Минобрнауки России

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Оренбургский государственный университет»**

Кафедра промышленной электроники и информационно-измерительной техники

**Методические указания**

для обучающихся по освоению дисциплины

*«Б1.Д.В.15 Электроника и микропроцессорная техника»*

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки

*12.03.04 Биотехнические системы и технологии*

(код и наименование направления подготовки)

*Инженерное дело в медико-биологической практике*

(наименование направленности (профиля) образовательной программы)

Тип образовательной программы

*Программа прикладного бакалавриата*

Квалификация

*Бакалавр*

Форма обучения

*Очная*

Оренбург, 2021

Составитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Хлуденев

Методические указания рассмотрены и одобрены на заседании кафедры промышленной электроники и информационно-измерительной техники

Заведующий кафедрой ПЭиИИТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.В. Худорожков

Методические указания являются приложением к рабочей программе дисциплины «Электроника и микропроцессорная техника», зарегистрированной в ЦИТ под учетным номером .

**1 Общие положения**

Дисциплина относится к обязательным дисциплинам (модулям) вариативной части блока Д «Дисциплины (модули)». Изучается в 8 семестре.

В результате изучения дисциплины студент должен:

**Знать:** основные научно-технические проблемы и перспективы развития медицинской электронной техники, ее взаимосвязь со смежными областями.

**Уметь:** выполнять информационный поиск перспективной элементной базы.

**Владеть:** средствами автоматизации проектирования объектов профессиональной деятельности.

Достижение планируемых результатов обучения обеспечивается добросовестным отношением обучающегося к изучению теоретического материала дисциплины с использованием материала лекций и рекомендованной литературы, ознакомлением с рекомендованными периодическими изданиями, своевременным выполнением лабораторных работ и заданий на практических занятиях, а также выполнением и защитой курсовой работы.

**2 Общие методические рекомендации**

Контроль достижения планируемых результатов обучения обеспечивается использованием оценочных средств, представленных в «Фонде оценочных средств …» (приложение к рабочей программе дисциплины), в учебном процессе во время занятий по расписанию, а также во время промежуточной аттестации.

**2.1 Теоретический материал, изучаемый при освоении дисциплины**

**Раздел 1 Элементная база**

Пассивные элементы. Полупроводниковые приборы: диоды, биполярные и полевые транзисторы. Характеристики и параметры полупроводниковых приборов

**Раздел 2 Аналоговые устройства**

Усилители постоянного и переменного тока, операционные усилители, усилители мощности. Операционные схемы. Активные фильтры. Генераторы.

**Раздел 3 Цифровые устройства**

Информационные основы цифровой электроники. Логические элементы. Функциональные узлы комбинационного типа. Триггеры. Функциональные узлы последовательностного типа. Полупроводниковые запоминающие устройства.

**Раздел 4 Микропроцессорная техника**

Архитектура и функционирование микропроцессорного устройства. Микроконтроллеры. Форматы команд и данных. Виды адресации. Средства разработки и отладки программного обеспечения. Периферийные модули: порты, таймеры, цифро-аналоговые и аналого-цифровые преобразователи. Прерывания. Построение встраиваемых микропроцессорных устройств.

**Вопросы, изучаемые самостоятельно с использованием рекомендуемой литературы:**

**Раздел 1 Элементная база**

Пассивные элементы. Характеристики и параметры ЭРЭ.

**Раздел 2 Аналоговые устройства**

Источники питания электронных устройств.

**Раздел 3 Цифровые устройства**

Интегральные схемы логических элементов и функциональных узлов.

**Раздел 4 Микропроцессорная техника**

Параметры цифро-аналоговых и аналого-цифровых преобразователей. Система команд PIC16/

**Контроль усвоения** изученного теоретического материала осуществляется методом опроса на лабораторных работах, практических занятиях, а также методом **тестирования**. Объем (тематику) проверяемого при тестировании материала и дату проведения тестирования определяет ведущий преподаватель.

**Рекомендуемая литература**:

1 Опадчий, Ю. Ф Аналоговая и цифровая электроника [Текст] : полный курс: учеб. для вузов / Ю. Ф. Опадчий, О. П. Глудкин, А. И. Гуров. - М. : Горячая линия-Телеком, 2005. - 768 с. : ил. - ISBN 5-93517-002-7.

2 Гусев, В. Г. Электроника и микропроцессорная техника [Текст] : учеб. для вузов / В. Г. Гусев, Ю. М. Гусев.- 5-е изд., стер. - Москва : Высш. шк., 2008. - 798 с. : ил. - Библиогр.: с. 786-787. - Прил.: с. 788-795. - ISBN 978-5-06-005680-8.

**2.2 Практикум**

Тематика лабораторных работ и практических занятий приведена в рабочей программе дисциплины. Отчет о выполненной лабораторной работе каждый обучающийся защищает индивидуально. Во время защиты студент должен быть готовым пояснить методику проведения исследований, показать умение анализировать результаты, полученные в ходе проведения исследований, быть готовым ответить на вопросы преподавателя по теме проводимых исследований.

Целью практических занятий по дисциплине является формирование умений решать задачи анализа и расчета характеристик электрических цепей, выполнять информационный поиск перспективной элементной базы. Тематика практических занятий связана с решением задач расчета характеристик и параметров элементов, функциональных звеньев и узлов. Практические задания студенты выполняют под руководством преподавателя с использованием методических рекомендаций (Приложение А). Методические рекомендации для лабораторных занятий приведены в приложении Б.

**2.3 Курсовая работа**

Целью выполнения курсовой проекта в шестом семестре является закрепление практических навыков самостоятельного выполнения маршрута проектирования, решения типовых проектных задач, развитие творческих способностей и овладение средствами проектирования электронных устройств, средствами автоматизации проектирования объектов профессиональной деятельности.

Темой курсовой работы является разработка частотно-избирательного фильтра (по вариантам). Студент должен, применяя принципы проектирования сложных технических объектов, сформировать электрическую функциональную схему фильтра, выполнить расчет параметров элементов. Корректность полученных результатов и выполнение всех требований задания должны быть подтверждены результатами анализа частотных характеристик. Варианты заданий приведены в фонде оценочных средств.

Над курсовой работой студенты работают самостоятельно – в свободное от занятий время. Возникающие в процессе работы вопросы студенты могут решать в часы консультаций с руководителем работы. При выполнении курсовой работы руководствоваться методическими указаниями (приложение В).

Срок сдачи преподавателю законченной курсовой работы на проверку – за 1 – 3 дня до защиты. Защита курсовой работы – 18 учебная неделя (согласно расписанию).

**Приложение А**

**Методические рекомендации к практическим занятиям**

**1 Характеристики полупроводниковых диодов**

**1.1 Краткие сведения из теории**

Цель работы - получить практические навыки исследования статических и динамических характеристик полупроводниковых диодов.

Схема для экспериментального определения статической вольт-амперной характеристики (ВАХ) диода приведена на рисунке 1.1.

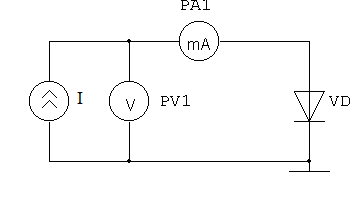


Рисунок 1.1

Статическая ВАХ описывает зависимость тока, протекающего через диод от приложенного напряжения

I = f(U) .

На практике ВАХ получают, задаваясь различными значениями I и измеряя U. При положительных значениях I и U снимают прямую ветвь ВАХ, при отрицательных – обратную. За положительное направление тока принимают – от анода к катоду. По статической ВАХ можно определить важнейшие параметры диода. Прямое падение напряжения при заданном прямом токе определяют на прямой ветви ВАХ. Обратный ток при заданном значении обратного напряжения определяют на обратной ветви ВАХ.

Свойства диода при его включении и выключении определяют по временной диаграмме тока, которую снимают при достаточно быстром изменении полярности проложенного напряжения. Типы исследуемых выпрямительных диодов и стабилитронов приведены в таблице 1.1

Таблица 1.1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Вариант | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Диод | KD102A | KD202D | KD212A | KD503A | KD521A | D237A | KD220A | KD510A |
| Стаби-литрон | KS133A | KS139A | KS147A | KS156A | KS168A | D814A | D814D | D815B |

1.2.1 Построить прямую ветвь статической ВАХ диода.

1.2.2 Построить обратную ветвь статической ВАХ диода.

1.2.3 Построить временную диаграмму переключения диода. Определить время обратного восстановления диода.

1.2.4 Построить обратную ветвь статической ВАХ стабилитрона.

**1.3 Порядок выполнения работы**

1.3.1 Открыть проект \Elecrtonics\LR1\Static командой **File>Open\Project**.

1.3.2 Открыть в менеджере проекта схему измерения статической ВАХ ROOT: PAGE1. В полях базы данных диода Value и Implementation задать тип диода.

1.3.3 Выполнить редактирование профиля моделирования командой **PSpice>Edit Simulation Profile** для расчета входных статических характеристик БТ.

Установить вид анализа (Analysis type) – DC Sweep.

Для опции Primary Sweep:

- задать Sweep Variable Current source Name: I1;

- задать закон изменения Sweep type: Linear и пределы изменения (Start Value =0, End Value = **100m** Increment = **1m**).

Закрыть окно редактирования нажатием кнопки OK.

1.3.4 Выполнить моделирование командой **PSpice>Run**. При этом выполняется запуск программы PSpice A/D.

1.3.5 Вывести на экран график зависимости падения напряжения на диоде U(I), для этого командой **Trace>Add Trace** открыть окно Add Trace и выбрать сигнал V(1), подтвердить выбор нажатием клавиши ОК. Зарисовать график. Удалить график командой **Trace>Delete All Traces**.

1.3.6 Выполнить редактирование профиля моделирования командой **PSpice>Edit Simulation Profile** для расчета обратной ветви ВАХ.

Задать предел изменения End Value = -**100u**, Increment = -**1u**).

Закрыть окно редактирования нажатием кнопки OK.

1.3.7 Выполнить моделирование командой **PSpice>Run**. При этом выполняется запуск программы PSpice A/D.

1.3.8 Вывести на экран график зависимости падения напряжения на диоде U(I), для этого командой **Trace>Add Trace** открыть окно Add Trace и выбрать сигнал V(1), подтвердить выбор нажатием клавиши ОК. Зарисовать график. Удалить график командой **Trace>Delete All Traces**.

1.3.9 В полях базы данных диода Value и Implementation задать тип стабилитрона. Повторить пункты 1.3.6 – 1.3.8 для значений обратного тока от 0 до -30 мА с шагом 0.1 мА.

Закрыть проект \Elecrtonics\LR1\Static командой **File>Close**.

1.3.10 Открыть проект \Elecrtonics\LR1\Dinamic командой **File>Open\Project**.

Выполнить редактирование профиля моделирования командой **PSpice>Edit Simulation Profile** для расчета во временной области, для этого на закладке Analysis:

- задать вид анализа (Analysis type) – Time Domain (Transient);

- задать значение интервала расчета Run to Time = 2us;

- закрыть окно редактирования нажатием кнопки OK.

1.3.11 Выполнить моделирование во временной области командой **PSpice>Run**.

1.3.12 Вывести на экран временную диаграмму I(t), для этого командой **Trace>Add Trace** открыть окно Add Trace и выбрать сигнал I(D1:A), подтвердить выбор нажатием клавиши ОК. Зарисовать график. Удалить график командой **Trace>Delete All Traces**.

1.3.13 Закрыть проект \Elecrtonics\LR1\\Dinamic командой **File>Close**.

**1.4 Оформление работы**

Отчет по лабораторной работе должен содержать:

- измерительные схемы;

- графики ветвей статической ВАХ диода и стабилитрона;

- временную диаграмму переключения диода.

**1.5 Контрольные вопросы**

1.5.1 Что показывает статическая ВАХ диода?

1.5.2 Как снимают статическую ВАХ на прямой и обратной ветви?

1.5.3 Какие параметры диода можно определить по статической ВАХ? Поясните, как это можно сделать.

1.5.4 По временной диаграмме переключения диода объясните процесс выключения диода.

**2 Характеристики биполярных транзисторов**

**2.1 Краткие сведения из теории**

Цель работы - получить практические навыки исследования статических характеристик биполярных транзисторов (БТ).

Схема для экспериментального определения статических характеристик n-p-n транзистора по схеме с общим эмиттером (ОЭ) приведена на рисунке 2.1.

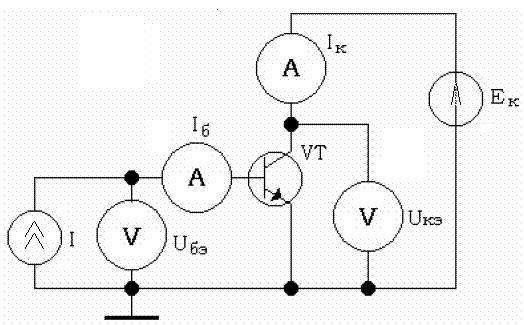


Рисунок 2.1

Входными статическими характеристиками транзистора, включенного с ОЭ, является семейство характеристик

IБ = f(UБЭ) при UКЭ=const.

На практике входные характеристики получают, задаваясь различными значениями IБ и измеряя UБЭ, при фиксированном значении UКЭ. Обычно ограничиваются семейством из двух характеристик при UКЭ = 0 и UКЭ > UБЭ max .

Выходными статическими характеристиками транзистора, включенного с ОЭ, является семейство характеристик

IК = f(UКЭ) при IБ=const.

Выходные характеристики снимают, задаваясь различными значениями UКЭ и измеряя IК, при фиксированном значении IБ.

Все задающие и измеряемые параметры должны находиться в допустимых пределах. Предельные значения параметров транзисторов приведены в таблице 2.1

Таблица 2.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вариант | Тип транзистора | Предельная расс. мощность, Вт | Предельное  значение IК,, мА | Предельное  значение UКЭ , В |
| 1 | КТ312Б | 0.225 | 30 | 40 |
| 2 | KT315b | 0.15 | 100 | 20 |
| 3 | KT3102a | 0.25 | 100 | 50 |
| 4 | KT3117a | 0.5 | 400 | 50 |
| 5 | КТ653Б | 1.0 | 1200 | 50 |
| 6 | КТ630В | 0.8 | 1000 | 150 |
| 7 | KT342a | 0.25 | 50 | 30 |
| 8 | KT503a | 0.35 | 150 | 25 |

По статическим характеристикам можно определить важнейшие параметры БТ. Статический коэффициент передачи тока базы определяется как отношение тока коллектора IК к току базы IБ  на пологих участках выходных ВАХ:

BIК/ IБ.

Коэффициент передачи по приращениям токов определяется как отношение приращения тока коллектора IК к приращению тока базы IБ на пологих участках выходных ВАХ:

IК/ IБ.

Дифференциальное входное сопротивление RВХ транзистора в схеме с общим эмиттером (ОЭ) определяется при фиксированном значении напряжения коллектор-эмиттер UКЭ >0. Оно может быть найдено как отношение приращения напряжения база-эмиттер к вызванному им приращению тока базы линейном участке водной ВАХ:

RВХ  = UБЭ/IБ = (UБЭ2 – UБЭ1)/(IБ2 –IБ1).

**2.2 Задание к выполнению работы**

2.2.1 Определить входные статические характеристики БТ.

2.2.2 Определить входные статические характеристики БТ.

2.2.3 Определить статические параметры БТ.

**2.3 Порядок выполнения работы**

2.3.1 Открыть проект \Elecrtonics\LR2\BT командой **File>Open\Project**.

2.3.2 Открыть в менеджере проекта схему измерения статических характеристик БТ ROOT: PAGE1. В поле Value задать тип транзистора.

2.3.3 Выполнить редактирование профиля моделирования командой **PSpice>Edit Simulation Profile** для расчета входных статических характеристик БТ.

Установить вид анализа (Analysis type) – DC Sweep.

Для опции Primary Sweep:

- задать Sweep Variable Current source Name: I1;

- задать закон изменения Sweep type: Linear и пределы изменения (Start Value =0, End Value = **500**u, Increment = **2**u).

Для опции Secondary Sweep:

- задать Sweep Variable Voltage source Name: V1;

- задать закон изменения Sweep type: Linear и пределы изменения (Start Value = 0, End Value = **10**, Increment = **10**).

На закладке Libraries в окне Filename выбрать библиотеку параметров модели транзистора и подключить ее нажатием кнопки Add to Design.

Закрыть окно редактирования нажатием кнопки OK.

2.3.4 Выполнить моделирование командой **PSpice>Run**. При этом выполняется запуск программы PSpice A/D.

2.3.5 Вывести на экран график зависимости тока коллектора, для этого командой **Trace>Add Trace** открыть окно Add Trace и выбрать сигнал Vb (Q1), подтвердить выбор нажатием клавиши ОК. Зарисовать график. Удалить график командой **Trace>Delete All Traces**.

2.3.6 По статическим входным ВАХ определить значение RВХ .

2.3.7 Выполнить редактирование профиля моделирования командой **PSpice>Edit Simulation Profile** для расчета выходных статических характеристик БТ.

Установить вид анализа (Analysis type) – DC Sweep.

Для опции Primary Sweep:

- задать Sweep Variable Voltage source Name: V1;

- задать закон изменения Sweep type: Linear и пределы изменения (Start Value = 0, End Value = **10**, Increment = **0.1**).

Для опции Secondary Sweep:

- задать Sweep Variable Current source Name: I1;

- задать закон изменения Sweep type: Linear и пределы изменения (Start Value = 0, End Value = **500u**, Increment = **100u**).

Закрыть окно редактирования нажатием кнопки OK.

2.3.8 Выполнить моделирование командой **PSpice>Run**. При этом выполняется запуск программы PSpice A/D.

2.3.9 Вывести на экран график зависимости тока коллектора, для этого командой **Trace>Add Trace** открыть окно Add Trace и выбрать сигнал IC(Q1), подтвердить выбор нажатием клавиши ОК. Зарисовать график. Удалить график командой **Trace>Delete All Traces**.

2.3.10 По статическим выходным характеристикам определить значения параметров Bи .

**2.4 Оформление работы**

Отчет по работе должен содержать:

- измерительную схему;

- статические входные и выходные характеристики БТ, определение основных статических параметров с необходимыми графическими построениями.

**2.5 Контрольные вопросы**

2.5.1 Как снимают статические характеристики БТ?

2.5.2 Отметьте на входных ВАХ участки, на которых транзистор работает в режимах отсечки, насыщения, активном.

2.5.3 Отметьте на выходных ВАХ участки, на которых транзистор работает в режимах отсечки, насыщения, активном.

2.5.4 Поясните как определяются основные параметры БТ?

**3 Расчет усилительного каскада переменного тока**

**3.1 Краткие сведения из теории**

Схема транзисторного усилительного каскада с общим эмиттером приведена на рисунке 3.1.

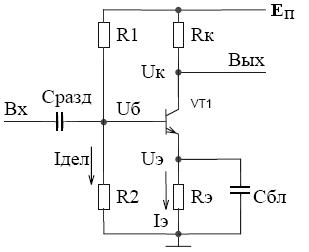


Рисунок 3.1

Схема содержит:

- транзистор VT1;

- сопротивление нагрузки Rк;

- сопротивление для стабилизации режима по постоянному току Rэ;

- делитель на резисторах R1, R2, задающий потенциал базы транзистора VT1;

- разделительный конденсатор Сразд, применяемый для разделения входной цепи транзистора и источника сигнала по постоянному току;

- блокировочный конденсатор Сбл, служащий для увеличения коэффициента усиления каскада в рабочей полосе частот.

Исходные данные для расчета приведены в таблице 3.1

Таблица 3.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вариант | Тип  транзистора | Предельная рассеиваемая мощность PК ДОП, Вт | Нижняя граничная частота Fн , Гц | Напряжение  питания  Eп, В |
| 1 | КТ312Б | 0.225 | 100 | 9 |
| 2 | КТ315Б | 0.15 | 200 | 6 |
| 3 | КТ3102А | 0.25 | 25 | 10 |
| 4 | КТ3117А | 0.5 | 50 | 12 |
| 5 | КТ653Б | 1.0 | 400 | 12 |
| 6 | КТ630В | 0.8 | 1000 | 15 |
| 7 | KT342a | 0.25 | 500 | 10 |
| 8 | KT503a | 0.35 | 150 | 15 |

**3.2 Методика расчета**

Усилительный каскад сначала рассчитывают по постоянному току

(статический режим). При этом виде расчета задаются положением точки покоя Uкэ0 , Iк0 на пологом участке выходной ВАХ транзистора. Рекомендуется выбирать

Uкэ0 ≈ (Eп - Uэ0)/2 + Uэ0 ,

Uэ0 = Iэ0Rэ = 0.1\*Eп.

При выборе Iк0  следует обеспечить симметричность положения рабочей точки покоя на нагрузочной прямой

Uкэ = Eп  - Iк (Rк + Rэ ) .

Также требуется обеспечить выполнение условия

P к0 = Uкэ0 \* Iк0 < Pк доп.

В результате расчета определяются сопротивления резисторов, обеспечивающих выбранные режимы:

Rэ = Uэ0 / Iэ0  ≈ Uэ0 / Iк0 ,

Rк = (Eп – Uкэ0)/Iк0  - Rэ,

Uб0 ≈ Iэ0Rэ + 0.7 В,

Iдел ≈ 0.1\*Iк0,

R1 = (Eп – Uб0)/Iдел,

R2 = Uб0/Iдел.

При расчете параметров элементов усилителя по переменному току обычно полагают, что в схеме выполняется условие

1/ ϖCбл << Rэ .

Т.е., в рабочей полосе частот емкостное сопротивление блокировочного конденсатора Сбл много меньше сопротивления резистора Rэ

Входное сопротивление транзистора в рабочей полосе частот равно

Rвх тр = β rэ + rб ,

где rэ = mϕT/Iэ0 – дифференциальное сопротивление прямосмещенного эмиттерного перехода,

ϕT=26 мВ – температурный потенциал,

m – коэффициент, характеризующий качество эмиттерного перехода (m=1.05 – 1.3).

Для определения входного сопротивления Rвх каскада необходимо учесть параллельное включение сопротивлений Rвх тр, R1 и R2. Емкость конденсатора Сбл выбирается из условия обеспечения нижней граничной частоты:

Cбл ≈ 1/(0.82 π Fн r э).

Емкость разделительного конденсатора также выбирается из условия

обеспечения нижней граничной частоты:

Cразд ≈ 1/(0.82 π Fн Rвх).

**3.3 Содержание отчета**

1) Схема усилительного каскада.

2) График выходных характеристик с указанием рабочей точки покоя и построением нагрузочной прямой.

3) Расчетные формулы.

4) Результаты расчета параметров.

**4 Характеристики усилительного каскада переменного тока**

**4.1 Краткие сведения из теории**

Цель работы - получить практические навыки исследования усилительного каскада по схеме с общим эмиттером.

Программа PSpice позволяет определить основные характеристики усилительного каскада, в том числе:

- режим по постоянному току в рабочей точке покоя (Bias Point);

- частотные характеристики (AC Sweep);

- переходные процессы при воздействии сигналов различной формы (Transient Analysis).

Расчет статического режима рабочей точки всегда предшествует другим видам расчетов. Для расчета частотных характеристик пользователю необходимо указать интересующие его частотный диапазон и масштаб изменения частоты. Все независимые источники напряжения V и тока I, для которых заданы параметры АС являются входными воздействиями. Остальные спецификации этих источников, в том числе параметры синусоидального сигнала SIN, не принимаются во внимание, они учитываются при анализе переходных процессов. Результаты анализа выводится на экран в виде графиков амплитудно-частотных и фазо-частотных характеристик (АЧХ и ФЧХ).

Анализ переходных процессов проводится при воздействии на входы схемы источников сигналов (гармонического, импульсного и других). Для анализа во временной области пользователю необходимо задать информацию о параметрах источников входных сигналов, временной интервал расчета. Результаты расчета выводятся на экран в виде временных диаграмм сигналов.

**4.2 Описание последовательности выполнения задания**

4.2.1 Открыть проект \Elecrtonics\lr6 командой **File>Open>Project**.

4.2.2 Открыть в менеджере проекта схему SCHEMATIC1: PAGE1.

4.2.3 Выполнить редактирование параметров элементов схемы (в соответствии с вариантом задания и по результатам расчета).

4.2.4 Выполнить редактирование профиля моделирования командой **PSpice>Edit Simulation Profile** для расчета рабочей точке покоя и частотных характеристик усилительного каскада, для этого на закладке Analysis:

- установить вид анализа (Analysis type) – AC Sweep/Noise;

- задать закон изменения AC Sweep Type: Logarithmic и пределы изменения частоты (Start Frequency, End Frequency, Points/Decade = 10);

* закрыть окно редактирования нажатием кнопки OK.

Значение Start Frequency задать равным значению заданного параметра Fн , ориентировочное значение End Frequency задать 100Meg (возможно потребуется уточнить для определения верхней границы полосы пропускания).

Выполнить моделирование командой **PSpice>Run**.

4.2.5 Активизировать окно программы OrCAD Capture. В меню **PSpice>Bias Points** установить флаги Enable Bias Current Display и Enable Bias Voltage Display. В окне SCHEMATIC1: PAGE1 отображаются результаты расчета рабочей точке покоя узловые напряжения и токи ветвей). Оцените по ним соответствие выбранному положению рабочей точке покоя на выходных ВАХ транзистора (Vкэ, Iк). В случае необходимости скорректируйте положение рабочей точке покоя подбором сопротивления R1. Занести результаты расчета рабочей точке покоя в отчет.

4.2.6 Активизировать окно программы PSpice AD. Вывести на экран график АЧХ, для этого командой **Trace>Add Trace** открыть окно Add Trace и в окне Trace Expression задать значение DB(V(Out)/V(In)), подтвердить выбор нажатием клавиши ОК. Зарисовать график АЧХ. Определить граничные частоты полосы пропускания по уровню минус 3 дБ, для этого включить электронный курсор (команда **Trace>Toggle Cursor**). Записать определенные значения. Удалить график командой **Trace>Delete All Traces**.

4.2.7 Вывести на экран график ФЧХ, для этого командой **Trace>Add Trace** открыть окно Add Trace и в окне Trace Expression задать значение P(V(Out)), подтвердить выбор нажатием клавиши ОК. Зарисовать график ФЧХ. Удалить график командой **Trace>Delete All Traces**.

4.2.8 Активизировать окно программы OrCAD Capture. Выполнить редактирование значений параметров VOFF=0, VAML=1mV, FREQ источника входного сигнала Vg. Задать значение FREQ в пределах полосы пропускания.

4.2.9 Выполнить редактирование профиля моделирования командой **PSpice>Edit Simulation Profile** для расчета во временной области, для этого на закладке Analysis:

- задать вид анализа (Analysis type) – Time Domain (Transient);

- задать значение интервала расчета Run to Time равным пяти периодам входного гармонического сигнала;

- закрыть окно редактирования нажатием кнопки OK.

Выполнить моделирование во временной области командой **PSpice>Run**.

4.2.10 Вывести на экран график выходного сигнала V(Out), для этого командой **Trace>Add Trace** открыть окно Add Trace и выбрать сигналы V(In) и V(Out), подтвердить выбор нажатием клавиши ОК. Зарисовать графики. Представить результаты преподавателю. Удалить график командой **Trace>Delete All Traces**.

Определить значение коэффициента усиления по напряжению на заданной частоте FREQ. Сравнить со значением коэффициента усиления, полученным по АЧХ.

**4.3 Оформление отчета**

Отчет должен содержать:

1) результаты расчета статического режима;

2) графики АЧХ и ФЧХ;

3) временную диаграмму выходного сигнала;

4) найденные значения параметров.

**4.4 Контрольные вопросы**

6.4.1 Поясните назначение основных элементов схемы усилительного каскада.

6.4.2 Поясните принцип работы усилительного каскада.

6.4.3 Каким образом можно определить положение рабочей точки покоя?

6.3.4 Каким образом можно определить частотные характеристики? Как по ним можно найти границы полосы пропускания?

6.4.5 Как можно определить значение коэффициента усиления по напряжению?

**5 Операционные схемы**

**5.1 Краткие сведения из теории**

Цель работы - получить практические навыки исследования операционных схем. На рисунке 5.1 приведена схема неинвертирующего звена – одной из распространенных операционных схем.

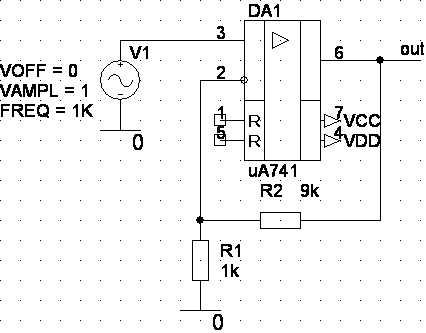


Рисунок 5.1

Допуская, что операционный усилитель идеальный, коэффициент передачи схемы по напряжению

K = 1 + R2/R1.

Варианты заданий для расчета параметров элементов схемы:

1) K = 5;

2) K = 15;

3) K = 20;

4) K = 30;

5) K = 25;

6) K = 50;

7) К = 2;

8) К = 35.

**5.2 Описание последовательности выполнения задания**

5.2.1 Открыть проект \Elecrtonics\OA командой **File>Open>Project**.

5.2.2 Открыть в менеджере проекта схему SCHEMATIC1: PAGE1.

5.2.3 Выполнить редактирование параметров элементов схемы.

5.2.4 Выполнить редактирование профиля моделирования командой **PSpice>Edit Simulation Profile** для расчета передаточной характеристики, для этого на закладке Analysis:

- установить вид анализа (Analysis type) – DC Sweep;

- для опции Primary Sweep задать Sweep Variable Voltage source Name: V1,

задать закон изменения Sweep type: Linear и пределы изменения (Start Value = -10, End Value = 10, Increment = 0.5).

Закрыть окно редактирования нажатием кнопки OK. Выполнить моделирование командой **PSpice>Run**.

5.2.5 Вывести на экран график передаточной характеристики, для этого командой **Trace>Add Trace** открыть окно Add Trace и в окне Trace Expression задать значение V(Out), подтвердить выбор нажатием клавиши ОК. Зарисовать график. Определить коэффициент передачи по напряжению (по наклону линейного участка). Удалить график командой **Trace>Delete All Traces**.

5.2.6 Выполнить редактирование профиля моделирования командой **PSpice>Edit Simulation Profile** для расчета частотных характеристик, для этого на закладке Analysis:

- установить вид анализа (Analysis type) – AC Sweep/Noise;

- задать закон изменения AC Sweep Type: Logarithmic и пределы изменения частоты (Start Frequency =1, End Frequency , Points/Decade = 10).

Закрыть окно редактирования нажатием кнопки OK. Выполнить моделирование командой **PSpice>Run**.

5.2.7 Вывести на экран график АЧХ, для этого командой **Trace>Add Trace** открыть окно Add Trace и в окне Trace Expression задать значение DB(V(Out)), подтвердить выбор нажатием клавиши ОК. Зарисовать график АЧХ. Определить граничную частоту полосы пропускания по уровню минус 3 дБ, для этого включить электронный курсор (команда **Trace>Toggle Cursor**). Удалить график командой **Trace>Delete All Traces**.

5.2.8 Вывести на экран график ФЧХ, для этого командой **Trace>Add Trace** открыть окно Add Trace и в окне Trace Expression задать значение P(V(Out)), подтвердить выбор нажатием клавиши ОК. Зарисовать график ФЧХ. Удалить график командой **Trace>Delete All Traces**.

5.2.9 Активизировать окно программы OrCAD Capture. Выполнить редактирование значений параметров VOFF=0, VAML, FREQ источника входного сигнала V1.

5.2.10 Выполнить редактирование профиля моделирования командой **PSpice>Edit Simulation Profile** для расчета во временной области, для этого на закладке Analysis:

- задать вид анализа (Analysis type) – Time Domain (Transient);

- задать значение интервала расчета Run to Time на протяжении 10 периодов;

- закрыть окно редактирования нажатием кнопки OK.

5.2.11 Выполнить моделирование во временной области командой **PSpice>Run**.

5.2.12 Вывести на экран график выходного сигнала V(Out), для этого командой **Trace>Add Trace** открыть окно Add Trace и выбрать сигнал V(Out), подтвердить выбор нажатием клавиши ОК. Зарисовать график. Представить результаты преподавателю. Удалить график командой **Trace>Delete All Traces**.

**5.3 Оформление отчета**

Отчет должен содержать:

1) график передаточной характеристики;

2) графики АЧХ и ФЧХ;

3) временную диаграмму выходного сигнала.

**5.4 Контрольные вопросы**

5.4.1 Перечислите известные Вам операционные схемы.

5.4.2 Что такое операционный усилитель?

5.4.3 От чего зависит коэффициент передачи масштабирующего звена?

**6 Функциональное проектирование фильтра**

**6.1Краткие сведения из теории**

Требуется построить структурную схему частотно-избирательного фильтра, для которого заданы требования к форме АЧХ. Для решения задачи метод каскадной реализации, который предусматривает:

- нахождение передаточной функции фильтра в операторной форме;

- разложение передаточной функции фильтра на сомножители второго и первого порядка;

- реализацию полученного разложения каскадным соединением звеньев вто­рого и первого порядка.

Сначала необходимо решить задачу аппроксимации заданной АЧХ фильтра. В ходе решения этой задачи определяют передаточную функцию фильтра в области комплексной частоты s в виде произведения сомножителей

k

H(s) = П Нi(s),

i=1

где k = n/2 для четных значений n, k = (n-1)/2 +1 для нечетных n.

Для фильтра нижних частот сомножители могут быть первого порядка

wi

Hi(s) = Hoi ———

s + wi

или второго порядка

wi2

Hi(s) = Hoi ——————— .

s2 + (wi/Qi)s + wi2

Для фильтра верхних частот сомножители могут быть первого порядка

s

Hi(s) = Hoi ———

s + wi

или второго порядка

s2

H(s) = Hoi ——————— .

s2 + (wi/Qi)s + wi2

Для полосового фильтра сомножители могут быть второго порядка

(wi /Qi)s

H(s) = Hoi ———————— .

s2 + (wi/Qi)s + wi2

Параметры передаточной функции:

- n – порядок фильтра;

- wi - частота пары комплексно-сопряженных полюсов или вещественного полюса;

- Qi - добротность пары комплексно-сопряженных полюсов

можно определить с помощью программы аппроксимации АЧХ.

В каче­стве элементов структурной схемы рекомендуется использовать элементы LAPLACE из библиотеки abm.olb.

Для установления соответствия характеристик проектируемого устройства требованиям технического задания необходимо выполнить анализ АЧХ фильтра в соответствующих диапазонах частот.

**6.2 Порядок выполнения задания**

6.2.1 Выполнить аппроксимацию АЧХ фильтра.

6.2.2 Построить структурную схему фильтра, задать значения параметров ее элементов.

6.2.3 Рассчитать АЧХ фильтра, сравнить с требованиями задания.

**6.3 Содержание отчета**

Отчет должен содержать:

- передаточную функцию и ее параметры;

- схему структурную фильтра;

- АЧХ фильтра, выводы.

**7 Схемотехническое проектирование фильтра**

**7.1Краткие сведения из теории**

Типовые схемы звеньев первого и второго порядка для ФНЧ, ФВЧ и ПФ хранятся в банке схемных решений (БСР). Поиск необходимой схемы выполняется по имени файла. Имя файла схемы состоит из полей:

- тип фильтра (lp - ФНЧ, hp - ФВЧ, bp - ПФ);

- порядок звена (1 или 2);

- добротность звена второго порядка (lq - низкая, mq - средняя, hq - высокая) или вариант реализации звена первого порядка (i - инвертирующее, n - неинвертирующее).

Для определения значений параметров элементов можно использовать инженерные методики расчета, реализованные на языке java script. Методики сохранены в форме html файлов с именами, совпадающими с именами схемных файлов звеньев.

Правильность выполнения задания контролируется расчетом АЧХ звена.

**7.2 Порядок выполнения задания**

7.2.1 Определить схемную реализацию звена фильтра.

7.2.2 Рассчитать значения параметров элементов звена.

7.2.3 Рассчитать АЧХ звена, сравнить с требованиями задания.

**7.3 Содержание отчета**

Отчет должен содержать:

- функциональную схему фильтра;

- расчет параметров элементов;

- АЧХ звена, выводы.

**8 Логические функции**

**8.1 Краткие сведения из теории**

Рассматривается задача реализации переключательной функции (ПФ) *y = f (x1, x2 , … , xn )*, то есть построения схемы, имеющей n-входов *x1, x2 , … , xn*и выход *y*, функционирующей в соответствии с функцией *f*.

Для снижения затрат оборудования при реализации ПФ используют их минимальные формы записи. Решение задачи синтеза КС сводится к выполнению следующих шагов:

- минимизация ПФ (построение МДНФ);

- преобразование нормальных форм к виду, удобному для реализации, то есть к суперпозиции функций из элементного базиса;

- построение схемы по формуле ПФ, то есть графа вычислений ПФ, вершинами которого являются соответствующие ЛЭ.

*Пример 1* **-** Синтезировать КС, реализующую ПФ, которая задана картой Карно (рисунок 8.1).

x1 x2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x3x4 | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | 1 | 1 |
| 01 | 1 | 1 | 1 | 1 |
| 11 | 0 | 0 | 0 | 0 |
| 10 | 1 | 0 | 1 | 0 |

Рисунок 8.1

В результате минимизации получим МДНФ для ПФ

*y =‾x1 ⋅‾x2 ⋅‾x4 + x1 ⋅ x2 ⋅‾x4 +‾x3 ,*

Варианты заданий приведены в таблице 8.1

|  |  |
| --- | --- |
| Таблица 8.1 | |
| Вариант | Переключательная функция |
| 1 | f(x1,x2,x3,x4) = (0, 1, 2, 4, 5, 6, 10, 12) |
| 2 | f(x1,x2,x3,x4)= (0, 2, 3, 8, 9, 10, 11, 14, 15) |
| 3 | f(x1,x2,x3,x4) = (0, 2, 4, 6, 7, 9, 12, 15) |
| 4 | f(x1,x2,x3,x4) = (0, 1, 2, 3, 8, 10, 11, 12, 13, 15) |
| 5 | f(x1,x2,x3,x4) = (1, 3, 5, 6, 7, 8, 10, 14, 15) |
| 6 | f(x1,x2,x3,x4) = (3, 4, 5, 9, 10, 12, 14, 15) |
| 7 | f(x1,x2,x3,x4) = (2, 4, 5, 6, 8, 9, 10, 14) |
| 8 | f(x1,x2,x3,x4) = (1, 4, 8, 9, 10, 12, 13, 14, 15) |
| 9 | f(x1,x2,x3,x4) = (1, 2, 4, 8, 9, 10, 12, 13, 15) |
| 10 | f(x1,x2,x3,x4) = (0, 4, 5, 6, 7, 8, 9, 13, 15) |
| 11 | f(x1,x2,x3,x4) = (4, 6, 7, 9, 10, 11, 12, 14, 15) |
| 12 | f(x1,x2,x3,x4) = (3, 4, 5, 6, 9, 10, 12, 13, 14) |
| 13 | f(x1,x2,x3,x4) = (2, 6, 7, 8, 11, 12, 13, 14, 15) |
| 14 | f(x1,x2,x3,x4) = (1, 2, 3, 4, 7, 9, 10, 11, 14, 15) |
| 15 | f(x1,x2,x3,x4) = (1, 4, 7, 8, 9, 10, 12, 13, 14, 15) |
| 16 | f(x1,x2,x3,x4) = (3, 5, 6, 7, 8, 9, 10, 12, 13) |
| 17 | f(x1,x2,x3,x4) = (4, 8, 9, 10, 11, 12, 14, 15) |
| 18 | f(x1,x2,x3,x4) = (2, 3, 4, 6, 7, 8, 9, 11, 12) |

**8.2 Оформление работы**

Отчет по работе должен содержать:

- таблицу истинности заданной функций;

- карту Карно, результаты минимизации.

**9 Синтез комбинационных узлов**

**9.1 Краткие сведения из теории**

Решение задачи синтеза КС сводится к выполнению следующих шагов:

- преобразование нормальных форм к виду, удобному для реализации, то есть к суперпозиции функций из элементного базиса;

- построение схемы по формуле ПФ, то есть графа вычислений ПФ, вершинами которого являются соответствующие ЛЭ.

*Пример 1* **-** Синтезировать КС, реализующую ПФ, которая задана картой Карно (рисунок 9.1).

При подаче на входы комбинации значений *x1, x2 , … , xn*, соответствующей входному набору ПФ, схема формирует на выходе значение сигнала *y*, соответствующее значению ПФ на этом наборе. Таким образом, рассматриваемые схемы, которые называют логическими схемами (ЛС) или комбинационными схемами (КС) с одним выходом, реализуют ПФ.

КС строят из логических элементов (ЛЭ), каждый из которых реализует элементарную ПФ – логическую операцию. Набор типов ЛЭ обычно ограничивают. Разрешенный набор типов ЛЭ называют базисом, а соответствующие типы - базовыми. Задача реализации ПФ является решаемой, если базис является функционально полным. На занятии рассматриваются способы построения КС в базисах элементов:

* И, ИЛИ, НЕ;
* И-НЕ.

Быстродействие КС оценивается задержкой распространения сигнала *tзр* , то есть временным интервалом, на который задержаны изменения значений выходного сигнала относительно вызвавших их изменений одного из входных сигналов. В общем случае значения задержек по разным входам могут быть разными, поэтому быстродействие оценивается по наихудшему случаю

*tзр = max tзр i ,*

*i=1,n*

где *tзр i* - задержка распространения по i–му входу, равная сумме задержек ЛЭ, лежащих на самом длинном пути от i–го входа до выхода схемы. Чтобы упростить определение быстродействия КС обычно считают, что все ЛЭ имеют одинаковые задержки *tлэ*. Тогда задержки можно определять в относительных единицах *tзр / tлэ*.

Она может быть использована для построения КС в базисе Буля (элементов И, ИЛИ, НЕ). На рисунке 9.1 приведен результат синтеза для МДНФ функции.

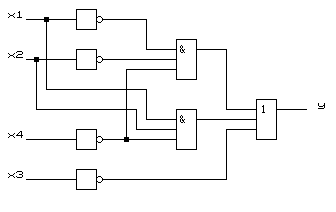


Рисунок 9.1

Для реализации ПФ в базисе И-НЕ используют МДНФ. Выполняют ее двойную инверсии и преобразование Де-Моргана

*y =‾x1 ⋅‾x2 ⋅‾x4 ⋅ x1 ⋅ x2 ⋅‾x4 ⋅ x3 ,*

Схема представлена на рисунке 9.2.

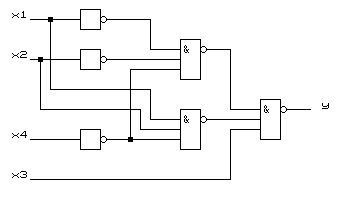


Рисунок 9.2

**9.2 Оформление работы**

Отчет по работе должен содержать:

- заданную функцию;

- результаты эквивалентных преобразований;

- построенные схемы.

**10 Синтез счетчиков**

**10.1 Задание**

На триггерах заданного типа построить синхронный счетчик, с поступлением каждого синхроимпульса выдающий набор сигналов, циклическая последовательность которых определяется вариантом задания. Исходные данные приведены в таблице 10.1.

Таблица 10.1

|  |  |  |
| --- | --- | --- |
| Вариант | Последовательность состояний | Тип триггера |
| 1 | 0 2 3 4 5 6 7 | RS |
| 2 | 0 2 3 4 5 6 7 | JK |
| 3 | 0 1 2 4 5 6 7 | D |
| 4 | 0 1 2 3 5 6 7 | JK |
| 5 | 0 1 2 3 4 6 7 | RS |
| 6 | 0 1 2 3 4 5 7 | D |
| 7 | 0 1 2 3 4 5 7 | RS |
| 8 | 1 2 3 4 5 6 7 | D |
| 9 | 0 1 2 3 4 5 6 7 | JK |
| 10 | 0 2 1 3 4 5 6 7 | D |
| 11 | 0 1 3 2 4 5 6 7 | D |
| 12 | 0 1 2 4 3 5 6 7 | RS |
| 13 | 0 1 2 3 5 4 6 7 | RS |
| 14 | 0 1 2 3 4 6 5 7 | RS |
| 15 | 0 1 2 3 4 5 7 6 | D |
| 16 | 0 1 2 3 4 5 | JK |
| 17 | 1 2 3 4 5 6 | D |
| 18 | 2 3 4 5 6 7 | JK |
| 19 | 0 3 4 5 6 7 | D |
| 20 | 0 1 4 5 6 7 | JK |

Содержание отчета:

- задание;

- совмещенная таблица состояний и функций возбуждения;

- карты Карно;

- аналитические представления функции возбуждения триггеров;

- функциональная схема счетчика.

**10.2 Порядок выполнения**

Синтез основан на использовании обратной таблицы переходов триггеров. По этой таблице можно определить, какие сигналы необходимо подать на управляющие входы, чтобы перевести триггер из одного состояния Qn-1 в другое Qn.

Таблица обратных переходов для JK -, SR - и D -триггеров

|  |  |  |  |
| --- | --- | --- | --- |
| Qn-1 Qn | J K | S R | D |
| 0 0 | 0 х | 0 х | 0 |
| 0 1 | 1 х | 1 0 | 1 |
| 1 0 | х 1 | 0 1 | 0 |
| 1 1 | х 0 | х 0 | 1 |

Для синтеза счётчика необходимо построить совмещённую таблицу состояний счётчика и функций возбуждения триггеров. Проследим этот процесс на примере синтеза синхронного счётчика по модулю 5 на JK-триггерах. Циклическая последовательность наборов выходных сигналов будет иметь вид: 0 – 1 – 2 – 3 – 4. Для кодирования 5 состояний счетчика будет достаточно трех разрядов (23 ≥ 5). Символ «х» представляет произвольное значение сигнала.

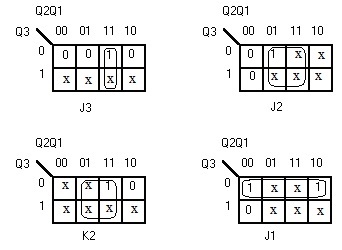
Совмещенная таблица

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N состояния | Q3Q2Q1 | J3K3 | J2K2 | J1K1 |
| 0 | 0 0 0 | 0 х | 0 х | 1 х |
| 1 | 0 0 1 | 0 х | 1 х | х 1 |
| 2 | 0 1 0 | 0 х | х 0 | 1 х |
| 3 | 0 1 1 | 1 х | х 1 | 0 х |
| 4 | 1 0 0 | х 1 | 0 х | 0 х |

При переходе из состояния 000 в состояние 001 триггер 1-го разряда переходит из состояния 0 в состояние 1. Для осуществления этого перехода по таблице обратных переходов для JK-триггера определяем что должны быть поданы J1=1, K1=х, т.е. уровень сигнала К1 не влияет на выполнение перехода из 0 в 1. Для 2-го и 3-го разрядов переход из состояния 000 в состояние 001 требует сохранить для триггеров 2-го и 3-го разрядов состояние 0, что соответствует наборам J2K2 =J3K3 = 0х. Аналогично заполняются строки таблицы для других состояний.

Для перехода из состояния 100 в начальное состояние 000 необходимо, чтобы 1-ый и 2-ой триггеры сохранили на своих выходах 0, а 3-ий перешёл из состояния 1 в состояние 0. Для этого необходимо обеспечить J1K1 = J2K2 = 0х, a J3K3 = х1.

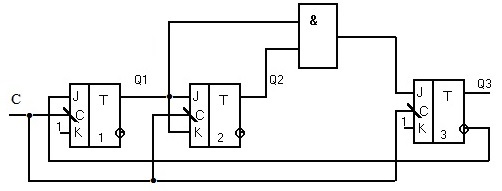
Далее совмещённая таблица интерпретируется как таблица истинности для функций возбуждения J1,K1 ,J2,K2,J3, K3, зависящих от переменных Q3Q2Q1. Для синтеза счётчика необходимо получить их аналитическое представление. Очевидно, что принимая для K3 и К1 вместо «х» значение «1», получим K3 = К1 = 1. Для остальных функций можно использовать карты Карно. Так как наборы переменных Q3Q2Q1 : 110 и 111 для данного счетчика не используются, то для функций возбуждения можно использовать безразличное значение «х», доопределяя его произвольно до «0» или «1» при поиске склеек.



После минимизации получаем ДНФ функций:

*J3 = Q2 · Q1, J2 =K2 = Q1, J1 =‾ Q3,*

В соответствии с этим результатом можно построить функциональную схему счётчика на JK-триггерах, синхронизируемых фронтом.



Для построения счётчиков на триггерах других типов (обязательно синхронизируемых фронтом) нужно использовать соответствующие столбцы таблицы обратных переходов.

**11 Алгоритмизация и кодирование алгоритма**

**11.1 Описание задания**

Необходимо разработать и кодировать на языке Ассемблера микроконтроллера (МК) PIC-micro алгоритм программы генератора чисел. Числовая последовательность формируется на линиях порта PORTD в соответствии с вариантом задания по таблице 11.1.

Таблица 11.1

|  |  |
| --- | --- |
| Вариант | Задание |
| 1 | Формирователь последовательности чисел Фибоначчи 0, 1,…,233 |
| 2 | Вычитающий двоичный счетчик по модулю 16 |
| 3 | Суммирующий восьмиразрядный счетчик в унитарном коде |
| 4 | Вычитающий восьмиразрядный счетчик в унитарном коде |
| 5 | Суммирующий восьмиразрядный счетчик в коде Джонсона |
| 6 | Вычитающий восьмиразрядный счетчик в коде Джонсона |

**1.2 Рекомендации по выполнению**

***Алгоритм*** – точный набор инструкций, описывающих последовательность действий некоторого исполнителя для достижения результата, решения некоторой задачи за конечное время. Если исполнителем является процессорное ядро МК, то в конечном итоге набор инструкций необходимо представить в виде последовательности машинных команд – загрузочного кода программы. Кодированное представление алгоритма получают путем последовательного преобразования исходного описания, отличающегося меньшей степенью детализации и представленного в форме, удобной для восприятия человеком.

Исходное представление алгоритм программы обычно формируют по результатам анализа ее функциональной спецификации. По заданию основные функции программы:

- модификация содержимого ячейки Cnt (в соответствии с вариантом задания);

- вывод содержимого ячейки Cnt в порт PORTD.

На рисунке 11.1 приведен пример - схема программы, реализующей алгоритм формирования на линиях порта PORTD числовой последовательности по закону суммирующего десятичного счетчика. При старте программа выполняет инициализацию порта вывода, инициализацию ячейки счетчика Cnt. После чего в теле цикла программы выполняется модификация (инкремент) содержимого ячейки Cnt и вывод ее содержимого в порт PORTD. Если содержимое счетчика превышает допустимое значение, определяемое модулем счета, счетчик обнуляется.

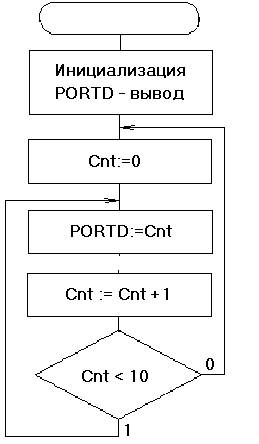


Рисунок 11.1

Теперь перейдем к кодированию алгоритма на языке Ассемблера - формированию исходного текста программы. Текст программы состоит из следующих разделов:

- директивы листинга, определения модели МК;

- директивы определения переменных;

- директивы указания вектора старта;

- операторы программы;

- директива окончания программы.

В программе используется одна переменная Cnt, выделим для нее первую свободную ячейку в банке 0, используя директиву EQU

Cnt EQU 0x20 ; счетчик

Последовательность операторов программы должна соответствовать алгоритму. При старте программы должны выполняться операторы инициализации, а затем операторы основного цикла.

Для настройки линий PORTD на вывод в регистр TRISD необходимо загрузить управляющее слово b'00000000' или очистить регистр TRISD. Регистр TRISD находится в банке 1, поэтому перед использованием оператора очистки регистра TRISD необходимо переключить соответствующие разряды (RP1,RP0) регистра STATUS. После очистки регистра TRISD необходимо вернуть исходное состояние измененных разрядов (RP0) регистра STATUS. Для наглядности вместо инструкций установки и сброса разрядов регистра STATUS можно использовать директивы

BANKSEL TRISD ; банк 1 (адреса 80h - FFh),

BANKSEL PORTD ; банк 0 (адреса 00h - 7Fh).

При трансляции эти директивы будут заменены соответствующими инструкциями.

Особенностью системы команд МК PIC-micro является отсутствие команд сравнения и условных переходов. Поэтому эти действия необходимо выполнить, используя реализованные операции системы команд.

Сравнение слов информации обычно реализуют путем вычитания без сохранения разности. Результат сравнения можно оценить по значениям признаков выполнения операции (флагов): C – признака переноса и Z – признака нулевого результата. Вычитание константы можно заменить сложением с константой противоположного знака. В результате сложения:

movlw -.10 ; число -10

addwf cnt,w ; прибавить к содержимому счетчика

флаг С установится, если условие cnt < 10 не выполняется. Ветвление алгоритма можно реализовать с помощью команд:

btfss STATUS,C ; если С=1

goto inc ; пропустить, иначе выполнить

goto loop ;.

Если С=0 (cnt<10), будет выполнен переход на метку Inc, иначе переход на метку Loop.

Сформированный исходный текст программы приведен ниже

list p=16f877a,st = OFF ; директива листинга

#include <p16f877a.inc> ; определение переменных МК

\_\_CONFIG \_CP\_OFF & \_WDT\_OFF & \_BODEN\_OFF & \_PWRTE\_ON & \_HS\_OSC & \_WRT\_OFF& \_LVP\_OFF & \_DEBUG\_ON & \_CPD\_OFF

; директива установки разрядов слова конфигурации

;\*\*\*\*\* Определение переменных

Cnt EQU 0x20 ; счетчик

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ORG 0x000 ; вектор старта

Main nop

BANKSEL TRISD ; банк 1 (адреса 80h - FFh)

clrf TRISD^80 ; все линии PORTD на вывод

BANKSEL PORTD ; банк 0 (адреса 00h - 7Fh)

Loop clrf Cnt ; обнулить счетчик

Inc movf Cnt,w ;

movwf PORTD

incf Cnt,f ; инкремент счетчика

movlw -.10 ; число -10

addwf Cnt,w ; прибавить к содержимому счетчика

btfss STATUS,C ; для сравнения

goto Inc ;

goto Loop

END ; директива окончания программы.

**12 Дискретный ввод-вывод**

**12.1 Описание задания**

Необходимо разработать и кодировать на языке Ассемблера алгоритм программы дискретного ввода-вывода для МК типа PIC16F877A.

Нажатие с последующим отпусканием кнопки RC1 должно вызывать модификацию содержимого ячейки Cnt. Содержимое ячейки Cnt должно отображаться на светоизлучающих диодах, подключенных к линиям порта PORTD. Варианты заданий приведены в таблице 11.1.

**12.2 Рекомендации по выполнению**

Функции программы:

- опрос кнопки, подключенной к линии порта PORTC (RC1);

- модификация содержимого ячейки Cnt;

- вывод содержимого ячейки Cnt в порт PORTD.

На рисунке 12.1 приведен пример - схема программы, реализующая алгоритм счетчика по модулю 10 числа нажатий на кнопку RC1.

При старте программы выполняется инициализация портов ввода и вывода, инициализация ячейки счетчика Cnt. После чего в теле цикла программы выполняется вывод содержимого ячейки Cnt в порт PORTD. Состояние линий порта PORTC копируется в ячейку PCh0.

Активному (нажатому) состоянию кнопки соответствует низкий потенциал на линии PORTC. Пассивному (отжатому) состоянию кнопки соответствует высокий потенциал, который обеспечивается подтягивающими резисторами. Все механические кнопки имеют негативное свойство, известное как «дребезг» контактов, которое обусловлено колебаниями упругих контактов при их замыкании и размыкании. Длительность колебаний составляет несколько миллисекунд. В устройствах на базе МК обычно используют программные способы подавления «дребезга» контактов. Простейший из них основан на ограничении минимальных интервалов времени между последовательными операциями анализа состояния входных линий порта. Для надежного считывания состояний кнопки величина этих интервалов не должна быть меньше 5 – 10 мс. Для подавления «дребезга» контактов в цикл основной программы включен оператор вызова подпрограммы задержки D5ms.

По комбинации значений 0 – 1– 1 соответствующего разряда в ячейках PCh0, PCh1, PCh2 (копии PORTC в текущем и двух предыдущих циклах программы) определяется активность линии входного порта. Для этого вычисляется значение переменной

*Key = ͞P͞C͞͞h͞0 & PCh1 & PCh2*

и анализируется состояние ее разряда Key,1. В случае если Key,1 = 1 (кнопка нажата), изменяется содержимое ячейки Cnt.

При вызове подпрограммы задержки D5ms происходит инициализация счетчика миллисекунд Count\_ms. Подпрограмма выполняется, пока Count\_ms не достигнет значения 0. Декремент переменной Count\_ms через одну миллисекунду выполняет подпрограмма обработки прерываний INT.

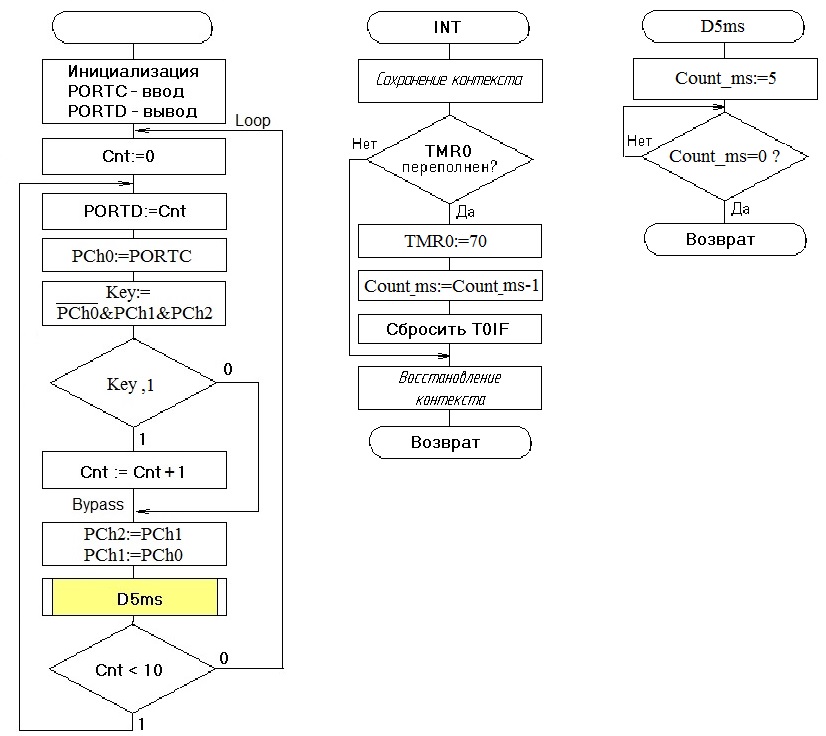


Рисунок 12.1

Режим работы таймера определяется состоянием разрядов регистра OPTION\_REG (таблица 12.1).

Если бит T0CS=0, то TMR0 работает от внутреннего тактового генератора, инкремент счетчика происходит в каждом машинном цикле. На входе TMR0 может быть включен предделитель частоты, если бит PSA=0. Коэффициент деления определяется значениями разрядов PS2:PS0. Значение коэффициента деления частоты 1:16 получается при <PS2:PS0> = 011. Для настройки TMR0 на данный режим в регистр OPTION\_REG необходимо загрузить управляющее слово b'10000011'.

Таблица 12.1



При переполнении таймера устанавливается флаг T0IF (регистр INTCON). После анализа флага, его необходимо программно сбросить и выполнить загрузку константы в таймер TMR0. При загрузке константы 70 при тактовой частоте МК F=12 МГц интервал времени между двумя переполнениями таймера составит:

ТTMR = 4⋅24⋅186/F = 4⋅16⋅(256-70)/12⋅106 = 0.992 мс.

При возникновении прерывания МК выполняет текущую команду, запоминает адрес следующей команды (точки возврата) в стеке и передает управление подпрограмме обработки прерывания. После обработки прерывания из стека извлекается адрес точки возврата, и управление передается прерванному процессу. Это позволяет МК своевременно реагировать на различные события, при условии, что обработка прерываний занимает не слишком много времени. При возникновении нескольких прерываний от различных источников, их обработка выполняется последовательно в соответствии с приоритетами.

Для данной задачи прерывания будут возникать при переполнении таймера TMR0. Микроконтроллеры семейства PIC16F87x имеют 14 источников прерываний. Регистр INTCON содержит флаги отдельных прерываний, биты разрешения этих прерываний и бит глобального разрешения прерываний GIE.

Если бит GIE (INTCON,7) установлен в ‘1’, разрешены все немаскированные прерывания. Если GIE=0, то все прерывания запрещены. Каждое прерывание в отдельности может быть разрешено/запрещено установкой/сбросом соответствующего бита разрешения. При сбросе микроконтроллера бит GIE сбрасывается в ‘0’.

В регистре INTCON находятся флаги прерываний: от внешнего сигнала INT, от изменения уровня сигнала на входах RB7:RB4, от переполнения TMR0 (таблица 12.2). В регистрах PIR1, PIR2 содержатся флаги прерываний периферийных модулей микроконтроллера, а в регистрах PIE1, PIE2 соответствующие биты разрешения прерываний. В регистре INTCON находится бит разрешения прерываний от периферийных модулей PEIE.

Таблица 12.2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Адрес | Имя | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 |
| 0Bh,8Bh,  10Bh,18Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF |
| 0Ch | PIR1 | PSPIF | ADIF | RCIF | TXIF | SSPIF | CCP1F | TMR2IF | TMR1IF |
| 8Ch | PIE1 | PSPIE | ADIE | RCIE | TXIE | SSPIE | CCP1E | TMR2IE | TMR1IE |
| 0Dh | PIR2 | - |  | - | EEIF | BCLIF | - | - | CCP2IF |
| 8Dh | PIE2 | - |  | - | EEIE | BCLIE | - | - | CCP2IE |

При переходе в подпрограмму обработки прерываний бит GIE аппаратно сбрасывается в ‘0’, запрещая прерывания, адрес возврата из подпрограммы обработки прерываний помещается в стек, а в счетчик команд PC загружается вектор прерывания 0х0004. Источник прерываний можно определить проверкой флагов прерываний. Чтобы избежать повторного перехода на обработку прерывания необходимо сбросить установленный флаг обрабатываемого прерывания.

При возвращении из подпрограммы обработки прерывания, по команде RETFIE, бит GIE аппаратно устанавливается в ‘1’, разрешая все немаскированные прерывания. Прерывание при переполнении таймера TMR0 можно запретить, сбросив бит T0IE в регистре INTCON.

Микроконтроллеры PIC16F87x не имеют специальных команд для сохранения содержимого аккумулятора и специальных регистров во время обработки прерывания с последующим их восстановлением. Автоматически сохраняется только значение программного счетчика для возврата из подпрограммы. Поэтому необходимо программно сохранять контекст (содержимое аккумулятора W и регистра STATUS). С учетом этой особенности шаблон программы с комментариями приведен ниже:

list p=16f877a,st = OFF ; директива листинга

#include <p16f877a.inc> ; определение переменных МК

\_\_CONFIG \_CP\_OFF & \_WDT\_OFF & \_BODEN\_OFF & \_PWRTE\_ON & \_HS\_OSC & \_WRT\_OFF& \_LVP\_OFF & \_DEBUG\_ON & \_CPD\_OFF

; директива установки разрядов слова конфигурации

;\*\*\*\*\* Определение переменных

w\_temp EQU 0x70 ; переменные для сохранения контекста

status\_temp EQU 0x71 ;

; директивы определения остальных переменных

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ORG 0x000 ; вектор старта

nop ; требуется для ICD

goto Main ; переход на начало программы

ORG 0x004 ; вектор прерывания

movwf w\_temp ; сохранение контекста

movf STATUS,w ;

movwf status\_temp ;

; операторы подпрограммы обработки прерывания

Ret movf status\_temp,w ; восстановление контекста

movwf STATUS ;

swapf w\_temp,f

swapf w\_temp,w ;

retfie ; возврат из подпрограммы

Main

; операторы программы

; операторы подпрограмм

END ; директива окончания программы

Использование команды swapf для загрузки и выгрузки значений обусловлено тем, что эта команда не изменяет состояние флага нуле­вого результата Z регистра STATUS. Применение команды movf со­кратило бы программный код, но тогда при сохранении акку­мулятора может быть изменено состояние бита Z, что в общем случае недопустимо. Корректное программирование требует, чтобы при воз­вращении из подпрограммы прерывания значения регистров W и STATUS были восстановлены абсолютно точно.

Операторы подпрограммы обработки прерывания:

btfss INTCON,T0IF ;прерывание по переполнению TMR0?

goto Ret ; нет

movlw .70

movwf TMR0 ;загрузить таймер

decf Count\_ms,f ;мс прошла

bcf INTCON,T0IF ;сбросить флаг прерывания.

Для настройки линий PORTC на ввод в регистр TRISC необходимо загрузить управляющее слово b'11111111'. Для настройки линий PORTD на вывод в регистр TRISD необходимо загрузить управляющее слово b'00000000'.

Операторы инициализации:

Main clrf PORTD ;инициализация модулей МК

BANKSEL TRISD ;bank1

movlw b'11111111' ;все ввод

movwf TRISC^80h

movlw b'00000000' ;все вывод

movwf TRISD^80h

movlw b'10000011' ;режимы TMR0

movwf OPTION\_REG^80h

BANKSEL PORTD ;bank0

bsf INTCON,GIE ;разрешить прерывания

bsf INTCON,T0IE ;разрешить прерывания от TMR0.

Операторы основного цикла программы:

Loop clrf Cnt ;основной цикл

Inc movf Cnt,w

movwf PORTD ;вывод на линейный индикатор

movf PORTC,w

movwf PCh0

comf PCh0,w

andwf PCh1,w

andwf PCh2,w

movwf Key

btfss Key,1 ;опрос кнопки "+"

goto Bypass ;кнопка "+" не нажата

incf Cnt,f ;инкремент счетчика

Bypass movf PCh1,w

movwf PCh2

movf PCh0,w

movwf PCh1

call D5ms ;задержка

movlw -.10 ;число -10

addwf Cnt,w ;прибавить к содержимому счетчика

btfss STATUS,C ;для сравнения

goto Inc

goto Loop ;в начало цикла.

Исходный текст подпрограммы задержки:

D5ms movlw .5

movwf Count\_ms ;загрузить счетчик миллисекунд

movf Count\_ms,f ;время закончилоcь?

btfss STATUS,Z

goto $-2 ; нет

return ;возврат в основную программу.

**13 Аналоговый ввод**

**13.1 Описание задания**

Необходимо разработать и кодировать на языке Ассемблера алгоритм программы аналогового ввода для МК типа PIC16F877A. Программа должна выполнять запуск модуля АЦП, получать результаты преобразования, обрабатывать и выводить на индикаторы. Для выполнения работы используется отладочный стенд, в котором предусмотрено подключение к входному каналу АЦП AN0 напряжения Vin с делителя напряжения, выполненного на потенциометре. Вход делителя напряжения подключен к напряжению питания +5 В. На входы опорного напряжения АЦП Vref также должно быть подано напряжение питания. Значение кодового эквивалента аналого-цифрового преобразования

Nout = (Vin / Vref )·2n ,

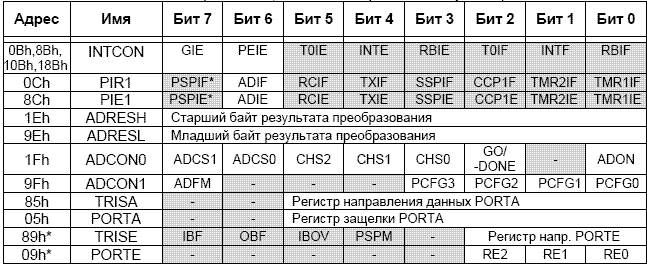
где n-число используемых разрядов АЦП.

**13.2 Рекомендации по выполнению**

В таблице 13.1 приведены регистры специального назначения, связанные с

модулем АЦП. Результат аналого-цифрового преобразования сохраняется в регистрах ADRESH (старший байт) и ADRESL (младший байт).

Таблица 13.1



Разряды регистра управления ADCON0 определяют:

- биты 7-6 ADCS1:ADCS0 **-** выбор источника тактового сигнала (00 = FOSC/2; 01 = FOSC/8; 10 = FOSC/32; 11 = FRC - внутренний RC генератор модуля АЦП);

- биты 5-3 CHS2:CHS0 **-** выбор аналогового канала AN0.. AN7;

- бит 2 GO/-DONE **-** бит статуса модуля АЦП (установка бита вызывает начало преобразования, аппаратно сбрасывается по завершении преобразования);

- бит 1 не используется;

- бит 0 ADON **-** включение модуля АЦП (1 - модуль включен, 0 - модуль выключен и не потребляет ток).

Разряды регистра управления ADCON1 определяют:

- бит 7 ADFM **-** формат сохранения 10-разрядного результата (пояснения на рисунке 13.1);

- биты 6-4 не используются**,** читаются как ‘0’;

- биты 3-0 PCFG3:PCFG0 **-** управляющие биты настройки каналов АЦП и многофункциональных выводов (таблица 13.2).

Когда преобразование завершено, 10-разрядный результат аналого-цифрового преобразования записывается в регистры ADRESH:ADRESL, после чего сбрасывается флаг GO/-DONE (ADCON0,2) и устанавливается флаг прерывания ADIF в регистре PIR1. Сброс флага GO/-DONE в '0' во время преобразования приведет к его прекращению.

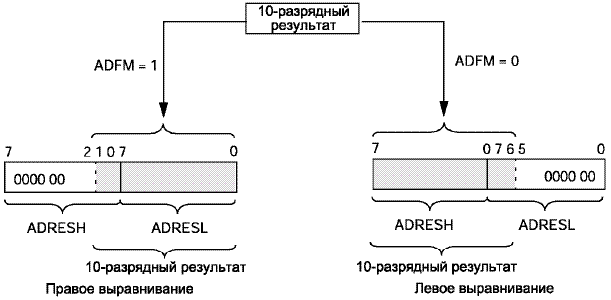


Рисунок 13.1

Таблица 13.2

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PCGF3:  PCGF0 | AN7  RE2 | AN6  RE1 | AN5  RE0 | AN4  RA5 | AN3  RA3 | AN2  RA2 | AN1  RA1 | AN0  RA0 | VREF+ | VREF- |
| 0000 | A | A | A | A | A | A | A | A | VDD | VSS |
| 0001 | A | A | A | A | VREF+ | A | A | A | RA3 | VSS |
| 0010 | D | D | D | A | A | A | A | A | VDD | VSS |
| 0011 | D | D | D | A | VREF+ | A | A | A | RA3 | VSS |
| 0100 | D | D | D | D | A | D | A | A | VDD | VSS |
| 0101 | D | D | D | D | VREF+ | D | A | A | RA3 | VSS |
| 011x | D | D | D | D | D | D | D | D | VDD | VSS |
| 1000 | A | A | A | A | VREF+ | VREF- | A | A | RA3 | RA2 |
| 1001 | D | D | A | A | A | A | A | A | VDD | VSS |
| 1010 | D | D | A | A | VREF+ | A | A | A | RA3 | VSS |
| 1011 | D | D | A | A | VREF+ | VREF- | A | A | RA3 | RA2 |
| 1100 | D | D | D | A | VREF+ | VREF- | A | A | RA3 | RA2 |
| 1101 | D | D | D | D | VREF+ | VREF- | A | A | RA3 | RA2 |
| 1110 | D | D | D | D | D | D | D | A | VDD | VSS |
| 1111 | D | D | D | D | VREF+ | VREF- | D | A | RA3 | RA2 |

Время получения одного бита результата определяется параметром TAD. Для 10-разрядного результата требуется как минимум 12TAD. Для получения корректного результата преобразования необходимо выбрать источник тактового сигнала АЦП, обеспечивающий время TAD не менее 1.6 мкс.

На рисунке 13.2 приведена схема программы, которая после инициализации модуля АЦП циклически выполняет:

- последовательный запуск модуля АЦП для выполнения преобразования напряжения из канала AN0;

- выводит код результата преобразования в PORTD.

Период дискретизации аналоговых сигналов определяется полным временем выполнения цикла.

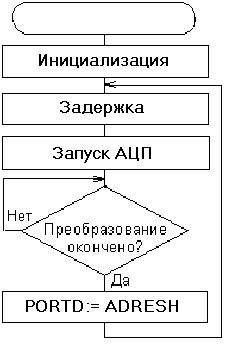


Рисунок 13.2

Исходный текст программы

#include p16f877a.inc

LIST p=16F877a, st=OFF

\_\_CONFIG \_CP\_OFF & \_WDT\_OFF & \_BODEN\_ON & \_PWRTE\_ON & \_HS\_OSC & \_WRT\_OFF & \_LVP\_OFF & \_DEBUG\_ON & \_CPD\_OFF

;---------------------- Переменные --------------------------

CBLOCK 0x20

Cnt1 ;счетчик

ENDC

;---------------------- Константы ---------------------------

Init\_PD EQU b'00000000' ;все вывод

Init\_ADC1 EQU b'00001110' ;результат АЦП влево, AN0 аналоговый вход

Init\_ADC0 EQU b'10000001' ;Fosc/32,канал AN0,разрешить A/D

;=====================Основная программа=====================

Begin

;--------------------------Инициализация---------------------------

clrf PORTD

movlw Init\_ADC0

movwf ADCON0 ;Fosc/32, канал AN0, разрешить A/D

BANKSEL TRISD

movlw Init\_PD

movwf TRISD^80h ;все линии - вывод

movlw Init\_ADC1

movwf ADCON1^80h ;результат АЦП вправо, вход AN0

BANKSEL PORTD

;----------------------основной цикл-----------------------

Main movlw .255 ;задержка 255 мкс

movwf Cnt1

Dly decfsz Cnt1,f

goto Dly

bsf ADCON0,GO ;Начало A/D преобразования

Wait btfsc ADCON0,GO ;Ожидание завершения

goto Wait ;A/D преобразования

movf ADRESH,w ;ст. байт A/D преобразования в

movwf PORTD

goto Main ;перейти в начало основного цикла

END

**14 Вывод символьной информации**

**14.1 Описание задания**

Необходимо разработать и кодировать на языке Ассемблера алгоритм программы вывода символьной информации для МК типа PIC16F877A.

Нажатие с последующим отпусканием кнопки RC1 должно вызывать модификацию содержимого ячейки Cnt. Содержимое ячейки Cnt должно отображаться на 7-сегментных индикаторах в десятичном коде для вариантов 1 – 3 и в шестнадцатеричном коде для вариантов 4 - 6. Варианты заданий приведены в таблице 11.1.

**14.2 Рекомендации по выполнению**

Символьные индикаторы являются элементом человеко-машинного интерфейса (HMI). В простых задачах используют 7-сегментные индикаторы для отображения десятичных цифр и других символов, например, некоторых букв латинского алфавита. Схема включения 4-разрядного 7-сегментного индикатора с общим катодом приведена на рисунке 14.1. Аноды светоизлучающих сегментов всех разрядов подключим к линиям порта PORTD. Общий катод каждого разряда управляются транзисторным ключом. Управление ключами будут обеспечивать четыре линии порта PORTB.

Данное включение предполагает использование динамической индикации: на линии порта PORTD поочередно выводятся коды символов разрядов и открывается соответствующий транзисторный ключ. Чтобы мерцание индикаторов не было заметно, частота сканирования разрядов должна быть не менее 50 Гц. Если принять, что в цикле программы длительностью 5 мс будет выводиться символ одного разряда, то период сканирования составит 20 мс.

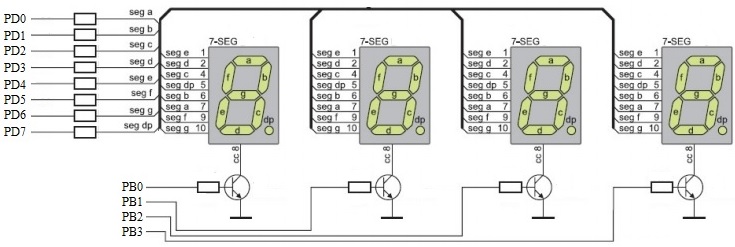


Рисунок 14.1

Схема алгоритма примера программы вывода в десятичном коде значения декадного счетчика приведена на рисунке 14.2. Выводимая информация – значение переменной Cnt преобразуется и выводится на 7-сегментные индикаторы.

Для этого используются:

- подпрограмма B2\_BCD для преобразования двоичного кода в двоично-десятичный (аргумент функции двухбайтный двоичный код в ячейках H\_byte, L\_byte; значение функции трехбайтный BCD в ячейках R0,R1,R2);

- подпрограмма Pack, выполняющая переформатирование данных (аргумент функции двухбайтный BCD в ячейках R1,R2; значение функции 7-сегментный код разрядов в ячейках Dig4 – Dig1);

- подпрограмма вывода данных через порты Dis7seg.

Алгоритм функции B2\_BCD и функции BINto7Seg, вызываемой из подпрограммы Pack для преобразования 4-разрядного двоичного кода символа в 7-сегментный код, на рисунке 14.2 не приведены. Программа использует подпрограмму задержки D5ms и подпрограмму обработки прерываний Int (алгоритм приведен на рисунке 14.2).

При кодировании алгоритма следует учесть, что линии PB3 – PB0 порта PORTB необходимо настроить на вывод, остальные на ввод:

movlw b'11110000' ;3-0 вывод

movwf TRISB^80h

Подпрограмма преобразования в двоично-десятичный код B2\_BCD должна использоваться, если информацию необходимо отображать в десятичном формате.

Если информация должна отображаться в шестнадцатеричном коде, то значение переменной Cnt необходимо копировать в ячейку R2. Ячейку R1 необходимо очистить.

При реализации подпрограммы BINto7Seg использован метод табличных вычислений. Значение аргумента передается в функцию через аккумулятор и суммируется с младшим байтом программного счетчика PCL. Функция возвращает через аккумулятор значение 7-сегментного кода при выполнении инструкции возврата retlw с соответствующим параметром.

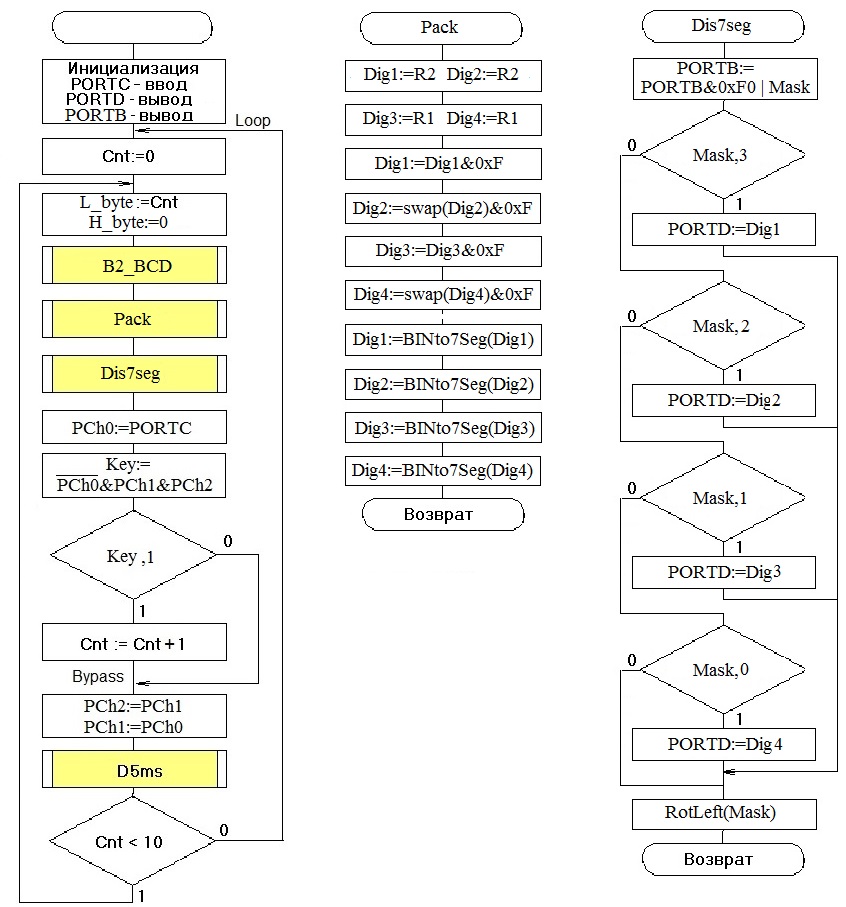


Рисунок 14.2

До основного цикла кроме инициализации портов, таймера и системы прерываний необходимо выполнить инициализацию переменной Mask

clrf Mask

incf Mask,f

Текст подпрограммы вывода на индикатор:

Dis7seg movlw 0xF0

andwf PORTB

movlw 0x0F

andwf Mask,w

iorwf PORTB

btfss Mask,3

goto $+4

movf Dig1,w ;вывести

movwf PORTD ;символ

goto Shift

btfss Mask,2

goto $+4

movf Dig2,w ;вывести

movwf PORTD ;символ

goto Shift

btfss Mask,1

goto $+4

movf Dig3,w ;вывести

movwf PORTD ;символ

goto Shift

btfss Mask,0

goto Shift

movf Dig4,w ;вывести

movwf PORTD ;символ

Shift bcf STATUS,C

rlf Mask,f

btfsc Mask,4

swapf Mask,f

return

**Приложение Б**

**Методические рекомендации к выполнению лабораторных работ**

**1 Характеристики логических элементов**

**1.1 Краткие сведения из теории**

Цель работы - получить практические навыки по исследованию статических и динамических свойств логических элементов. В работе исследуется элемент транзисторно-транзисторной логики (ТТЛ).

Основные параметры логических элементов можно определить по их статическим и переходным характеристикам. Статическая передаточная характеристика *Uвых = f(Uвх)* отражает зависимость потенциала на выходе от потенциала на одном из входов при постоянных значениях потенциала U0 или U1 на остальных входах, где U0 - уровень потенциала, соответствующий логическому "0" и U1 - уровень потенциала, соответствующий логической "1".

Типичная передаточная характеристика инвертирующего логического элемента показана на рисунке 1.1, и имеет три явно выраженных участка:

1) соответствующий состоянию Uвых = U1 ;

2) соответствующий состоянию Uвых = U0 ;

3) соответствующий неопределенному состоянию.

Значения потенциала Uвх, соответствующие границам участков, называются пороговыми уровнями V0п и V1п . Их положение определяют из условия dUвых/dUвх = -1. Средний порог переключения Vп можно определить как проекцию на ось Uвх точки, лежащей на пересечении передаточной характеристики и прямой равного усиления Uвых = Uвх.

Если на входах схемы установлены логические уровни U0 или U1 , то при воздействии положительной потенциальной помехи величиной U+п> V0п – U0 или отрицательной помехи U-п > U1 - V1п происходит ложное переключение логического элемента. Максимально допустимые величины потенциальной помехи, не вызывающие сбоев в цифровой схеме, называются уровнями помехозащищенности для помехи положительной полярности U+п = V0п – U0 и помехи отрицательной полярности U-п = U1 – V1п .

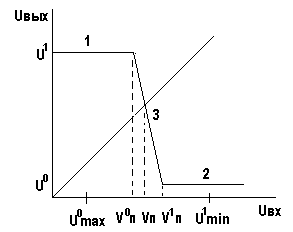


Рисунок 1.1

Передаточные характеристики реальных логических элементов имеют разброс, обусловленный технологическим разбросом, влиянием внешних факторов (нагрузки, температуры и т.д.). В этом случае значения уровней U0 и U1 и пороговых уровней V0п , V1п принадлежит некоторым диапазонам:

U0min < U0 < U 0max , V0п min < V0.п < V0.п max ,

U1min < U1 < U1max , V1.п min < V1.п < V1.п max .

Помехоустойчивость определяется для наихудшего сочетания факторов

U+п = V0.п min – U0max ,

U-п = U1min – V1п max .

Для элементов ТТЛ предельные значения уровня логической “1” U1min не менее 2.4 В, логического нуля U0max “0” – не более 0,4 В.

Передаточную статическую характеристику можно снять, подавая на вход логического элемента напряжение с регулируемого источника 0 - 5 В.

Чтобы оценить быстродействие логических элементов в условиях, соответствующих их работе в цифровых устройствах, измерение переходных характеристик следует производить в цепи последовательно включенных элементов. По переходным характеристикам возможно определить основные динамические параметры: задержки распространения при включении tзр01 и выключении tзр10. Эти задержки определяются как интервалы времени между моментами пересечения соответствующих фронтов сигналов порога переключения Vп. Среднее время задержки распространения

*tзр = 0.5⋅ ( tзр01 + tзр10)*

определяет среднее время выполнения логических операций. Наибольшее влияние на значения *tзр* оказывает паразитная емкость нагрузки.

**1.2 Задание к выполнению работы**

1.2.1. Снять статическую передаточную характеристику элемента заданного типа и определить основные статические параметры.

1.2.2. Снять переходную характеристику элемента заданного типа и определить динамические параметры.

**1.3 Порядок выполнения работы**

1.3.1 Открыть проект \Electronics\TTL командой **File>Open\Project**.

1.3.2 Открыть в менеджере проекта схему измерения характеристик логического элемента ТТЛ ROOT: PAGE1.

1.3.3 Выполнить редактирование профиля моделирования командой **PSpice>Edit Simulation Profile** для расчета статических характеристик логического элемента ТТЛ, для этого:

- установить вид анализа (Analysis type) – DC Sweep;

- задать Sweep Variable Voltage source Name: Vin1;

- задать закон изменения Sweep type: Linear и пределы изменения (Start Value = 0, End Value = 4, Increment = 0.1);

- закрыть окно редактирования нажатием кнопки OK.

1.3.4 Выполнить моделирование командой **PSpice>Run**. При этом выполняется запуск программы PSpice A/D.

1.3.5 Вывести на экран график зависимости сигнала V(Out1), для этого командой **Trace>Add Trace** открыть окно Add Trace и выбрать сигнал V(Out1), подтвердить выбор нажатием клавиши ОК. Зарисовать график. Удалить график командой **Trace>Delete All Traces**.

1.3.6 По статическим характеристикам определить следующие параметры:

- порог переключения Vп;

- пороговые уровни "нуля" V0п и "единицы" V1п;

- уровни помехозащищенности U-п и U+п .

1.3.7 Активизировать окно программы OrCAD Capture. Выполнить редактирование параметров источника импульсного сигнала Vin1 - задать значения параметров V1=0, V2=3.75, TD=0, TR=10ns, TF=5ns, PW=50ns, PER=200ns.

1.3.8 Выполнить редактирование профиля моделирования командой **PSpice>Edit Simulation Profile** для расчета переходной характеристики, для этого:

- задать вид анализа (Analysis type) – Time Domain (Transient);

- задать значение времени моделирования Run to time = 120ns;

- закрыть окно редактирования нажатием кнопки OK.

1.3.9 Выполнить расчет переходных характеристик командой **PSpice>Run**.

1.3.10 Вывести на экран временные диаграммы сигналов In, Out1, Out2, для этого командой **Trace>Add Trace** открыть окно Add Trace и выбрать сигналы V(In), V(Out1), V(Out2) подтвердить выбор нажатием клавиши ОК. Определить значения:

- задержки распространения при включении tзр01 и выключении tзр10;

- средней задержки распространения tзр.

Зарисовать графики. Удалить график командой **Trace>Delete All Traces**.

**1.4 Оформление работы**

Отчет по работе должен содержать:

- статические характеристики элемента ТТЛ, определение основных статических параметров с необходимыми графическими построениями;

- переходные характеристики элемента ТТЛ и определение основных динамических параметров.

**1.5 Контрольные вопросы**

1.5.1 Как работает ТТЛ элемент со сложным инвертором?

1.5.2 Какой из логических элементов:2ИЛИ-НЕ, 2ИЛИ, 4И, 4И-НЕ, 4И-2ИЛИ-НЕ наиболее просто реализовать в схемотехнике ТТЛ?

1.5.3 Перечислите основные статические и динамические параметры и характеристики элементов ТТЛ.

1.5.4 Перечислите варианты модификаций элементов ТТЛ (ТТЛШ) и их особенности.

**2 Анализ комбинационных узлов**

**2.1 Краткие сведения из теории**

Цель занятия - получить практические навыки решения типовых задач анализа комбинационных узлов.

Методы и средства функционально-логического моделирования позволяют эффективно решать основные задачи анализа цифровых РЭУ: логическую и временную верификацию. Логическая верификация направлена на выявление соответствия работы цифрового РЭУ заданному алгоритму функционирования. Временная верификация направлена на оценку быстродействия.

Чтобы выполнить логическую верификацию, необходимо определить установившиеся значения выходных сигналов (выходные наборы) для всех заданных комбинаций входных сигналов (входных наборов). Промежуточные значения сигналов, формируемые во время переходных процессов, во внимание не принимаются. Если все полученные значения совпадают с заданными, то это позволяет сделать вывод о правильности синтеза цифрового РЭУ. Выявленные отличия свидетельствует о допущенных ошибках при синтезе.

Упорядоченную последовательность входных наборов называют тестом. Для цифровых РЭУ комбинационного типа последовательность поступления входных наборов не имеет значения. Длительность формирования каждого из входных наборов определяется по условию завершения переходных процессов между последовательными изменениями входных сигналов.

Для временной верификации выполняют анализ характера переходных процессов при переключении элементов цифрового РЭУ. В качестве оценки быстродействия цифровых РЭУ комбинационного типа используют:

*tзр = max tзр ij ,* (2.1)

*j=1,m; i=1,n*

где *tзр ij*- задержка распространения сигнала от *i–*го входа до *j-*го выхода. Величину *tзр*  определяют как задержку фронта выходного сигнала *yj* относительного вызвавшего его фронта входного сигнала *xi* .

**2.2 Описание последовательности выполнения задания**

2.2.1 Логическая верификация

Активизировать окно программы OrCAD Capture. Открыть созданный проект командой **File>Open>Project**. Открыть в менеджере проекта схему цифрового РЭУ SCHEMATIC1: PAGE1.

Сформировать тест для логической верификации. Запустить программу PSpice Stimulus Editor. Выполнить команду **File>New**.

Последовательно выполнить формирование цифровых сигналов командой **Stimulus>New**,для этого в открывшейся панели выбрать тип сигнала (Signal) и в строке Name указать его имя, которое должно совпадать с обозначенным на схеме (значение параметра Implementation соответствующего элемента DigStim1).

При необходимости выполнить редактирование цифровых сигналов, для этого активировать клавишу Add a new point or Transition to a Stimulus (при этом курсор примет вид карандаша), подвести курсор к графику сигнала, где необходимо поставить переход (изменить его значение), и нажать левую кнопку мыши. После завершения расстановки переходов нажать правую кнопку мыши (при этом курсор примет обычный вид). При необходимости предварительно выделенный переход (нажатием левой кнопки мыши) можно удалить командой **Edit>Delet** или “отбуксировать” при нажатой левой клавиши мыши.

Сохранить файл сигналов командой **File>Save As**, назначив имя Vd<n>, где <n> - номер варианта. Завершить работу с программой Stimulus Editor.

Активизировать окно программы OrCAD Capture. Выполнить редактирование профиля моделирования командой **PSpice>Edit Simulation Profile** для моделирования цифрового устройства, для этого на закладке Analysis:

- задать вид анализа (Analysis type) – Time Domain (Transient);

- задать значение времени моделирования Run to time;

- на закладке Stimulus выбрать нажатием на кнопку Browse и подключить к проекту файл сигналов нажатием на кнопку Add to Design, закрыть окно редактирования нажатием кнопки OK.

Выполнить моделирование цифрового РЭУ командой **PSpice>Run**.

Вывести на экран временные диаграммы входных и выходных сигналов, для этого командой **Trace>Add Trace** открыть окно Add Trace и выбрать соответствующие сигналы, подтвердить выбор нажатием клавиши ОК. Зарисовать временные диаграммы в отчет. Определить задержки. Удалить диаграммы командой **Trace>Delete All Traces**. Сделать вывод о соответствии работы схемы алгоритму функционирования.

**2.3 Оформление отчета**

Отчет должен содержать:

- временные диаграммы входных и выходных сигналов комбинационной схемы;

- анализ результатов моделирования (вывод о правильности работы схемы и значения задержек).

**2.4 Контрольные вопросы**

2.4.1 Перечислите задачи анализа комбинационных схем. Кратко сформулируйте их содержание.

2.4.2 Как сформировать тест для логической верификации схемы комбинационного типа?

2.4.3 Как можно оценить задержки распространения сигналов в комбинационных схемах?

**3 Двоичный сумматор**

**3.1 Краткие сведения из теории**

Цель работы - исследовать работу многоразрядного сумматора с последовательным переносом.

На рисунке 3.1 приведена функциональная схема четырехразрядного сумматора с последовательным переносом, построенная на основе одноразрядных сумматоров.

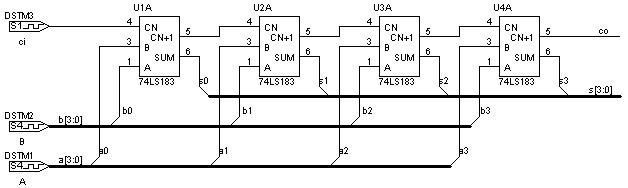


Рисунок 3.1

Сумматоры являются важнейшим арифметическим узлом. Их быстродействие во многом определяет быстродействие цифровых вычислительных устройств. Быстродействие многоразрядного сумматора характеризуется задержками распространения tAS, tAC, tCS, tCC и определяется по наихудшему случаю.

Схема сумматора с последовательным переносом сохранена в файлах sum\_sc.opj и sum\_sc.dsn.

**3.2 Задание к выполнению работы**

3.2.1 Определить входные наборы четырехразрядного сумматора, смена которых позволит определить матрицу задержек одноразрядного сумматора, а также задержки формирования суммы многоразрядных сумматоров.

3.2.2 С помощью программы PSpice Stimulus Editor сформировать тест для исследования быстродействия сумматоров.

3.2.3 Выполнить моделирование работы схемы сумматора в среде программы PSPICE AD. Зарисовать временные диаграммы сигналов и определить значения задержек tAS, tAC, tCS, tCC многоразрядного сумматора.

3.2.4 Ответить на контрольные вопросы, оформить отчет о выполненной работе.

**3.3 Порядок выполнения работы**

3.3.1 Откройте проект Electronics\Sum\sum\_sc.

3.3.2 Откройте в менеджере проекта схему сумматора SCHEMATIC1: PAGE1.

3.3.3 Запустите программу PSpice Stimulus Editor. Сформируйте тест для исследования работы схемы. Подберите значения слов данных A[3:0], B[3:0] и сигнала Ci при смене которых будут изменяться выходные сигналы сумматора младшего разряда, а задержка формирования многоразрядной суммы будет максимальной (распространение сигналов будет происходить по самому длинному пути). Сформируйте модели сигналов A[3:0], B[3:0], Ci. Сохраните файл сигналов. Завершите работу с программой PSpice Stimulus Editor.

3.3.4 Активируйте окно программы OrCAD Capture. Выполните редактирование профиля моделирования, для этого на закладке Analysis:

- задайте вид анализа (Analysis type) – Time Domain (Transient);

- задайте требуемое значение времени моделирования Run to time;

- на закладке Stimulus выбрать нажатием на кнопку Browse и подключите к проекту файл тестовых сигналов нажатием на кнопку Add to Design, закройте окно редактирования нажатием кнопки OK.

3.3.5 Выполните моделирование цифрового устройства.

3.3.6 По временным диаграммам входных и выходных сигналов определите, как работает схема. Зарисуйте (скопируйте) временные диаграммы в отчет. Определите значения задержек распространения сигналов. Закройте проект.

**3.4 Содержание отчета**

Отчет должен содержать:

- исследуемая схема сумматора;

- результаты моделирования (временные диаграммы, значения задержек)

- выводы.

**3.5 Контрольные вопросы**

3.5.1 Почему схемы многоразрядных сумматоров строятся на основе одноразрядных?

3.5.2 Какие функции реализует одноразрядный сумматор?

3.5.3 Как необходимо включить многоразрядный сумматор для сложения чисел, представленных в прямом, обратном и дополнительном коде?

**4 Исследование счетчиков**

**4.1 Краткие сведения из теории**

Счетчики – автоматы, используемые для подсчета числа поступивших на их вход импульсов. По модулю счета различают двоичные и двоично-кодированные счетчики. По направлению счета различают суммирующие, вычитающие и реверсивные счетчики. По особенностям структуры различают асинхронные и синхронные счетчики. В асинхронных счетчиках счетные импульсы непосредственно подаются на синхровход триггера только младшего разряда. Для переключения триггеров других разрядов импульсы формируются элементами схемы счетчика. Так как элементы имеют задержки распространения сигналов, то триггеры различных разрядов начинают переключаться в различные моменты времени – асинхронно. В синхронных счетчиках триггеры всех разрядов переключаются синхронно, так как на их синхровходы подаются счетные импульсы.

**4.2 Описание последовательности выполнения задания**

4.2.1 Открыть файл async\_ctr.

4.2.2 Задать имя dsrc1.st в поле Data File окна параметров источника сигнала счетных импульсов. Задать значение параметра Time Delta 1.0e-8. Для открытия окна параметров необходимо навести курсор на элемент источника G и нажать ЛКМ.

4.2.3 Командой **Анализ > Переходной процесс (Tran)** выполнить расчет во временной области, для этого в окне Расчет переходного процесса (Tran):

- задайте значение интервала расчета Время анализа, равное 3.2e-7 с.

В окне Графики для выходных переменных задать через пробел имена узлов clk q0+10 q1+20 q2+30 q3+40. Нажать кнопку **Расчет**. Результат расчета будет представлен в виде временных диаграмм входного и выходного сигналов.

По диаграммам сигналов определить, как работают триггеры. Зарисовать временные диаграммы в отчет. Определить время установления счетчика. Удалить диаграммы. Закрыть файл LR7\_1.

4.2.4 Сделать выводы о работе асинхронного счетчика.

4.2.5 Открыть файл sync\_ctr.

4.2.6 Выполнить действия, приведенные в п.п. 4.5.2 – 4.5.3.

4.2.8 Сделать вывод ы о работе синхронного счетчика.

**4.3 Содержание отчета**

Отчет должен содержать:

- исследуемые схемы;

- временные диаграммы сигналов, полученные при исследовании счетчиков;

- анализ результатов моделирования (выводы).

**4.4 Контрольные вопросы**

Как отличается работа синхронного и асинхронного счетчика?

Поясните причины этих отличий.

Каким образом можно определить время установления счетчика?

1. **Трансляция и тестирование программы**

**5.1 Описание работы**

Цель работы:

- освоить работу в среде системы MPLAB IDE;

- приобрести практические навыки отладки простых программ.

Объектом отладки в работе является программа генератора чисел. Числовая последовательность (в соответствии с вариантом задания) формируется на линиях порта PORTD.

MPLAB IDE – это интегрированная среда разработки программ для микроконтроллеров (МК) PICmicro фирмы Microchip, позволяющая писать, компилировать и отлаживать программы.

**5.2 Создание проекта и трансляция программы**

В процессе разработки программы формируется проект - множество взаимосвязанных файлов, включая исходный текст программы <имя>.asm, загрузочный модуль <имя>.hex и другие файлы. Все эти файлы удобно хранить в отдельной папке. Взаимосвязи файлов содержатся в корневом файле проекта.

В папке PIC создайте свою рабочую папку, а в ней папку lr5. Скопируйте в нее подготовленный файл с исходным текстом программы lr5.asm. Создайте новый проект, для этого выберите команду *Project > Project Wizard*и последовательно выполните действия:

- задайте тип микроконтроллера PIC16F877A;

- задайте транслятор Microchip MPASM Toolsuite;

- задайте имя проекта lr5 и папку для нового проекта;

- включите в проект исходный текст программы lr5.asm.

Откройте окно с исходным текстом программы, для этого в окне менеджера проекта lr5.mcw дважды щелкните по ярлыку lr5.asm в папке *Source Files* или выполните команду *File>Open*.

Выполните трансляцию программы (команда *Project>Make*). На закладке *Build* в окне *Output* будет приведена командная строка, посланная ассемблеру, сообщения об ошибках или успешном завершении трансляции.

Если программа содержала синтаксические и другие формальные ошибки, в окне будет выведен список сообщений об ошибках (ERROR). Если дважды щелкнуть мышкой на сообщении об ошиб­ке, то курсор автоматически перейдет к соответствующему оператору в исход­ном тексте. Если в программе нет подобных ошибок, ассемблер выдаст сообщение BUILD SUCCEEDED. Это означает, что получен загрузочный модуль программы, готовый для выполнения микроконтроллером. Но это не означает, что программа не содержит логических ошибок, и будет работать правильно. Проверку работы программы, поиск и исправление возможных ошибок выполняют в ходе отладки.

**5.3 Отладка программы**

В данной работе предусмотрено выполнение только автономной отладки программы на программно-логической модели МК. Активировать инструмент *MPLAB-SIM Simulator*можно командой *Debugger>Select Tool>MPLAB SIM*.

Существуют различные методы тестирования программ, направленные на проверку их правильности и выявление возможных ошибок. Но практически все они основаны на отслеживании выполнения программы и сравнении с ожидаемым ходом выполнения. Для отладки программ обычно применяют два режима:

- пошаговая отладка;

- выполнение программы (прогон) до точки останова.

При пошаговой отладке выполняется один оператор программы, и затем контролируются те переменные, на которые должен воздействовать данный оператор. Отдельный участок программы можно выполнить в режиме прогона. Точку останова устанавливают на операторе, при выполнении которого необходимо проверить значения переменных или просто проконтролировать, передаётся ли управление данному оператору.

Для отслеживания содержимого ограниченного набора ячеек памяти удобно использовать смотровое окно. Чтобы создать смотровое окно выполните команду *View> Watch*. В верхней части окна *Watch* находятся прокручиваемые списки регистров специальных функций (SFR) и идентификаторов, используемых в программе (Symbol). Из левого списка выберите ячейку PORTD, добавьте в окно наблюдения нажатием кнопки *Add SFR*. Из правого списка выберите ячейку Cnt, добавьте в окно наблюдения нажатием кнопки *Add Symbol*.

Инициализируйте МК нажатием кнопки *Reset*. Программный счетчик бу­дет сброшен в ноль. Средствами символьного отладчика строка исходного кода, соответствующая этому адресу, отмечена указателем (зеленой стрелкой). В строке состояния MPLAB IDE отображаются:

- режим работы отладчика (MPLAB SIM);

- тип МК (PIC16F877a);

- содержимое программного счетчика PC;

- содержимое аккумулятора W;

- состояние флагов Z dc c (заглавной буквой – установлен, строчной – сброшен);

- тактовая частота МК (12 MHz);

- активный банк памяти данных (bank 0).

Нажмите кнопку *Step Into*. Будет выполнена первая команда программы. Программный счетчик будет показывать PC:0x01, указатель пе­реместится на следующую исполняемую команду. Для пошагового выполнения можно также использовать функциональную клавишу <F7>. Нажимайте клавишу <F7> несколько раз и наблюдайте, как перемещается по тексту указатель, изменяются показания программного счетчика и значения переменных в смотровом окне.

Установите курсор на строке с оператором goto Loop. Нажмите правую кнопку мыши и в появившемся меню выберите опцию *Set Breakpoint* (точка останова). Слева от этой строки появится символ “B” красного цвета, сигнализируя, что сразу перед исполнением команды в этой строке выполнение програм­мы остановится. Продолжите выполнение программы в режиме прогона командой *Run* или клавишей <F9>. При достижении точки оста­нова симулятор остановится на выделенном операторе (но эта команда не выполнится).

Если нажать клавишу <F9> еще раз, симуляция программы продолжится до следующего достижения точки останова. Разумеется, когда программа еще не достигла точки останова, ее можно остановить командой *Halt* или нажатием клавиши <F5>.

По содержимому окна *Watch*  определите корректность выполнения программы. При выявлении несоответствия хода выполнения программы ожидаемому, необходимо локализовать ошибку в программе. Для этого необходимо:

- определить шаг (и соответствующий оператор программы), начиная с которого выполнение программы отклоняется от ожидаемого;

- проанализировать причину этого отклонения;

- исправить ошибку в исходном тексте;

- выполнить повторную трансляцию и проверку корректности выполнения программы.

После выполнения работы закройте проект командой *Project>Close*.

**5.4 Содержание отчета**

Отчет должен содержать:

- схему алгоритма программы;

- листинг трансляции программы;

- выводы о корректности программы.

**5.5 Контрольные вопросы**

Поясните по схеме алгоритма работу программы.

Почему алгоритм программы не заканчивается?

Каким образом можно закодировать ветвление алгоритма?

Поясните содержание листинга трансляции программы?

По каким признакам можно сделать вывод, что программа выполняется правильно?

**6 Отладка программы дискретного ввода-вывода**

**6.1 Описание работы**

Цель работы - получить практические навыки выполнения автономной и комплексной отладки программ дискретного ввода-вывода в среде MPLAB IDE.

Содержанием работы является отладка программы для МК типа PIC16F877A, которая анализирует состояние кнопки, подключенной к линиям порта RC1. Нажатие с последующим отпусканием кнопки RC1 вызывает модификацию содержимого ячейки Cnt (в соответствии с вариантом задания), которое отображается на светоизлучающих диодах, подключенных к линиям порта PORTD. Активному (нажатому) состоянию кнопки соответствует низкий потенциал на линии порта PORTC. Пассивному (отжатому) состоянию кнопки соответствует высокий потенциал, который обеспечивается подтягивающими резисторами.

Для отладки программы предлагается использовать технологию нисходящей отладки. Сначала выполняется отладка основной программы, а затем подпрограмм.

**6.2 Создание проекта и трансляция программы**

6.2.1 В папке PIC создайте свою рабочую папку. Скопируйте в нее подготовленный файл с исходным текстом программы lr6.asm. Создайте новый проект с помощью мастера *Project>Project Wizard*. Задайте имя проекта lr6, включите в проект исходный текст программы lr6.asm.

6.2.2 Откройте окно с исходным текстом программы. При выполнении автономной отладки основной программы необходимо закомментировать оператор вызова подпрограммы задержки D5ms. Выполните трансляцию программы (команда *Project>Make*). Если программа содержала синтаксические и другие формальные ошибки, исправьте их и повторите трансляцию.

**6.3 Отладка программы**

6.3.1 Активируйте отладчик MPLAB IDE для работы с симулятором (команда *Debugger>Select Tool>MPLAB SIM*). Настройте симулятор для выполнения автономной отладки данного приложения (команда *Debugger>Setting*). В окне *Simulator Setting* на закладке *Osc/Trace* задайте значение тактовой частоты *Processor Frequency* 12 MHz.

Выполните команду *View> Watch*, чтобы создать окно наблюдения интересующих ячеек регистрового файла: PORTC, PORTD, Cnt.

6.3.2 Для автономной отладки программы необходимо иметь средства имитации нажатия кнопок. Для этих целей можно использовать асинхронные стимулы. Чтобы создать асинхронный стимул, выберите пункт меню *Debugger> Stimulus>New Workbook* для открытия диалогового окна настройки стимулов. В открывшемся окне *Stimulus* на закладке *Asynch* содержится таблица, каждая строка которой позволяет задать атрибуты одного стимула. Установите курсор на первой строке в столбце *Pin/SFR* и задайте линию порта ввода RC1. В столбце *Action* выберите значение *Set High* (высокий уровень). Это значение будет имитировать действие подтягивающего резистора. В следующей строке таблицы создайте стимул для RC1 для имитации воздействия кнопки. Для этого в столбце *Action* выберите значение *Pulse Low* (импульс низкого потенциала), а в столбце *Width* значение длительности импульса в единицах времени, указанных в столбце *Units*. Длительность этих импульсов в реальном устройстве определяется временем нажатия кнопки и составляет доли секунды. При моделировании целесообразно задать существенно меньшие значения, чтобы сократить время моделирования. Вполне достаточно установить 100 cyc.

6.3.3 Необходимо выполнить тестирование программы. Сначала следует задать исходное (пассивное) значение разряда RC1=1. Произвольно изменить содержимое PORTC не удастся, поэтому произведите сброс МК (*Debugger> Reset*), затем нажмите кнопку *Fire* стимула RC1 типа *Set High*, выполните одну команду программы (*Debugger>Step Into*). Проконтролируйте установку значения разряда 1 порта PORTC в окне наблюдения.

Проверку работы программы удобно выполнять в режиме анимации (быстрого пошагового режима), который активируется командой *Debugger >Animate*. В этом режиме после выполнения очередной команды обновляется содержимое всех открытых окон системы MPLAB IDE и только затем управление передается следующей команде программы.

Нажмите кнопку *Fire* стимула RC1 типа *Pulse Low* на панели контроллера стимулов. Проследите за изменением соответствующего разряда PORTC в окне наблюдения. Убедитесь, что «нажатие» кнопки вызывает необходимую модификацию переменной Cnt. Повторите эту проверку необходимое число раз. Занесите в отчет формируемую последовательность значений переменной Cnt. Остановите процесс выполнения программы нажатием кнопки *Halt*.

6.3.4 Далее по плану выполняется отладка подпрограмм INT и D5ms. Восстановите операторы вызова подпрограммы задержки D5ms в исходном тексте программы. Выполните трансляцию программы.

Контролировать сохранение точки возврата можно в окне стека. Откройте окно стека командой *View>Hardware Stack*. Для контроля формируемых интервалов времени откройте окно секундомера *Debugger>Stopwatch*.

6.3.5 Чтобы проконтролировать возникновение и обработку прерываний в режиме анимации, потребуется слишком много времени. Более рационально использовать режим прогона до точки останова. Установите точку останова на векторе прерывания. Запустите выполнение программы командой *Debugger> Run*. Убедитесь, что происходит передача управления на вектор прерывания. По показаниям секундомера определите интервалы времени между последовательными прерываниями. Измените адрес точки останова так, чтобы можно было определить время выполнения цикла программы. Занесите результаты измерений в отчет.

6.3.6 Используя кабели, подключите:

- к разъему порта PORTC отладочного стенда периферийный модуль с установленными кнопками и часами реального времени;

- к разъему порта PORTD отладочного стенда периферийный модуль с установленными светоизлучающими диодами. Подключите отладочный стенд к USB порту персональному компьютеру.

Активируйте отладчик MPLAB IDE для работы с PICkit2 (команда *Debugger>Select Tool>PICkit2*). Если в окне *Output* на закладке *MPLAB PICkit2* появится сообщение об ошибке, проверьте разъемы и соединительные кабели.

После установления связи и самотестирования выполните загрузку целевого приложения в микроконтроллер отладочного стенда по команде *Debugger>Program*.

Нажмите кнопку панели *Run* или выберите команду *Debugger>Run*. По состоянию линейки светодиодов, подключенных к выводам PORTD, проверьте реакцию МК на нажатие кнопки. Сравните с заданным поведением. Нажмите кнопку панели *Halt* или выберите команду *Debugger>Halt*, чтобы остановить выполнение программы. Сбросьте МК, выбрав *Debugger> Reset> Processor Reset*.

6.3.7 Если результаты проверки Вас разочаровали, то в данном случае, вероятнее всего, причина в настройке портов или дребезге контактов. Проверьте правильность управляющих слов, загруженных в регистры TRISC, TRISD, OPTION\_REG. Проверьте правильность реализации функции подавления дребезга контактов. После исправления ошибок и трансляции программы, повторите загрузку целевого приложения в МК и тестирование программы.

В случае успеха отключите отладчик системы MPLAB IDE командой *Debugger> Select Tool>None*. Отключите кабель от USB-порта компьютера. Закройте проект командой *Project>Close*. В противном случае необходимо локализовать и устранить ошибки в программе.

**6.4 Оформление работы**

Отчет по лабораторной работе должен содержать:

- схему алгоритма программы;

- исходный текст программы;

- расчетные и измеренные значения задержек;

- выводы.

**6.5 Контрольные вопросы**

Поясните по схеме алгоритма работу программы.

Как выполняется отладка программ, имеющих сложную структуру?

Как в среде симулятора можно имитировать изменения внешних сигналов на линиях портов ввода МК?

Каким образом выполняется программное подавление «дребезга контактов»?

Поясните, каким образом можно настроить таймер TMR0 на требуемый режим работы.

Как выполняется обработка запросов прерываний?

Каким образом можно оценить длительность формируемых временных интервалов с использованием симулятора?

Каким образом можно отслеживать выполнение программы с использованием внутрисхемного отладчика PICkit2?

**7 Отладка программы аналогового ввода**

**7.1 Описание работы**

Объектом отладки в работе является программа, которая выполняет запуск модуля АЦП, обрабатывает результаты преобразования и выводит их на светодиодные индикаторы.

**7.2 Создание проекта и трансляция программы**

7.2.1 В папке PIC создайте свою рабочую папку. Скопируйте в нее подготовленный файл с исходным текстом программы lr7.asm. Создайте новый проект с помощью мастера *Project>Project Wizard*. Задайте имя проекта lr7, включите в проект исходный текст программы lr7.asm.

7.2.2 Откройте окно с исходным текстом программы. Выполните трансляцию программы (команда *Project>Make*). Если программа содержала синтаксические и другие формальные ошибки, исправьте их и повторите трансляцию.

**7.3 Отладка программы**

7.3.1 Используя кабели, подключите:

- к разъему порта PORTA отладочного стенда потенциометр;

- к разъему порта PORTD отладочного стенда периферийный модуль с установленными светоизлучающими диодами.

7.3.2 Подключите отладочный стенд к USB порту персонального компьютера. Активируйте отладчик MPLAB IDE для работы с PICkit 2 (команда *Debugger>Select Tool>* *PICkit 2*). После установления связи и самотестирования выполните загрузку целевого приложения в микроконтроллер отладочного стенда по команде *Debugger>Program*.

Нажмите кнопку панели *Run* или выберите команду *Debugger>Run*. По состоянию линейки светодиодов, подключенных к выводам PORTD, проверьте реакцию МК на изменение положения движка потенциометра. Определите границы диапазона кодовых эквивалентов входного напряжения. Проведите измерение значений входного напряжения и зафиксируйте в отчете значения кодовых эквивалентов. Нажмите кнопку панели *Halt* или выберите команду *Debugger>Halt*, чтобы остановить выполнение программы.

Если результаты проверки Вас разочаровали, то в данном случае, вероятнее всего, причина в настройках портов или дребезге контактов. Проверьте правильность управляющих слов, загруженных в регистры TRISA, TRISD. После исправления ошибок и трансляции программы, повторите загрузку целевого приложения в МК и тестирование программы.

В случае успеха отключите отладчик системы MPLAB IDE командой *Debugger> Select Tool>None*. ***После этого можно отключить кабель от USB-порта компьютера.*** Закройте проект командой *Project>Close*.

**7.4 Оформление работы**

Отчет по лабораторной работе должен содержать:

- схему алгоритма и исходный текст программы;

- результаты тестирования программы;

- выводы.

**7.5 Контрольные вопросы**

Поясните по схеме алгоритма работу программы.

Как при автономной отладке можно проконтролировать корректность взаимодействия с модулем АЦП?

Как в среде симулятора можно имитировать выполнение аналого-цифрового преобразования?

**8 Отладка программы символьного вывода**

**8.1 Описание работы**

Цель работы - получить практические навыки выполнения автономной и комплексной отладки программы вывода символьной информации.

Объектом отладки в работе является программа, которая должна выполнять модификацию содержимого ячейки Cnt. Содержимое ячейки Cnt должно отображаться на 7-сегментных индикаторах. Для управления индикаторами в отладочном стенде применен режим динамического управления.

**8.2 Создание проекта и трансляция программы**

8.2.1 В папке PIC создайте свою рабочую папку. Скопируйте в нее подготовленный файл с исходным текстом программы lr8.asm. Создайте новый проект с помощью мастера *Project>Project Wizard*. Задайте имя проекта lr8, включите в проект исходный текст программы lr8.asm.

Откройте окно с исходным текстом программы. Выполните трансляцию программы (команда *Project>Make*). Если программа содержала синтаксические и другие формальные ошибки, исправьте их и повторите трансляцию.

**8.3 Отладка программы**

8.3.1 Используя кабели, подключите:

- к разъему порта PORTC отладочного стенда периферийный модуль с установленными кнопками и часами реального времени;

- к разъемам портов PORTB и PORTD отладочного стенда периферийный модуль с установленными 7-сегментными индикаторами. Подключите отладочный стенд к USB порту персональному компьютеру.

Активируйте отладчик MPLAB IDE для работы с PICkit2 (команда *Debugger>Select Tool>PICkit2*). Если в окне *Output* на закладке *MPLAB PICkit2* появится сообщение об ошибке, проверьте разъемы и соединительные кабели.

После установления связи и самотестирования выполните загрузку целевого приложения в микроконтроллер отладочного стенда по команде *Debugger>Program*.

8.3.2Активируйте выполнение программы в режиме реального времени. Проверьте реакцию МК на нажатие кнопки, фиксируйте результат на 7-сегментных индикаторах. Проконтролируйте правильность отображения информации. Нажмите кнопку панели *Halt* или выберите команду *Debugger>Halt*, чтобы остановить выполнение программы.

В случае успеха отключите отладчик системы MPLAB IDE командой *Debugger> Select Tool>None*. Отключите кабель от USB-порта компьютера. Закройте проект командой *Project>Close*. В противном случае необходимо локализовать и устранить ошибки в программе.

**8.4 Оформление работы**

Отчет по лабораторной работе должен содержать:

- схему алгоритма программы;

- исходный текст программы;

- результаты тестирования программы;

- выводы.

**8.5 Контрольные вопросы**

Поясните по схеме алгоритма работу программы.

Каким образом выполняется вывод информации на 7-сегментные индикаторы отладочного стенда?

**Приложение В**

**Методические рекомендации к выполнению курсовой работы**

**1Анализ технического задания**

Представления об объекте проектирования необходимо разделить на иерархические уровни: функциональный и схемотехнический. При этом проектируемое устройство рассматривается как сис­тема, которую можно разделить на подсистемы более низкого порядка: проектируемое устройство функциональные подсхемы (звенья, узлы)  электрорадиоэлементы (транзисторы, резисторы и т.д.).

При проектировании аналоговых устройств на функциональном иерархиче­ском уровне разрабатываются структурные или функциональные схемы, эле­ментами которых являются типовые аналоговые звенья и узлы. В процессе схе­мотехнического проектирования аналоговых устройств выполняют разработку функциональных подсхем звеньев и узлов, а также функциональные и принципи­альные электрические схемы для аналогового устройства в целом, элементами которых являются серийные интегральные схемы и электрорадиоэлементы.

**2 Функциональное проектирование**

Методика функционального проектирования аналоговых устройств во мно­гом определяется их принадлежностью к определенному классу устройств. Одним из распространенных методов синтеза активных RC-фильтров явля­ется каскадная реализация, которая предусматривает: разложение передаточной функции фильтра на сомножители второго и первого порядка и реализацию полученного разложения каскадным соединением звеньев вто­рого и первого порядка, взаимодействие между которыми пренебрежимо мало.

Сначала необходимо решить задачу аппроксимации заданной АЧХ фильтра. В ходе решения этой задачи определяют передаточную функцию фильтра в области комплексной частоты s в виде произведения сомножителей

n/2 wi2

H(s) = П Hoi ——————— , (5)

i=1 s2 + (wi/Qi)s + wi2

w1 (n+1)/2 wi2

H(s) = Ho1 ——— П Hoi ——————— (6)

s + w1 i=2 s2 + (wi/Qi)s + wi2

для фильтра нижних частот четного и нечетного порядка;

n/2 s2

H(s) = П Hoi ——————— , (7)

i=1 s2 + (wi/Qi)s + wi2

s (n+1)/2 s2

H(s) = Ho1 ——— П Hoi ———————— (8)

s + w1 i=2 s2 + (wi/Qi)s + wi2

для фильтра верхних частот четного и нечетного порядка;

n/2 (wi /Qi)s

H(s) = П Hoi ———————— (9)

i=1 s2 + (wi/Qi)s + wi2

для симметричного полосового фильтра четного порядка,

где n - порядок фильтра;

wi - частота пары комплексно-сопряженных полюсов;

w1 - частота вещественного полюса;

Qi - добротность пары комплексно-сопряженных полюсов.

Исходные данные для решения задачи аппроксимации представляются в виде графика допусков (рисунок 1), но поля допусков неравномерности АЧХ в по­лосе пропускания и задерживания следует задавать меньше чем в ТЗ для обес­печения запасов работоспособности. Запасы работоспособности необходимы для обеспечения работоспособности фильтра при отклонениях параметров эле­ментов от их номинальных значений из-за технологического разброса, изменения условий эксплуатации и старения.

Следующими проектными процедурами являются процедуры структурного и параметрического синтеза, в ходе выполнения которых формируют структур­ную схему фильтра и определяют численные значения параметров ее элементов. В каче­стве элементов структурной схемы целесообразно использовать звенья с пере­даточными функциями сомножителей Нi(s), входящих в состав функций (5) – (9). В результате передаточная функция n-го порядка представляется в виде

k

H(s) = П Нi(s), (11)

i=1

где k = n/2 для четных значений n, k = (n-1)/2 +1 для нечетных n.

При выполнении этих проектных процедур имеется две степени свободы:

- порядок следования звеньев (m звеньев можно каскадно соединить m! различными способами);

- распределение усиления между звеньями при выполнении условия

n/2

П H0i = H' = const, (12)

i=1

причем для ФНЧ и ФВЧ H' = Ho, для ПФ - H' > Ho. Этими степенями свободы можно воспользоваться для оптимизации харак­теристик фильтра по такому критерию, как максимум динамического диапазона для исключения перегрузки звеньев фильтра при линейном режиме работы по­следнего звена. При этом максимумы каждой k-й промежуточной АЧХ (т.е. мо­дуля передаточной функции первых k звеньев) должны быть как можно ближе равны максимуму АЧХ всего фильтра. Как возможный вариант получения оптимального по данному критерию решения можно использовать порядок следования звеньев с нарастающей добротностью. То есть, на входе схемы следует включать (если требуется) звено первого порядка, а затем звенья второго порядка в порядке увеличения добротности полюсов передаточной функции.

Для установления соответствия характеристик проектируемого устройства требованиям технического задания необходимо выполнить анализ АЧХ всего фильтра в соответствующих диапазонах частот. Для оценки оптимальности сформированной схемы по критерию максимума динамического диапазона не­обходимо выполнить анализ промежуточных АЧХ в полосе пропускания. Опти­мальное решение можно найти путем частичного или полного перебора различ­ных вариантов. В случае, если требования задания не выполняются, необходимо принять решение относительно дальнейшего хода проектирования:

- проверить расчет параметров звеньев схемы;

- проверить правильность формирования структурной схемы;

- заново решить задачу аппроксимации.

Для решения задачи аппроксимации АЧХ фильтров можно использовать программу APPR1. Программа запрашивает у пользователя исходные данные (требования к АЧХ, вид аппроксимации), определяет порядок фильтра и рассчи­тывает параметры передаточной функции (5) - (9) с точностью до постоянного множителя. Т.е., коэффициенты H0i программа не рассчитывает, их необходимо определить самостоятельно из условия (12).

Формирование структурной схемы необходимо выполнить в среде схем­ного графического редактора Capture. Звено с передаточной функцией, представленной в операторной форме, входит в библиотеку Abm.lib под именем Laplace. Аналитические представления полиномов передаточной функции каждого звена должны быть заданы в полях NUM для числителя и DENOM для знаменателя.

Пример - При решении задачи аппроксимации были выбраны значения неравномерности АЧХ в полосе пропускания  = 0.5 дБ и подавления в полосе задерживания 42 дБ. Этим обеспечиваются запасы рабо­тоспособности:

- в полосе пропускания (1 - 0.5)/2 = + - 0.25 (дБ);

- в полосе задерживания минус (40 - 42) = 2 (дБ). Программа APPR1 при использовании чебышевской аппроксимации рассчитала следующие пара­метры передаточной функции (5):

n = 6;

w1 = 7.469E3 , Q1 = 6.836E-1, w1 / Q1 = 1.093E4, w1 2 = 5.578E7;

w2 = 1.448E4 , Q2 = 1.810, w2 / Q2 = 7.998E3, w2 2 = 2.096E8;

w3 = 1.906E4 , Q3 = 6.513, w3 / Q3 = 2.927E3, w3 2 = 3.635E8.

Структурная схема фильтра шестого порядка представлена на рисунке 2. Она содержит три звена второго порядка.

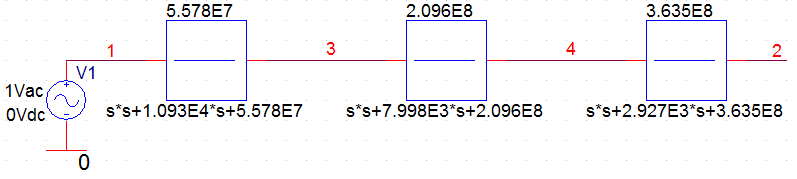


Рисунок 2

Корректность результатов синтеза следует проверить выполнением анализа – параметрической верификации. Для этого необходимо рассчитать АЧХ схемы, по результатам расчета определить основные выходные параметры и определить выполняются ли условия работоспособности (4). Для расчета АЧХ с помощью программы PSPICE к входу схемы следует подключить источник напряжения V1 типа VAC из библиотеки Source. Результаты расчета АЧХ фильтра шес­того порядка, выполненные с помощью программы PSPICE, приведены на ри­сунках 3 и 4.

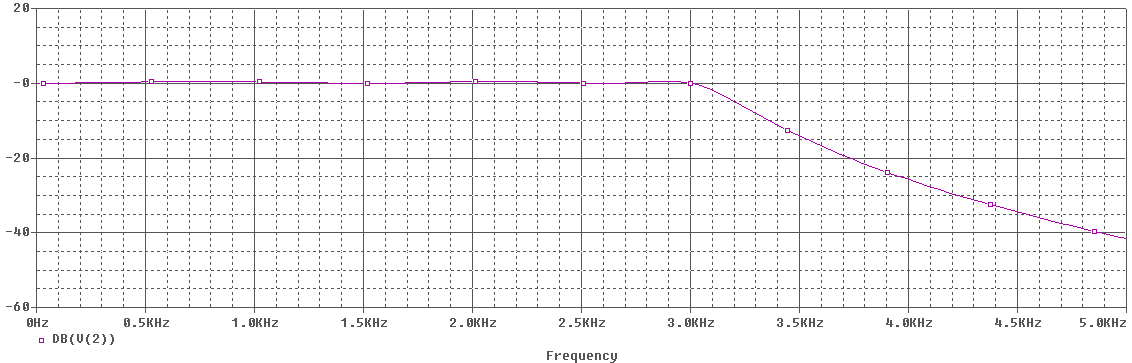


Рисунок 3

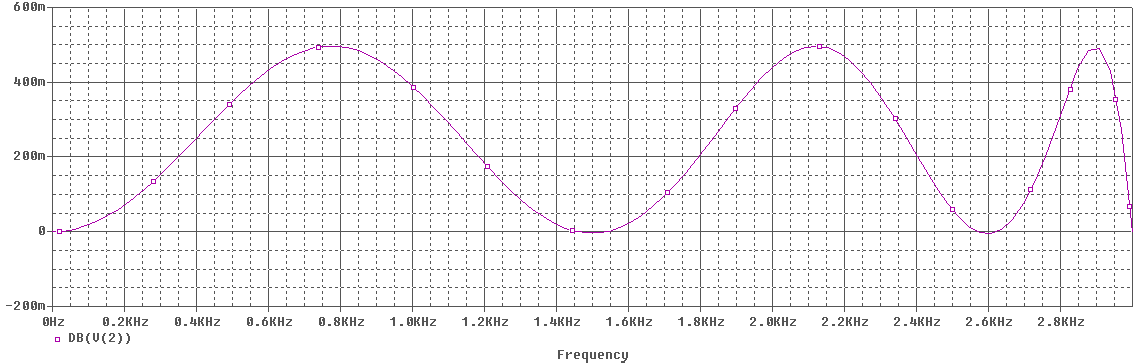


Рисунок 4

Из полученных характеристик следует, что АЧХ имеет неравномерность в полосе пропускания (0 - 3000 Гц) 0.5 дБ, в полосе задерживания (свыше 5000 Гц) коэффициент пере­дачи не превышает минус 41.6 дБ. Т.к. требования задания выполняются, данный ва­риант структурной схемы можно рассматривать как результат этапа функционального проектирования и использовать его как часть исходных данных при схемотехническом проектировании.

1. **Схемотехническое проектирование**

Исходными данными для этапа схемотехнического проектирования являются требования к основным параметрам проектируемого устройства, а также результаты функционального проектирования, представленные в виде структурной схемы и требований к функциям и параметрам ее звеньев. Схемо­техническое проектирование аналоговых устройств начинают с уточнения исходных данных, а затем выполняют проектную процедуру синтеза варианта структуры (схемного решения).

Типовые схемы звеньев первого и второго порядка для ФНЧ, ФВЧ и ПФ хранятся в банке схемных решений (БСР). Функциональную электрическую схему фильтра можно построить из найденных в БСР подсхем, соединяя их входы и вы­ходы в соответствии со структурной схемой.

Как правило, численные значения параметров элементов схем, найденных в БСР, не обеспечивают необходимые характеристики проектируемого устройства. Поэтому требуется решать задачи параметрического синтеза. Проектная проце­дура параметрического синтеза в общем случае предусматривает решение задач определения значений параметров, их допусков и типов элементов схем. Первая из этих задач может быть решена, используя инженерные методики, основанные на про­стых аналитических моделях.

При промышленном проектировании расчетные значения параметров эле­ментов необходимо заменить ближайшими номинальными значениями, соответствующими стандартным рядам (для постоянных резисторов и для постоянных конденсаторов ГОСТ 28884-90). Реальные значения параметров элементов рас­пределяются случайным образом около номинальных значений в пределах тех­нологических допусков и приводят к производственному разбросу выходных параметров электронных устройств. Поэтому требуется решать задачу расчета оптимальных допусков на параметры элементов.

Оценку основных свойств полученных решений и проверку их работоспособности выполняют в ходе проектной процедуры анализа. На этапе схемотехнического проектирования обычно выполняют анализ статического режима, переходных процессов и частотных характеристик. В случаях, когда результаты анализа не соответствуют требованиям ТЗ, необходимо установить причины и устранить их путем корректировки результатов параметрического синтеза или модификации структуры схемы. Для анализа определяющих характеристик аналоговых устройств используются методы схемотехнического моделирования.

Пример - Для рассматриваемого примера в БСР были найдены схемы звеньев ФНЧ второго порядка низкой (Q<2) добротности lp2lq1 (рисунок 4), средней (2<Q<20) добротности lp2mq1 (рисунок 5).

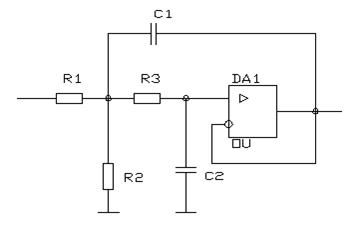


Рисунок 4

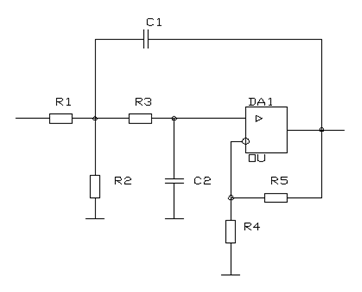


Рисунок 5

Параметры элементов звеньев были рассчитаны по инженерным методикам с использованием шаблонов расчета из файлов lp2lq1.txt и lp2mq1.txt соответственно.

Методика расчета схемы LP2LQ1:

//Исходные данные

K=1

W=7469

Q=0.6836

//Задаться значениями C1 и C2 < C1/(4\*Q^2)

C1=0.00000001

C2=0.0000000047

//Вычислить значения

A=C1/(2\*Q^2\*C2)-1

P=A+sqrt(A^2-1)

R=1/(W\*sqrt(P\*C1\*C2))

R3=P\*R;

if (K.eq.1) then

R1=R

// - а R2 - исключается из схемы;

endif

if (K.lt.1) then

R1=R/K;

R2=R/(1-K)

endif

end.

Методика расчета схемы LP2MQ1:

//Исходные данные

K=1

Q=6.513

W=19065

//Задаться значениями C и R;

C1=0.00000001

С2=0.00000001

R=2200

//Вычислить значения

P=C1/C2/(36\*Q^2)\*(sqrt(1+12\*Q^2\*(1+C2/C1))+1)^2;

R=1/(W\*sqrt(P\*C1\*C2));

R3=P\*R;

R5=R4\*(C2/C1\*(1+P)-sqrt(P\*C2/C1)/Q);

K0=1+R5/R4;

X=Q\*K0^2\*sqrt(C1/P/C2)

Если K>K0, то R1=K0/K\*R; R2=K0/(K0-K)\*R

иначе K=K0; R1=R, а R2 - исключается из схемы

end.

Результаты расчета звеньев по данным методикам приведены в таблице 1. Для звеньев 1 и 2 из схемы lp2lq1.sch исключается резистор R2.

Таблица 1

Звено 1 2 3

Подсхема lp2lq1 lp2lq1 lp2mq1

Элементы Значения параметров

C1, Ф 1.0E-8 1.0E-8 1.0E-8

C2, Ф 4.7E-9 7.5E-10 1.0E-8

R1, Ом 1.357E4 2.210E4 1.607E4

R2, Ом - - 1.017E3

R3, Ом 2.810E4 2.877E4 4.420E3

R4, Ом - - 2.2E3

R5, Ом - - 3.477Е3

Сформированная функциональная схема фильтра приведена на рисунке 6, элементы схемы имеют сквозную нумерацию позиционных обозначений.

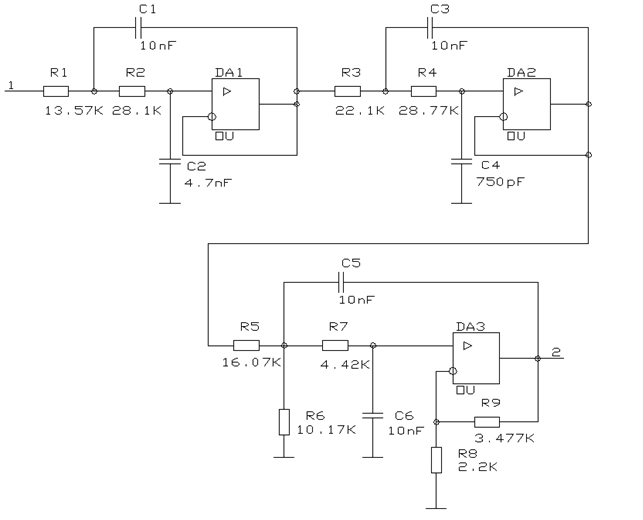


Рисунок 6

Корректность результатов синтеза функциональной схемы следует проверить выполнением анализа – рассчитать АЧХ. Если при построении схемы и расчете параметров элементов не были допущены ошибки, то форма и параметры АЧХ должны быть, как у структурной схемы. В противном случае следует найти и исправить ошибки.

**4 Контрольные вопросы для защиты работы**

1) Поясните назначение разработанной схемы.

2) Каким образом задаются требования к АЧХ фильтров?

3) Определите форму и параметры выходного сигнала фильтра (по варианту задания), если на его вход длительное время подается сигнал

Uвх(t) = U0 + Um ·Sin 2πft ,

значения параметров U0 , Um , f получить у преподавателя.

4) Перечислите выполненные этапы проектирования.

5) Перечислите выполненные на каждом этапе проектные процедуры. Поясните их содержание.