**Пояснительная записка**

к курсовой работе по дисциплине

«Цифровые устройства и микропроцессоры»

Студент Ксендзов А. В.

Группа 712 специальности 2007

2001

**Введение**

С внедрением в промышленность цифровых технологий появилась возможность строить устройства обработки оцифрованных сигналов вычислительным методом. Такой способ обладает рядом важных преимуществ, таких как повышенная точность обработки, меньшая зависимость параметров от внешних условий, а также возможность реализации таких обрабатывающих устройств, которые невозможно или трудно было реализовать в аналоговом виде. К таким устройствам относятся и цифровые фильтры, для которых стало возможным построение разнообразных частотных характеристик путем их аналитической задачи. При этом реализуемы и фильтры традиционных типов: нижних частот, верхних частот, полосовые и режекторные.

Сглаживающий фильтр относится к особому типу фильтров и призван убирать из сигнала импульсные проявления (пики и щелчки), сглаживая их. Импульс имеет широкий (в идеале бесконечный) частотный спектр, однако острота его формы определяется именно высокочастотными составляющими.

По заданию требуется спроектировать цифровой сглаживающий фильтр, используя набор К1821, включающий микропроцессор ВМ85, ПЗУ и порты ввода-вывода на микросхеме РФ55, ОЗУ, таймер счетчик и порты на микросхеме РУ55. Входной сигнал аналоговый, оцифровывается с помощью 10-разрядного АЦП 1108ПВ1, после чего имеет вид отсчетов, следующих через интервал дискретизации. После обработки сигнал в дополнительном цифровом коде должен передаваться в порт ввода-вывода по алгоритму с квитированием. Частота дискретизации FД = 5 кГц, разрядность выходного сигнала 8. Обработка должна происходить в реальном масштабе времени. При проектировании следует придерживаться двух критериев: минимальная аппаратная конфигурация устройства и минимальное время операций обработки сигнала (вычисления). При определенных условиях эти критерии противоречат друг другу.

Фильтр есть линейное устройство, производящее операции над частотным спектром исходного сигнала. Для понимания процесса обработки сигнала в цифровом фильтре полезно знать его основные характеристики. Передаточная функция фильтра, определяющая отношение выходного сигнала к входному при конкретном значении комплексной частоты, по заданию имеет вид:

H(z) = 0.4 (0.1 + 0.65z –1 + z –2 + 0.65z –3 + 0.1z -4 ) (1),

здесь z = e p Tд- характеризует задержку отсчета на время, равное интервалу дискретизации ТД = 1/FД = 0.2 мс. Для данного случая это означает, что амплитуда очередного выходного отсчета равна сумме амплитуд пяти последних входных отсчетов, помноженных на соответствующие коэффициенты. Когда на вход поступает очередной отсчет, по этому правилу сразу вычисляется очередной выходной. По сути это и есть алгоритм работы цифрового фильтра, который требуется спроектировать.

По этому алгоритму, при подаче на вход единственного единичного отсчета фильтр ответит пятью отсчетами, следующими через ТД и имеющими амплитуды, равные соответствующим коэффициентам передаточной функции:

Видно, что высокий острый импульс сглаживается и растягивается во времени.

Если на вход подается последовательность из пяти и более единичных отсчетов, то на выходе будет наблюдаться последовательность нарастающих по амплитуде импульсов, причем каждый последующий больше предыдущего на соответствующий коэффициент, а начиная с пятого все отсчеты имеют единичную амплитуду:



Строго говоря, выражение (1) является аналитическим описанием дискретного режекторного гребенчатого фильтра нерекурсивного типа ( с конечной импульсной характеристикой ).

Гребенчатость фильтра связана со стробоскопическим эффектом при оцифровке сигнала, когда бесконечное множество аналоговых колебаний различных частот имеют одинаковую дискретную реализацию в отсчетах и, следовательно, обрабатываются фильтром одинаково. Но только одно из таких колебаний имеет частоту, меньшую частоты дискретизации, а значит, остальные колебания не отвечают требованию теоремы Котельникова, согласно которой верхняя частота сигнала должна быть как минимум в два раза меньше частоты дискретизации, иначе может произойти потеря информации, закодированной в дискретном сигнале. По заданию, FД = 5 кГц , следовательно, верхняя частота сигнала, а значит, и фильтра составляет



FВ = 2.5 кГц.

1. **Анализ и формализация задачи**

Согласно техническому заданию и выражению (1), обобщенная структура, описывающая работу фильтра, выглядит следующим образом:

xn

Z-1

Z-1

Z-1

Z-1

b4

b3

b0

b2

b1

yn

C

**Σ**

где xn – входные отсчеты, yn – выходные отсчеты,

z –1 – задержка на ТД , bm и С – коэффициенты.

Как видно, фильтр должен держать в памяти пять отсчетов одновременно и оперировать с ними, производя арифметические операции. С выполнением поставленной задачи может справиться микропроцессорная система, действующая по заданной программе. Она составляет основу цифрового фильтра и включает в себя микропроцессор, оперативную память для хранения отсчетов и постоянную память для хранения кода программы, реализующей алгоритм фильтрации. Согласно заданию, микропроцессорная система строится на основе комплекта К1821, включающего микропроцессор ВМ85, микросхему РФ55 ( ПЗУ и порты ввода-вывода ) и микросхему РУ55 (ОЗУ, таймер-счетчик и порты ввода-вывода ).

Исходный сигнал, который фильтр должен обрабатывать, имеет аналоговую форму, поэтому его необходимо оцифровать. Для этой цели используется аналогово-цифровой преобразователь (АЦП) последовательного приближения 1108ПВ1. Данный АЦП работает с униполярным сигналом от 0 до 3 вольт, поэтому необходимо обеспечить масштабирование и сдвиг на +1.5 В аналогового сигнала, который по заданию изменяется в пределах от -2 до +2 В. Это осуществляется с помощью высокостабильного резистивного делителя и развязывающего повторителя, выполненного на операционном усилителе.

АЦП 110ПВ1 – десятиразрядный, данные выдаются в прямом параллельном коде. Вычисление начинается с отрицательным перепадом тактового импульса (ТИ) при условии низкого уровня на входе запуска ST. в течении первого тактового импульса происходит сброс регистров и установка опорных напряжений на входе внутреннего селектора АЦП. Далее в течении 10 ТИ осуществляется последовательное приближение, и в последнем 12-м ТИ формируется низкоуровневый сигнал RAD, служащий для оповещения внешнего устройства о готовности данных ( присутствии их в выходном буфере). АЦП может работать и в 8-разрядном режиме, если его вход SE10/8 подключен к шине отрицательного источника питания, при этом приближение происходит за 8 ТИ и таким образом, на преобразование затрачивается 10 тактов. Следует заметить, что у данного АЦП может быть внешнее или внутреннее тактирование, в последнем случае частота следования тактовых импульсов определяется величиной конденсатора CCLK , при ССLK = 10 пФ fmax = 27 МГц, τmin = 37 нс, время преобразования tпр= 10τmin = 0.37 мкс, отсюда следует, что максимально возможное время обработки микропроцессором данных tобр= TД – tпр = 200-0.37=199.63 мкс, то есть, время выборки занимает ничтожную часть общего времени ТД, отведенного на получение очередного входного отсчета и вычисление выходного.

Однако за малое время tпр сигнал на входе АЦП может меняться, и в результате приближений выходная информация не будет соответствовать действительности. Необходимо, чтобы во время аналогово-цифрового преобразования уровень входного сигнала был постоянен, или хотя бы не изменялся более чем на 0.5h , где h – шаг квантования

Максимальное изменение будет происходить на верхней частоте, равной согласно теореме Котельникова

fв = 0.5 FД = 2.5 кГц. Скорость этого изменения можно рассчитать, взяв производную от испытательного колебания с частотой fв и размахом Uвх max = 3 B.

S(t) = 3 sin (2π \* 2500 t ); S’(t) = 3\*2π \*2500 cos (2π \*2500 t );

Максимальная скорость изменения равна 3\*2π\*2500 ≈ 47124 В/с.

Δu = 47124 \* 3.7\*10-7 = 17.4 мВ > 5.86 = 0.5h.

Чтобы поддерживать сигнал на входе АЦП постоянным на время преобразования, требуется устройство выборки и хранения (УВХ), которое фиксирует уровень с момента прихода на управляющий вход тактирующего строба и удерживает этот уровень в течении всей его длительности. В простейшем случае УВХ представляет собой аналоговый ключ и емкость хранения:

В настоящее время производятся готовые УВХ в виде микросхем, например, серия К1100. Микросхема К1100СК2 представляет собой УВХ аналогового сигнала, изменяющегося в пределах ±5 В. При подключаемой емкости хранения 1 нФ время выборки не превышает: tвыборки<10 мкс. Скорость изменения выходного напряжения при хранении не более 5 мВ/мс, то есть при использовании УВХ Δumax = 1.85 мВ < 5.86 мВ.

Другая проблема состоит в том, что входной сигнал является биполярным и изменяется в пределах ±2 В, а заданный АЦП требует униполярный сигнал на входе в пределах 0…3 В. То есть необходимо перед подачей входного сигнала на АЦП промасштабировать его на ¾ и поднять постоянную составляющую на 1.5 В. С этой задачей справится аналоговый сумматор на операционном усилителе:

Коэффициент передачи такого сумматора определяется выражением:



Возможен вариант короткого замыкания цепи ОС,

R3 = 0, ROC = ∞. Тогда, если положить, что U1 = UBX , a U2 = Uпит , то очевидно, что коэффициент масштабирования U1 , равный ¾ , можно обеспечить, взяв R1 = 1 кОм, R2 = 3 кОм. Тогда коэффициент при U2 составляет ¼ . Учитывая, что требуемое UСМ = 1.5 В, получаем Uпит = 6 В = ½ Епит = +12 В. То есть, требуемое Uпит можно получить с шины питания с помощью симметричного резистивного делителя, при этом надо исключить влияние делителя на сумматор, поставив между ними повторитель напряжения на операционном усилителе.

Как уже отмечалось, при проникновении на вход АЦП составляющих с частотами, не отвечающими условию теоремы Котельникова, f1 + m FД , m = 1; 2; 3; … , может возникать стробоскопический эффект, выражающийся в перенесении этих составляющих в частотную область, отвечающую данной теореме, f 1 < FД .

Поэтому частота входного сигнала должна отвечать теореме Котельникова. Поскольку в техническом задании не оговариваются частотные свойства входного сигнала, то на входе необходимо поставить аналоговый фильтр с частотой среза, равной ½ FД = 2500 Гц, который подавлял бы высокочастотные составляющие и в какой-то мере устранял бы подобные проявления. В простейшем случае это ФНЧ 1-го порядка – RC-цепь и операционный усилитель.

R<<RBX oy , R = 1 кОм ; С = 1 / (πRFД ) = 63.66 нФ .

Таким образом, аналоговая входная часть проектируемого устройства представляет собой ФНЧ, грубо ограничивающий спектр сигнала, схему сдвига и масштабирования, устройство выборки и хранения и, наконец, АЦП.

Синхронизацию выборки с частотой FД целесообразно осуществить аппаратным способом, с помощью таймера-счетчика, встроенного в микросхеме РУ-55, настроив его период счета на ТД. Таймер может работать в двух периодических режимах: импульс низкого уровня длится половину периода счета (режим 1) или же один такт общей синхронизации ТТ микропроцессорной системы независимо от периода счета (режим 3). Если тактовая частота составляет fT = 5 МГц, то ТТ = 0.2 мкс. На выборку УВХ необходимо 10 мкс, то есть, в этом плане выгодней режим 1. Однако, для запуска АЦП со строгой периодичностью ТД необходимо, чтобы импульс ST длился не более tпр=0.37 мкс, то есть, для работы АЦП как нельзя лучше подходит режим 3 ( ровный меандр ). Кроме того, короткого импульса может не хватить на сброс УВХ.

Из положения можно выйти, настроив таймер на режим 1. На УВХ меандр подавать через схему задержки, состоящую из интегрирующей цепи и инвертора, на АЦП – через схему формирования короткого отрицательного импульса, состоящую из дифференцирующей цепи и инвертора. В результате импульс ST запустит АЦП, когда уровень сигнала на выходе УВХ уже стабилизировался и поддерживается постоянным.

При этом должны выполняться следующие временные соотношения:

t1 > tпр = 0.37 мкс;

½ TД – t1 > tвыборки ⇒ t1 < 90 мс;

t2 > τmin = 37 нс;

t2 < tпр = 0.37 мкс.

Выберем t1 = 10 мкс. Учитывая, что порог срабатывания инвертора составляет 2.5 В, можно рассчитать постоянную времени RC-цепи:

2.5 = 5 (1 - е – ( t1 / τ1 ) ); τ1 = - t1 / ln 0.5 = 14 мкс.

При R = 1 кОм С = τ / R ≈ 14 нФ.

Аналогично, положим t2 = 0.1 мкс. Постоянная времени дифференцирующей RC-цепи находится из выражения:

2.5 = 5 е – ( t2 / τ2 ) ; τ2 = - t2 ln 0.5 = 0.144 мкс.

При R = 1 кОм С = τ / R ≈ 144 пФ.

1. **Определение общего алгоритма функционирования устройства**

Работу цифрового фильтра определяет программа, состоящая из инструкций, предназначенных для микропроцессора, и хранящаяся в ПЗУ независимо от наличия питающего напряжения. Поскольку ПЗУ – единственное устройство, способное хранить информацию при отсутствии питания, то в нем должны быть также заложены данные о конфигурации и синхронизации устройства. При включении питания фильтр в первую очередь должен быть сконфигурирован на правильную, то есть соответствующую требуемой работу. Это осуществляется с помощью программы инициализации. При включении питания микропроцессор обнуляет счетчик команд, и выполнение инструкций тогда происходит с нулевого адреса ПЗУ, находящегося в области векторов прерываний. Туда следует в качестве инструкции поместить ссылку на адрес ПЗУ, с которого начинается программа инициализации. Когда управление будет передано ей, она должна настроить порты на ввод и вывод данных, таймер-счетчик на частоту синхронизации выборки, а также обнулить рабочие ячейки ОЗУ, исключив тем самым быстроменяющиеся переходные процессы. Программа инициализации не должна прерываться, поэтому на время ее выполнения прерывания следует запретить.

Чтобы настроить таймер на частоту дискретизации и режим работы 1, необходимо 16-разрядное управляющее слово:

0100001111001000(2) = 83С4(16)

Дав старших бита 01 определяют режим 1, а остальное представляет собой 14-разрядное число N = ТД / ТТ = 1000(10) .

Старший байт этого слова записывается по адресу 7004(16), а младший – по адресу 7005(16).

Для ввода и вывода данных могут использоваться соответственно порты А и В микросхемы РУ55, а 6-разрядный порт С можно использовать как линии готовности и квитирования портов А и В. Для установки используется управляющее слово, имеющее вид:

1 1 1 1 1 1 1 0 (2)

Попарно разряды этого слова, начиная с младшего, означают следующее:

Порт А работает на ввод, порт В – на вывод;

порт С работает как порт служебных сигналов;

разрешить прерывания по портам А и В;

запуск таймера.

Линии порта С передают следующие сигналы:

PC0 – INTRA – запрос прерывания по порту А;

PC1 – BFA – состояние буфера порта А;

PC2 – STBA – строб записи в порт А;

PC3 – INTRB – запрос прерывания по порту В;

PC4 – BFB – состояние буфера порта В;

PC5 – STBB – сигнал квитирования с периферии.

4-й и 5-й биты управляющего слова разрешают вывод сигналов INTRA и INTRB.

Управляющее слово необходимо для инициализации записать по адресу 7000(16).

Конечный шаг инициализации - устанавливается маска прерываний и с них снимается запрет. В случае, если используется аппаратное прерывание RST7.5, маска имеет вид:

0 0 0 0 1 0 1 1 (2)

0-й и 1-й разряды содержат единицы, маскирующие прерывания 5.5 и 6.5, нуль во 2-м оставляет доступным прерывание 7.5; единица в 3-м разряде разрешает прерывания.

Для установки этой маски используется специальная команда SIM микропроцессора ВМ85.

Программа инициализации заканчивается стандартно – командой HLT – останов. После этого процессор будет запускаться только аппаратным прерыванием RST 7.5 – при положительном перепаде напряжения на соответствующем входе. Сигналом запуска для микропроцессора должен служить сигнал наличия данных в буфере порта А BFA. Данный сигнал переходит на высокий уровень по положительному (заднему) фронту сигнала готовности данных АЦП RAD, представляющего собой импульс низкого уровня длительностью чуть больше такта АЦП τmin, то есть когда оцифрованный отсчет загружается в буфер порта А. Сигнал BFA обнуляется, когда происходит программное обращение к порту А с целью считать данные во внутренний регистр микропроцессора (аккумулятор).

При перепаде на высокий уровень, если прерывание не замаскировано и разрешено, произойдет сохранение счетчика команд в стеке, и микропроцессор начнет работу с ячейки 003С(16). В эту ячейку следует записать инструкцию перехода на начальный адрес основной программы, которая будет считывать входной отсчет из порта А, обрабатывать его согласно алгоритму фильтрации и выводить в порт В. Основная программа должна отработать до прихода в порт А очередных данных, а до этого не прерываться. Первое обеспечивается эффективностью алгоритма и быстродействием программы. Второе – хранением высокого уровня на входе RST 7.5 триггером, поэтому ложная импульсная помеха, попавшая на этот вход во время выполнения программы, не прервет и не перезапустит ее. Но чтобы обеспечить возможность нового запуска программы при приходе очередных данных, необходимо сбросить триггер, что осуществляется переустановкой маски, описанной выше. Кроме того, необходимо записать исходное число 50FF в указатель стека, так как при каждом прерывании в стек записывается счетчик команд, а указатель стека дважды декрементируется, что в конечном итоге приведет к достижению им рабочей области и засорению ее неверной информацией. Завершается основная программа командой HLT, и процессор ожидает поступления на вход RST 7.5 очередного положительного перепада.

**3. Разработка конкретных алгоритмов работы устройства**

Как уже оговаривалось, при рестарте микропроцессор начинает исполнять инструкции, находящиеся в области векторов прерываний ПЗУ в зависимости от типа рестарта. При включении питания исполняется команда, записанная в ячейке с адресом 0000(16), а при аппаратном прерывании RST 7.5. - начальный адрес 003С(16). В эти ячейки следует поместить команды безусловного перехода на определенный адрес JMP. В первом случае ссылка осуществляется на программу инициализации, во втором – на основную программу обработки оцифрованного сигнала. Обе программы ориентировочно небольшие по сравнению с размером ПЗУ ( 2кбайт ), поэтому их можно рассредоточить в адресном пространстве. Например, программу инициализации поместить с адреса 0200(16), а основную программу – начиная с адреса 0500(16). Тогда ориентировочная карта памяти выглядит следующим образом:

|  |  |
| --- | --- |
| 0000 | JMP 0200 |
| 0001 |  |
| … |  |
| 003C | JMP 0500 |
| 003D |  |
| … |  |
| 0200 | Инициализация |
| 0201 | … |
| … |  |
| 0500 | Осн. программа |
| 0501 | … |
| … |  |
| 5000 | ОЗУ |
| 5001 |  |
| … |  |
| 7001 | Порт А |
| 7002 | Порт В |
| 7003 |  |
| … |  |

ПЗУ, ОЗУ и порты ввода-вывода имеют одно адрестое пространство, поэтому обращение к ним можно производить с помощью одних и тех же инструкций: LDA (загрузка ) и STA (запоминание). При этом данные соответственно загружаются в аккумулятор и копируется из него.

Алгоритм программы инициализации имеет следующий вид:

Основная программа должна начинаться с загрузки в аккумулятор входного отсчета из порта А. Входные данные, получаемые от АЦП, представлены в беззначном коде (минимальное число 00000000(2), максимальное 11111111(2)). Чтобы перейти к рабочему коду процессора – дополнительному, достаточно прибавить к входному число 80(16) = 10000000(2), это эквивалентно отсечению постоянной составляющей. Далее предстоит умножение числа на коэффициенты и суммирование согласно выражению (1). Для этих целей необходимо хранить предыдущие отсчеты в памяти ОЗУ. Возможен следующий вариант организации оперативной (рабочей) памяти:

|  |  |  |
| --- | --- | --- |
| Отсчет\*0.4 | \*коэффиц. | Адрес(16) |
| Xi-4 | 0.1 | 5000 |
| Xi-3 | 1 | 5001 |
| 0.65 | 5002 |
| 0.1 | 5003 |
| Xi-2  **Σ** | 1 | 5004 |
| 0.65 | 5005 |
| 0.1 | 5006 |
| Xi-1 | 1 | 5007 |
| 0.65 | 5008 |
| 0.1 | 5009 |
| Xi | 1 | 500А |
| 0.65 | 500В |
| 0.1 | 500С |

ОЗУ начинается с адреса 5000(16) , тут и можно разместить рабочую область. Преобразованный отсчет умножается на 0.4, помещается в ячейку с адресом 500А, далее умножается на 0.65 и помещается в ячейку 500В. Эти два умножения выполняются стандартно – сдвигом множителя и сложением.

Коэффициент 0.4 можно представить в 8-разрядной сетке приближенно: 0.0110011 = 0.25+0.125+0.015625+0.0078125 = 0.3984325. Алгоритм умножения на 0.4:

Аналогично производится умножение на коэффициент 0.65, который в двоичном виде можно представить как 0.1010011 = 0.5+0.125+0.015625+0.0078125=0.6484375.

Как видно, алгоритмы схожи. Деление на два может быть реализовано с помощью арифметического сдвига аккумулятора путем сочетания инструкций RLC, RAR, RAR. Такая последовательность команд исключает потерю знака из 7-го разряда, а значит, может применяться для положительных и отрицательных чисел. Округление представляет собой сложение с элементом, оказавшимся после деления вне разрядной сетки. При арифметическом сдвиге этот элемент переходит во флаг переноса С, следовательно, для округления может быть использована команда сложения с переносом ADC E, где Е – заранее обнуленный регистр, либо ADC B, когда непосредственно за округлением в алгоритме следует сложение с регистром В. Чтобы не потерять множитель-отсчет, перед сложением с регистром В его необходимо сохранить, например, в регистр С, после сложения восстановить в аккумулятор и продолжать операции арифметического сдвига.

Умножение на 0.1 можно выполнить более просто, если учесть, что С(b0+b1+b2+b3+b4)=С(2b0+2b1+b2)=1 – это легко проверить. Домножив обе части равенства на величину преобразованного в дополнительный код отсчета, получим:

Xi = 2Cb0Xi + 2Cb1Xi +Cb2Xi = 2\*0.4\*0.1Xi + 2\*0.4\*0.65Xi +0.4Xi

Значение 0.4\*0.1\*Xi является искомым и должно быть записано в ячейку 500С:

Алгоритм умножения на 0.1 должен реализовать данное выражение, причем исходные данные в числителе известны, если преобразованный отсчет Xi перед умножением на 0.4 сохранить в регистре, например, D. Значение 0.4\*0.65\*Xi на данном этапе находится в аккумуляторе. Умножение его на 2 можно выполнить последовательностью инструкций RAL, RAL, RRC, ANI 11111110(2) , позволяющей сохранить знак числа и исключить нечетность из-за перехода единицы в нулевой разряд аккумулятора из флага переноса.

Теперь рабочая область ОЗУ подготовлена к суммированию в соответствии с выражением (1), как показано на рис. . Суммируется содержимое ячеек с адресами 500С, 5008, 5004, 5002, 5000. Для этого можно использовать инструкцию ADD M, а в качестве указателя на адреса – регистр HL. После суммирования данные готовы к выводу в порт.

Далее содержимое ячеек 5003 – 500С подлежит переносу на три ячейки вверх – таким образом текущий отсчет становится предыдущим, предыдущий – препредыдущим и т. д. Сдвиг удобно осуществлять парами, используя для этого инструкции LHLD и SHLD.

Последней операцией основной программы является сброс входного триггера входа микропроцессора RST 7.5., после чего вход будет воспринимать сигнал прерывания как команду рестарта микропроцессора. Для сброса триггера просто переустанавливается маска прерываний, описанная выше.

Алгоритм работы основной программы ( алгоритм фильтрации ) выглядит следующим образом:

**4. Разработка и отладка программного обеспечения**

Программа, реализующая приведенные выше алгоритмы, разработана для микропроцессора КР1821ВМ85. При создании программы использовался в качестве компилятора кросс-ассемблер ASM80, позволяющий с помощью директив и меток, располагаемых в тексте программы, уйти от прямых физических адресов. Для пошагового анализа работы программы, выявления и исправления ошибок в реализации алгоритмов использовался отладчик DEB80. Исходный текст программы, а также ее листинг с указанием физических адресов, приведены ниже. Основная часть программы, реализующая алгоритм фильтрации, занимает ячейки ПЗУ с адреса 0500(16) по 0598(16) включительно, то есть 152 байта памяти. На исполнение основной части программы с момента рестарта RST 7.5. уходит 727 машинных тактов при максимально допустимом числе их N=1000, то есть программа, работая, укладывается в интервал дискретизации с запасом по времени ≈ 25 %, что означает выполнение одного из требований к устройству – обработка сигнала в реальном масштабе времени.

Ввиду отсутствия в обобщенной структуре фильтра обратных связей и конечности цифровой импульсной характеристики фильтр абсолютно устойчив, то есть любое входное возмущение не приведет к генерации.

Исходя из того, что переходная характеристика не имеет выброса и стремится к единице, можно утверждать, что при условии корректной реализации алгоритма не будет происходить переполнения разрядной сетки, то есть переходной процесс не превышает постоянный входной уровень, а установившийся режим в точности повторяет его. Это относится и к максимально допустимым цифровым значениям входных отсчетов. Переполнение может иметь место при некоторых упущениях в реализованной программе, например, когда результаты всех умножений ( на 0.4, 0.65, 0.1 ) округлятся в большую сторону. Однако особый способ реализации умножения на 0.1 ( вычитанием из единицы ) исключает такую возможность.

С целью проверки на переполнение был осуществлен ручной и автоматический расчет работы программы. В качестве исходных принимались два критических случая – минимального и максимального постоянных уровней на входе.

В первом случае от АЦП приходил максимальный отсчет FFh, который после перехода к дополнительному коду принимал значение 7Fh. Далее это значение умножалось согласно алгоритму на 0.4, и результат 33h записывался в ячейку 500А. Он же умножался затем на 0.65 ( результат 23h – в ячейку 500В ) и на 0.1 ( результат 03h – в ячейку 500С ). Перед суммированием эти результаты были занесены в ячейки 5000 – 5009 согласно рис. . В итоге суммирование дало результат, равный входному: 03h + 23h + 33h + 23h + 03h = 7Fh – он и был отправлен в порт В.

Во втором случае входной отсчет имел значение 00h, после преобразования 80h, после умножения - CDh, E0h и FАh в ячейках 500А, 500В и 500С соответственно. Эти же значения помещаются в другие рабочие ячейки, соответствующие алгоритму суммирования. Результат суммирования: FAh + E0h + CDh + E0h + FAh = 81h – был отправлен в порт.

И в том, и в другом случае переполнения не произошло.

**5. Погрешность расчета, связанная с конечным представлением коэффициентов**

Ранее уже отмечалось, что заданные коэффициенты представлены округленно: 0.4 как 0.3984325, 0.65 как 0.6484375. Нетрудно показать, что умножение на 0.1 фактически является умножением на 0.1015625. Поэтому реальная амплитудно-частотная характеристика фильтра будет отличаться от заданной с помощью передаточной функции. Однако отличия истинных коэффициентов от заданных столь мало, что реальная АЧХ практически не отличается от заданной, показанной на рис. .

Другим существенным фактором искажений является конечное представление самих отсчетов. Так, после умножения на 0.4, а затем на 0.1, существенными остаются только разряды с 5-го по 7-й, причем 7-й – знаковый. Остальные 5 разрядов по сути отбрасываются, то есть возможна ситуация, когда при изменении значения отсчета в пределах 1Fh результат умножения на коэффициент 0.1 остается неизменным. Это также влияет на погрешность при расчете.

**6. Описание принципиальной схемы устройства**

Принципиальная схема цифрового фильтра поясняет электрические соединения между отдельными элементами, которыми являются:

1. цифровые микросхемы комплекта КР1821, обеспечивающие минимальную конфигурацию микропроцессорной системы;
2. микросхема аналогово-цифрового преобразователя;
3. микросхема устройства выборки и хранения;
4. схема формирования управляющих стробов, состоящая из двух RC-цепей и элементов И-НЕ, использующихся в качестве инверторов;
5. операционные усилители, служащие в качестве развязывающих элементов и основа схемы смещения и масштабирования;
6. четыре разъема:

а) для входного сигнала ( может использоваться стандартный разъем типа JACK );

б) для вывода параллельного кода отсчета;

в) разъем питающих напряжений и заземления;

г) разъем синхронизации с внешним устройством.

1. вспомогательные элементы.

**Заключение**

В данной курсовой работе был спроектирован цифровой сглаживающий фильтр на основе комплекта микросхем КР1821ВМ85. По результатам разработки фильтра можно выработать следующие рекомендации по диагностике неисправностей в процессе эксплуатации.

1. Если данные не выводятся в выходной порт, необходимо проверить наличие сигнала синхронизации выборки на выходе TOUT ( вывод 6 микросхемы DD3 ), а также его форму и период.
2. Если данные есть, но они явно искажены, возможная причина неисправности заключается в рассогласовании работы цепи формирования управляющих стробов, и необходимо проверить наличие и форму сигналов на входах 22 DD5 и 13 DA3, их соответствие диаграммам.

Возможно также, что причиной рассинхронизации является перегрузка выхода СИНХР фильтра: рекомендуется подключать этот выход не более чем к одному внешнему устройству.

Кроме того, следует проверить наличие аналогового сигнала на выходах операционных усилителей и УВХ, соответствие его требованиям ко входному сигналу АЦП.

В остальных случаях причиной неисправности может являться отказ вспомогательных элементов или микросхем.

Список литературы

1. Рафикузаман М. Микропроцессоры и машинное проектирование микропроцессорных систем: В 2-х кн. Кн.1. Пер. с англ.-М.: Мир , 1988.

2. Р. Токхайм. Микропроцессоры. Курс и упражнения. Пер. с англ. В.Н Грасевиче и Л.А. Ильяшенко. М.: Энергоатомиздат, 1988

3. Щелкунов Н.Н., Дианов А.П. Микропроцессорные средства и системы.- М.: Радио и связь,1989.

4. Федорков Б.Г., Телец В.А. Микросхемы ЦАП и АЦП: функционирование, параметры, применение.- М.: Энергоатомиздат, 1990.

5. Микропроцессорный комплект К1810: Структура, программирование, применение: Справочная книга. Под ред. Ю.М.Казаринова.- М.: Высш. шк., 1990.

6. Программирование для микропроцессора К580: Методические указания к лабораторным работам / РГРТА.; Сост.: В.Н.Локтюхин, Ю.А.Сосулин. Рязань, 1996.

7. Перельман Б.П. Отечественные микросхемы и их зарубежные аналоги : справочник. М: 1998