**Министерство образования и науки Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «мэи»**

**Институт Радиотехники и электроники им. В.А. Котельникова**

**Кафедра Формирования и обработки радиосигналов**

КУРСОВОЙ ПРОЕКТ

**по дисциплине «Методы и устройства цифровой обработки сигналов»**

Тема: **«Цифровая обработка сигналов на микроконтроллере»**

|  |  |  |
| --- | --- | --- |
| Студент, гр. ЭР-15-15 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Жеребин В.Р. |
| Руководитель, ст. преподаватель | (подпись)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Филатов В.А. |
|  | (подпись) |  |

Москва

2018**содержание**

|  |  |  |
| --- | --- | --- |
|  | Задание | 3 |
| 1. | Самостоятельная домашняя проработка | 4 |
| 1.1. | Определение структуры, основных требований и параметров к микропроцессорной системе (МПС) обработки сигналов. | 4 |
| 1.2. | Полная принципиальная схема МПС. | 5 |
| 1.3. | Алгоритмы основной программы и необходимых подпрограмм. | 6 |
| 1.4. | Теоритический анализ свойств цифрового фильтра. | 13 |
| 1.5. | Реализация и отладка алгоритмов основной программы и подпрограмм на языке Ассемблера и в кодах микроконтроллера PIC18F2520. Полная программная документация. | 17 |
| 2. | Моделирование и отладка проекта в MPLAB X. | 21 |
| 2.1. | Ввод подготовленной программы и исходных данных в память микроконтроллера для отладки в MPLAB. | 21 |
| 2.2. | Отладка в симуляторе MPLAB SIM. | 21 |
| 2.3 | Проверка правильности полученных результатов. | 38 |
| 3. | Проверка правильности функционирования МПС при помощи модуля PICkit | 39 |
|  | Заключение | 42 |

**Задание**

Цель проекта – познакомится с аппаратным построение цифровых систем обработки сигналов с использованием механизма прерываний и встроенных в микроконтроллер устройств ввода-вывода (АЦП, таймер, последовательный порт USART), практическое освоение приемов цифровой обработки сигналов с использованием микроконтроллера PIC18F2520 и выполнение программы с использованием отладчика PICkit. Индивидуальное задания предусматривает обработку отрезка сигнала длительностью 1 секунда, состоящего из 256 8-ми битных отчета, которые необходимо записать в массив ***А***. Результат обработки массива А сохранить в массив ***В***, а так же вывести значения массива В через последовательный порт с заданной скоростью.

1. **Самостоятельная домашняя проработка**

**1.1. Определение структуры, основных требований и параметров к микропроцессорной системе (МПС) обработки сигналов.**

* Канал приема и канал передачи сигналов.
* Тактирование МПС.
* Задействованные аппаратные модули МК и связанные с ним внешние выводы.
* Питание МК.
* Необходимые внешние соединения, цепи обеспечения и согласования с периферийными устройствами.
* Функциональное распределение выводов микросхемы МК.

Для формирования массива А в МПС использую вход ***AN0*** АЦП, с которого по нажатии кнопки подлеченной к выводу (***RB0/INT0***), происходит прерывание основной программы, оцифровка аналогово сигнала с частотой 256 выборок в секунду и запись старших 8 бит в массив А. Во время формирования массива включается светодиод на ***RB2***. После записи 256 отчетов сигнала в массив А светодиод RB2 продолжает гореть и происходит возврат из прерывания в основную программу для обработки этого отрезка сигнала, при этом включается светодиод на ***RB3***. После завершения обработки и записи в массив В, включается светодиод на ***RB4*** и запускается передача массива по USART. По завершении передачи включается светодиод на ***RB5*** и ожидается нажатие кнопки на ***RB1***. Тактирование МПС происходит от внутреннего тактового генератора с стабильной частотой 4 МГц. Питание МК производится от постоянного напряжение +5В. Этот же источник питания используется и для подачи постоянной составляющей сигнала на вход АЦП.

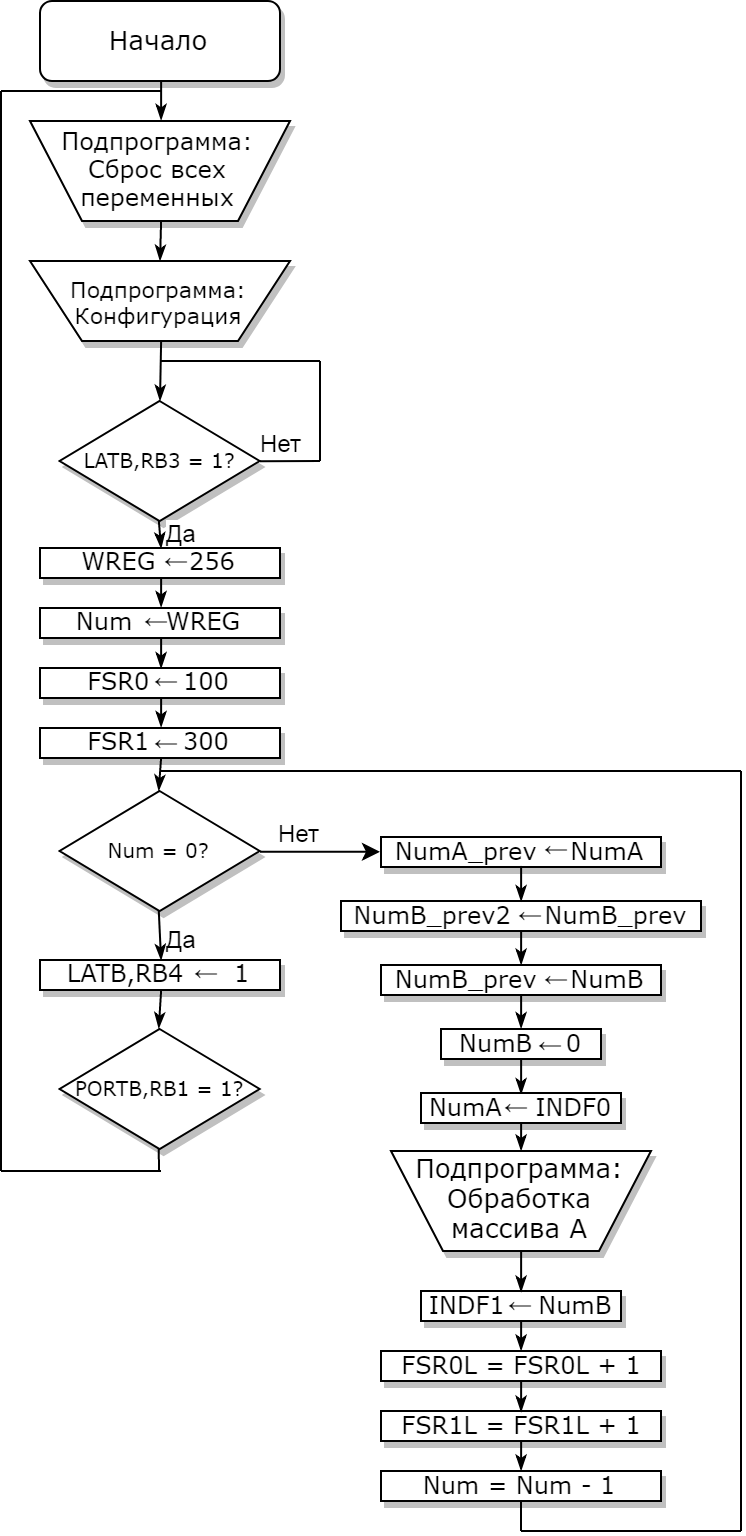
* 1. **Полная принципиальная схема МПС.**

Полная принципиальная схема МПС представлена на рисунке 1.1

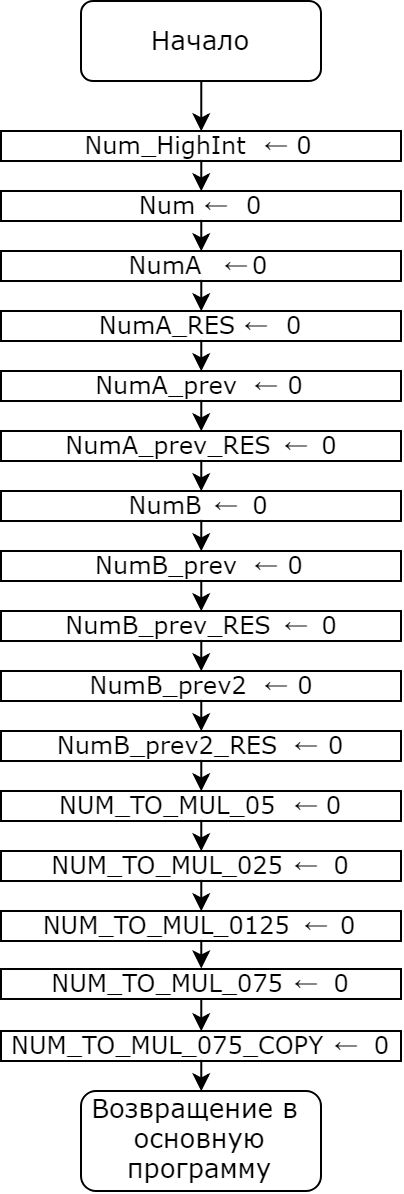
|  |
| --- |
|  |
| Рисунок 1.1 – Полная принципиальная схема МПС. |

* 1. **Алгоритмы основной программы и необходимых подпрограмм.**

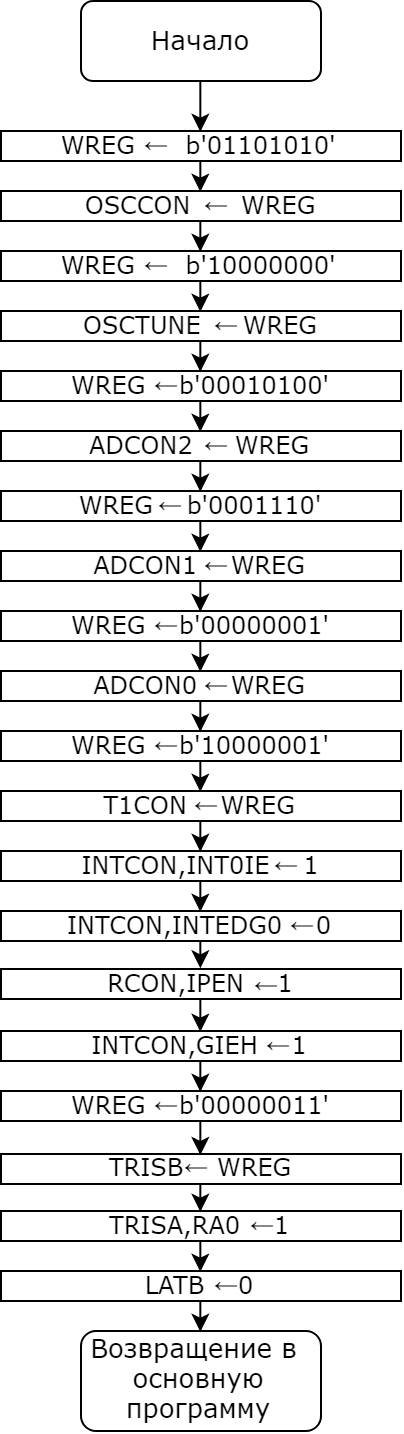
Алгоритм основной программы:



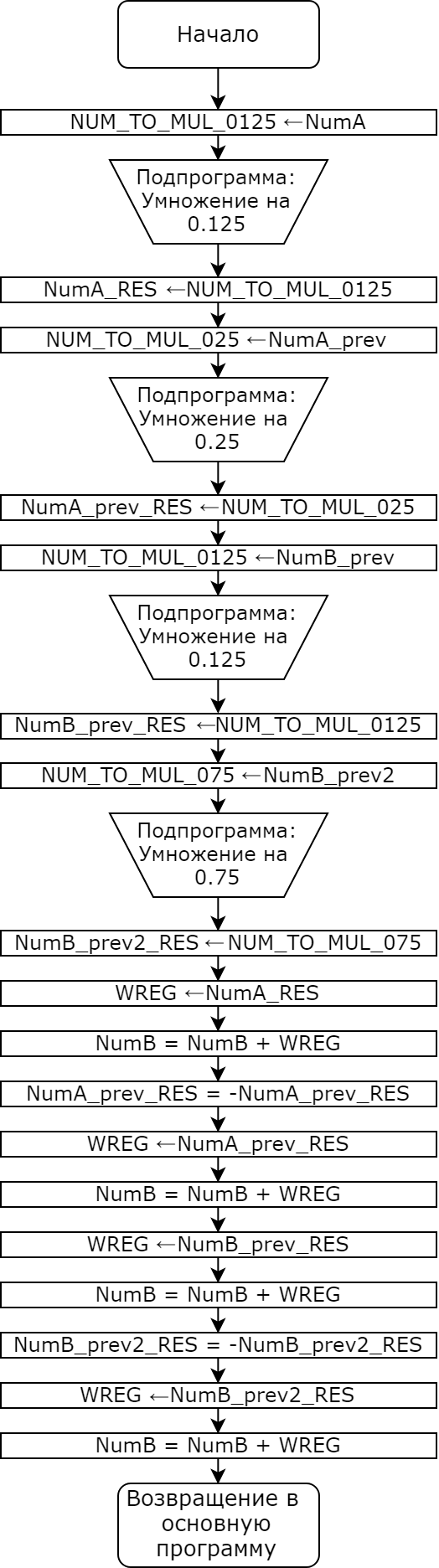
Алгоритм подпрограммы – Сброс всех переменных:



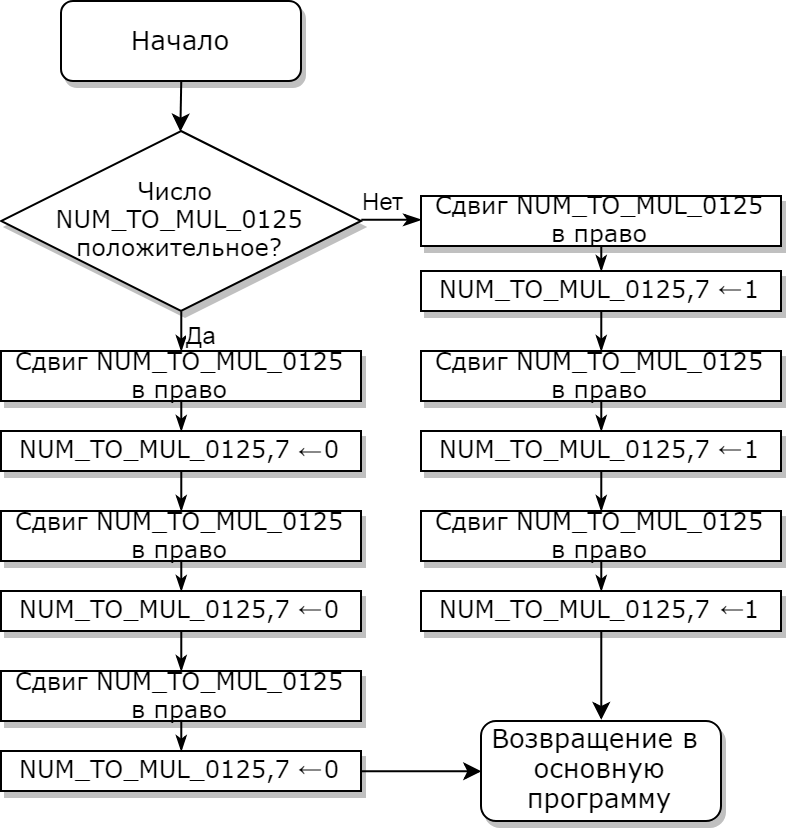
Алгоритм подпрограммы – Конфигурация:



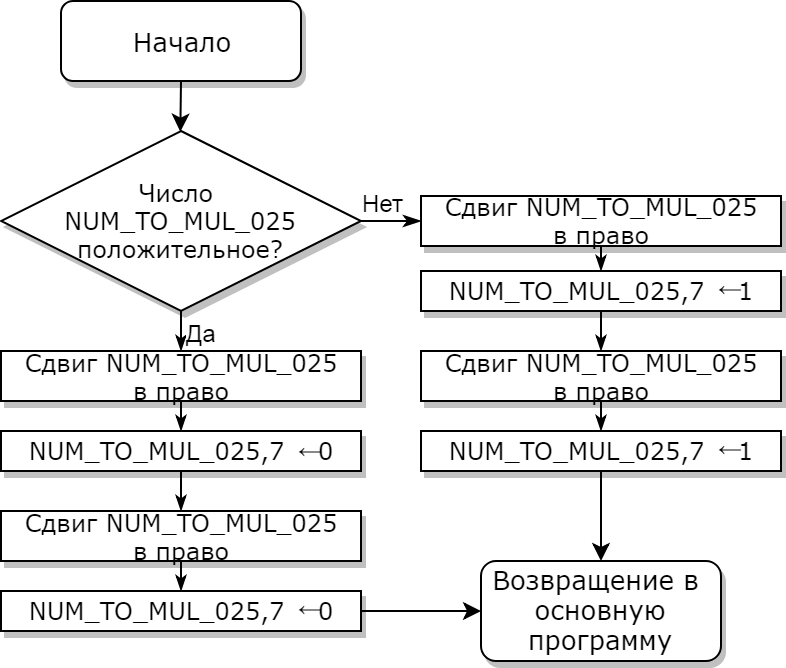
Алгоритм подпрограммы – Обработка массива А:



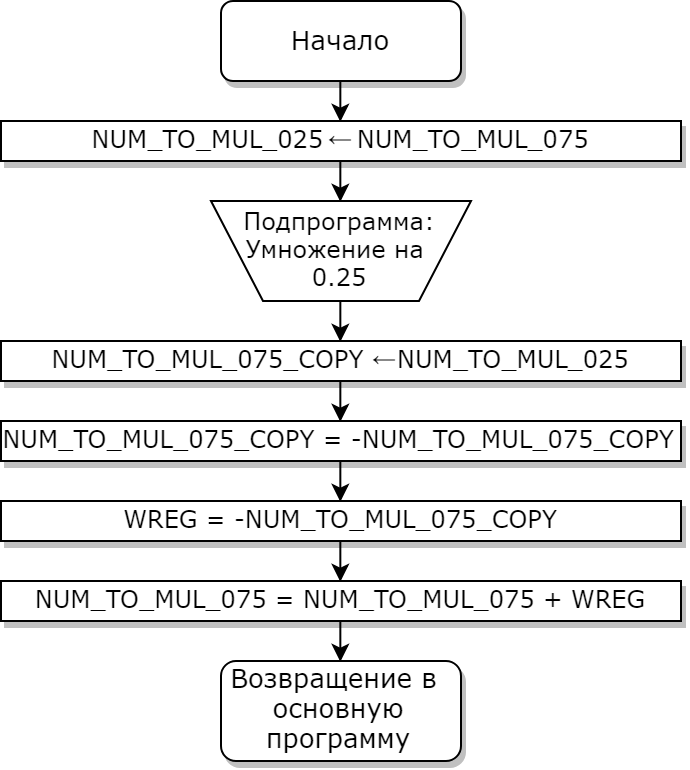
Алгоритм подпрограммы – Умножение на 0.125:



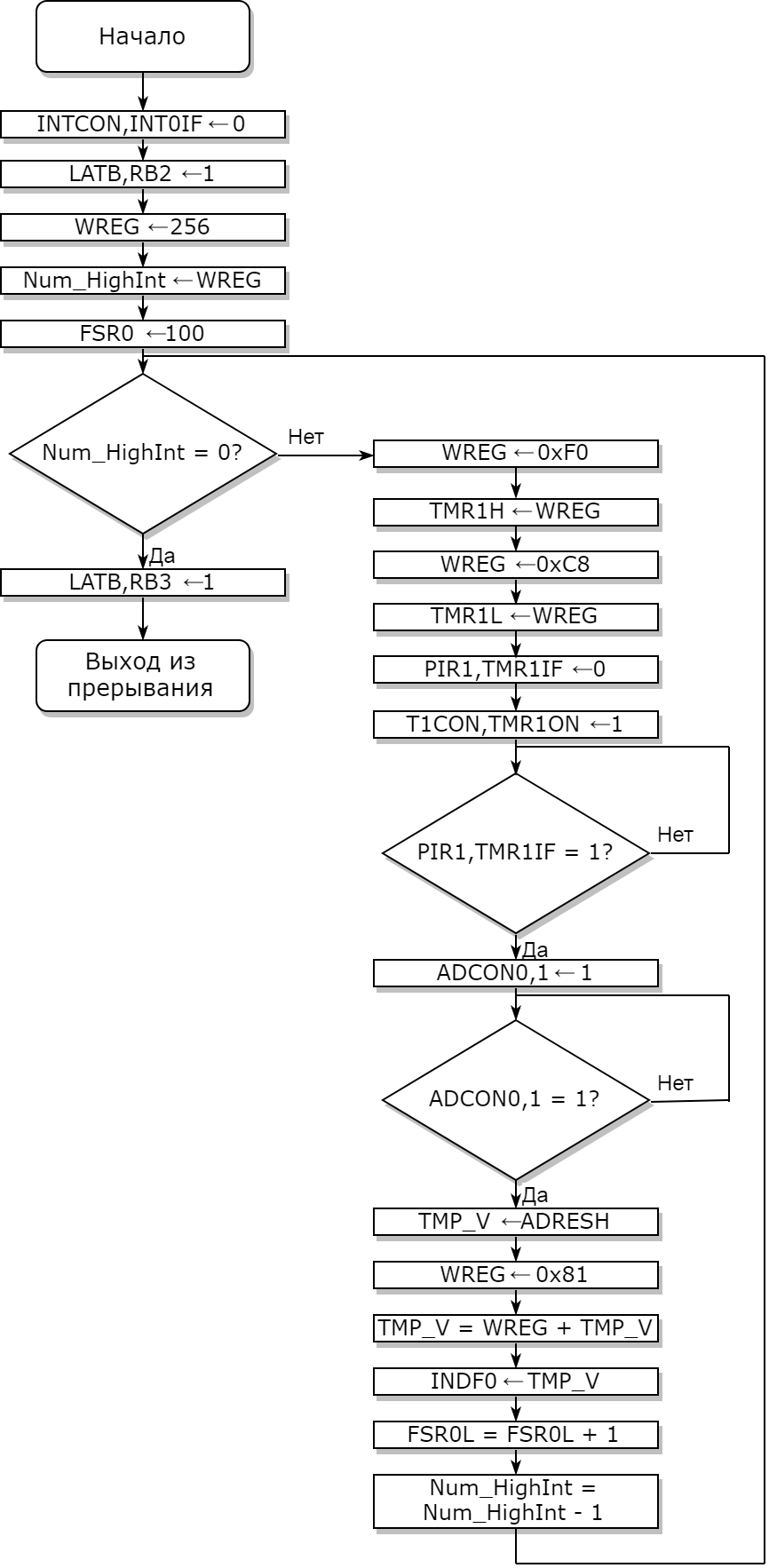
Алгоритм подпрограммы – Умножение на 0.25:



Алгоритм подпрограммы – Умножение на 0.75:



Алгоритм прерывания:



* 1. **Теоритический анализ свойств цифрового фильтра.**

Заданный алгоритм цифровой обработки сигнала:



Где *Аi* и *Bi* элементы массива **А** и **В** соответственно; *i* – номер элемента массива, изменяется от 0 до 255.

По заданному алгоритму можем получить разносное уравнение ЦФ:



Где Y – отсчеты выходного сигнала, X – отсчеты входного сигнала.

Проведем обратное Z-преобразование, для переходя к временным отсчетам:



По полученному выражению получаем импульсную и передаточную характеристики ЦФ. На рисунке 1.2 представлена импульсная характеристика ЦФ. На рисунке 1.3 представлена передаточная характеристика ЦФ.

|  |
| --- |
|  |
| Рисунок 1.2 – Импульсная характеристика ЦФ. |

Импульсная характеристика имеет колебательный затухающий характер. Следовательно данный фильтр – полосовой. Период колебания импульсной характеристики составляет 4 такта. При условии, что за 1 секунду АЦП обрабатывает 256 отчетов, резонансная частота фильтра составит Гц. Импульсная характеристика к 20 такту затухает, значит фильтр имеет конечную импульсную характеристику (КИХ).

|  |
| --- |
|  |
| Рисунок 1.3 – Передаточная характеристика ЦФ. |

Зная разносное уравнение, получим передаточную функцию ЦФ:



Для получение частотных характеристик, проведем замену: 



Где  – нормированная цифровая частота, *fд* = 265 Гц – частота дискретизации. С учетом того, получим следующее выражение:



По данной формуле построим нормированную амплитудно-частотную (рисунок 1.4 а), б)) и фазо-частотную (рисунок 1.5) характеристики ЦФ.

|  |
| --- |
| *f*, Гц      а)    *f*, Гц  б) |
| Рисунок 1.4 – Нормированная амплитудно-частотная характеристика ЦФ.  а) Шкала амплитуды в разах. б) Шкала амплитуды в децибелах. |

|  |
| --- |
| *f*, Гц |
| Рисунок 1.5 – Фазо-частотная характеристика ЦФ. |

* 1. **Реализация и отладка алгоритмов основной программы и подпрограмм на языке Ассемблера и в кодах микроконтроллера PIC18F2520. Полная программная документация.**

Реализация алгоритма основной программы:

Main:

CALL CLRF\_ALL; Вызов подпрограммы: Сброс всех переменных

NOP

CALL CONF; Вызов подпрограммы: Конфигурация

M1: BTFSS LATB,RB3; Ожидание включения светодиода на RB3

BRA M1

MOVLW .256; Запись 256 в WREG

MOVWF Num; Запись из WREG в переменную Num

LFSR FSR0,0x100; Задаем базовый адрес для массива А - 0x100

LFSR FSR1,0x300; Задаем базовый адрес для массива В - 0x300

M2: MOVFF NumA,NumA\_prev; Перенос значения А(n) в А(n-1)

MOVFF NumB\_prev,NumB\_prev2; Перенос значения B(n-1) в B(n-2)

MOVFF NumB,NumB\_prev; Перенос значения B(n) в B(n-1)

CLRF NumB; Обнуление значения B(n)

MOVFF INDF0,NumA; Перенос значения из памяти МК в А(n)

CALL MAIN\_FUNC; Вызов подпрограммы: Обработка массива А

MOVFF NumB,INDF1; Перенос значения B(n) в память МК

INCF FSR1L,F; Переход на следующий адрес массива В

INCF FSR0L,F; Переход на следующий адрес массива А

DECFSZ Num,F; Уменьшение на 1 количество операций, и проверка на нулевое значение

BRA M2

BSF LATB,RB4; Включение светодиода на RB4

M3: BTFSC PORTB,RB1; Ожидание нажатия кнопки на RB1

BRA M3

GOTO Main; Перезапуск основной программы

Реализация алгоритма подпрограммы – Сброс всех переменных:

CLRF\_ALL:

CLRF Num\_HighInt; Сброс переменной Num\_HighInt

CLRF Num; Сброс переменной Num

CLRF NumA; Сброс переменной NumA

CLRF NumA\_RES; Сброс переменной NumA\_RES

CLRF NumA\_prev; Сброс переменной NumA\_prev

CLRF NumA\_prev\_RES; Сброс переменной NumA\_prev\_RES

CLRF NumB; Сброс переменной NumB

CLRF NumB\_prev; Сброс переменной NumB\_prev

CLRF NumB\_prev\_RES; Сброс переменной NumB\_prev\_RES

CLRF NumB\_prev2; Сброс переменной NumB\_prev2

CLRF NumB\_prev2\_RES; Сброс переменной NumB\_prev2\_RES

CLRF NUM\_TO\_MUL\_05; Сброс переменной NUM\_TO\_MUL\_05

CLRF NUM\_TO\_MUL\_025; Сброс переменной NUM\_TO\_MUL\_025

CLRF NUM\_TO\_MUL\_0125; Сброс переменной NUM\_TO\_MUL\_0125

CLRF NUM\_TO\_MUL\_075; Сброс переменной NUM\_TO\_MUL\_075

CLRF NUM\_TO\_MUL\_075\_COPY; Сброс переменной NUM\_TO\_MUL\_075\_COPY

CLRF TMP\_V; Сброс переменной TMP\_V

RETURN; Возвращение в основную программу

Реализация алгоритма подпрограммы – Конфигурация:

CONF:

;OSCILLATOR

MOVLW b'01101010'; Fosc = 4 MHz

MOVWF OSCCON

MOVLW b'10000000'; F = 8 MHz

MOVWF OSCTUNE

;ADC

MOVLW b'00010100'; Включаем АЦП

MOVWF ADCON2

MOVLW b'00001110'; Аналоговый вход AN0, остальные - цифровые

MOVWF ADCON1

MOVLW b'00000001'; Используем канал AD0

MOVWF ADCON0

;TIMER

MOVLW b'10000001';16-битный, Т=1мкс

MOVWF T1CON

BSF INTCON,INT0IE; Нажатие кнопки вызывает прерывания

BCF INTCON2,INTEDG0; По отрицательному фронту

BSF RCON,IPEN; Разрешение прерываний

BSF INTCON,GIEH; Разрешение глобальных прерываний

; Порты входных индикаторов

BSF TRISA,RA0

BSF TRISB,RB0

BSF TRISB,RB1

; Порты выходных индикаторов

BCF TRISB,RB2

BCF TRISB,RB3

BCF TRISB,RB4

BCF TRISB,RB5

CLRF LATB; Сброс всех значений в регистре LATB

RETURN; Возвращение в основную программу

Реализация алгоритма подпрограммы – Обработка массива А:

MAIN\_FUNC:

MOVFF NumA,NUM\_TO\_MUL\_0125; Запись значения NumA в переменную NUM\_TO\_MUL\_0125

CALL FUNC\_MUL\_TO\_0125; Вызов подпрограммы: Умножение на 0.125

MOVFF NUM\_TO\_MUL\_0125,NumA\_RES;

MOVFF NumA\_prev,NUM\_TO\_MUL\_025; Запись значения NumA\_prev в переменную NUM\_TO\_MUL\_025

CALL FUNC\_MUL\_TO\_025; Вызов подпрограммы: Умножение на 0.25

MOVFF NUM\_TO\_MUL\_025,NumA\_prev\_RES; Запись значения NUM\_TO\_MUL\_025 в переменную NumA\_prev\_RES

MOVFF NumB\_prev,NUM\_TO\_MUL\_0125; Запись значения NumB\_prev в переменную NUM\_TO\_MUL\_0125

CALL FUNC\_MUL\_TO\_0125; Вызов подпрограммы: Умножение на 0.125

MOVFF NUM\_TO\_MUL\_0125,NumB\_prev\_RES; Запись значения NUM\_TO\_MUL\_0125 в переменную NumB\_prev\_RES

MOVFF NumB\_prev2,NUM\_TO\_MUL\_075; Запись значения NumB\_prev2 в переменную NUM\_TO\_MUL\_075

CALL FUNC\_MUL\_TO\_075; Вызов подпрограммы: Умножение на 0.75

MOVFF NUM\_TO\_MUL\_075,NumB\_prev2\_RES; Запись значения NUM\_TO\_MUL\_075 в переменную NumB\_prev2\_RES

MOVF NumA\_RES,W; Запись значения NumA\_RES в WREG

ADDWF NumB,F; Операция сложения WREG и NumB. Результат записывается в NumB

NEGF NumA\_prev\_RES; Перевод NumA\_prev\_RES в дополнительный код

MOVF NumA\_prev\_RES,W; Запись значения NumA\_prev\_RES в WREG

ADDWF NumB,F; Операция сложения WREG и NumB. Результат записывается в NumB

MOVF NumB\_prev\_RES,W; Запись значения NumB\_prev\_RES в WREG

ADDWF NumB,F; Операция сложения WREG и NumB. Результат записывается в NumB

NEGF NumB\_prev2\_RES; Перевод NumA\_prev\_RES в дополнительный код

MOVF NumB\_prev2\_RES,W; Запись значения NumB\_prev2\_RES в WREG

ADDWF NumB,F; Операция сложения WREG и NumB. Результат записывается в NumB

RETURN; Возвращение в основную программу

Реализация алгоритма подпрограммы – Умножение на 0.125:

FUNC\_MUL\_TO\_0125:

BTFSS NUM\_TO\_MUL\_0125,7; Проверка старшего бита в переменной NUM\_TO\_MUL\_0125

BRA K1\_0125

BRA K2\_0125

K1\_0125:RRCF NUM\_TO\_MUL\_0125,F; Сдвиг переменной NUM\_TO\_MUL\_0125 на 1 разряд

BCF NUM\_TO\_MUL\_0125,7; Обнуление старшего разряда

RRCF NUM\_TO\_MUL\_0125,F; Сдвиг переменной NUM\_TO\_MUL\_0125 на 1 разряд

BCF NUM\_TO\_MUL\_0125,7; Обнуление старшего разряда

RRCF NUM\_TO\_MUL\_0125,F; Сдвиг переменной NUM\_TO\_MUL\_0125 на 1 разряд

BCF NUM\_TO\_MUL\_0125,7; Обнуление старшего разряда

BRA K3\_0125

K2\_0125:RRCF NUM\_TO\_MUL\_0125,F; Сдвиг переменной NUM\_TO\_MUL\_0125 на 1 разряд

BSF NUM\_TO\_MUL\_0125,7; Установка старшего разряда

RRCF NUM\_TO\_MUL\_0125,F; Сдвиг переменной NUM\_TO\_MUL\_0125 на 1 разряд

BSF NUM\_TO\_MUL\_0125,7; Установка старшего разряда

RRCF NUM\_TO\_MUL\_0125,F; Сдвиг переменной NUM\_TO\_MUL\_0125 на 1 разряд

BSF NUM\_TO\_MUL\_0125,7; Установка старшего разряда

BRA K3\_0125

K3\_0125:RETURN; Возвращение в основную программу

Реализация алгоритма подпрограммы – Умножение на 0.25:

FUNC\_MUL\_TO\_025:

BTFSS NUM\_TO\_MUL\_025,7; Проверка старшего бита в переменной NUM\_TO\_MUL\_025

BRA K1\_025

BRA K2\_025

K1\_025: RRCF NUM\_TO\_MUL\_025,F; Сдвиг переменной NUM\_TO\_MUL\_025 на 1 разряд

BCF NUM\_TO\_MUL\_025,7; Обнуление старшего разряда

RRCF NUM\_TO\_MUL\_025,F; Сдвиг переменной NUM\_TO\_MUL\_025 на 1 разряд

BCF NUM\_TO\_MUL\_025,7; Обнуление старшего разряда

BRA K3\_025

K2\_025: RRCF NUM\_TO\_MUL\_025,F; Сдвиг переменной NUM\_TO\_MUL\_025 на 1 разряд

BSF NUM\_TO\_MUL\_025,7; Установка старшего разряда

RRCF NUM\_TO\_MUL\_025,F; Сдвиг переменной NUM\_TO\_MUL\_025 на 1 разряд

BSF NUM\_TO\_MUL\_025,7; Установка старшего разряда

BRA K3\_025

K3\_025: RETURN; Возвращение в основную программу

Реализация алгоритма подпрограммы – Умножение на 0.75:

FUNC\_MUL\_TO\_075:

MOVFF NUM\_TO\_MUL\_075,NUM\_TO\_MUL\_025; Запись значения NUM\_TO\_MUL\_075 в переменную NUM\_TO\_MUL\_025

CALL FUNC\_MUL\_TO\_025; Вызов подпрограммы: Умножение на 0.25

MOVFF NUM\_TO\_MUL\_025,NUM\_TO\_MUL\_075\_COPY; Запись значения NUM\_TO\_MUL\_025 в переменную NUM\_TO\_MUL\_075\_COPY

NEGF NUM\_TO\_MUL\_075\_COPY; Перевод NUM\_TO\_MUL\_075\_COPY в дополнительный код

MOVF NUM\_TO\_MUL\_075\_COPY,W; Запись значения NUM\_TO\_MUL\_075\_COPY в WREG

ADDWF NUM\_TO\_MUL\_075,F; Операция сложения WREG и NUM\_TO\_MUL\_075. Результат записывается в NUM\_TO\_MUL\_075

RETURN; Возвращение в основную программу

Реализация алгоритма прерываний:

HighInt:

; \*\*\* high priority interrupt code goes here \*\*\*

BCF INTCON,INT0IF; Сброс флага прерывания

BSF LATB,RB2; Включение светодиода на RB2

MOVLW .256; Запись 256 в WREG

MOVWF Num\_HighInt; Запись из WREG в переменную Num\_HighInt

LFSR FSR0,0x100; Задаем базовый адрес для массива А - 0x100

Mint1: MOVLW 0xF0; Запись 0xF0 в WREG

MOVWF TMR1H; Запись из WREG в переменную TMR1H

MOVLW 0xC8; Запись 0xC8 в WREG

MOVWF TMR1L; Запись из WREG в переменную TMR1L

BCF PIR1,TMR1IF; Сброс флага таймера

BSF T1CON,TMR1ON; Включение таймера

Mint2: BTFSS PIR1,TMR1IF; Задержка ~3.9 мс

BRA Mint2

BSF ADCON0,1; Запуск АЦП

Mint3: BTFSS ADCON0,1; Проверка на завершение оцифровки

BRA Mint3

MOVFF ADRESH,TMP\_V; Запись значений в массив А

MOVLW 0x81; Запись 0x81 в WREG

ADDWF TMP\_V,F; Операция сложения WREG и TMP\_V. Результат записывается в TMP\_V

MOVFF TMP\_V,INDF0; Перенос значения из TMP\_V в память МК

INCF FSR0L,F; Переход на следующий адрес массива А

DECFSZ Num\_HighInt,F; Уменьшение на 1 количество операций, и проверка на нулевое значение

BRA Mint1

BSF LATB,RB3; Включение светодиода на RB3

RETFIE FAST; Возращение из прерывания

Список пользовательских констант:

NUM\_TO\_MUL\_05

NUM\_TO\_MUL\_025

NUM\_TO\_MUL\_0125

NUM\_TO\_MUL\_075

NUM\_TO\_MUL\_075\_COPY

Num\_HighInt

Num

NumA

NumA\_prev

NumB

NumB\_prev

NumB\_prev2

NumA\_RES

NumA\_prev\_RES

NumB\_prev\_RES

NumB\_prev2\_RES

TMP\_V

Список подпрограмм:

CLRF\_ALL

CONF

MAIN\_FUNC

FUNC\_MUL\_TO\_0125

FUNC\_MUL\_TO\_025

FUNC\_MUL\_TO\_075

Список источников прерываний:

RB0/INT0 – 21 вывод микросхемы.

Список служебных регистров:

|  |  |
| --- | --- |
| WREG  BSR  STATUS  INDF0  FSR0  FSR0L  INDF1  FSR1  FSR1L  LATB  PORTB  TRISB  TRISA | OSCCON  OSCTUNE  ADCON2  ADCON1  ADCON0  T1CON  INTCON  INTCON2  RCON  TMR1H  TMR1L  PIR1  ADRESH |

1. **моделирование и отладка проекта в mplab x.**
   1. **Ввод подготовленной программы и исходных данных в память микроконтроллера для отладки в MPLAB.**

В пакете MPLAB X v4.00 создаем новый проект с названием, к примеру, Main.X. В этом проекте создаем файл формата .ASM с названием KP. Вводим подготовленную программу и проверяем на наличие ошибок. Ошибок не обнаружено, значит собираем проект и запускаем симуляцию. Для получения импульсной характеристики создадим в памяти МК на 100 адресе значение равное 10010 или 6416. Для получения переходной характеристики создадим файл тестового сигнала с постоянным значением равным 10010 или 6416.

* 1. **Отладка в симуляторе MPLAB SIM**

Для отображения результата в наглядном виде будем использовать плагин **DMCI Window.** Это позволит нам получать значения в графическом виде, что существенно облегчит анализ цифровой обработки.

**Симуляция импульсной характеристики.**

|  |
| --- |
|  |
| Рисунок 2.1 – Результат симуляции импульсной характеристики. |

Общий вид и значения полностью совпадают с рассчитанными в пункте **1.4**.

**Симуляция передаточной характеристики.**

|  |
| --- |
|  |
| Рисунок 2.2 – Результат симуляции переходной характеристики. |

Общий вид и значения так же полностью совпадают с рассчитанными в пункте **1.4**.

**Симуляция сигналов разной частоты.**

|  |
| --- |
| C:\Users\User-PC\AppData\Local\Microsoft\Windows\INetCache\Content.Word\sin1hz.png |
| C:\Users\User-PC\AppData\Local\Microsoft\Windows\INetCache\Content.Word\sin1hzres.png |
| Рисунок 2.3 – Результат симуляции сигнала с частотой 1 Гц. |

|  |
| --- |
|  |
|  |
| Рисунок 2.4 – Результат симуляции сигнала с частотой 8 Гц. |

|  |
| --- |
|  |
|  |
| Рисунок 2.5 – Результат симуляции сигнала с частотой 16 Гц. |

|  |
| --- |
|  |
|  |
| Рисунок 2.6 – Результат симуляции сигнала с частотой 32 Гц. |

|  |
| --- |
|  |
|  |
| Рисунок 2.7 – Результат симуляции сигнала с частотой 48 Гц. |

|  |
| --- |
|  |
|  |
| Рисунок 2.8 – Результат симуляции сигнала с частотой 56 Гц. |

|  |
| --- |
|  |
|  |
| Рисунок 2.8– Результат симуляции сигнала с частотой 64 Гц. |

|  |
| --- |
| C:\Users\User-PC\AppData\Local\Microsoft\Windows\INetCache\Content.Word\sin72hz.png |
|  |
| Рисунок 2.9 – Результат симуляции сигнала с частотой 72Гц. |

|  |
| --- |
|  |
|  |
| Рисунок 2.10 – Результат симуляции сигнала с частотой 96 Гц. |

|  |
| --- |
|  |
|  |
| Рисунок 2.11 – Результат симуляции сигнала с частотой 120 Гц. |

**Симуляция суммы сигналов с разными частотам**

|  |
| --- |
|  |
|  |
| Рисунок 2.12 – Результат симуляции сигналов с частотами 5 Гц и 64 Гц. |

|  |
| --- |
|  |
|  |
| Рисунок 2.13 – Результат симуляции сигналов с частотами 5 Гц, 30 Гц и 56 Гц. |

|  |
| --- |
|  |
|  |
| Рисунок 2.14 – Результат симуляции сигналов с частотами 8 Гц, 24 Гц и 40 Гц. |

|  |
| --- |
|  |
|  |
| Рисунок 2.15 – Результат симуляции сигналов с частотами 16 Гц, 32 Гц и 48 Гц. |

* 1. **Проверка правильности полученных результатов.**

По рисункам 2.3 – 2.11, на которых присутствует только один сигнал, можно определить отношение амплитуд. Это отношение и будет значением передаточной функции на определённой частоте. Там, где отношение будет максимальным, будет резонансная частота фильтра.

Таблица 2.1. Отношение амплитуд для различных частот.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| f, Гц | 1 | 8 | 16 | 32 | 48 | 56 | 64 | 72 | 90 | 120 |
|  | 0,078 | 0,078 | 0,109 | 0,156 | 0,375 | 0,703 | 0,781 | 0,594 | 0,25 | 0,188 |

Теперь нормируем полученные значение и нанесем их на график АЧХ фильтра

|  |
| --- |
| *f*, Гц |
| Рисунок 2.16 – Нормированная амплитудно-частотная характеристика ЦФ. |

Как видно по рисунку 2.16 результаты симуляции совпадают с теоритическим анализом с небольшой погрешностью. Резонансная частота в обоих случаях составляет 64 Гц.

1. **ПРоверка правильности функционирования мпс при помощи модуля PICKit.**

В пакете MPLAB X переключим Debub Tool с симуляции на модуль PICkit 3.0. Теперь мы можем подключить МПС к данному модулю. После подключения устанавливаем питание +5В. Теперь на МПС подаем постоянное напряжение смещения с помощью делителя напряжения для того, что бы подавать на АПЦ МК только положительные значения напряжения. Подключаем на вход МПС генератор аналогово сигнала. Результат оцифровки АЦП записываем в массив А. После оцифровки обрабатываем сигнала и записываем его в массив В. Результаты обоих массивов выводим с помощью плагина DMCI.

|  |
| --- |
|  |
|  |
| Рисунок 3.1 – Результат оцифровки и обработки сигнала. Частота сигнала 2 Гц. |

|  |
| --- |
|  |
|  |
| Рисунок 3.2 – Результат оцифровки и обработки сигнала. Частота сигнала 40 Гц. |

|  |
| --- |
| C:\Users\User-PC\Desktop\7 семестр\Микроконтроллеры\Курсовой проект\KP\2 V 63,5 Hz.png |
| C:\Users\User-PC\Desktop\7 семестр\Микроконтроллеры\Курсовой проект\KP\2 V 63,5 Hz out.png |
| Рисунок 3.3 – Результат оцифровки и обработки сигнала. Частота сигнала 2 Гц. |

**заключение**

Амплитуда входного сигнала во всех 3 случая (Рисунки 3.1 – 3.3) составляет 2 В, но так как на входе МПС стоит конденсатор и резисторы, они образуют некий фильтр, который меняет амплитуду оцифрованного сигнала. Эти данные полностью совпадают и с теоритическими расчетами, и с симуляцией. Следует, что данная программа работает исправно, и работа продела верно.

Время работы АЦП составляет 1 секунда, что и требуется по заданию. Время на обработку массива необходимо в разы меньше, оно составляет 29,387 мс. Если изменить время работы АЦП до уровня обработки сигнала, данная программа будет работать с максимальной скоростью.