросхему ЦАП (схема включения, назначение выводов, диаграммы записи и т.д.)

Выводы.

5 Контрольные вопросы

5.1 От чего зависит быстродействие ЦАП?

5.2 На какой частоте работают современные микросхемы ЦАП?

5.3 Зачем для ЦАП необходим источник опорного напряжения (ИОН)?

5.4 Какие выводы у ЦАП MAX503 образуют шину данных, какие шину адреса, а какие шину управления?

5.5 Нарисуйте типичную характеристику преобразования ЦАП.

5.6 От чего зависит максимальное выходное напряжение у ЦАП МАХ503?

6 Задания.

Требуется заданную микросхему ЦАП подключить к микроконтроллеру, написать процедуры обмена. Документацию на микросхемы, при необходимости, загрузить с сайта производителя.

№ Варианта	Тип ЦАП	Производитель ЦАП
1	МАХ504	Maxim-ic
2	МАХ515	Maxim-ic
3	МСР4921	Microchip
4	LTC1450	Linear Technology
5	LTC1451	Linear Technology
6	LTC1456	Linear Technology
7	LTC1655	Linear Technology
8	МСР4922	Microchip
9	AD5305	Analog Devices
10	AD5310	Analog Devices
11	AD5315	Analog Devices
12	AD5320	Analog Devices
13	AD5611	Analog Devices
14	AD5325	Analog Devices
15	МСР4921	Microchip
16	МСР4821	Microchip
17	МСР4725	Microchip
18	МАХ5822	Maxim-ic
19	МАХ5821	Maxim-ic
20	МАХ5820	Maxim-ic
21	МАХ5541	Maxim-ic

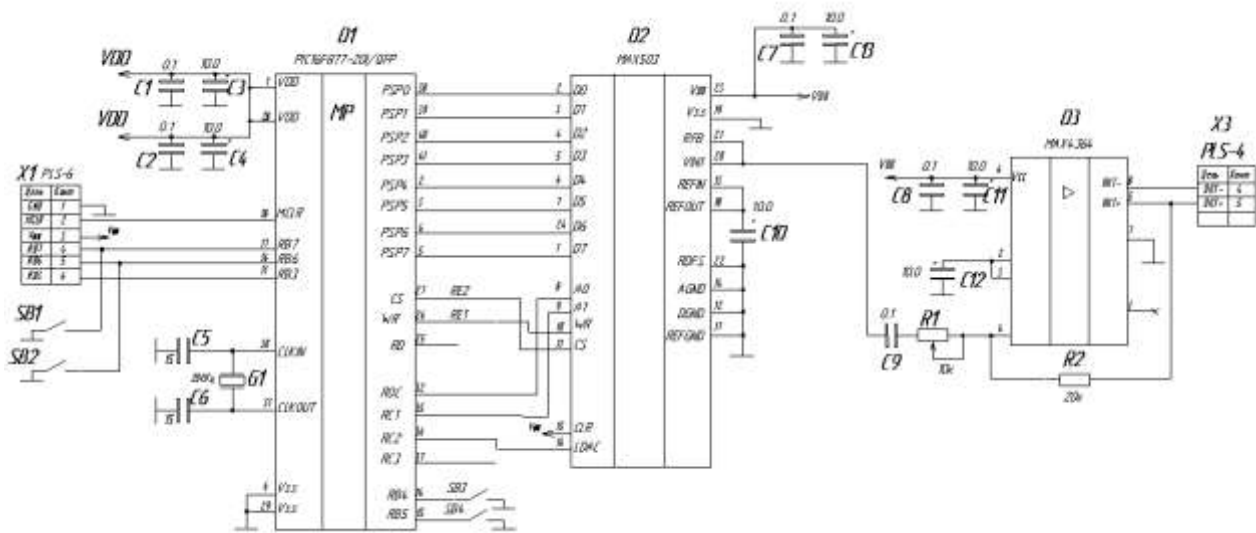


Рисунок 4 – Схема включения МАХ503 в лабораторном макете.