**ВВЕДЕНИЕ**

В связи с повсеместным использованием цифровых управляющих систем постоянно растет необходимость разработки и усовершенствования их.

Большинство цифровых систем строится на микропроцессорах либо на микроконтроллерах. При помощи микропроцессорных систем происходит управление различными технологическими процессами и операциями. Данные системы практически универсальны, так как они имеют очень высокое быстродействие, и достаточную разрядность для различных выполнения различных расчетов на производстве. Используя в данных системах ППЗУ, возможно, при помощи одной компьютерной системы управление различным оборудованием. То есть необходимо изменение только программы управления.

Центральное место в структуре микропроцессорного устройства занимает микропроцессор, который выполняет арифметические и логические операции над данными, программное управление процессором обработки информации, организует взаимодействие всех устройств, входящих в систему. Микропроцессор представляет собой функционально законченное устройство, состоящее из одной или нескольких программно-управляемых БИС и предназначенное для выполнения операций по обработке информации и управления вычислительным процессом.

В курсовой работе разрабатывается устройство на базе микроконтроллера фирмы AVR семейства Mega.

Эти контроллеры характеризуются наиболее развитой периферией, наибольшими среди всех микроконтроллеров AVR объемами памяти программ и данных, поддерживают несколько режимов пониженого энергопотребления, имеют блок прерываний, сторожевой таймер и допускают программирование непосредственно в готовом устройстве. Они предназначены для использования в мобильных телефонах, контроллерах различного периферийного оборудования (принтеры, сканеры, современные дисковые накопители, приводы CD-ROM / DVD-ROM и т. п.), сложной офисной технике и т. д.

Микроконтроллер АТmega16 выбран на основании того, что он полностью удовлетворяет требованиям для реализации заданного устройства, а также имеет доступную цену и широко распространен.

**1. АНАЛИЗ ПОСТАВЛЕННОЙ ЗАДАЧИ**

Процесс проектирования интегрирующего устройства состоит из тех же этапов, что и процесс проектирования аналоговых фильтров. Сначала формулируются требования к желаемым характеристикам интегратора, по которым затем рассчитываются его параметры. Амплитудная и фазовая характеристики формируются аналогично аналоговым фильтрам. Ключевое различие между аналоговым и цифровым интеграторами заключается в том, что, вместо вычисления величин сопротивлений, емкостей и индуктивностей для аналогового интегратора, рассчитываются значения коэффициентов для цифрового. Иными словами, в цифровом интеграторе числа заменяют физические сопротивления и емкости аналогового. Эти числа являются коэффициентами, они постоянно находятся в памяти и используются для обработки (фильтрации) дискретных данных, поступающих от АЦП.

Интегрирующее устройство, работающий в реальном масштабе времени, оперирует с дискретными по времени данными в противоположность непрерывному сигналу, обрабатываемому аналоговым интегратором. При этом очередной отсчет, соответствующий отклику интегратора, формируется по окончании каждого периода дискретизации.

Вначале сигнал должен быть оцифрован с помощью АЦП для получения выборки x(n). Далее эта выборка поступает на цифровой интегратор. Отсчеты выходных данных y(n) используются для восстановления аналогового сигнала с использованием ЦАП с низким уровнем ложного сигнала.

В дискретных системах, даже с высокой степенью избыточной дискретизации, требуется наличие аналоговых ФНЧ перед АЦП и после ЦАП для устранения эффекта наложения спектра.

**1.1 Описание интегратора 1-го порядка**

Интегрирующее устройство (ИУ), интегратор – вычислительное устройство для определения интеграла, например вида где х и у — входные переменные. Входными переменными величинами могут быть механическое перемещение, давление, электрический ток (напряжение), число импульсов, температура и т. п.

Интегратор используется как самостоятельное вычислительное устройство при решении математических задач методами интегрирования; может служить элементом системы автоматического регулирования (интегрирующее звено); входить в состав вычислительной машины; использоваться для моделирования физического процесса и т. д. Интегратор применяют для изучения фильтрации.

Цифровые интеграторы входят в состав цифровых дифференциальных анализаторов, а также некоторых специализированных вычислительных устройств, например интерполяторов. Интегрирование функции в цифровых ИУ заменяется операцией суммирования конечного числа последовательных значений этой функции (её приращений), заданных в дискретных точках. При этом входная и выходная числовая информация представляется в виде электрических импульсов, а интегрирование осуществляется суммированием этих импульсов. Выбирая цену импульсов достаточно малой, можно обеспечить практически необходимую точность при замене интеграла суммой; точность аналогового ИУ ограничена.

ИУ описывается системой уравнений

x1[n]=x[n];

x2 [n]= x1[n] + x3[n-1];

x3 [n]= x1[n] + x2[n-1];

x4 [n]= kx1[n];

y[n]= x4[n] .

где y[n], x[n] – входная и выходная последовательности;

xi[n] – состояние i-го узла графа схемы усреднения;

 - постоянный коэффициент усреднения,

где δ – заданная погрешность восстановления идеальной АЧХ;

fd - частота прихода значений входной последовательности;

N – число циклов усреднения.

Частота сигнала, поступающего на вход равна 50 Гц, тогда минимальная частота дискретизации должна быть не менее 100Гц.

Рис.1.2 – Структурная схема интегратора

**1.2 Аналитический обзор микроконтроллера AТmega16**

AТmega16 представляет собой 8-разрядные микроконтроллеры с 16 Кбайтами внутрисистемной программируемой Flash памяти. Он обладает следующими характеристиками:

* 8-разрядный высокопроизводительный AVR микроконтроллер с малым потреблением
* Прогрессивная RISC архитектура

**1.** 130 высокопроизводительных команд, большинство команд выполняется за один тактовый цикл,

**2.** 32 8-разрядных рабочих регистра общего назначения

Полностью статическая работа

**3.** Производительность приближается к 16 MIPS (при тактовой частоте 16 МГц)

**4.** Встроенный 2-цикловый перемножитель

* Энергонезависимая память программ и данных

1. 16 Кбайт внутрисистемно программируемой Flash памяти (In-System Self-Programmable Flash) :

-обеспечивает 1000 циклов стирания/записи

- дополнительный сектор загрузочных кодов с независимыми битами блокировки

- Внутрисистемное программирование встроенной программой загрузки

- Обеспечен режим одновременного чтения/записи (Read-While-Write)

2. 512 байт EEPROM:

- Обеспечивает 100000 циклов стирания/записи

3. 1 Кбайт встроенной SRAM

- Программируемая блокировка, обеспечивающая защиту программных средств пользователя

* Интерфейс JTAG (совместимый с IEEE 1149.1)

**1.** Возможность сканирования периферии, соответствующая стандарту JTAG

**2.** Расширенная поддержка встроенной отладки

**3.** Программирование через JTAG интерфейс: Flash, EEPROM памяти,перемычек и битов блокировки

* Встроенная периферия

1. Два 8-разрядных таймера/счетчика с отдельным предварительным делителем, один с режимом сравнения

2. Один 16-разрядный таймер/счетчик с отдельным предварительным делителем и режимами захвата и сравнения

3. Счетчик реального времени с отдельным генератором

4. Четыре канала PWM

5. 8-канальный 10-разрядный

6. Байт-ориентированный 2-проводный последовательный интерфейс

7. Программируемый последовательный USART

8. Последовательный интерфейс SPI (ведущий/ведомый)

9. Программируемый сторожевой таймер с отдельным встроенным генератором

10. Встроенный аналоговый компаратор

* Специальные микроконтроллерные функции

1.Сброс по подаче питания и программируемый детектор кратковременного снижения напряжения питания

2.Встроенный калиброванный RC-генератор

3. Внутренние и внешние источники прерываний

4. Шесть режимов пониженного потребления: Idle, Power-save, Power-down, Standby, Extended Standby и снижения шумов ADC

* Выводы I/O и корпуса

32 программируемые линии ввода/вывода

40-выводной корпус PDIP и 44-выводной корпус TQFP

* Рабочие напряжения 4,5 - 5,5 В
* Рабочая частота 0 - 16 МГц (ATmega16)

Рис.1.3 - Расположение выводов контролера ATmega16


## Рис.1.4 - Архитектура модели AVR ATmega16

**1.4 Доопределение набора аппаратных средств**

Кроме контролера в состав устройства входят:

- АЦП

- ЦАП

- ФНЧ на 100 Гц

- ФНЧ на 75 Гц

**1.4.1 АЦП**

Микроконтроллер ATmega16 оснащен 10-разрядным АЦП, который имеет следующие характеристики:

- АЦП последовательного приближения;

- работает с тактовой частотой в диапазоне от 50 до 200 кГц;

- Интегральная нелинейность 0,5LSB;

- Абсолютная ошибка ±2LSB;

- Время преобразования 65-250 мкс;

- Максимальная разрешающая способность 15 преобразований в секунду;

- 8 мультиплексированных каналов;

- 2 дифференциальных канала с встроенным усилителем, который имеет 3 фиксированных коэффициента усиления: 1, 10, 200;

- Диапазон входного сигнала: 0 … Uп;

- 2 режима работы:

* Режим однократного преобразования,
* Режим циклического преобразования;

 - Режим преобразования по автопереключению;

 - Прерывание – на завершение преобразования;

 - Режим сохранения энергии для обеспечения понижения шумов.

АЦП присоединен к 8-канальному аналоговому мультиплексору, позволяющему использовать любой вывод порта А в качестве входа АЦП. Ниже приведена блок схема АЦП на рис 1.5.

Рис. 1.5 – Блок-схема АЦП

Данного АЦП вполне достаточно для преобразования входного аналогового сигнала в цифровой. Т.к. верхняя граница частотного диапазона, в котором изменяется сигнала, поступающий с выхода аналогового фильтра, составляет 50 Гц. При этом частота дискретизации должна быть не меньше 2fв, а АЦП может обрабатывать сигнал с fт до 200 кГц. При максимальной тактовой частоте АЦП мы получаем высокую точность обработки сигнала, которая достаточна для данных целей.

Точность АЦП оценивается относительной погрешностью δАЦП

где N- число двоичных разрядов

Наилучшая точность преобразования аналогового сигнала в цифровой код получается, когда используется вся шкала АЦП,т.е. в том случае, когда

Где это максимально е значение сигнала на аналоговом входе , а - шкала АЦП.

**1.4.2 ФНЧ с частотой среза 100 Гц**

На входе АЦП необходимо поставить низкочастотный фильтр для устранений эффекта наложения спектра шумов от цифровой аппаратуры на аналоговый сигнал. ФНЧ пропускает заданный диапазон низких частот (0 до 100 Гц) и подавляет все остальные, которые не входят в этот диапазон.

В качестве ФНЧ используем фильтр Баттерворта 2-го порядка, который обеспечивает затухание вне полосы пропускания равное -40 дБ/дек. Передаточная характеристика определяется следующим выражением:

,

где - коэффициент усиления каскада на нулевой частоте,

- частота среза ,

 - относительная частота.

Ниже приведена принципиальная схема ФНЧ, которая реализована на операционном усилителе LF\_147.

Рис. 1.6 – Схема электрическая принципиальная ФНЧ

Так как нет необходимости в усиление входного сигнала, то коэффициент усиления α = 1. Частота среза равна . Зададимся значениями резистора R1 из диапазона (10-100) kОм. Рассчитаем значения остальных элементов фильтра:

R2 = R1 = 100 кОм,

**1.4.3 ЦАП**

АП выбран фирмы MAXIM. Это микросхема МАХ504, которая представляет собой последовательный 10-разряднай ЦАП напряжения. Он имеет следующее характеристики:

- Униполярное питание +5В;

- Буферизованные выходы напряжения;

- Встроенный источник опорного напряжения 2.048 В;

- Относительная погрешность ±0.5 LSB (макс)

- Гарантированная погрешность в диапазоне рабочих температур

- Гибкий диапазон вывода: 0 … Uп;

- Инициализация при подаче питания;

- Время преобразования 25мкс;

- Последовательный вывод данных, с возможностью последовательного соединения устройств в цепь.

Функциональная схема MAX504 приведена на рис 1.8.

Рис 1.8 – Функциональная схема ЦАП MAX504

**1.4.4** **ФНЧ с частотой среза 75 Гц**

На выходе ЦАП необходимо поставить низкочастотный фильтр для устранений эффекта наложения спектра шумов от цифровой аппаратуры на аналоговый сигнал. ФНЧ пропускает заданный диапазон низких частот 0 до 75 Гц.

Расчет данного фильтра аналогичен предыдущему фильтру ФНЧ. Передаточная характеристика определяется следующим выражением:

,

Ниже приведена принципиальная схема ФНЧ, которая реализована на операционном усилителе LF\_147.

Рис. 1.6 – Схема электрическая принципиальная ФНЧ

Так как нет необходимости в усиление входного сигнала, то коэффициент усиления α = 1. Частота среза равна . Зададимся значениями резистора R1 из диапазона (10-100) kОм. Рассчитаем значения остальных элементов фильтра:

R2 = R1 = 100 кОм,

**2. ПРОЕКТИРОВАНИЕ ПРИНЦИПИАЛЬНОЙ СХЕМЫ**

**УСТРОЙСТВА**

**2.1 Схема включения микроконтроллера**

В микроконтроллере AT mega16 используется напряжение 5 В, подаваемое на вход VCC. Для питания АЦП используются вход AVCC, подключаемый к напряжению питания через фильтр низких частот. Вход AREF используется для подключения внешнего фильтрующего конденсатора для повышения помехозащищенности.

Рис.2.1 - Схема подключения питания микроконтроллера

**2.2 Формирование тактовых импульсов**

Тактировать микроконтроллер можно несколькими способами: использовать внешний тактирующий сигнал, либо подключать внешний кварцевый резонатор или RC-цепочку, либо использовать встроенный генератор с внутренней RC- цепочкой. Всё зависит от той задачи, которую мы решаем. В данном случае нам не нужна особая точность вычислений и нет согласования с другими тактируемыми устройствами, поэтому вполне можно воспользоваться встроенным генератором с внутренней RC- цепочкой.

Использование встроенного RC-генератора с внутренней время задающей RC-цепочкой является наиболее экономичным решением, так как при этом не требуется никаких внешних компонентов.

Внутренний RC-генератора микроконтроллеров семейства мега может работать на нескольких фиксированных частотах.

Частота работы внутреннего RC-генератора определяется значениями битов CKSEL3-0 согласно табл.1.

Табл.2 Режимы работы внутреннего RC-генератора.

В разрабатываемом устройстве высокая частота, а соответственно и высокая скорость работы не нужна, поэтому вполне достаточно будет выбрать частоту тактирования равную 2.0 МГц. Снижение частоты приведёт к уменьшению количества бессмысленных циклов обработки сигналов, а соответственно и к уменьшению энергопотребления.

**2.3 Организация сброса**

В данном случае нет смысла использовать отдельную кнопку сброса, сброс будет осуществляться по включению питания. При включении устройства подаётся питание на контроллер и автоматически осуществляется сброс, при выключении устройства, просто снимается питание с контроллера.

Для предотвращения сбоев работы контролера и выполнения программы при скачке напряжения, используем встроенную схему сброса при снижении питания BOD, которая отслеживает напряжение питания. Если работа этой схемы разрешена, то при снижении питания ниже некоторого уровня она переводит контроллер в состояние сброса. Когда напряжение вновь увеличится до порогового значения, запускается таймер задержки сброса. После формирования задержки tTOUT внутренний сигнал сброса снимается и происходит запуск микроконтроллера. Временные диаграммы соответствующие сбросу от схемы BOD, показаны на рис.7.

Рис.2.2 - Временные диаграммы формирования сброса

по снижению питания

Включением/выключением схемы BOD управляет конфигурационная ячейка BODEN. Для разрешения работы схемы эта ячейка должна быть запрограммирована в «0». Порог срабатывания VBOT определяется состоянием конфигурационной ячейки BODLEVEL, при «1» порог срабатывания 2.7 В, при «0» порог 4 В.

Для уменьшения вероятности ложных срабатываний порог напряжения переключения схемы имеет гистерезис, равный 50 мВ. Кроме того, срабатывание схемы BOD происходит только в том случае, если период провала больше 2 мкс.

Задание длительности задержки сброса tTOUT определяется значением конфигурационных ячеек, и включает в себя две составляющих: ts – выход на рабочий режим и стабилизация частоты тактового генератора, tr – для установки напряжения питания. При использовании встроенного RC-генератора с внутренней RC-цепочкой при включённой схеме BOD, биты конфигурации SUT1-0 установлены в значение «00», при этом ts= 6 тактов, tr –соответственно не используется.

**2.4 Схемы входных и выходных устройств**

На вход МК подается аналоговый сигнал с ФНЧ. Сигнал подается на один из выводов порта А, так как АЦП подключен к этим выводам. Для подключения аналогового фильтра будем использовать вывод PA1.

Данные с микроконтроллера подаются на ЦАП. Для выходных данных будем использовать выводы порта В и С. Младшие разряды обработанных данных подаются через порт В, а старшие – через порт С на ЦАП. Сигналы управления для ЦАП будут подаваться через порт D.

**3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ**

**3.1 Разработка алгоритма программы**

Рис.3.1 – Алгоритм работы устройства

**3.2 Проектирования модуля инициализации портов ввода-вывода**

Порт настроим на ввод информации. Порты В и С на вывод. Мы будем использовать только младшую тетраду порта D, то инициализируем только ее на ввод информации.

void GlobalInitialize(void)

{

DDRA = 0x00; //Port A pins as output

PORTA = 0x01; //Turn ON PullUP for PortA pin

DDRB = DDR\_SPI; //Port B pins as output

PORTB = 0xff; //Send 0xFF to PortB output pins

DDRC = 0x01; //Port C pins as output

PORTC = 0x01; //Send 0x03 to PortC output pins

}

**3.3 Инициализация АЦП**

За настройку работы АЦП отвечают три регистра:

* ADCSRA – регистр управления и состояния,
* ADMUX – регистр управления мультиплексором,
* SFIOR – регистр специальных функций.

Форматы регистров управления работой АЦП приведены на рис.6.

Рис.3.2 Формат регистров ADCSRA, ADMUX, SFIOR

Инициализация регистра ADCSRA:

Бит 7 ADEN =”1” включение АЦП.

Бит 6 ADSC =”1” запуск первого преобразования в режиме непрерывного преобразования.

Бит 5 ADATE позволяет выбрать режим работы АЦП

Бит 4 ADIF флаг прерывания, устанавливается при завершении преобразования

Бит 3 ADIE=”1” разрешение прерывания по завершению преобразования

Биты 2…0 ADPS2..0=”010” выбран делитель частоты на 16, тогда при тактовой частоте контроллера 2 МГц, частота тактов АЦП будет составлять 125 кГц, использование такой частоты позволяет увеличить точность преобразования.

Инициализация регистра ADMUX:

Биты 7,6 REFS1:REFS0= “11” при этом в качестве опорного напряжения используется внутренний источник с Uоп = 2,56 В.

Бит 5 ADLAR результат преобразования выравнивается по левой границе 16-ти разрядного слова

Биты 4..0 MUX4..0 управляют входным мультиплексором, а так как у нас используются входы ADC1, то эти разряды могут принимать значение “00001” .

Инициализация регистра SFIOR:

Биты 7..5 ADTS2..0=”000” определяют режим непрерывного преобразования. В процессе выполнения которого есть возможность изменять содержимое разрядов MUX2..0, что позволяет осуществлять последовательное преобразование сигналов нескольких каналов.

Бит 4 ADHSM=”0” при “1” увеличивает скорость работы АЦП, однако в этом мы не нуждаемся.

Результат преобразования будет определяться выражением:

ADC=1024\*Vin/Vref

void Init\_ADC(void)

{

ADCSRA |= (1 << ADIE); //Enable ADC Interrupt

ADCSRA |= (1 << ADPS1); //ADC Timing /16

ADCSRA |= (1 << ADEN); //Enable ADC

ADMUX |= (1 << MUX0); //выбор канала - мультирлексированный ADC1

ADMUX |= (1 << REFS0); //Internal Reference 2,56V selected

ADMUX |= (1 << REFS1);

SFIOR = 0;

 #asm("sei");

}

**3.4 Проектирование процедуры чтения данных с АЦП**

Сигнал со входа АЦП преобразовывается по установке бита ADSC в 1. Затем ожидается выставления флага конца преобразования ADIF в 1. После того, как ADIF = 1 данные сохраняются во временной переменной.

unsigned int ReadADC()

{

unsigned int TMP;

unsigned int ADC;

ADCSRA |= 1<<ADSC; //запустить АЦП на преобразование

if (!(ADCSRA & (1<<ADIF)) )//флаг конца преобразования

//если преобразование закончилось

{

TMP=ADCH;

TMP=(TMP<<8);

TMP=TMP + ADCL;

ADCSRA=(ADCSRA | (1<<ADSC)); //запуск АЦП снова

}

ADC=TMP;

CLRBIT(ADCSRA,ADEN); //выключить АЦП

return ADC;

}

**3.5 Проектирование процедуры работы интегратора**

unsigned int Integrator(unsigned int in)

{

static unsigned int KH=8;

static unsigned int KL=10; //K=0.8 for economy memory pri uveli4enii to4nosti

static unsigned int x\_in, x1, x2[2], x3[2], x4, y\_out;

x\_in=0;

x2[0]=0; x2[1]=0; //init old state

x3[0]=0; x3[1]=0; //init old state

y\_out=0;

//your start variable here

x\_in=in;

x1=x\_in;

x2[1]=x1+x3[0];

x3[1]=x1+x2[0];

x4=x2[1]\*KH/KL;

//save old state

x3[0]=x3[1];

x2[0]=x2[1];

//==============

return x4;

}

**3.6 Процедура передачи данных ЦАП**

Так как ЦАП имеет последовательный вход, то данные необходимо передавать посылками. Для этого используется интерфейс SPI (Serial Peripheral Interface) - полнодуплексный скоростной синхронный трёхпроводной интерфейс.

Рис.3.2- Конфигурация SPI

SPI в реализации Atmel имеет четыре задаваемые программно скорости передачи, может передавать байты от старшего к младшему биту и наоборот, обнаруживает ошибки пакета.

За работу интерфейса отвечают 3 регистра:

* Регистр SPCR - регистр управления

SPIE SPE DORD MSTR CPOL CPHA SPR1 SPR0

SPIE - разрешение прерывания по окончанию передачи.

SPE - SPI включен. Если установлен этот бит, то выводы #SS, MISO, MOSI и SCK работают как выводы SPI, иначе - как простые выводы порта.

DORD - направление передачи данных. Если установлен, то передача идёт с младшего бита, если сброшен - со старшего.

MSTR - если установлен, контроллер работает как Master, если сброшен - как Slave. Управляется также выводом #SS, если он настроен на ввод - при подаче "0" на #SS бит MSTR сбрасывается.

CPOL - определяет уровень на выходе SCK в режиме ожидания - SCK = CPOL

CPHA - если установлен, передача и приём бита производится по обратному фронту, если сброшен - по прямому.

SPR1, SPR0 - делитель тактовой частоты (F - частота генератора контроллера).

SPR1 SPR0 Частота

0 0 F/4

0 1 F/16

1 0 F/64

1 1 F/128

* Регистр SPSR - регистр состояния
* Регистр SPDR - регистр данных

void Init\_SPI(void)

{

// ÌÀÑÒÅÐ

DDR\_SPI=(1<<MOSI)|(1<<SCK); // Â: è ÑLK

SPCR |= (1 << SPIE); // SPI

SPCR |= (1 << SPE); //

SPCR |= (1 << MSTR); //

SPCR |= (1 << SPR0); // /16

}

void SendSPI(unsigned int x)

{

SPDR = x; // SPI

while(!SendDataReady); // ÷è

SendDataReady = FALSE; //è â 0

}

interrupt [SPI\_STC] void SPI\_interrupt(void)

{

SendDataReady = TRUE;

}

**3.7 Процедура main( )**

void main(void)

{

unsigned int ADC\_conv;

unsigned int y\_out;

PORTC = 0x01;

GlobalInitialize();

Init\_ADC();

Init\_SPI();

while(1)

{

PORTC = 0x00;

ADC\_conv=ReadADC();

y\_out=Integrator(ADC\_conv);

SendSPI(y\_out); //ïåðåäà÷à äàííûõ ÀÖÏ

delay\_us(6); //for ADC

}

}

**Заключение**

В курсовой работе разработано устройство цифровой фильтрации на основе микроконтроллера фирмы AVR ATmega16. Разработанный фильтр пропускает сигнал в диапазоне от 0 до 200 Гц. Данное устройство обладает высокой точностью преобразования. Также имеет малые габариты, вес, и обладает низким энергопотреблением.

Одним из главных достоинств цифровых фильтров, является высокая точность преобразования, так же они не имеют дрейфа вследствие изменения параметров компонентов. Но они имеют ограничения при работе в реальном режиме времени, т.к. вычисление должно быть завершено в период дискретизации. Аналоговые ж фильтры необходимо использовать для устранения эффекта наложения спектров и на высоких частотах.