ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

КАФЕДРА РАДИОТЕХНИЧЕСКИХ СИСТЕМ

УТВЕРЖДАЮ

Зав. кафедрой радио-

технических систем

\_\_\_\_\_\_\_\_\_\_\_\_ В.И.Кошелев

"\_\_\_"\_\_\_\_\_\_\_\_\_\_ 2007 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по дисциплине:

# «ЦИФРОВЫЕ УСТРОЙСТВА и МИКРОПРОЦЕССОРЫ»

на тему: Цифровой фильтр высокой частоты

Автор работы Косс А.В.

Специальность 210305 гр. 410

Руководитель Соколов Ю.П.

Работа защищена \_\_\_\_\_\_\_\_\_\_\_\_\_

Члены комиссии: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка, дата)

Рязань 2007 г

РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра радиотехнических систем

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ**

По дисциплине «Цифровые устройства и микропроцессоры»

Студент *Косс А.В.*  код *F0b12\_70* группа *410*

1. Тема *Цифровой фильтр высокой частоты*

2. Срок представления проекта к защите « *23* » *апреля* 2007 г.

3. Исходные данные для проектирования:



3.1.  *Линейное разностное уравнение фильтра:*

3.2.  *Порядок фильтра: M = 12*

3.3.  *Коэффициенты фильтра:*

|  |  |  |  |
| --- | --- | --- | --- |
| b1  b2  b3  b4  b5  b6  b7 | 0.023815103093366  0.014780763864104  -0.067269084707391  0.054715869201380  0.089243271343775  -0.295613381939767  0.394551759773816 | b8  b9  b10  b11  b12  b13 | -0.295613381939767  0.089243271343775  0.054715869201380  -0.067269084707391  0.014780763864104  0.023815103093366 |

3.4.  *Разрядность данных и коэффициентов: n = 8*

3.5.  *Входной сигнал – 8-разрядный параллельный дополнительный код*

3.6.  *Выходной сигнал – аналоговый, диапазон изменения (0… + 2,5)В*

3.7.  *Цифро-аналоговый преобразователь – AD9708*

3.8.  *Частота дискретизации: 2500 Гц*

3.9.  *Тактовая частота микроконтроллера: 12 МГц*

3.10.  *Микроконтроллер КР1830ВЕ51*

4. Содержание пояснительной записки курсового проекта

4.1. Титульный лист

4.2. Задание на курсовой проект

4.3. Содержание

4.4. Введение

4.5. Анализ, формализация и декомпозиция задачи

4.6. Разработка и описание общего алгоритма функционирования устройства

4.7. Разработка и обоснование структурной схемы устройства

4.8. Составление и описание принципиальной схемы устройства

4.9. Разработка и отладка программы на языке ассемблера

4.10. Определение быстродействия, импульсной и переходной характеристик устройства

4.11. Заключение

4.12. Список использованных источников

4.13. Приложение

5. Перечень графического материала: *схема принципиальная электрическая*

**Руководитель проекта** *05.02.2007 г. Соколов Ю.П.*

подпись

**Задание принял к исполнению** « » *2007 г*.

инициалы, фамилия

дата

подпись

**Содержание**

**цифровой фильтр программа ассемблер**

1 Введение.

2 Анализ, формализация и декомпозиция задачи

2.1. Распределение портов и организация взаимодействия с внешними устройствами

2.2Выбор режима таймера-счетчика

2.3Масштабирование коэффициентов и упрощение разностного уравнения.

2.4 Разработка и описание общего алгоритма функционирования устройства.

2.5. Электрическая принципиальная схема фильтра

3 Разработка и отладка программы на языке ассемблера

4 Оценка быстродействия и определение импульсной и переходной характеристик

5 Заключение

6 Список литературы

**1. Введение**

Цифровые методы обработки информации все более широко внедряются во многие области техники связи и управления, в частности, в область фильтрации сигналов.

Это обусловлено тем, что эти методы обеспечивают значительно более высокую точность по сравнению с методами аналоговой обработки, а также тем, что цифровые устройства надежны и удобны в эксплуатации.

Особо важное место, среди различных методов фильтрации занимают линейные методы; устройства, реализующие цифровые методы фильтрации сигналов, получили название цифровых фильтров. На практике цифровые фильтры реализуются либо в виде программы на универсальной цифровой вычислительной машине, либо в виде специализированного вычислительного устройства с электронными схемами ввода и вывода информации.

По сравнению с аналоговыми фильтрами цифровые фильтры обладают рядом важных достоинств. К ним, прежде всего, относятся высокая стабильность и точность, не зависящие от воздействия внешних условий, простота изменения характеристик и возможность использования в качестве адаптивных устройств; при эксплуатации цифровых фильтров не возникают задачи согласования нагрузок, они могут работать в диапазоне сверхнизких частот; они могут обладать линейными фазовыми характеристиками и т.д.

Важно отметить, что цифровые фильтры практически реализуются на интегральных цифровых логических элементах, вследствие чего они могут быть компактными, недорогими и высоконадежными устройствами. Вместе с тем следует отметить, что в отличие от аналоговых цифровым фильтрам присущи некоторые специфические погрешности, обусловленные дискретизацией и квантованием аналоговых сигналов (при выполнении арифметических операций в вычислительных устройствах).[1]

В задании на курсовой проект даны частота дискретизации, разностное уравнение и коэффициенты фильтра, что в свою очередь определят характеристики фильтра.

Следовательно в курсовом проекте необходимо:

1. Разработать электрическую принципиальную схему фильтра;
2. Разработать рабочую программу на языке ассемблера, которая будет обеспечивать взаимодействие и работоспособность БИС входящих в состав фильтра, для реализации заданных свойств фильтра;
3. Провести заключительный анализ характеристик спроектированного фильтра, и если они не удовлетворяют заданным внести корректировки в рабочую программу.

**2. Анализ, формализация и декомпозиция задачи**

Формализация задачи при выполнении всех этапов проектирования предполагает представление выполнения функций фильтра на языке логических и математических связей, в виде логических, математических моделей и т.п. Задачей формализации является согласование структурное, логическое, алгоритмическое, электрическое всех аппаратных и программных модулей устройства, обеспечивающее разработку работоспособной электрической схемы (аппаратная часть) и работоспособной программы фильтра, согласованной с аппаратной частью [2].

**2.1. Распределение портов и организация взаимодействия с внешними устройствами**

В техническом задании на курсовой проект заданы базовый микроконтроллер, АЦП и формат входных и выходных сигналов. Микроконтроллер КР1830ВЕ51 представлен на рис. 1.



рис. 1[3][[1]](#footnote-1)\*

Для ввода и вывода данных будем использовать два двунаправленных порта: P2 – для ввода данных, P0 – для вывода. Необходимо организовать взаимодействие микроконтроллера и внешнего устройства таким образом, что бы готовые данные поступали с заданной частотой дискретизации. Нулевой бит порта P3 (P3.0) будем использовать в качестве линии запуска внешнего устройства. По этой линии на вывод внешнего устройства будет поступать импульс нулевого или высокого (в зависимости от внешнего устройства) уровня. Примем в программе импульс запуска - низкого уровня. Сигнал о готовности входного отсчета будет поступать на вывод P3.2 микроконтроллера. В программе будем считать, что данные готовы, если данный бит равен 0. Примем, что входное внешнее устройство имеет третье состояние выходных портов, тогда для того, что бы входные отсчеты поступили на порт микроконтроллера будем подавать импульс низкого уровня на соответствующий вход внешнего устройства. Выделим для этого линию P3.1. Запуск внешнего устройства целесообразно делать в начале программы, что позволит сэкономить время необходимое для ожидания готовности отсчета. Другими словами внешнее устройство параллельно с выполнением программы подготовит входные отсчеты еще до момента требующего их ввода. Или же, в случае если внешнее устройство не успеет подготовить отсчеты к тому времени, когда в программе предусмотрен их ввод; мы сэкономим время равное времени выполнения программы до этого момента.

В отличие от ввода отсчетов, их вывод осуществляется сразу «по готовности». Заданный в теоретическом задании ЦАП работает в состоянии постоянной готовности. И что бы ЦАП не считывала «неправильные» данные с выходного порта (это связано с тем, что выходной отсчет приходит неравномерно на каждый бит порта), будем формировать импульс считывания поданный на соответствующий вход ЦАП. Выделим для этого линию P1.0.

Для промежуточного хранения данных в микроконтроллере предусмотрено 4 регистровых банка RB0…RB3 по восемь регистров в каждом (R0…R7) (рис. 2).

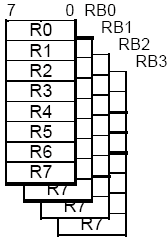


рис. 2[3]

Регистровые банки переключаются полем RS слова состояния программы PWS. [3]

Выберем для основной программы 1 регистровый банк (RB1). Для рабочей программы определим третий регистровый банк (RB3).

**2.2 Выбор режима таймера-счетчика**

Для настройки таймера счетчика T/C0 необходимо определить интервал дескретизации:

TД=1/FД=400мкс

и количество машинных циклов, укладывающихся на интервале дискретизации:

N= TД/TМЦ=400.

Следовательно режим с автоматической перезагрузкой (второй режим) не подойдет, т.к. в нем 8-ми разрядная сетка (максимальное количество машинных циклов 256). Поэтому выберем первый режим таймера-счетчика, который использует 16-ти разрядный счетный регистр. При этом перезагрузку 16-азрядного счетчика необходимо будет делать программно перед каждым циклом вычисления.

Запишем в память микроконтроллера число -400(10)=FE70(16)

Time0L <= 70h -Младший байт числа -400 в ДК

Time0H <= FEh -Старший байт числа -400 в ДК

**2.3 Масштабирование коэффициентов и упрощение разностного уравнения**

Для проектирования фильтра в ТЗ дано его разностное уравнение и соответствующие коэффициенты. Так как ячейка памяти может содержать 8ми разрядно двоичное число, причем старший разряд – знаковый, то необходимо провести округление коэффициентов фильтра до семи разрядов.

При помощи программы Matlab выберем способ округления. Для этого построим АЧХ фильтров при различном округлении (рис. 3) и сравним с АЧХ заданного фильтра. На основе сравнения выберем тот способ, который лучше удовлетворяет ТЗ и виду фильтра (ВЧ).

Возьмем способ округления – отбрасыванием дробной части ( зеленый график) (рис. 4).

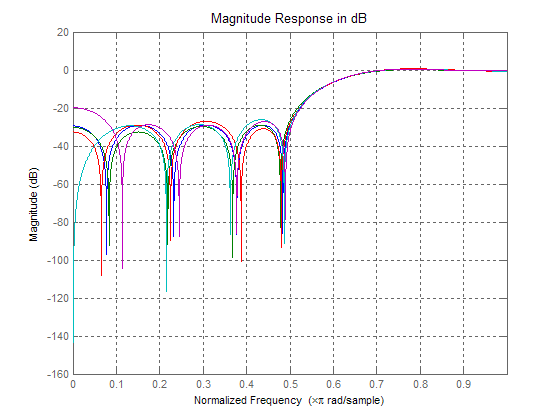


рис. 3

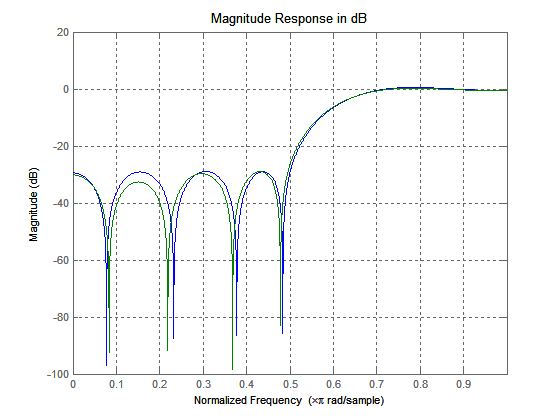


рис. 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Исходные коэффициенты (Dec) | Исходные коэффициенты (Bin) | Округленные коэффициенты\* (Dec) | Округленные коэффициенты\* (Bin) (ПК) | Округленные коэффициенты (Hex) |
| B1 | 0.023815103093366 | 0,00000110 | 3 | 00000011 | 03 |
| B2 | 0.014780763864104 | 0,00000011 | 1 | 00000001 | 01 |
| B3 | -0.067269084707391 | -0,00010001 | -8 | 10001000 | 88 |
| B4 | 0.054715869201380 | 0,00001110 | 7 | 00000111 | 07 |
| B5 | 0.089243271343775 | 0,00010110 | 11 | 00001011 | 0B |
| B6 | -0.295613381939767 | -0,01001011 | -37 | 10100101 | A5 |
| B7 | 0.394551759773816 | 0,01100101 | 50 | 00110010 | 32 |
| B8 | -0.295613381939767 | -0,01001011 | -37 | 10100101 | A5 |
| B9 | 0.089243271343775 | 0,00010110 | 11 | 00001011 | 0B |
| B10 | 0.054715869201380 | 0,00001110 | 7 | 00000111 | 07 |
| B11 | -0.067269084707391 | -0,00010001 | -8 | 00001000 | 88 |
| B12 | 0.014780763864104 | 0,00000011 | 1 | 00000001 | 01 |
| B13 | 0.023815103093366 | 0,00000110 | 3 | 00000011 | 03 |

Сумма модулей исходных коэффициентов:

1.48542670807338

Сумма модулей коэффициентов числителя:

1.43750000000000

В ТЗ дано следующее разностное уравнение:



,М=12

Т.к. заданные коэффициенты симметричны, то разностное уравнение можно преобразовать к следующему виду:



Такое преобразование разностного уравнения позволит сократить программу вычисления отсчета на 7 циклов.

Необходимо учесть, что в некоторых случая сумма двух отсчетов может переполнить 8ми разрядную сетку. Поэтому примем, что входные отсчеты по модулю меньше 0.5(10). Тогда сумма этих отсчетов не будет приводить к переполнению разрядной сетки.

**2.4 Разработка и описание общего алгоритма функционирования устройства**

Представим алгоритм функционирования фильтра следующим образом:

Аппаратно- реализуемые операции Программно- реализуемые операции

Сформировать импульса сброса для входа RST

ИНИЦИАЛИЗАЦИЯ

Основная программа

Таймер\счетчик

Программа

вычисления

отсчетов

ЦАП

Источник сигнала

вкл. питания

сброс при включении питания

Прерывание от TC/0

Запрос данных

Сигнал о готовности данных

Передача данных

Импульс считывания

Вычисленный отсчет

Выходной аналоговый сигнал

При включении фильтра происходит формирование импульса сброса, который обнуляет счетчик команд и приводит фильтр к исходному состоянию.

При инициализации настраивается таймер счетчик, загружается PSW основной программы, очищаются память выделенная под остчеты, устанавливается вершина стека. Необходимо, чтобы программа инициализации начиналась с нулевого адреса (0000h).

Основная программа представляет собой бесконечный цикл

После идет программа вычисления отсчета. В которой предусмотрено: ввод данных (xn), их обработка, в соответствии с разностным уравнением, и вывод данных (yn).

**2.5 Электрическая принципиальная схема фильтра.**

Вых сигнал

Ист.

сигнала

МК

ЦАП

Преобразователь выходного сигнала

1

2

3

5

6

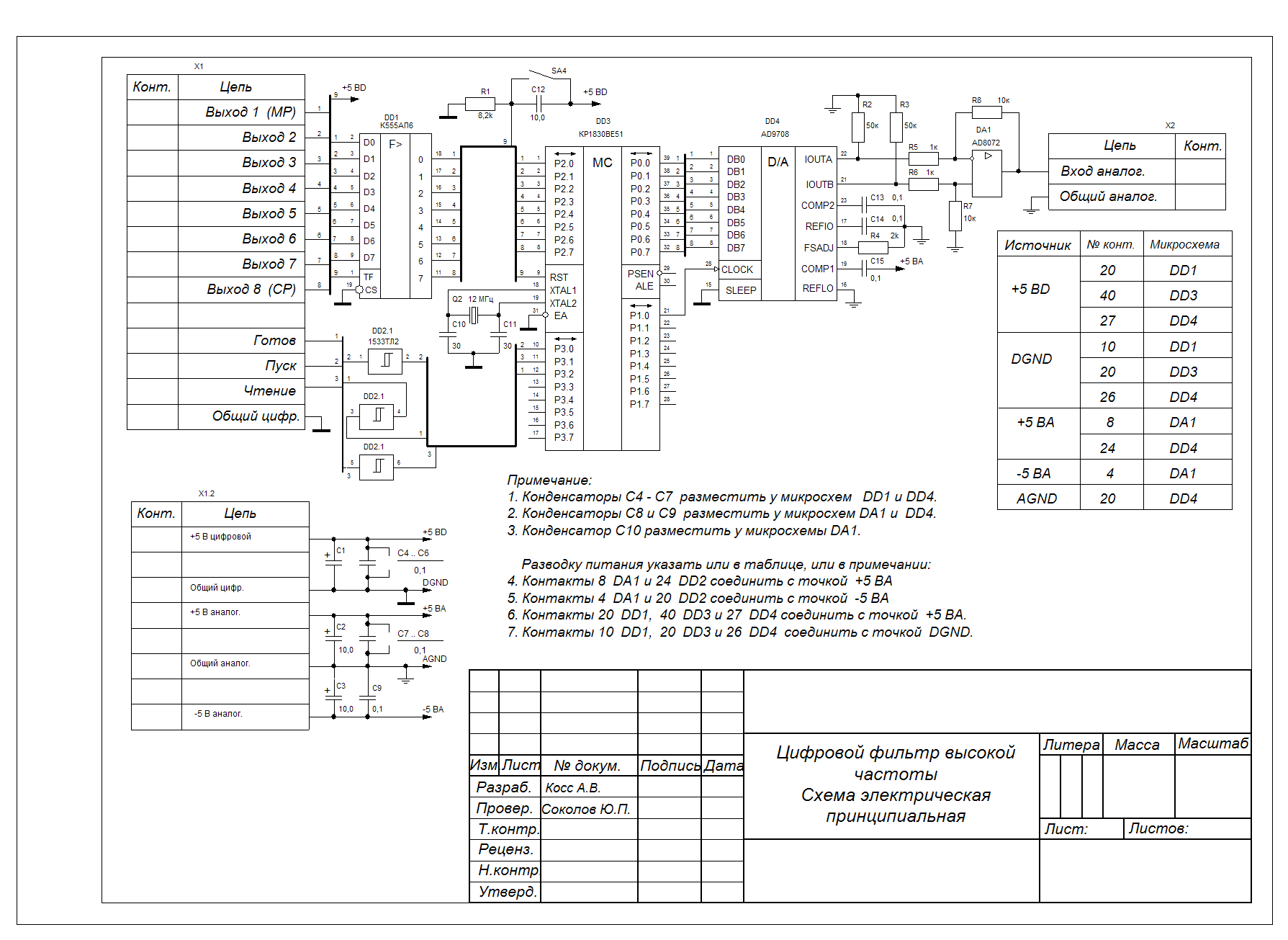
7

4

По линии 1 происходит запуск внешнего устройства. Линия 2 служит для сообщения от внешнего устройства о готовности входных данных. По линии 4 подается импульс, который заставляет внешнее устройство подать готовые данные на вход микроконтроллера.

Между микроконтроллером и ЦАП существует 2 линии связи: линия номер 5 – служит для выдачи готовых данных с МК, а по линии 6 подается импульс, инициирующий чтение готовых данных ЦАП.

По линии 7 преобразованный в аналоговый сигнал поступает на «Преобразователь выходного сигнала», который преобразует сигнал к заданному виду.



В состав схемы, в соответствии с теоретическим заданием, входит микроконтроллер КР1830BE51 и ЦАП АD9708. Подсоединим к микроконтроллеру кварцевый резонатор с частотой 12 МГц, для задания тактовой частоты микроконтроллера. Для его сброса, к входу RST подсоединим RC-цепочку, которая при замыкании ключа будет формировать импульс сброса микропроцессора. Поставим между микроконтроллером и входным устройством буфер К555АП6, служащий для развязки данных, умощнения входного сигнала и повышения помехоустойчивости.

На выходе ЦАП будем использовать дифференциальное включение выходов ЦАП, при котором напряжение *UДИФ* = *UOUT А* – *UOUT В* изменяется в пределах от +0,5 В до -0,5 В. Что бы преобразовать выходное напряжение к заданному виду (*(0… + 2,5)*) необходимо подать напряжение смещения +1.75В и усилить выходной сигнал в 2.5 раза. Для этого используем ДУ AD8072.

Для повышения помехоустойчивости между входным внешним устройством и МК в линии ПУСК, ЧТЕНИЕ, ГОТОВ поставим триггеры Шмитта.

**3. Разработка и отладка программы на языке ассемблера**

Первым делом необходимо разработать секцию инициализации, которая обязательно должна начинаться с нулевого адреса. В секции инициализации предусмотрим следующие макросы:

InitSteck – инициализация стека, запись вершины стека

InitInt – инициализация прерываний, разрешение всех прерываний и прерываний от таймера счетчика, в частности.

InitTimer – инициализация таймера, выбор первого таймера/счетчика, выбор первого режима работы, запрет внешнего управления. Выбираем функцию таймера.

InitCoeff – инициализация коэффициентов, запись констант значений коэффициентов в память (при этом коэффициенты располагаются в памяти последовательно)

InitPSW – инициализация PSW основной программы и программы вычисления отсчетов. В память программы записываются соответствующие значения PSW.

ClearXn – инициализация отсчетов, очистка памяти выделенной под отсчеты.

ReloadTimer – инициализация таймера, остановка таймера, загрузка значений младшего и старшего байтов интервала дискредитации.

После инициализации идет основная программа.

MainProgramm – бесконечный цикл основной программы. Выход по прерыванию от TC0.

Дальше идет секция исполнения, в которой происходит обработка входных данных и вывод их на выходной порт.

RunACP - макрос запуска внешнего устройства. По линии P3.0 на соответствующий контакт устройства подается импульс нулевого уровня.

ReloadTimer – то же что и в секции инициализации

SavePSW – сохранение PSW основной программы и загрузка PSW рабочей программы.

CheckACP – проверка готовности данных от внешнего устройства. Если по линии P3.2 установлен 0, то данные готовы.

InputData – Ввод входных данных. Т.к. выводы внешнего входного устройство имеют третье состояние, то для того что бы данные попали на порт МК, необходимо по линии P3.1 сформировать импульс низкого уровня.

PrepareData – макрос, необходимый для упрощения разностного уравнения. Здесь происходит суммирование отсчетов стоящих у симметричных коэффициентов и запись суммы последовательно в память.

ComputeSample – макрос вычисления отсчета. Содержит в себе макрос очистки двухбайтовой суммы и макрос суммирования с ее накоплением.

ShiftL – сдвиг результата на 1 разряд влево.

OutputData – макрос вывода данных. Здесь результат переводится из дополнительного кода в прямой и затем старший байт отправляется на выходной порт. Формируется импульс по линии P1.0, для фиксации выходного отсчета на ЦАП.

PrepareCycl – последовательный сдвиг отсчетов в памяти.

ReloadPSW – запись PSW рабочий программы и восстановление PSW основной.

RETI – возвращение в основную программу.

Текст программы

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Цифровой фильтр высокой частоты \*

\* Курсовой проект по дисциплине "Цифровые устройства и микропроцессоры" \*

\* Код F0b12\_70 \*

\* Выполнил студент группы 410 Косс Андрей Владимирович \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SPACES ON ;Разрешить пробелы между операндами

FILLCHAR 00h ;Заполнить промежутки между секциями символами 00

\*\*\*\*\*\*\*\*\*\*\*\* ОПРЕДЕЛЕНИЕ КОНСТАНТ \*\*\*\*\*\*\*\*\*\*\*\*

Time0L: EQU 70h ;Младший байт числа -400 в ДК

Time0H: EQU FEh ;Старший байт числа -400 в ДК

Order: EQU 07h ;Число циклов суммирования (общее число слагаемых)

MODE1: EQU 01h ;установка TC0 в режим 1, таймер, НЕ разрешено внешн. упр

Datab1: EQU 06h ;значение коэффициента b1

Datab2: EQU 03h ;значение коэффициента b2

Datab3: EQU 91h ;значение коэффициента b3

Datab4: EQU 0Eh ;значение коэффициента b3

Datab5: EQU 16h ;значение коэффициента b3

Datab6: EQU CBh ;значение коэффициента b3

Datab7: EQU E5h ;значение коэффициента b3

MRBank: EQU 01h ;регистровый банк основной прграммы

RRBank: EQU 18h ;регистровый банк прграммы вычисления отсчёта

Maska1: EQU 7Fh ;маска для получения беззнаковых чисел

Maska2: EQU 00h ;маска для очистки двухбайтового сумматора

Maska3: EQU 00h ;маска для очистки ячеек хранения отсчёта

Shift: EQU 01h ;количество разрядов на которое необходимо сдвинуть результат вычисления

\*\*\*\*\*\*\*\*\*\*\*\* ОПРЕДЕЛЕНИЕ НЕПЕРЕМЕЩАЕМЫХ РЕГИСТРОВ \*\*\*\*\*\*\*\*\*\*\*\*

IE: REG A8h ;Присвоить имя IE регистру разрешения прерываний (ячейка A8h)

TCON: REG 88h ;Присвоить имя TCON регистру управления таймерами/счетчиками

TMOD: REG 89h ;Присвоить имя TMOD регистру режима таймеров/счетчиков

InPort: REG P0 ;Присвоить имя InPort порту P0

OutPort: REG P1 ;Присвоить имя OutPort порту P1

SP: REG 81h ;Присвоить имя SP указателю стека (ячейка 81h)

PSW: REG D0h ;Присвить имя PSW регистру слова состояния МК

\*\*\*\*\*\*\*\*\*\*\*\* ОПРЕДЕЛЕНИЕ НЕПЕРЕМЕЩАЕМЫХ БИТ \*\*\*\*\*\*\*\*\*\*\*\*

ET0: REG IE.1 ;Присвоить имя ET0 биту 1 регистра IE

EA: REG IE.7 ;Присвоить имя EA биту 7 регистра IE

ITO: REG TCON.0 ;Присвоить имя IT0 биту 0 регистра TCON

IEO: REG TCON.1 ;Присвоить имя IE0 биту 1 регистра TCON

TR0: REG TCON.4 ;Присвоить имя TR0 биту 4 регистра TCON

RD: REG P3.1 ;Присвоить имя RD биту 0 порта P3

B0: REG P3.0 ;Присвоить имя B0 биту 1 порта P3

CheckBit: REG P3.2 ;Присвоить имя CheckBit биту 2 порта P3

WR: REG P1.0 ;Присвоить имя WR биту 0 порта Р1

SignA: REG E0h.7 ;Присвоить имя SignA (знак регистра А) старшему биту аккумулятора

SignB: REG F0h.7 ;Присвоить имя SignB (знак регистра B) старшему биту регистра В

\*\*\*\*\*\*\*\*\*\*\*\* ОПРЕДЕЛЕНИЕ ПЕРЕМЕЩАЕМЫХ РЕГИСТРОВ \*\*\*\*\*\*\*\*\*\*\*\*

RSECT ;Секция байт, располагается в DSEG

b1: DS 1 ;Коэффициенты числителя

b2: DS 1 ;

b3: DS 1 ;

b4: DS 1 ;

b5: DS 1 ;

b6: DS 1 ;

b7: DS 1 ;

DS 09h ;

Xn1: DS 1 ;Текущие отсчеты

Xn2: DS 1 ;

Xn3: DS 1 ;

Xn4: DS 1 ;

Xn5: DS 1 ;

Xn6: DS 1 ;

Xn7: DS 1 ;

Xn8: DS 1 ;

Xn9: DS 1 ;

Xn10: DS 1 ;

Xn11: DS 1 ;

Xn12: DS 1 ;

Xn13: DS 1 ;

ds 01h ;

X1: DS 1 ;Отсчеты соответствующие упрощенному уравнению

X2: DS 1 ;

X3: DS 1 ;

X4: DS 1 ;

X5: DS 1 ;

X6: DS 1 ;

X7: DS 1 ;

DS 09h ;

Yn1 DS 1 ;Выходной отсчет

MainPSW: DS 1 ; Регистр для записи PSW основной программы

RunPSW: DS 1 ; Регистр для записи PSW программы вычисления отчёта

VarA: DS 1 ;Байт для промежуточного хранения аккумулятора

SumH: DS 1 ;Старший байт суммы

SumL: DS 1 ;Младший байт суммы

AddrSP: DS 1 ;Начало стека

\*\*\*\*\*\*\*\*\*\*\*\* ОПРЕДЕЛЕНИЕ ПЕРЕМЕЩАЕМЫХ БИТ \*\*\*\*\*\*\*\*\*\*\*\*

SignM: DS 1 ;Регистр для хранения знака произведения

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* МАКРОСЫ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;---------------------------------------------------------------------------------------------------------------;Определение знака произведения

;Определяет знак произведения двух множителей, представленных в прямом

;коде и расположенных в регистрах А и В. SignM = SignA XOR SignB

;Автор: библиотека

;---------------------------------------------------------------------------------------------------------------SignMUL: MACRO

MOV C, SignA

ANL C, /SignB

MOV SignM, C

MOV C, SignB

ANL C, /SignA

ORL C, SignM

MOV SignM, C

ENDM

;---------------------------------------------------------------------------------------------------------------;Преобразование произведения в дополнительный код

;[B,A](БезЗн)=>

;Автор: библиотека

;---------------------------------------------------------------------------------------------------------------PK\_DK\_MUL: MACRO

JNB SignM, M1#

CPL A

ADD A, #01h

MOV VarA, A

MOV A, B

CPL A

ADDC A, #00h

MOV B, A

MOV A, VarA

M1#: ENDM

;---------------------------------------------------------------------------------------------------------------;Накапливающий сумматор двухбайтовых чисел [B,A]+[SumH,SumL]=>[SumH,SumL]

;Автор: библиотека

;---------------------------------------------------------------------------------------------------------------Sum2Byte: MACRO

ADD A, SumL

MOV SumL, A

MOV A, B

ADDC A, SumH

MOV SumH, A

ENDM

;---------------------------------------------------------------------------------------------------------------;Умножение C накоплением двух чивел, адреса которых находятся в регистрах R0 и R1

;текущего регистрового банка

;Автор:библиотека

;---------------------------------------------------------------------------------------------------------------MAC: MACRO

MOV A, @R0

MOV B, @R1

SignMUL

ANL A,#Maska1

ANL B,#Maska1

MUL AB

PK\_DK\_MUL

Sum2Byte

ENDM

;---------------------------------------------------------------------------------------------------------------;Очистка 2-байтового накопительного сумматора

;Автор: библиотека

;---------------------------------------------------------------------------------------------------------------ClearSum2Byte: MACRO

MOV SumH, #Maska2

MOV SumL, #Maska2

ENDM

;---------------------------------------------------------------------------------------------------------------;Вычисление отсчетов упрощенного уравнения

;Автор: Косс А.В.

;---------------------------------------------------------------------------------------------------------------PrepareData: MACRO

MOv A, Xn1

ADD A, Xn13

MOV X1, A

MOv A, Xn2

ADD A, Xn12

MOV X2, A

MOv A, Xn3

ADD A, Xn11

MOV X3, A

MOv A, Xn4

ADD A, Xn10

MOV X4, A

MOv A, Xn5

ADD A, Xn9

MOV X5, A

MOv A, Xn6

ADD A, Xn8

MOV X6, A

MOV X7, Xn7

ENDM

;---------------------------------------------------------------------------------------------------------------;Вычисление отсчета

;Автор: библиотека

;---------------------------------------------------------------------------------------------------------------ComputeSample: MACRO

ClearSum2Byte

MOV R0, #b1

MOV R1, #X1

MOV R4, #Order

M2# MAC

INC R0

INC R1

DJNZ R4, M2#

ENDM

;---------------------------------------------------------------------------------------------------------------;Подготовка цикла

;Автор: библиотека

;---------------------------------------------------------------------------------------------------------------PrepareCycl: MACRO

MOV Yn1, SumH

MOV Xn13, Xn12

MOV Xn12, Xn11

MOV Xn11, Xn10

MOV Xn10, Xn9

MOV Xn9, Xn8

MOV Xn8, Xn7

MOV Xn7, Xn6

MOV Xn6, Xn5

MOV Xn5, Xn4

MOV Xn4, Xn3

MOV Xn3, Xn2

MOV Xn2, Xn1

MOV Xn1, InPort

ENDM

;---------------------------------------------------------------------------------------------------------------;Инициализация прерываний

;EA=1 - разрешить все прерывания

;ET0=1 - разрешить прерываний TC0

;Автор: Косс А.В.

;---------------------------------------------------------------------------------------------------------------InitInt: MACRO

SETB EA

SETB ET0

ENDM

;---------------------------------------------------------------------------------------------------------------;Перезагрузка таймера

;Автор: Косс А.В.

;---------------------------------------------------------------------------------------------------------------ReloadTimer: MACRO

CLR TR0

MOV TL0, #Time0L

MOV TH0, #Time0H

SETB TR0

ENDM

;---------------------------------------------------------------------------------------------------------------;Инициализация таймера

;Автор: Косс А.В.

;---------------------------------------------------------------------------------------------------------------InitTimer: MACRO

ANL TMOD,#F0h

ORL TMOD,#MODE1

ENDM

;---------------------------------------------------------------------------------------------------------------;Инициализация стека ;установка вершины стека

;Автор: библиотека

;---------------------------------------------------------------------------------------------------------------

InitSteck: MACRO

MOV SP, #AddrSP

DEC SP

ENDM

;---------------------------------------------------------------------------------------------------------------;Инициализация PSW

;Автор: Косс А.В.

;---------------------------------------------------------------------------------------------------------------InitPSW: MACRO

MOV MainPSW,#MRBank

MOV RunPSW,#RRBank

ENDM

;---------------------------------------------------------------------------------------------------------------;Инициализация коэффициентов

;Автор: Косс А.В.

;---------------------------------------------------------------------------------------------------------------InitCoeff: MACRO

MOV b1, #Datab1

MOV b2, #Datab2

MOV b3, #Datab3

MOV b4, #Datab4

MOV b5, #Datab5

MOV b6, #Datab6

MOV b7, #Datab7

ENDM

;---------------------------------------------------------------------------------------------------------------;Инициализация отсчётов

;Автор: библиотека

;---------------------------------------------------------------------------------------------------------------ClearXn: MACRO

MOV Xn1, #Maska3

MOV Xn2, #Maska3

MOV Xn3, #Maska3

MOV Xn4, #Maska3

MOV Xn5, #Maska3

MOV Xn6, #Maska3

MOV Xn7, #Maska3

MOV Xn8, #Maska3

MOV Xn9, #Maska3

MOV Xn10, #Maska3

MOV Xn11, #Maska3

MOV Xn12, #Maska3

MOV Xn13, #Maska3

ENDM

;---------------------------------------------------------------------------------------------------------------;Основная программа

;Автор: библиотека

;---------------------------------------------------------------------------------------------------------------MainProgramm: MACRO

M3#: SJMP M3#

ENDM

;---------------------------------------------------------------------------------------------------------------;Преобразование данных из дополнительного в прямой код

; InByte(ДК)=> OutByte(ПК)

;Автор: библиотека

;---------------------------------------------------------------------------------------------------------------DK\_PK: MACRO InByte, OutByte

MOV A, InByte

JNB SignA, M4#

XRL A, #7Fh

INC A

M4#: MOV OutByte, A

ENDM

;---------------------------------------------------------------------------------------------------------------;Сохранение PSW основной программы и загрузка PSW программы вычисления отчета

;Автор: Косс А.В.

;---------------------------------------------------------------------------------------------------------------SavePSW: MACRO

MOV MainPSW,PSW

MOV PSW,RunPSW

ENDM

;---------------------------------------------------------------------------------------------------------------;Восстановление PSW основной программы

;Автор: Косс А.В.

;---------------------------------------------------------------------------------------------------------------ReloadPSW: MACRO

MOV RunPSW,PSW

MOV PSW,MainPSW

ENDM

;---------------------------------------------------------------------------------------------------------------;Запуск входного внешнего устройства

;Автор: Косс А.В.

;---------------------------------------------------------------------------------------------------------------

RunACP: MACRO

CLR B0

SETB B0

ENDM

;---------------------------------------------------------------------------------------------------------------;Проверка готовности данных входного внешнего устройства

;Автор: Косс А.В.

;---------------------------------------------------------------------------------------------------------------CheckACP: MACRO

M6#: JB CheckBit,M6#

ENDM

;---------------------------------------------------------------------------------------------------------------;Сдвиг результата вычисления на 1 разряд в лево

;Автор: Косс А.В.

;---------------------------------------------------------------------------------------------------------------

ShiftL: MACRO

MOV R7, #Shift

M1#: CLR C

MOV A, SumL

RLC A

MOV SumL, A

MOV A, SumH

RLC A

MOV SumH, A

DJNZ R7, M1#

ENDM

;---------------------------------------------------------------------------------------------------------------;Ввод данных из P0 в Xn1

;Автор: Косс А.В.

;---------------------------------------------------------------------------------------------------------------InputData: MACRO

SETB RD

DK\_PK InPort, Xn1

CLR RD

ENDM

;---------------------------------------------------------------------------------------------------------------;Перевод двухбайтового числа (результат вычисления отсчёта) из дополнительного в прямой код

;Автор: Косс А.В.

;---------------------------------------------------------------------------------------------------------------

DK\_PK\_Rez: MACRO

MOV A,SumH

JNB SignA,M1#

MOV A,SumL

CLR C

SUBB A,#01h

CPL A

MOV SumL,A

MOV A,SumH

SUBB A,#00h

XRL A,#7fh

M1#: MOV SumH,A

ENDM

;---------------------------------------------------------------------------------------------------------------

;Вывод данных

;Автор: Косс А.В.

;---------------------------------------------------------------------------------------------------------------

OutputData: MACRO

DK\_PK\_Rez

MOV OutPort, SumH

SETB WR

CLR WR

ENDM

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ПРОГРАММА \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.CODE ;Секция кода, располагается в СSEG

;Таблица векторов прерываний

;Прерывание RESET - стартовый адрес при сбросе микроконтроллера

ORG 0h

LJMP Init ;инициализация программы

;Прерывание TIMER0 - Прерывание таймера/счетчика 0

ORG 000Bh

LJMP Run

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;Программа инициализации

InitSect: SECTION ;Секция инициализации

Init: InitSteck

InitInt

InitTimer

InitCoeff

InitPSW

ClearXn

ReloadTimer

MainProgramm

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

RunSect: SECTION ;Секция исполнения

Run: RunACP

ReloadTimer

SavePSW

CheckACP

InputData

PrepareData

ComputeSample

ShiftL

OutputData

PrepareCycl

ReloadPSW

RETI

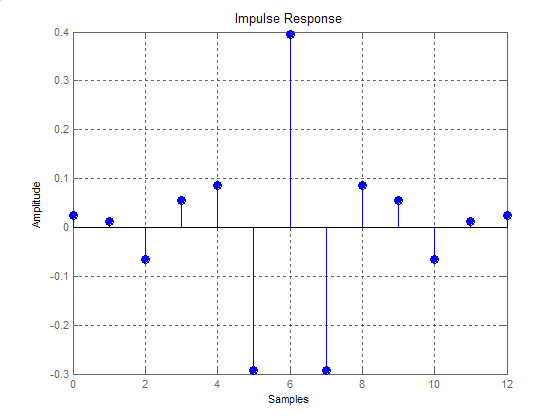
**4. Оценка быстродействия и определение импульсной и переходной характеристик**

Сосчитаем число машинных циклов, необходимых для вычисления положительного и отрицательного входных отсчетов:

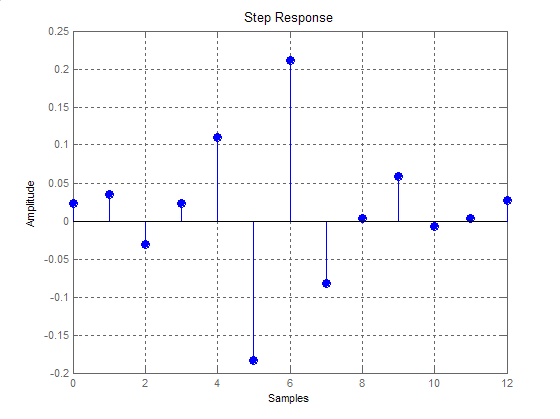
Положительный: 400 машинных циклов

Отрицательный: 418 машинных циклов

Построим импульсную и переходную характеристики фильтра в программе MatLab и сравним их со значениями вычисленными программно:



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № отчета | g(t)  MatLab | g(t)  3Ch (6010) | g(t) | g(t)  Округл | g(t) |
| 0 | 0,0234 | 0168h | 0,0234375 | 0100h | 0,0166 |
| 1 | 0,0117 | 00B4h | 0,01171875 | 0000h | 0 |
| 2 | -0,0664 | 83FCh | -0,06640625 | 8300h | -0,05 |
| 3 | 0,0547 | 0348h | 0,0546875 | 0300h | 0.05 |
| 4 | 0,0859 | 0528h | 0,0859375 | 0500h | 0,0833 |
| 5 | -0,293 | 9194h | -0,29296875 | 9100h | -0,2833 |
| 6 | 0,395 | 97ACh | переполнение | 9700h | переполнение |
| 7 | -0,293 | 9194h | -0,29296875 | 9100h | -0,2833 |
| 8 | 0,0859 | 0528h | 0,0859375 | 0500h | 0,0833 |
| 9 | 0,0547 | 0348h | 0,0546875 | 0300h | 0.05 |
| 10 | 0,0664 | 83FCh | -0,06640625 | 8300h | -0,05 |
| 11 | 0,0117 | 00B4h | 0,01171875 | 0000h | 0 |
| 12 | 0,0234 | 0168h | 0,0234375 | 0100h | 0,002037 |



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № отчета | h(t)  MatLab | h(t)  3Ch (6010) | h(t) | h(t)  Округл | h(t) |
| 0 | 0,0234 | 0168h | 0,0234375 | 0100h | 0,0166 |
| 1 | 0,0352 | 021Сh | 0,03515625 | 0200h | 0,0333 |
| 2 | -0,0313 | 81E0h | -0,03125 | 8100h | -0,0166 |
| 3 | 0,0234 | 0168h | 0,0234375 | 0100h | 0,0166 |
| 4 | 0,109 | 0690h | 0,109375 | 0600h | 0,1 |
| 5 | -0,184 | 8B04h | -0,18359375 | 8B00h | - 0,1833 |
| 6 | 0,211 | A2B0h | переполнение | A200h | переполнение |
| 7 | -0,082 | B444h | переполнение | B400h | переполнение |
| 8 | 0,00391 | AF1Ch | переполнение | AF00h | переполнение |
| 9 | 0.0586 | ABD4h | переполнение | AB00h | переполнение |
| 10 | -0.00781 | AFD0h | переполнение | AF00h | переполнение |
| 11 | 0.00391 | AF1Ch | переполнение | AF00h | переполнение |
| 12 | 0,0273 | ADB4h | переполнение | AD00h | переполнение |

По полученным результатам можно сказать, что программа фильтра работает правильно. Однако при относительно больших значениях, происходит переполнение. Из этого следует вывод, что фильтр будет функционировать только при малых входных отсчетах. Или же следует предусмотреть защиту от переполнения, однако это не входит в задачу курсового проекта.

**Заключение**

В данном курсовом проекте мной был разработан цифровой, нерекурсивный фильтр высоких частот. В процессе выполнения курсового проекта были выполнены все пункты теоретического задания.

**Список литературы**

1. Л.М. Гольденберг, Ю.П. Левчук, М.Н. Поляк Цифровые фильтры. М.: Связь, 1974
2. Щелкунов Н.Н., Дианов А.П. Микропроцессорные средства и системы. М.: Радио и связь, 1989
3. Соколов Ю.П. Микроконтроллеры семейства MSC-51: Архитектура, программирование, отладка: Учебное пособие / Рязанский государственная радиотехническая академия. Рязань, 2007
4. Сальников Н.И. Цифровые устройства и микропроцессоры: Методические указания к курсовому проекту / Рязанский государственные радиотехнический университет. Рязань, 2007

1. \* Графические материалы взяты из соответствующей номеру литературы. [↑](#footnote-ref-1)