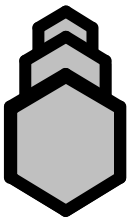


ХИМИКОТЕХНОЛОГИЧЕН И МЕТАЛУРГИЧЕН УНИВЕРСИТЕТ
ФАКУЛТЕТ ПО ХИМИЧНО И СИСТЕМНО ИНЖЕНЕРСТВО
КАТЕДРА "АВТОМАТИЗАЦИЯ НА ПРОИЗВОДСТВОТО"

ДИПЛОМНА РАБОТА

ТЕМА: ПРОЕКТИРАНЕ, ИЗРАБОТВАНЕ И ТЕСТВАНЕ НА СИСТЕМА ЗА ЦИФРОВА ФИЛТРАЦИЯ НА СИГНАЛИ

Ръководител катедра :
.....
(доц. д-р Венцислав Цочев)

Научен ръководител :
.....
(доц. д-р Венцислав Цочев)

Дипломант:.....
..
(инж. Иво П. Георгиев, фак.№ МС 145)

София, 2005

1.Теоретична част

1.1 Филтрация на сигнали

Филтрацията присъства като неизменна част в системите от заобикалящия ни свят. Филтрация като цяло е твърде общо понятие и би трябвало предварително да се конкретизира вида на величините, които ще се филтрират. Отстраняването на механични вибрации при движение на превозни средства, премахването на части от спектъра на светлината и отделянето на електрически сигнали с определени честоти са типични примери за филтрация, но твърде различни по начина си на реализация. Ако за механичната система решението би могло да бъде амортисьор, хидравлика или пружина, то за останалите два случая ще е необходимо да се използват оптически и електрически филтри.

Въпреки физическите разлики на величините, които се подлагат на филтрация, би могло да се даде общо определение на понятието като се вземе предвид неговата същност. Следователно може да се каже, че филтрацията е процес, при който се ограничава честотния спектър на периодичните сигнали (величини) до предварително зададени граници.

При измерване на физични величини обикновенно се извършва филтрация на сигнала получен на изхода на първичните преобразуватели с цел да се отдели полезения сигнал, носещ информация за интересуващата ни величина. Когато става дума за филтрация на електрически сигнали, обикновено нещата опират до проектирането на аналогови филтри. В много от случаите покриването на изискванията по отношение на стръмност на сряза и желано затихване в лентите на задържане са свързани с проектиране на филтри от много висок ред, което оскъпява значително решението на проблема. В много от случаите желаните показатели се оказват дори непосилни за аналоговите филтри. Едно решение на този проблем е използването на цифрови филтри, които в последните години масово се внедряват в системите.

Излизането на пазара на висококоростни цифрови сигнални процесори **DSP** и програмируеми логически схеми **PLD** дава нови възможности на инженерите да реализират идеите си на едно високотехнологично ниво. Може би най-голямо приложение **DSP** са намерили в сферата на комуникациите, обработката на аудио и видео изображения и автоматизацията.

В настоящата дипломна работа ще бъде разработена хардуерна микроконтролерна система за цифрова филтрация на сигнали, както и софтуера необходим за нея. Ще бъдат разгледани различните аспекти на проектирането и реализацията както на хардуера, така и на софтуера.

1.2Проектиране на нерекурсивни цифрови филтри по метода на прозорците

Нерекурсивните цифрови филтри (**НРЦФ**) представляват дискретни системи, които имат крайна импулсна характеристика **КИХ[6]**. Съкращението използвано в западната литература за този вид системи е **FIR**(Finite Impulse Response).

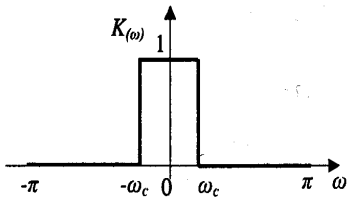
Задачата на проектирането на FIR филтри по метода на прозорците се свежда до определяне на импулсната преходна характеристика по зададени изисквания към коефициента на предаване **КП**. В основата на този метод стоят връзките между коефициента на предаване и импулсната характеристика на цифровия филтър

$$\dot{K}(\omega) = \sum_{n=-\infty}^{\infty} h(n) e^{-j\omega n} \qquad (1.1)$$

$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \dot{K}(\omega) e^{-j\omega n} d\omega \ , \qquad (1.2)$$

където $\dot{K}(\omega)$ е коефициент на предаване на дискретната система, който е периодична функция на честотата с период 2π . Изразът (1.1) представя периодичната функция $\dot{K}(\omega)$ в комплексен ред на Фурие. Коефициентите на този ред представляват дискретите на импулсната преходна характеристика h(n) и се определят на база на израза (1.2), който представлява обратно Фурие преобразуване (**ОФП**). На практика **ОФП** може да се използва за проектиране на нерекурсивни цифрови филтри по предварително зададен коефициент на предаване. Най-общо казано този метод ни дава импулсната преходна характеристика (във времевата област) на база на желания коефициент на предаване (честотна област).

За онагледяване на дадения метод ще бъде разгледано проектирането на нискочестотен филтър (**НЧФ**) със срязваща честота ω_c (фиг1.1). Нека изискванията към $\dot{K}(\omega)$ са:



$$\dot{K}(\omega) = \{1, \ |\omega| \le \omega_c; \ 0, \quad \omega_c < |\omega| < \pi \ \} \qquad (1.3)$$

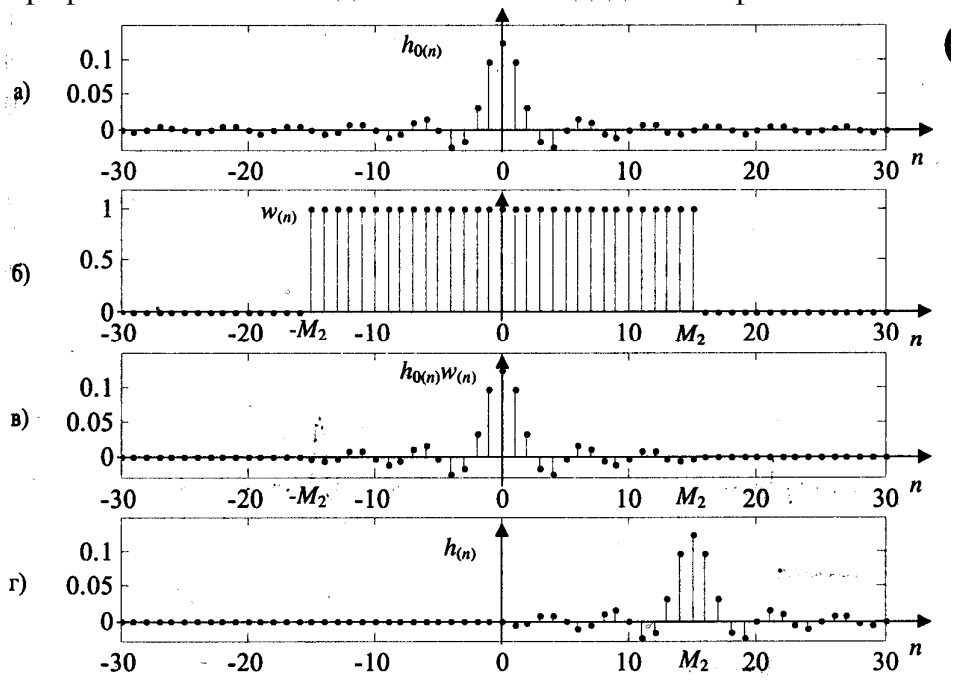
фиг 1.1

Този запис предполага, че филтърът не внася закъснение в сигнала т.е $\varphi(\omega)=0$. На база на израза (1.2) се изчислява ИХ $h_0(n)$ на филтъра. Индексът "0" показва, че това е оригиналната ИХ, която ще се коригира.

$$h_0(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}\dot{K}(\omega)e^{-j\omega n}d\omega=\frac{1}{2\pi}\int_{-\omega_c}^{\omega_c}e^{-j\omega n}d\omega=\frac{1}{j2\pi n}e^{j\omega n}\Big|_{-\omega_c}^{\omega_c}=$$

$$=\frac{1}{j2\pi n}\left(e^{j\omega_c n}-e^{-j\omega_c n}\right)=\frac{\sin(\omega_c n)}{\pi n}=\frac{\omega_c}{\pi}\frac{\sin(\omega_c)}{\omega_c n} \qquad (1.4)$$

Графиката на ИХ на идеалния НЧФ е дадена на фиг 1.2 а



фиг.1.2 Преобразуване на ИХ на нерекурсивен филтър.

а) оригинална ИХ

б) тегловна функция прозорец

в) отрязана ИХ получена чрез умножение на оригиналната с прозорец

г) реализуема ИХ

Оригиналната ИХ $h_0(n)$ има два основни недостатъка, които правят невъзможна реализацията на филтъра. Първо, оригиналната ИХ има безкраен брой дискрети и второ - в нея присъстват, дискрети при n<0,

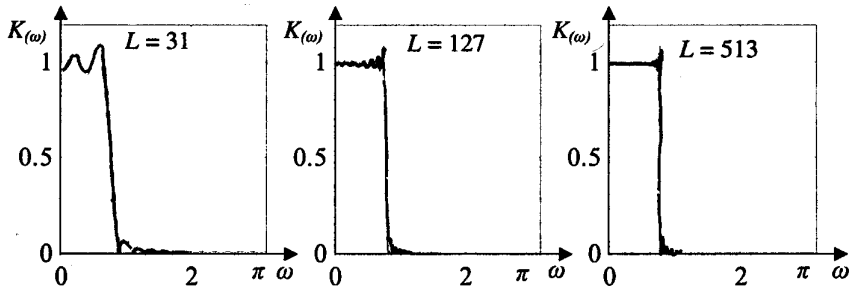
което означава получаване на чисто изпреварване и прави филтъра нереализуем (некаузален).
Решение на първия проблем е умножаването на ИХ с прозоречна тегловна функция (правоъгълен прозорец 1.2.б) $w_n = 1 \quad |n| \le M_2$ във времевата област . По този начин ще се получи отсичане на ИХ и получаването на краен брой дискрети различни от 0 (фиг 1.2.в).

Решението на втория проблем е отместването на ИХ надясно с M_2, което внася закъснение в преминаващите през филтъра сигнали, но прави филтъра практически реализуем. Така получената ИХ има дължина от L дискрета, $L = 2M_2 + 1$, като на графиката $M_2 = 15$. Отместването на сигнала във времето не внася изкривявания, но ограничаването на ИХ до краен брой дискрети води до пулсации на честотната характеристика в лентата на пропускане и задържане. Увеличаването на дискретите на ИХ не води до подтискане на пулсациите в честотната лента, а просто на изместването им към преходната област на амплитудно-честотната характеристика АЧХ. При увеличаване на дължината на прозореца L→∞ честотата на пулсациите се увеличава много и те се преобразуват в безкрайно тесен отскок наречен ефект на Гибс.

Тези изкривявания могат да се определят като се използва свойството на преобразуването на Фурие, че на умножаване на числови редици във времевата област съответства конволюция на техните Фурие преобразувания. Затова на произведението $h_0(n)w(n)$ съответства конволюция в честотната област $\dot{K}(\omega) \otimes W(\omega, L)$, където $W(\omega, L)$ е Фурие преобразуването на използвания прозорец. За коефициента на предаване на филтъра с отсечена ИХ се плучава:

$$\dot{K}(\omega, \Omega) = \dot{K}(\omega) \otimes W(\omega, L) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \dot{K}(\Omega) W(\omega - \Omega, L)\, d\Omega \qquad \textbf{(1.5)}$$

Вид на АЧХ в зависимост от дължината L на ИХ е даден на (фиг 1.3)



фиг 1.3 Влияние на дължината на ИХ L върху АЧХ при правоъгълен прозорец

Максималните пулсации на АЧХ в лентата на задържане и пропускане зависят от срязващата честота ω_c и от дължината на ИХ.

Когато изискванията по отношение на пулсациите на АЧХ в лентата на пропускане и към затихването в лентата на задържане не са високи, използването на правоъгълен прозорец е допустимо. Дължината на прозореца се избира така, че да се осигури достатъчно стръмна преходна област на АЧХ.

Обикновено изискванията към характеристиките на филтрите са по - строги и се налага модифициране на ИХ чрез умножаването и с подходяща по форма и дължина прозоречна функция с форма различна от правоъгълната. Преобразуването на Фурие на прозоречната функция трябва да има малки странични листи и тесен централен лист. Тесният централен лист води до тясна преходна област на АЧХ , а малките странични листи определят малки пулсации на АЧХ в лентата на пропускане и силно затихване в лентата на задържане. Изискванията за тесен централен лист и малки странични на Фурие преобразуването на прозоречната функция са взаимно противоречиви, затова се търси компромисно решение чрез използване на подходящ за конкретната задача прозорец.Поради големия брой прозорци за проектиране на нерекурсивни цифрови филтри, тук ще бъдат дадени само два от най-широко използваните:

- **Прозорец на Хеминг**

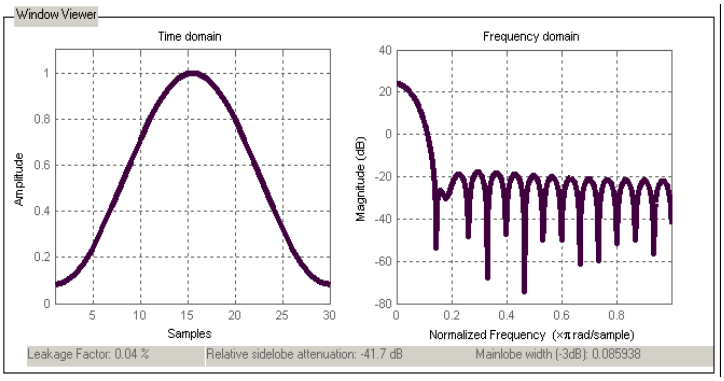
  Прозорецът на Хеминг е от групата на повдигнатите косинусоиди и се определя с израза:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{L-1}\right) = 0.54 - 0.46\left(\frac{\pi n}{M_2}\right) \ , n = 0,..,L-1 \qquad (1.6)$$

Импулсната характеристика на НЧФ проектиран с прозорец на Хеминг е:

$$h(n) = \left\{0.54 - 0.46\left(\frac{\pi n}{M_2}\right)\right\}\frac{\omega_c}{\pi}\frac{\sin[\omega_c(n-M_2)]}{\omega_c(n-M_2)}, n = 0,....., L-1 \qquad (1.7)$$

Прозорецът на Хеминг във времевата и честотната област при L=31 е показан на фиг.1.4



Фиг.1.4 Прозорец на Хеминг при дължина L=31

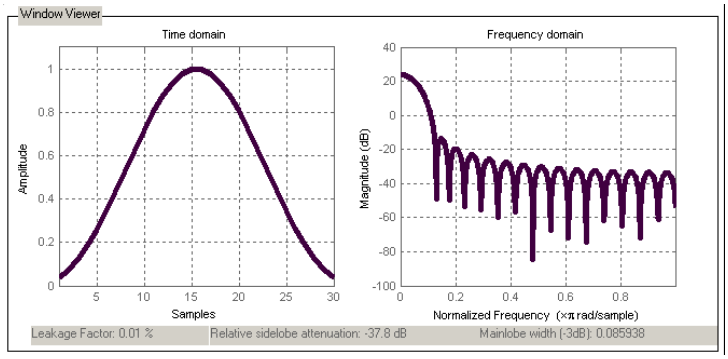
- **Прозорец на Кайзер**

Този прозорец е изведен при оптимизация по критерий за минимум на ширината на централния лист и при ограничаване на енергията на страничните листи и зададена дължина на прозореца. Изразът за прозоречна функция предложен от Кайзер (1.8) използва модифицираната Беселова функция от първи род нулев ред $I_{0(x)}$ (1.9) . Видът на прозореца зависи изключително много от стойността на коефициента β. При β=0 прозорецът на Кайзер съвпада с правоъгълния. При увеличаване на β страничните листи на образа му в честотната област намаляват, но се увеличава и ширината на централния лист.

$$w(n) = \frac{1}{I_{0(\beta)}} \left[\beta \sqrt{1 - \left(\frac{n - M_2}{M_2} \right)} \right] , n = 0,.., L-1 \qquad (1.8)$$

$$I_0(x) = 1 + \sum_{k=1}^{\infty} \left(\frac{x^k}{2^k \, k!} \right)^2 \qquad (1.9)$$

Прозорецът на Кайзер във времевата и честотната област при дължина на прозореца L=31 и β=5 е показан на фиг.1.4.

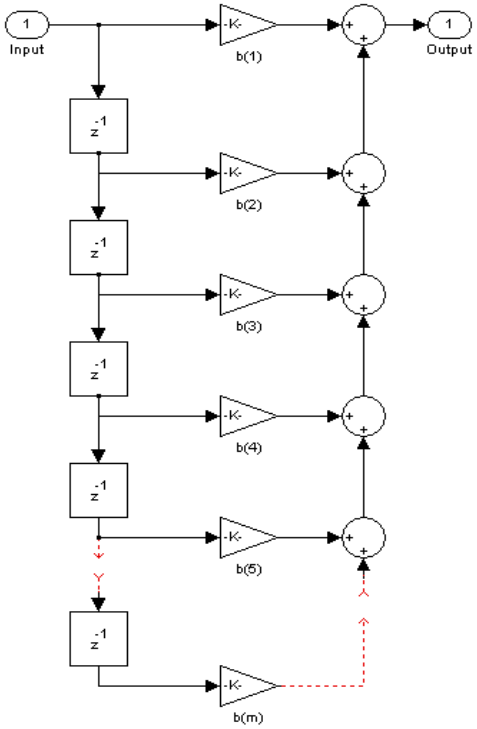


Фиг.1.5 Прозорец на Кайзер при дължина L=31 и β=5

1.2.1 Форми на реализация на FIR филтри

Като цяло структурните схеми на цифровите филтри (формите на реализация) показват начина, по който се обработват дискретите на входния сигнал, запомнянето на междинни резултати, по-предни дискрети на входния , изходния или междинен сигнал. Съществуват различни форми на структурните схеми, които реализират предавателната функция на филтрите. Структурните схеми се изграждат на базата на три основни елемента: клетки памет, умножители и суматори. Структурните схеми могат да се получат на базата на предавателната функция получена при синтеза на филтъра. В зависимост от начина на реализация на предавателната функция на филтъра формите са преки, каскадни, паралелни или решетъчни. Тук ще бъде разгледана пряката форма на реализация.

Предавателната функция на една дискретна система дадена в z^{-1} пространството най-общо може да се запише във вида:



$$W(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{\sum_{i=0}^{M} b_i z^{-i}}{\sum_{j=0}^{N} a_j z^{-j}} = \frac{b_0 + b_1 z^{-1} + ... + b_M z^{-M}}{1 + a_1 z^{-1} + ... + a_N z^{-N}} \quad , N \geq M$$

(1.10)

За нерекурсивни системи, с които се описват FIR филтрите $A(z^{-1}) = 1$, при което за предавателната функция на филтъра се получава:

$$W(z^{-1}) = \frac{B(z^{-1})}{1} = b_0 + b_1 z^{-1} + ... + b_M z^{-M}$$

(1.11)

Изходът на FIR филтъра даден във времевата област на базата на уравнение (1.11) е:

$$y(k)=\sum_{m=0}^{M} b(m)x(k-m)=x(k)b(0)+x(k-1)b(1)+\ldots+x(k-m)b(m)$$

(1.12)

където b, x, y, M са съответно коефициентите на филтъра, входния сигнал, изходния сигнал и реда на филтъра. Структурната схема реализираща тази пряка форма на FIR филтрите е показана на фиг 1.6

Фиг.1.6 Пряка форма на FIR

1.2.2 Предимства и недостатъци на FIR филтрите

- Предимство на **FIR** филтрите е, че имат линейна фазова характеристика **ФХ** в лентата на пропускане, т.е. не внасят изкривявания в сигнала, а само въвеждат закъснение.
- Недостатъците са висок ред на реализация на филтрите, който е свързан с въвеждането на големи времезакъснения. Големият ред от друга страна означава използването на повече памет и увеличаване на изчислителното време.

1.3 Проектиране на рекурсивни цифрови филтри

За разлика от **НРЦФ**, при които за формиране на изходния сигнал се използват само входните дискрети на сигнала, то при рекурсивните цифрови фиктри **РЦФ** се използват и миналите стойности на изходните дискрети[6]. В западната литература съкращението, което се използва за този тип филтри е **IIR** (Infinite Impulse Response). Има съществени разлики при проектирането на **IIR** и **FIR** филтри. За разлика от **FIR** филтрите, които нямат свои аналогови еквиваленти, **IIR** филтрите могат да се проектират на базата на аналогов филтър прототип, чрез дискретизация на предавателната му функция. Друго свойство, което може да се използва при проектирането на **IIR** филтрите е прилагането на така наречената филтрова трансформация. Това е преобразувание, при което от НЧФ прототип, могат да се получат предавателните функции на всички останали видове фитри (ВЧ- високочестотен, ЗФ- заграждащ, ЛФ-лентов).

Предавателната функция с която се описват **IIR** филтрите е дадена в (1.10) по отрицателните степени на z .

$$W\left(z^{-1}\right)=\frac{B\left(z^{-1}\right)}{A\left(z^{-1}\right)}=\frac{\sum_{i=0}^{M} b_i z^{-i}}{\sum_{j=0}^{N} a_j z^{-j}}=\frac{b_0+b_1 z^{-1}+\ldots..+b_M z^{-M}}{1+a_1 z^{-1}+\ldots.+a_N z^{-N}} \quad ,N \geq M \qquad (1.10)$$

Диференчното уравнение даващо изхода на дискретната система във времевата област е:

$$y(k)=\sum_{i=0}^{M} b_i x(k-i) - \sum_{j=0}^{N} a_j y(k-j) = b_0 x_0 + b_1 x_1 + ... + b_M x_{k-M} - a_1 y_1 - ..... - a_N y_{k-N} \quad (1.13)$$

1.3.1 Дискретизация на аналогови филтри

За да получим предавателната функция на дискретния филтър трябва да дискретизираме предавателна функция на аналогов филтър, получена с апроксимация по някой от познатите методи[6]. Някои от основните методи за апроксимация са Бътъруърт, Чебишев I и II , Бесел. Тук ще бъдат разгледани накратко първите два от тях:

- **Апроксимация по метода на Бътъруърт**

Апроксимацията по този метод се характеризира с максимално гладка АЧХ в лентата на пропускане[11,1].

Функцията на Бътъруърт има вида:

$$F(\omega)=\left|W_{butt}(j\omega)\right|^2 = A_{butt}(\omega)^2 = \frac{1}{1+(\omega/\omega_c)^{2n}} \quad , \quad (1.14)$$

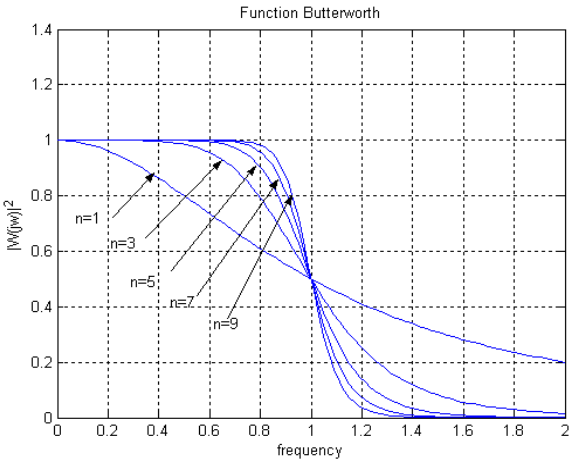
където n е реда на функцията, а ω_c е срязващата честота на филтъра.

Независимо от реда на филтъра е в сила равенството (1.15) (фиг.1.7)

$$F(\omega)=\left|W_{butt}(j\omega_c)\right|^2 = 0.5 , \quad (1.15)$$

но редът на филтъра определя стръмността на спусане на АЧХ след срязващата честота.

Предавателната функция на филтър апроксимиран по метода на Бътъруърт може да се представи във вида:



фиг1.7 $\left|W_{butt}(j\omega)\right|^2$ **във функция от честотата**

$$W_{butt}(p)=\frac{\kappa}{(p-p_1)(p-p_2).....(p-p_n)} \quad (1.16)$$

където $p_1, p_2, ..., p_n$ са полюсите на предавателната функция

- **Апроксимация по метода на Чебишев** I тип

Филтрите базирани на тази апроксимация използват полиномите на Чебишев:

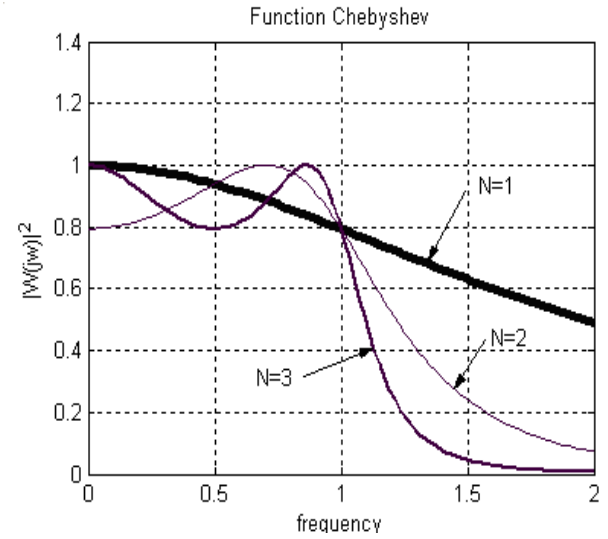
$$T_{N(x)} = \begin{cases} \cos\left(N \arccos(x)\right), & при \quad |x| \le 1, \\ ch\left(N\, arch(x)\right), & при \quad |x| > 1 \end{cases} \qquad (1.17)$$

При $|x| \le 1$ се получават пулсации с единична амплитуда, а при $|x| > 1$ монотонно нарастване.

Филтърът се конструира на основата на полинома на Чебишев, като квадратът на АЧХ на филтър от N ред се дефинира с израза(1.18):

$$\left|W_{butt}(j\omega)\right|^2 = A_{butt}(\omega)^2 = \frac{1}{1+\varepsilon^2 T^2_{N(\omega/\omega_c)}} \qquad (1.18)$$

Тази функционална връзка трансформира поведението на $T^2_{N(x)}$ в честотната характеристика по следния начин: при изменение на честотата в интервала $0 \le \omega \le \omega_c$ $T^2_{N(\omega/\omega_c)}$ пулсира в диапазона [0,1] , а знаменателят има пулсации от 1 до $1+\varepsilon^2$. Съответно квадратът на АЧХ пулсира в обхвата от 1 до $\frac{1}{1+\varepsilon^2}$. Амплитудата на пулсациите зависи от стойността на ε^2, а честотата им от реда на филтъра N.



Квадрата на модула на АЧХ във функция от честотата е даден на фиг.1.8. Предавателната функция на филтъра на Чебишев може да се представи във вида.

$$W_{Cheb}(p) = \frac{\kappa}{(p-p_1)(p-p_2).....(p-p_n)} \qquad (1.19)$$

фиг1.8 $\left|W_{Cheb}(j\omega)\right|^2$ **във функция от честотата**

Като цяло филтрите апроксимирани по метода на Чебишев се характеризират с голям наклон на среза в преходната област и силно затихване на сигналите в лентата на задържане.

Съществуват различни методи за дискретизация на аналогови предавателни функции[5].

- **Ойлерова трансформация**

Използването на тази трансформация дава възможност за определяне на дискретната предавателна функция по следната две формули:

$$W(z) = W(p)\Big|_{p=\frac{(z-1)}{T_0}} \qquad \text{(права разлика)} \qquad (1.20)$$

$$W(z) = W(p)\Big|_{p=\frac{(z-1)}{T_0 z}} \qquad \text{(обратна разлика)} \qquad (1.21)$$

където T_0 е времето на дискретизация.

- **Билинейна трансформация**

Получаването на дискретната предавателна функция при Билинейната трансформация се основава на субституцията $p = \frac{1}{T_0}\frac{z-1}{z+1}$

$$W(z) = W(p)\Big|_{p=\frac{1}{T_0}\frac{z-1}{z+1}} \qquad (1.2.2)$$

1.3.2 **Форми на реализация на IIR филтри**

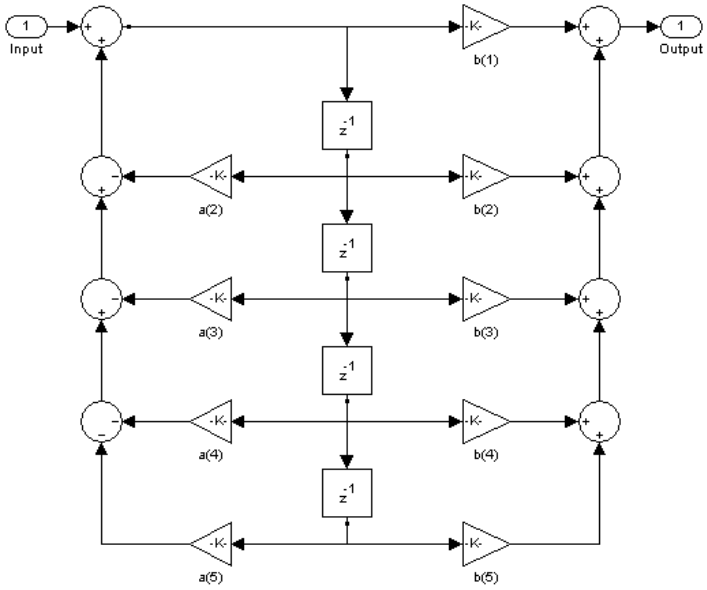
Формата на реализация на цифровите филтри зависи от начина на представяне на дискретната предавателна функция.

- **Пряка форма на реализация на IIR филтри**

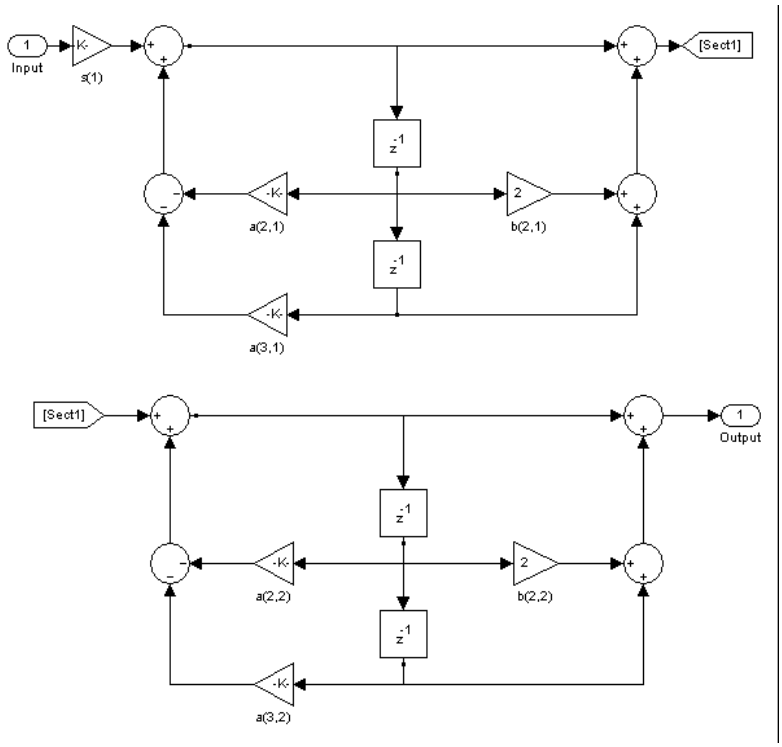
Тази форма се базира на записа на предавателната функция даден в уравнение (1.13). Графично пряката форма на реализация на IIR филтър от четвърти ред е показана на фиг.1.9

- **Реализация на IIR филтри със звена от втори ред**

Реализацията се базира на разбиването на предавателната функция на цифровия филтър (1.13) на каскадно свързани звена от първи и втори ред. Графично тази форма за филтър от четвърти ред е показана на фиг.1.10



фиг.1.9 Пряка форма на реализация на IIR



фиг.1.10 Реализация на IIR филтри с каскадно свързване на звена от втори ред

1.3.3 **Предимства и недостатъци на цифровите IIR филтри**

- **Предимства**

В сравнение с **FIR** филтрите, **IIR** имат съществени предимства, като по-нисък ред и произтичащите от това бързодействие, малко времезакъснение на изходния сигнал и по-малко използвана памет.

- **Недостатъци**

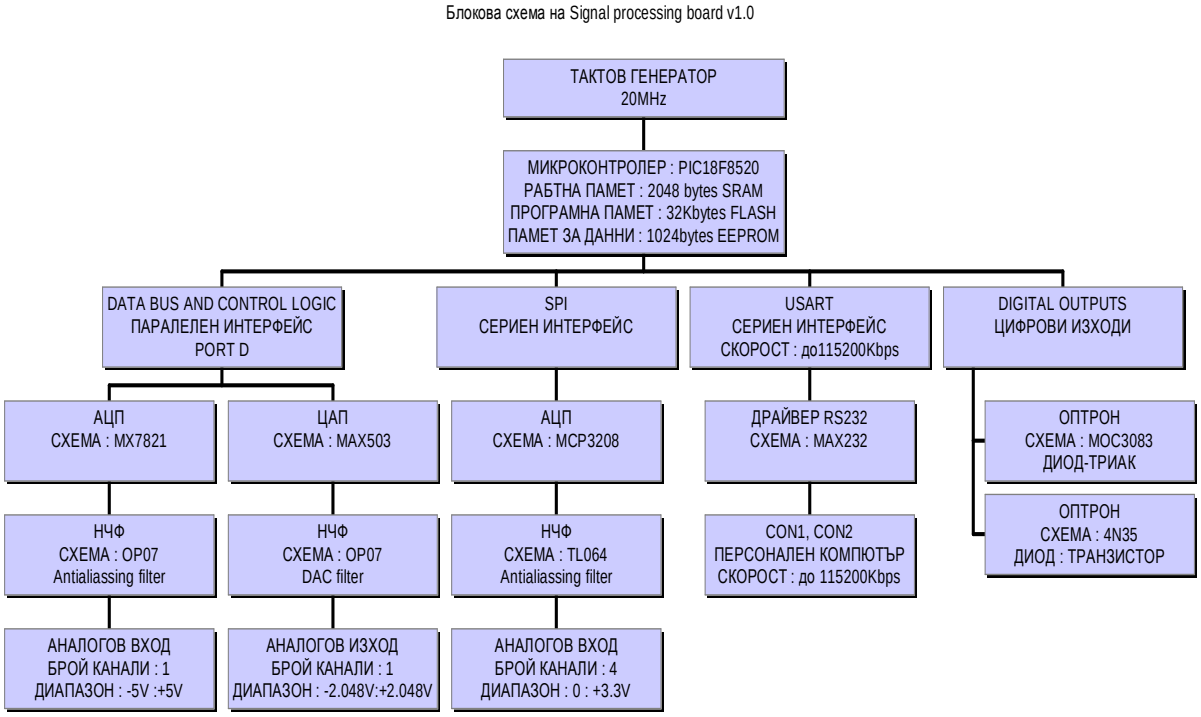
Единственият недостатък на **IIR** филтрите е нелинейната им фазово-честотна характеристика в лентата им на пропускане, което внася изкривявания в преминаващите през филтъра сигнали.

2. Проектиране на хардуера

2.1 Проектиране на електронни схеми и печатни платки с програмния пакет ORCAD 10.0.
Етапи на проектирането на Signal processing board v1.0

Проектираната хардуерна система, има за цел да покаже практически как се реализират и свързват помежду си основните модули, необходими за реализирането на една цялостна система за събиране, обработка и синтезиране на сигнали. Практическо приложение системата може да намери както в системи реализиращи цифрова филтрация на сигнали в реално време, така и в реализирането на цифрови контролери (PID,PI,FUZZY,PLC) за целите на управлението. Конкретното приложение, за което ще бъде използвана зависи единствено от програмата, която ще бъде заложена в микроконтролера.

Първоначалният етап от разработката е да се направи блокова схема на системата (фиг.2.1), на база на която ще бъде съставена електронната схема на устройството.



фиг.2.1 Блок схема на Signal processing board v1.0

Ядрото на системата е програмируем микроконтролер, който управлява свързаните към него периферни устройства:

- едноканален АЦП с паралелна комуникация
- четириканален АЦП със серийна комуникация
- едноканален ЦАП с паралелна комуникация
- драйверна схема за серийна комуникация с персонален компютър
- оптрони за получаване на галванично разделени цифрови изходи

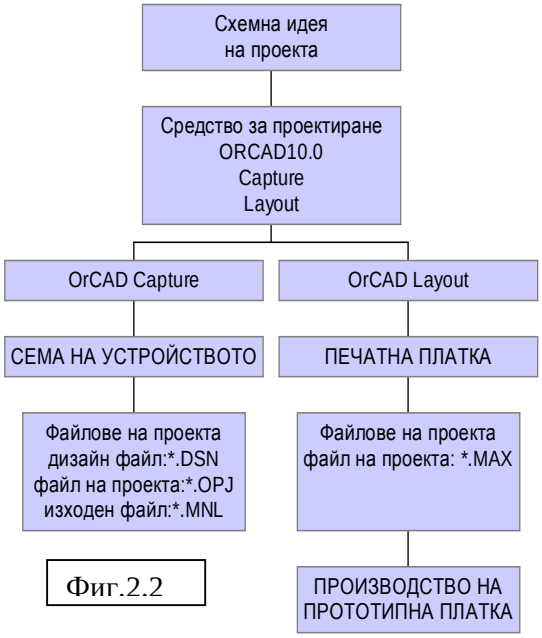
Към входовете на **АЦП** и изходите на **ЦАП** са свързани **НЧФ**,необходими за аналоговата филтрация на входящите и изходящи сигнали.

Схемата има и захранващ блок, който осигурява стабилизирани напрежения, необходими за работата на цифровите и аналоговите интегрални схеми.

Проектирането на електронната схема и печатната платка на устройството **Signal processing board v1.0** е извършена в средата на **ORCAD 10.0** [12]. Принципната електрическа схема, печатната платка на системата и снимка на системата са дадени в **Приложение 1, 2 и 3.**

ORCAD 10.0 е специализиран софтуер, предназначен за проектиране на електронни схеми и печатни платки и е може би най-популярния от този клас продукти.

ЕТАПИ ОТ ПРОЕКТИРАНЕТО НА ХАРДУЕРА НА SIGNAL PROCESSING BOARD V1.0



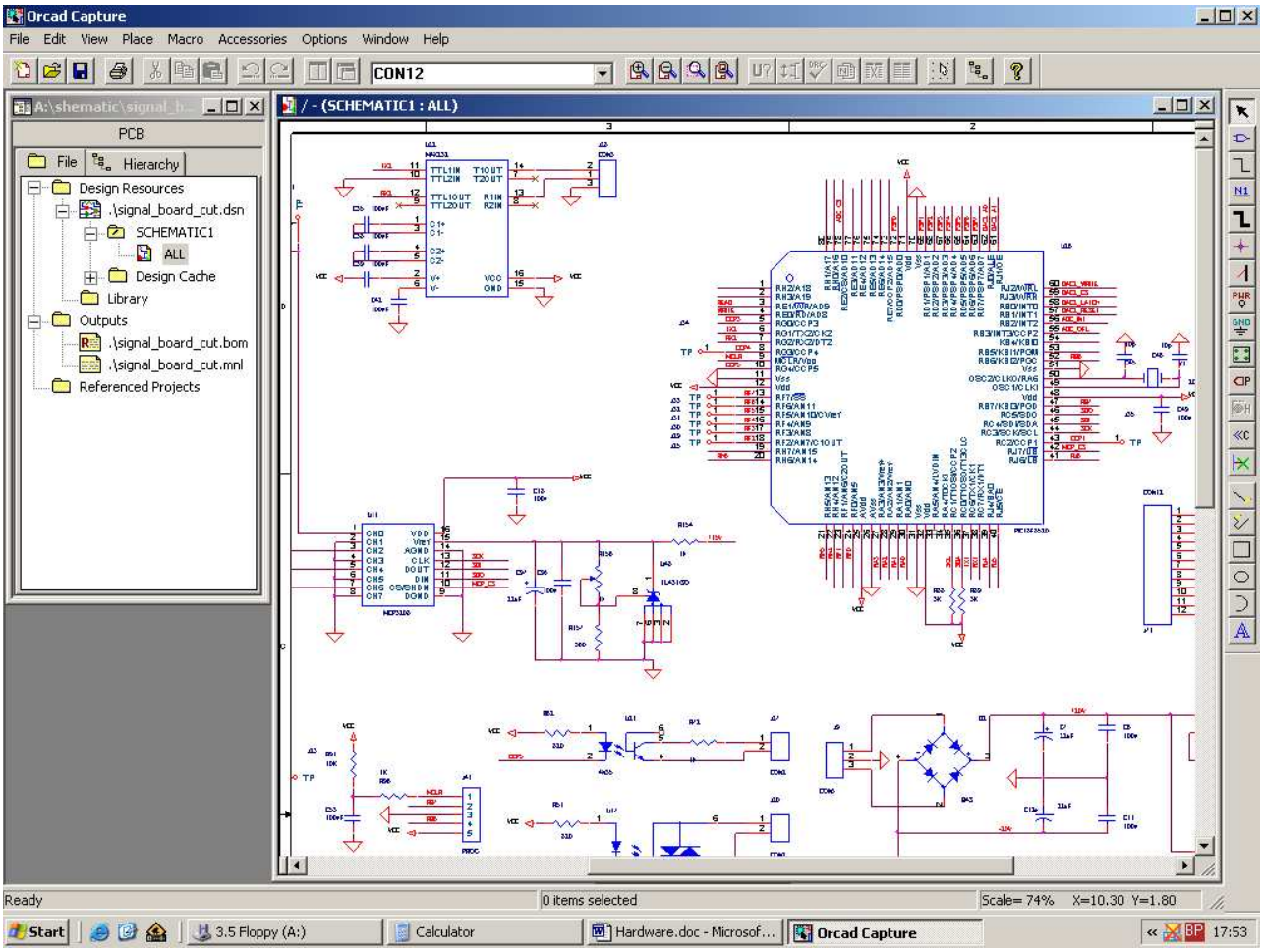
Фиг.2.2

Проектирането в среда на **ORCAD** е основен етап от автоматизираното проектиране на електронни схеми. Софтуерът, управляващ съвременните автоматизирани производства на печатни платки в повечето случаи е съвместим с файловете, генерирани при разработките на проектите в **ORCAD.**

На блоковата схема на фиг.2.2 са показани етапите през които се преминава при проектирането на електронната схема и печатната платка на **Signal processing board v1.0**.

За схемната идея и поставените цели се съставя блок-схема на устройството(фиг2.1), като по този начин се прави декомпозиция на задачата. След това за всеки модул поотделно се съставят няколко възможни решения и се избира оптималното по отношение на цена, качество, габарити на елементите и др.

Общото описание и действието на програмите **Capture** и **Layout ,** в които на практика се реализират схемата и печатната платка на проектираното устройство са представени накратко както следва:

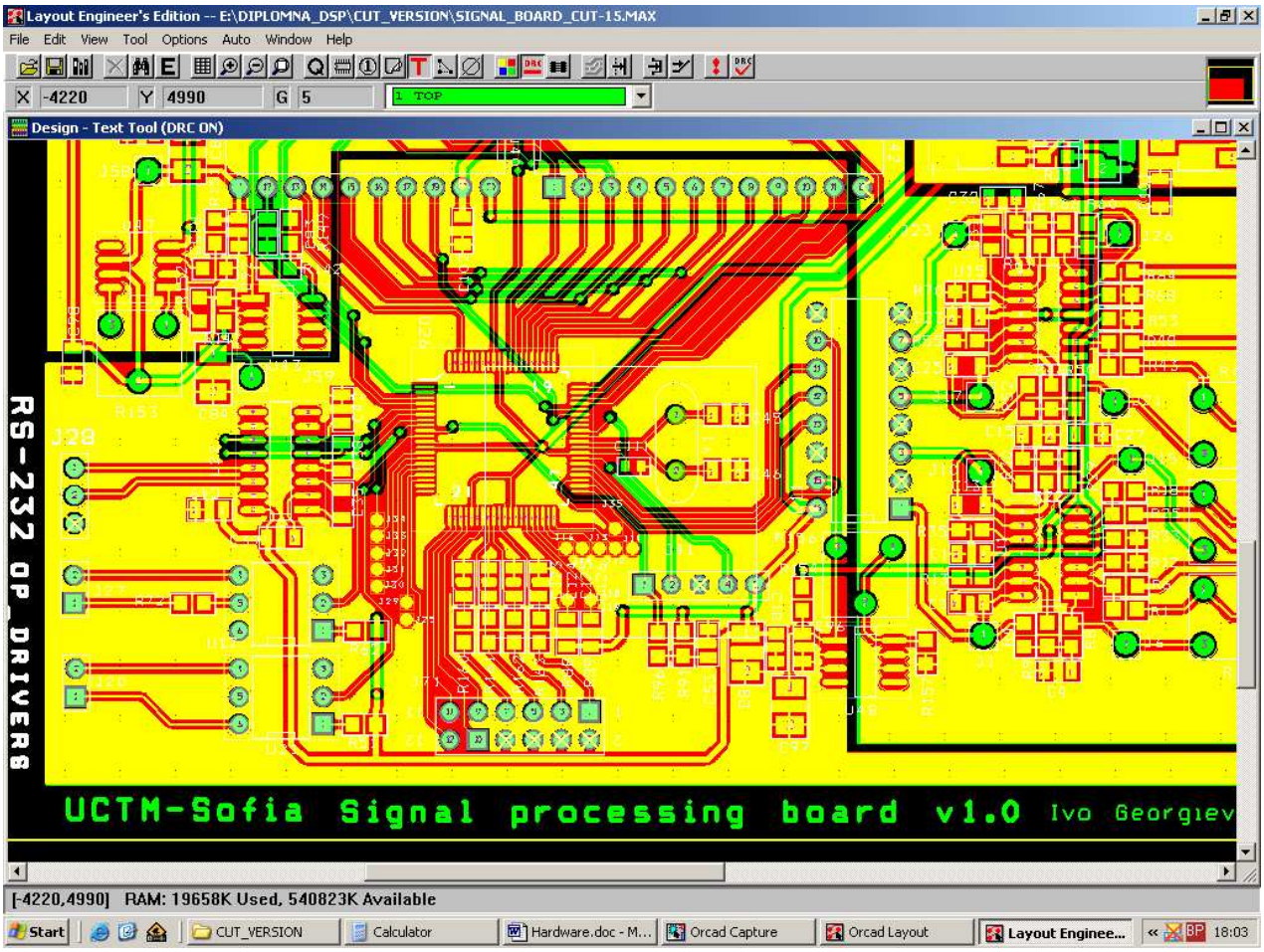


фиг.2.3 Графична среда на програмата OrCAD Capture

- **ORCAD Capture CIS** представлява графична среда (фиг.2.3) за създаване, редактиране и съхраняване на електронни схеми[8]. Позволява бърз достъп до всички страници, елементи, файлове и други средства посредством вградения системен мениджър на проекта(вляво на фигурата). В базата данни на програмата има голям брой библиотеки на електронни компоненти, като на потребителя е осигурен бърз достъп до всеки от тях. Освен това потребителят има възможност за директен достъп до данни на елементи от производителя през **Internet**, както и възможността да създава свои собствени такива.

  На база на проектираната и начертана в **Capture CIS** електронна схема, се генерира така наречения **netlist** файл(с разширение **.MNL),** който дава връзките между компонентите в схемата, необходими за създаването на файла(***.MAX**), който се използва в програмата **ORCAD Layout** при създаването на печатната платка .

  Програмата има възможност да генерира и други файлове: **BOM**(bill of materials) и др.



фиг.2.4 Графична среда на програмата за проектиране на печатни платки ORCAD Layout

- **ORCAD Layout** е продукт за проектиране на печатни платки. За входен файл той приема netlist файла(*.mnl) , генериран на база на създадената вече електрическа схема. На базата на този файл, **Layout** създава файл с разширение (*.MAX), в който се извършва проектирането на печатната платка.

  В началния етап от проектирането в прозореца на програмата са разпръснати компонентите (в избраните корпуси) на схемата и връзките между тях. Целта е компонентите да бъдат подредени и връзките между тях да бъдат опроводени. Програмата има възможност за автоматично подреждане и опроводяване, но резултатите не винаги са задоволителни. По тази причина подреждането и опроводяването на платката **Signal processing board v1.0** е направено ръчно. Когато опроводяването на всички компоненти е завършено, се прави проверка за грешки (от позициониране, опроводяване и др.). Ако няма такива, изходният файл(*.MAX) може вече да се използва за производството на платката. Производството на платката на база на MAX файла е напълно автоматизирано и в България има фирми, които предлагат тази услуга.

Конкретно изпълнението на поръчката за изработка на двуслойна платка за **Signal processing board v1.0** на база на проектираната в средата на **ORCAD10.0** електронна схема и печатна платка е възложено на фирма **“МИКРОН”.**

2.2 Избор на микроконтролер

При проектирането на хардуерната система е отделено голямо внимание при избора на микроконтролер, който да изпълнява изискванията по отношение на бързина, памет и възможности за комуникация с периферни устройства[9,2,7].

За конкретното приложение и целите на дипломната работа е избран микроконтролер **PIC18F8520,** производство на фирмата **Microchip**. Този микроконтролер притежава следните по-важни характеристики:

- 8 битов RISC микроконтролер
- програмна памет от тип Flash, която позволява многократно записване на информация 100 000 цикъла триене/запис.Това го прави изключително подходящ при проектиране на вградени системи и развойна дейност
- памет за данни EEPROM 1 000 000 цикъла триене/запис
- тактова честота на централния процесор до 40Mhz
- четири хардуерни таймера
  - Таймер 0 – 8/16 битов таймер/брояч
  - Таймер 1 – 16 битов таймер/брояч
  - Таймер 2 – 8 битов таймер/броячТаймер 3 - 16 битов таймер/брояч
  - Таймер 4 - 8 битов таймер/брояч
- пет модула за широчинно-импулсна модулация CCP/PWM
- синхронен сериен порт (Master Synchronous Serial Port MSSP) с два режима на работа
  - трипроводен SPI режим
  - двупроводен I2C режим
- универсален адресируем синхронен асинхронен приемо-предавател Universal Synchronous Asynchronous Receiver Transmitter (USART)
  - поддържа RS-232 и RS-485
- 10 битов , 16 канален аналогово-цифров преобразувател
- четири външни извода за прекъсвания
- 68 входно/изходни извода

Общите параметри на фамилията микроконтролери **18FXX20** са дадени в таб.2.1

Device	Program Memory		Data Memory		I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8-bit/16-bit	Ext Bus	Max Fosc (MHz)
	Bytes	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI	Master I^2C				
PIC18F6520	32K	16384	2048	1024	52	12	5	Y	Y	2	2/3	N	40
PIC18F6620	64K	32768	3840	1024	52	12	5	Y	Y	2	2/3	N	25
PIC18F6720	128K	65536	3840	1024	52	12	5	Y	Y	2	2/3	N	25
PIC18F8520	32K	16384	2048	1024	68	16	5	Y	Y	2	2/3	Y	40
PIC18F8620	64K	32768	3840	1024	68	16	5	Y	Y	2	2/3	Y	25
PIC18F8720	128K	65536	3840	1024	68	16	5	Y	Y	2	2/3	Y	25

таб.2.1 Параметри на серията микроконтролери 18FXX20

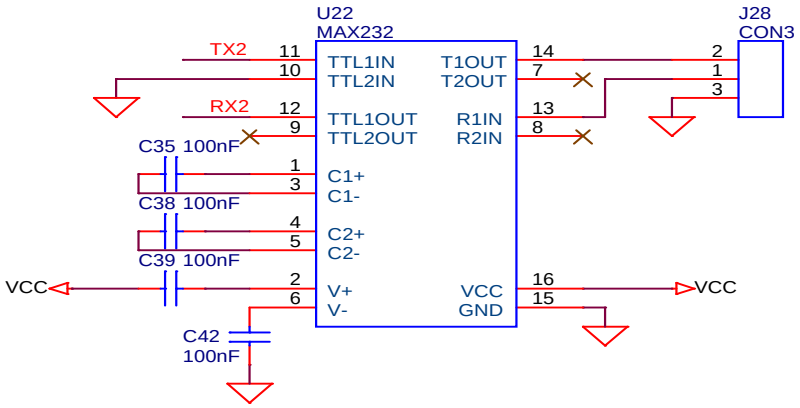
2.3 Комуникация на микроконтролера с персонален компютър по RS-232 интерфейс

За комуникацията на платката **Signal processing board v1.0** с персонален компютър се използва RS-232 интерфейс, който е поддържан от микроконтролера PIC18F8520. Той разполага с два USART модула, които са локализирани съответно:

- USART 1 на PORT C (RC6/TX1/CK1 и RC7/RX1/DT1)
- USART 2 на PORT G (RG1/TX2/CK2 и RG2/RX2/DT2)

За да могат да комуникират контролера и персоналния компютър е необходимо да се свърже драйверна схема, която да преобразува TTL нивата, с които работи микроконтролера в нива +13/-13V, необходими за прехвърлянето на данните . Има много производители на такива драйверни схеми, но в случая е избрана интегралната схема MAX232 на фирмата Maxim Dalas[10].

От съображения за разположението на входовете и изходите на печатната платка **Signal processing board v1.0** се избира драйверната схема MAX232 да се свърже към USART 2 на микроконтролера посредством връзките (**TX2,RX2**). Свързването на драйвера е показано на фиг.2.5.



фиг.2.5 Свързване на драйверната интегрална схема MAX232

2.4 Избор на аналогово-цифрови преобразуватели

При проектирането на хардуерната система особено внимание е отделено на избора на АЦП. Разучени са както различните по принцип на действие АЦП, така и типа на комуникацията , чрез която обменят данни с микроконтролерите и микропроцесорите.

Комуникацията между микроконтролера и АЦП се извършва по два типа интерфейс - сериен и паралелен.

2.4.1 Избор на АЦП със серийна комуникация

При този тип комуникация данните се предават серийно(бит по бит). Командите за управление и данните от АЦП са конфигурирани в пакети, вида на които зависи от конкретния протокол за комуникация. Скоростта за трансфер на данните зависи от тактовата честота на осцилатора.

Двата режима за серийна комуникация между интегрални схеми, които поддържа PIC18F8520 са:

- SPI интерфейс за обмен на данни (сериен периферен интерфейс)

Това е трипроводен интерфейс за комуникация като названието на трите линиии е съответно :

-SDI - Serial Data In
-SDO - Serial Data Out
-SCK - Serial Clock

Този интерфейс се характеризира с голяма скорост на обмен на данни, зависеща от тактовата честота и възможност за едновременно приемане и предаване на данни.

- I2C интерфейс за обмен на данни

I2C е двупроводен интерфейс за комуникация между интегрални схеми.

-SDA - Serial Data
-SCL - Serial Clock

Характерното при **I2C** е, че всяко устройство (интегрална схема) има свой хардуерен седем битов адрес. Това позволява към двете линии (**SDA, SCL**) да се свържат 2^7 броя интегрални схеми. Микроконтролера се обръща към всяка от тях с нейния собствен адрес. Като бързина **I2C** интерфейса отстъпва на **SPI**. Основните режими по отношение на скорост за I2C са:

- нормален – 100 kHz (100kbit/s)
- бърз – 400 kHz (400kbit/s)

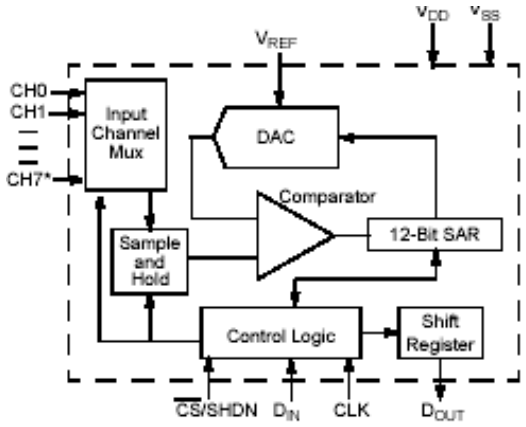
По отношение на приоритет в режим **I2C** контролерът може работи в следните режими:

- Master (главен)
- Slave (подчинен)

За **Signal processing board v1.0** е избрана интегрална схема **АЦП MCP3208** със серийна комуникация, производство на фирмата **Microchip[9]**.

MCP3208 е 12 битов, 8 канален **АЦП** със SPI интерфейс. Този **АЦП** може да измерва(преобразува) напрежения на всеки от входовете си, както спрямо маса (single-ended), така и между двойки входове(диференциално свързване). Различните конфигурации се определят софтуерно посредством подаването на команди по SPI интерфейса. Протоколите за управление и комуникация могат да се видят в документацията на интегралната схема.

Скоростта на дискретизация за този **АЦП MCP3208** може да достигне до 100 ksps при захранващо напрежение VDD = 5V и 50 ksps при VDD = 2.7V. Блоковата схема на **АЦП** е дадена на фиг.2.6



фиг.2.6 Схема на свързване на MCP3208 и опорното му напрежение

MCP3208 се свързва към MSSP на микроконтролера посредством три проводника необходими за SPI интерфейса :

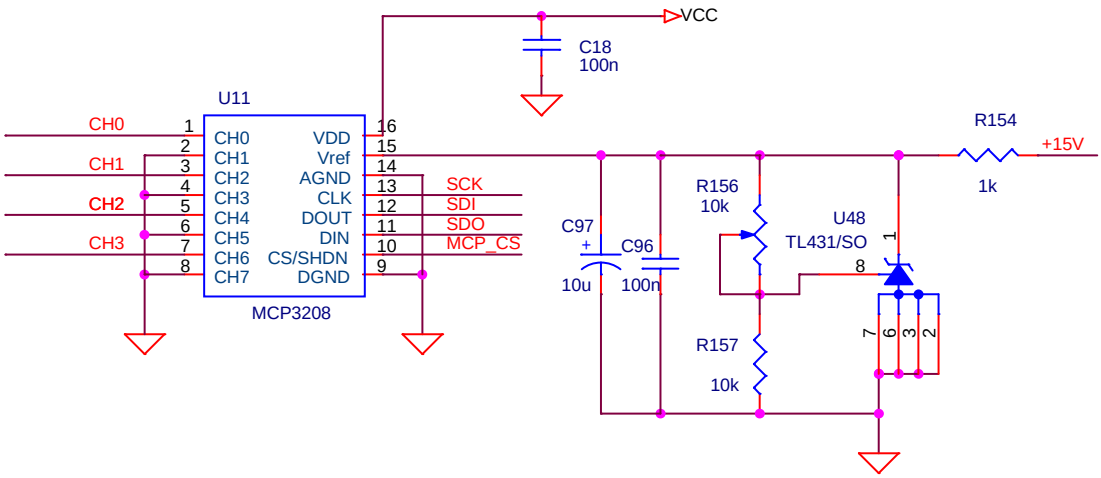
- SDO - PORT C(RC5/SDO)
- SDI - PORT C(RC4/SDI/SDA)
- SCK - PORT C(RC3/SCK/SCL)

и един за активиране на АЦП:

- MCP_CS–PORT J(RJ7/UB)

Аналоговите входове на схемата (CH0,CH1,CH2,CH3) се свързват към четирите аналогови изхода на ФПФ(филтри за предварителна филтрация), които ще бъдат разгледани отделно в т.2.6.

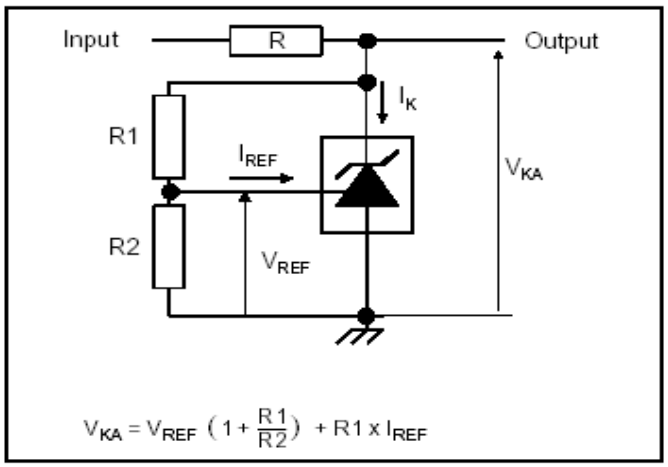
При хардуерната реализация използвана в Signal processing board v1.0 MCP3208 е проектиран да работи като четириканален АЦП с диференциални входове и външно опорно напрежение. Схемата на свързване е дадена на фиг.2.7.



фиг.2.7 Свързване на MCP3208 в схемата

Свързването на изводи (2,4,6,8) към маса означава, че независимо от софтуерната конфигурация на **АЦП** (спрямо маса или диференциално), той ще измерва напреженията постъпващи на входовете му (1,3,5,7) спрямо маса.

За надеждно и точно преобразуване **АЦП** изискват стабилни източници на опорно напреженеие, които да са нечувствителни към промяна на околната температура. За тази цел е избрана интегралната схема **TL431 (**фиг.2.8). Тя се отличава с добра стабилност по отношение на промяна на околната температурата и промяна на входното напрежение, а също така с широк диапазон на регулиране на изходно напрежение (до 36 V).



фиг.2.8 Схема на свързване на опорния източник TL431

Опорното напрежение се изчислява по формула 2.1:

$$V_{out} = V_{ref}(1 + \frac{R_1}{R_2}) + R_1 * I_{ref} \qquad (2.1),$$

където Vref = 2.5V.

Произведението $R_1 * I_{ref}$ може да се пренебрегне поради малката стойност на тока Iref.

Ако изберем R2 = 10kΩ и R1(тример) в диапазона 0-10 kΩ за максималната и минималната стойност на опорното напрежение ще получим:

$$V_{out\max} = V_{ref}(1 + \frac{10000}{10000}) = 5V$$

$$V_{out\min} = V_{ref}(1 + \frac{0}{10000}) = V_{ref} = 2.5V$$

По този начин се получава регулируемо опорно напрежение в диапазона 2.5 - 5V. За нашия случай стойностите на елементите са:

R156 = 10k; R157 = 10k;

2.4.2 Избор на АЦП с паралелна комуникация

При този вид интерфейс данните се предават паралелно. В зависимост от разрядността на управляващия контролер (8,16 битов), шината за данни може да бъде с различна големина (4,8,12,16 битова). Например при използване на 8 битов микроконтролер и шина и 16 битов АЦП може да прочетем информацията на два пъти по 8 бита. Обикновено освен шината за паралелно предаване на данни чиповете имат и изводи за управляваща логика(CS-Chip select; RD-Read mode; WR- Write mode и др.).

АЦП работещи със паралелен интерфейс позволяват многократно увеличаване на скоростта на предаване (приемане) на данни. Това ги прави по-широко използвани и предпочитани при разработката на системи за цифровата обработка на сигнали, изискващи по-голяма бързина. Предимствата на паралелния пред серийния интерфейс са най-ясно изразени при дискретизацията на високочестотни сигнали (GSM).

Недостатък при използване на паралелен интерфейс е значително повишения брой на използваните изводи на микроконтролера. Съпоставката между използваните изводи при паралелна и серийна комуникация показва 12,15 извода за паралелна при 3 за серийна (SPI).

За АЦП с паралелен интерфейс е избрана интегралната схема **MX782**1 на фирмата Maxim Dalas[10].

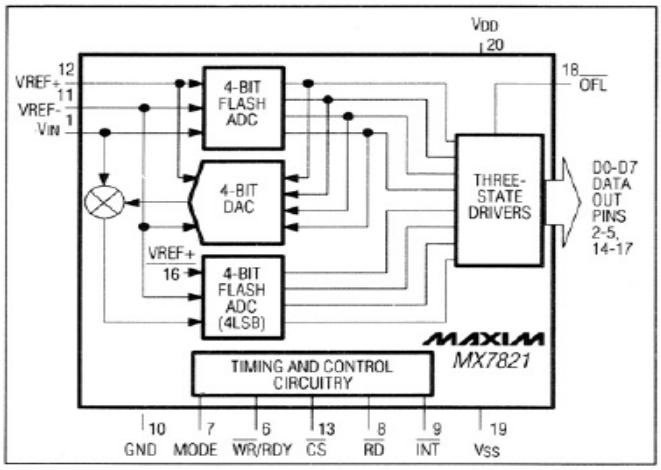
MX7821 е висококоростен 8 битов АЦП със следните по-важни характеристики:

- 8 битова резолюция
- съвместим с контролери 8 битов паралелен интерфейс
- 660ns – време за преобразуване
- еднополярно +5V или двуполярно ±5V захранване
- преобразуване на биполярни/еднополярни входни сигнали
- честотна лента на входните сигнали – 100kHz
- честота на дискретизация до 1 Msps
- два режима на работа на АЦП по отношение на комуникацията с контролера и управлението му

Някои от приложенията на интегралната схема MX7821, дадени от фирмата производител са:

- цифрова обработка на сигнали
- телекомуникации
- висококоростно серво-управление
- аудио-системи

Блоковата схема на АЦП е дадена на фиг.2.9:



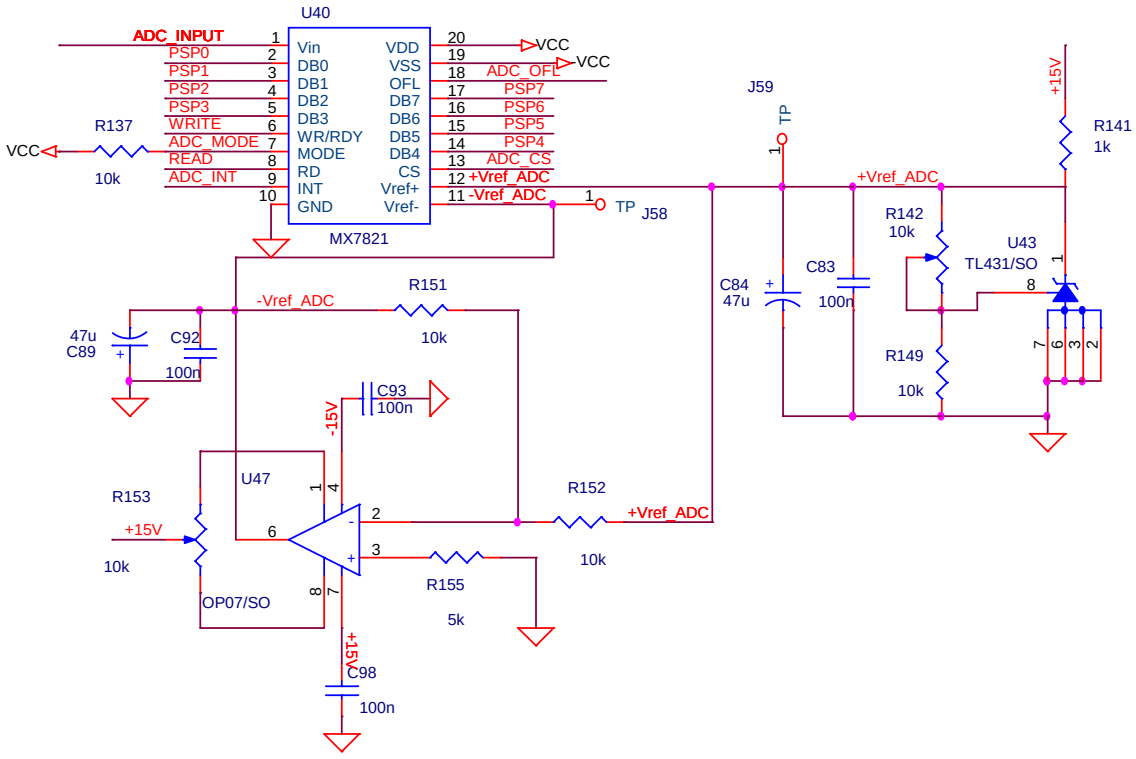
фиг.2.9 Блокова схема на АЦП MX7821

На блок схемата дадена на фиг.2.9 се виждат входовете и изходите на АЦП. Описание на изводите на MX7821 е дадено в табл.2.2. За повече информация относно **MX7821 (maxim dalas)**

табл.2.2

Изводи(pins)	название	Определение (функция)
1	Vin	Аналогов вход: диапазон : VREF(–) <= VIN <= VREF(+)
2	DB0	Цифров изход за данни(LSB)
3-5	DB1-DB3	Цифрови изходи за данни
6	WR/RDY	WRITE- вход за контрол или READY-статус на преобразуването(в зависимост от състоянието на входа MODE)
7	MODE	Състоянието на този вход определя режима на работа на АЦП(WR-RD или RD)т.е. софтуерното управление от страна на контролера. RD режим, (MODE=0) WR-RD режим, (MODE=1)
8	RD	READ- вход за стартиране на операция четене
9	INT	INTERRUPT изход .Генерира прекъсване към контролера, когато преобразуването е приключило
10	GND	GROUND
11	Vref(-)	Отрицателно опорно напрежение
12	Vref(+)	Положително опорно напрежение
13	CS	CHIP SELECT активира АЦП
14-16	DB4-DB6	Цифрови изходи за данни
17	DB7	Цифров изход за данни(MSB)
18	OVF	OVERFLOW . Цифров изход отчитащ препълването на АЦП
19	Vss	Отрицателно захранващо напрежение Vss=0V ;при еднополярно захранване Vss=-5V; при двуполярно захранване
20	Vdd	Положително захранващо напрежение Vdd= 5V

Хардуерната схема на свързване на MX7821 в **Signal processing board v1.0** е дадена на фиг.2.10.



фиг.2.10 Схема на свързване на MX7821 в Signal processing board v1.0

Аналоговият вход на MX7821 се свързва посредством връзката (ADC_INPUT) към изхода на прилежащия му филтър за предварителна филтрация на сигнала. Изчислението на филтъра е дадено в т.2.6.

Изчисляването на схемата за изработване на положително опорно напрежение с интегрална схема TL431 е аналогично с това при изчислението на опорния източник на MCP3208. Стойностите на компонентите за получаване на напрежения в интервала 2.5-5V са следните:

R142 – 10kΩ (тример)

R149 - 10kΩ

Кондензаторите (C83,C84) включени в изхода на схемата са филтрови и имат за цел да подтискат високочестотни смущения.

За получаване на отрицателно опорно напрежение, необходимо за работата на АЦП с двуполярни входни сигнали се използва полученото вече положително опорно напрежение. Това напрежение се подава на входа на инвертиращ операционен усилвател U47(OP07) с коефициент на усилване k= -1, вследствие на което на изхода му се получава напрежение еднакво по стойност, но с противоположен знак на входното. Напрежението получено на изхода на операционния усилвател се използва за отрицателно опорно напрежение необходимо на схемата MX7821.

Коефициента на усилване по напрежение на U47 при стойности на стъпротивленията R151= 10kΩ ; R152=10kΩ е :

$$k = -\frac{R151}{R152} = -\frac{10000}{10000} = -1$$

Тример-потенциометъра R153 се използва за компенсация на несиметрията в операционния усилвател. Компенсацията на несиметрията е необходима в случай, че на входа на операционния са подадени 0V(замасен вход), изхода е отличен от 0. При това положение изхода на ОУ се установява в нула с помощта на R153. Избрания ОУ тип OP07 има специални изводи за включване на нулиращ тример.

Напрежението постъпващо на входа на АЦП ще се преобразува в 8 битово число. Напрежението, на което ще отговаря 1бит зависи от източниците на опорно напрежение Vref(-) и Vref(+).

Например, ако Vref(-) = 0V, а Vref(+) = 3.3V за 8 битов АЦП ще получим, че на 1 бит отговаря напрежение:

$$U_{1bit} = \frac{V_{ref}(+)}{2^N - 1} = \frac{3.3}{2^8 - 1} = \frac{3.3}{255} \approx 0.01294V / bit \quad (2.2)$$

Свързването на АЦП MX7821 към микроконтролера се осъществява със следните връзки (NETS):

табл.2.3

MX7821	NETS	PIC18F8520
Изводи за данни		
DB0	PSP0	RD0/PSP0/AD0
DB1	PSP1	RD1/PSP1/AD1
DB2	PSP2	RD2/PSP2/AD2
DB3	PSP3	RD3/PSP3/AD3
DB4	PSP4	RD4/PSP4/AD4
DB5	PSP5	RD5/PSP5/AD5
DB6	PSP6	RD6/PSP6/AD6
DB7	PSP7	RD7/PSP7/AD7
Изводи за контрол И статус		
WR/RDY	WRITE	RE0/RD/AD8
MODE	ADC_MODE	Хардуерно с Pull-up резистор
RD	READ	RE1/WR/AD9
INT	ADC_INT	RB2/INT2
OFL	ADC_OFL	RB3/INT3/CCP2
CS	ADC_CS	RE2/CS/AD10

За шина за данни е избран да се ползва PORT D на микроконтролера.

Изводите даващи информация за статуса на преобразуването

(INT и OFL) се свързват към входовете на микроконтролера, които могат да генерират прекъсване при промяна на състоянието им. Първото прекъсване(INT) има за цел контролера да разбере кога е завършило преобразуването на АЦП. Така ще е известно кога точно да бъде прочетен резултата . Второто прекъсване(OFL) може да извести контролера, че подаденото входно напрежение е по-голямо от обхвата на АЦП.

Подробно управляващия софтуер на MX7821 ще бъде разгледан в софтуерната част на дипломната работа и главно в програмата реализираща цифровите филтри.

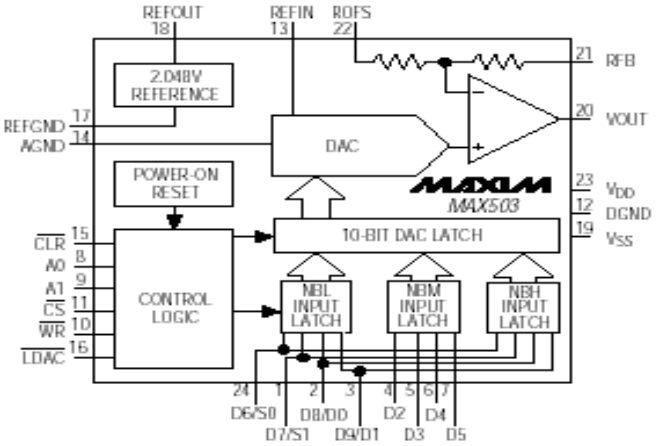
2.5 Избор на цифрово-аналогов преобразувател

Цифрово-аналоговите преобразуватели (ЦАП) имат за цел да преобразуват цифровите сигнали в пропорционални на тях електрически.

Избора на **ЦАП** за **Signal processing board v1.0 e** спрян на интегрална схема с паралелна комуникация (MAX503), която да бъде свързана към шината за данни използвана от АЦП(MX7821)[10]. По този начин към шината за данни (PORT D на контролера) ще се свързват две устройства, които обаче не трябва никога да са активни по едно и също време. Активацията на устройсвата ще се управлява софтуерно, използвайки изводите (Chip select) на всяко от устройствата. Някои от по-важните характеристики на ЦАП MAX503 са :

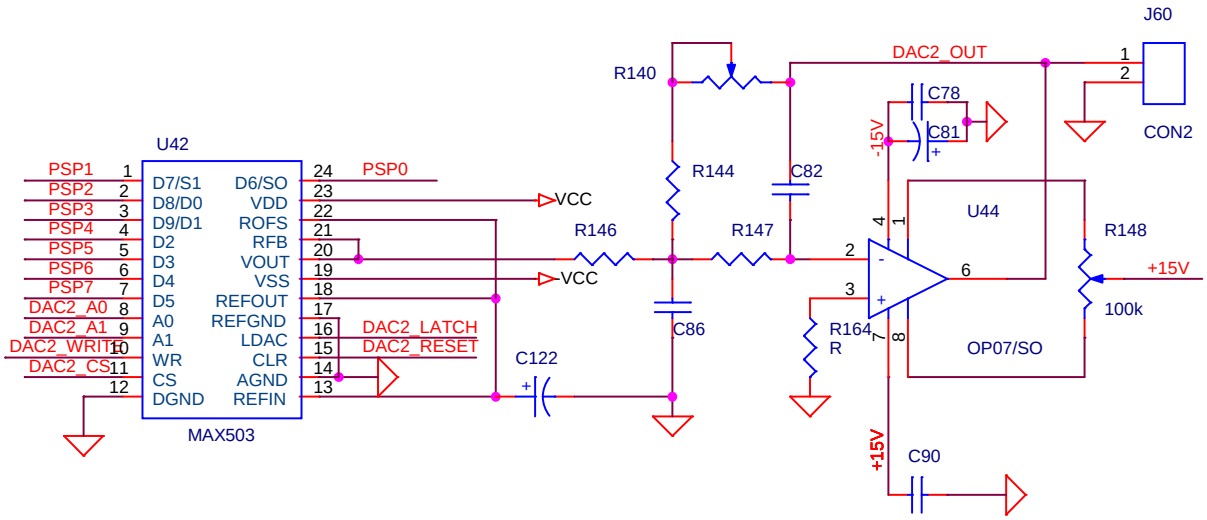
- буфериран напрежителен изход
- работа с еднополярно +5V и двуполярно ±5V напрежение
- използва 4 или 8 битов паралелен интерфейс
- вътрешен опорен източник на напрежение 2.048V
- ниска консумация 250μA
- време за установяване на изхода (settling time) 25μs
- грешка от ±1/2LSB за целия температурния диапазон от 0 до 70 градуса по целзий
- възможност за реализация на 4 квадрантно умножение без включването на външни елементи

Блоковата схема на преобразувателя е дадена на фиг.2.11:



фиг2.11 Блок схема на MAX503

Хардуерната реализация на ЦАП и изходния филтър е дадена на фиг2.12.



фиг.2.12 Свързване на MAX503и изходния филтър в Signal processing board

Връзките между ЦАП и микроконтролера са дадени в табл.2.4:

табл.2.4

MAX503	NETS	PIC18F8520
Изводи за данни		
D6/SO	PSP0	RD0/PSP0/AD0
D7/S1	PSP1	RD1/PSP1/AD1
D8/D0	PSP2	RD2/PSP2/AD2
D9/D1	PSP3	RD3/PSP3/AD3
D2	PSP4	RD4/PSP4/AD4
D3	PSP5	RD5/PSP5/AD5
D4	PSP6	RD6/PSP6/AD6
D5	PSP7	RD7/PSP7/AD7

Изводи за контрол		
A0	DAC2_A0	RJ0/ALE
A1	DAC2_A1	RJ1/OE
WR	DAC2_WRITE	RJ2/WRL
CS	DAC2_CS	RJ3/WRH
CLR	DAC2_RESET	RB1/INT1
LDAC	DAC2_LATCH	RBO/INT0

2.6 Изчисляване на филтрите за предварителна филтрация на сигналите

Филтрите за предварителна филтрация(ФПФ) на сигналите са аналогови (активни или пасивни) нискочестотни филтри , имащи за цел да отрежат честотната лента на входния сигнал при честота равна на половината от честотата на дискретизация на АЦП:

$$f_{cutoff} = \frac{f_s}{2},$$

където fs е честотата на дискретизация на АЦП.

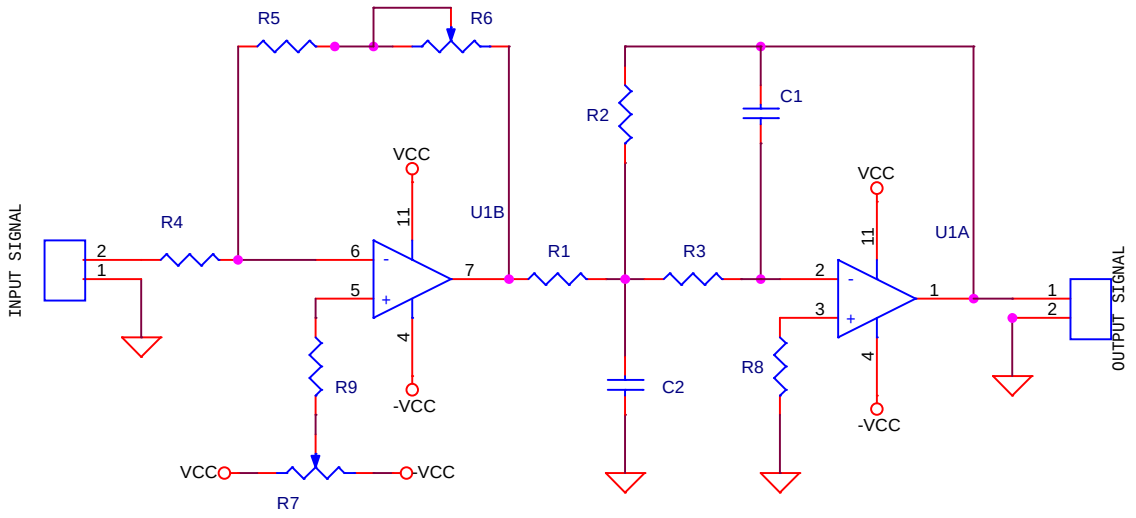
Отрязването на сигнали с честота f>fs/2 се прави с цел **АЦП** да не преобразува сигналите с честоти, които след това не могата да бъдат възстановени . Противното би довело до появата на "**псевдо-сигнали**", които на практика не съществуват в спектъра на сигнала.

Хардуерната система **Signal processing board v1.0** има два аналогово-цифрови преобразувателя, които имат общо пет аналогови входа. Четири от тях са за U11(MCP3208) и един за U40(MX7821). Това означава, че трябва да се проектират общо 5 филтъра. За филтрация на сигнала от изхода на **ЦАП** също трябва да се проектира един филтър със същата структора(**НЧФ**).

ФПФ, включени в системата са проектирани да работят като нискочестотни активни филтри с отрицателна обратна връзка (**ООВ**) от типа **MFB**(Multiple feedback). Към тях са включени и инвертиращи **ОУ** с регулируем коефициент на усилване, в случай че е необходимо предварително усилване на сигнала.

На фиг.2.13 е дадена електрическата схема за един канал (усилвател и филтър) изпълнен по схемата **MFB**. Схемата може да реализира практически предавателни функции на НЧФ от втори ред получени чрез апроксимация по познатите методи (Чебишев, Бътъруърт, Бесел). В дадената схема входния ОУ, реализира инвертиращ усилвател с коефициент на усилване:

$$k_1 = -\frac{R5 + R6}{R4}$$



фиг.2.13 Принципна схема ФПФ с буферен усилвател

Ако се изберат стойности за съпротивленията R5=10kΩ ; R6(тример) =50kΩ и R4=10kΩ за минималния и максималния коефициент на усилване ще получим:

$$k_1(\min) = -\frac{10000+0}{10000} = -1$$

$$k_1(\max) = -\frac{10000+50000}{10000} = -6$$

Конкретните стойности съответстващи на (R4,R5, R6) в схемата са:

- на R4 съответстват R6=R25=R48=R64=R123=10kΩ
- на R5 съответстват R1=R18=R43=R60=R118= 10KΩ
- на R6 съответстват R2=R21=R45=R131=50kΩ

Вторият ОУ и прилежащите му компоненти (R1,R2,R3,R8,C1,C2) реализират активен НЧФ с предавателна функция:

$$W_{MFB}(p) = \frac{\frac{R2}{R1}}{C1*C2*R2*R3*p^2 + C1(R2+R3+\frac{R2*R3}{R1})*p+1} \quad (2.3)$$

За изчертаване на честотните и фазовите характеристики на филтъра е написана малка програма в среда на MATLAB. За входни параметри тя приема стойностите на компонентите, участващи в предавателната функция (2.3), а връща полюсите и Боде-диаграмата на филтъра.

Програмата за изчертаване е:

```
function  [sys,p]=mfb(r1,r2,r3,c1,c2)
%MFB FILTER RESPONSE
%[sys,p] = mfb(r1,r2,r3,c1,c2)
%sys- filter transfer function
%p - poles on transfer function
%r1,r2,r3 -ohm
%c1,c2 - farads
sys= tf(r2/r1,[c1*c2*r2*r3   c1*(r2+r3+((r2*r3)/r1))  1 ]);
p=pole(sys);
bode(sys);grid
```

Компонента R8 не участва в предавателната функция на филтъра и се приема за R8= 5kΩ.

За изчисляването на стойностите на елементите (R1,R2,R3,C1,C2), се използва софтуер на фирмата **BURR-BROWN**®. Програмата работи под **DOS** и като входни параметри приема желаната срязваща честота, ред на филтъра, тип на схемата на филтъра(**MFB**) , тип на апроксимацията, и точност на компонентите. Програмата връща стойностите на компонентите на филтъра. Освен това може да изчертава АЧХ и ФЧХ на получения филтър.

В табл.2.5 са дадени стойностите на компонентите (R1,R2,R3,C1,C2) за петте канала, при зададена срязваща честота и действителна номерация на компонентите в електрическата схема на **Signal processing board v1.0**

табл.2.5

	fcut=fs/2 (kHz)	R1	R2	R3	R8	C1	C2
Канал 0 на U40-MX7821	fcut1 = 2	R136 3.6k	R130= 3.6k	R143= 15.4k	R139 5k	C95= 3.3nF	C85= 22nF
Канал 0 на U11-MCP3208	fcut2 = 2	R8= 3.6k	R4= 3.6k	R9= 15.4k	R13 5k	C3= 3.3nF	C4= 22nF
Канал 1 на U11-MCP3208	fcut3 = 2	R29= 3.6k	R22= 3.6k	R30= 15.4k	R35 5k	C13= 3.3nF	C15= 22nF
Канал 2 на U11-MCP3208	fcut4 = 1	R50= 9.1k	R46= 9.1k	R52= 11k	R55 5k	C25= 6.8nF	C27= 22nF
Канал 3 на U11-MCP3208	fcut5 = 1	R66= 9.1k	R61= 9.1k	R67= 11k	R70 5k	C31= 6.8nF	C32= 22nF
Канал ЦАП MAX503	fcut = 2	R146 3.6k	R144 3.6k	R147 15.4k	R164 5k	C82 3.3nF	C86 22nF

За типа на апроксимацията се задава Бесел поради факта,че филтрите проектирани по този метод имат най-линейната **ФЧХ** в сравнение с другите апроксимации (Чебишев и Бътъруърт). Следователно сигналите подавани на входа на тези филтри се предават на изхода им максимално точно, изместени единствено по фаза. Това означава, че филтрите апроксимирани по метода на Бесел внасят най-малки изкривявания в сигнала.

За филтрите, които ще работят с **АЦП MCP3208** е удобно да се използват интегрални схеми, които съдържат четири **ОУ** в един корпус. В случая е избрана интегралната схема **TL064**/SO, като за реализирането на четирите аналогови канала се използват две интегрални схеми (за усилвателите и филтрите).

За филтъра и усилвателя, който ще работи с АЦП MX7821 са избрани операционни усилватели тип OP07.

Елементите включени във филтъра(U44), свързан към изхода на ЦАП MAX503 също са дадени в табл.2.5

2.7 Захранващ блок на Signal processing board v1.0

Изискванията за работата на АЦП MX7821 при положение, че ще се обработват двуполярни входни напрежения, както и използваните интегрални схеми(ОУ ЦАП), налагат да се проектира захранващ модул, който да изработва следните по големина стабилизирани напрежения:

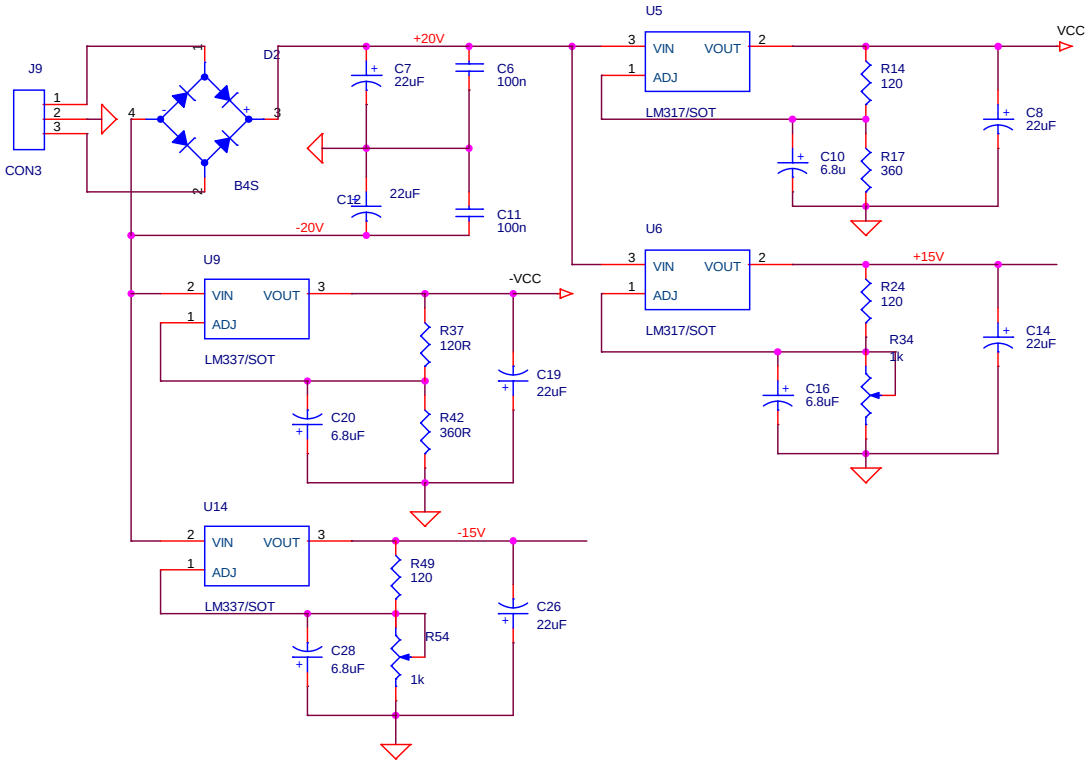
- ±5V
- ±12V

Напрежението +5V се използва за захранването на всички цифрови схеми на платката. На фиг.2.14 е показан използвания в системата захранващ модул.

За реализиране на схемата са използвани линейни стабилизатори за положително (LM1117) и отрицателно(LM337IMP) напрежение.
Положителните захранващи напрежения за стабилизатора LM1117 се изчисляват както следва:

$$V_{out}(+5V) = V_{ref}(1 + \frac{R_{17}}{R_{14}}) = 1.25(1 + \frac{360}{120}) = 5V$$

$$V_{out}(+12V) = V_{ref}(1 + \frac{R_{34}}{R_{24}}) = 1.25(1 + \frac{1000}{120}) = 11.67V$$



фиг.2.14 Захранващ блок на системата Signal processing board v1.0

Аналогично се изчисляват и отрицателните захранващи напрежения за LM337IMP:

$$V_{out}(-5V) = -V_{ref}(1+\frac{R_{42}}{R_{27}}) = -1.25(1+\frac{360}{120}) = -5V$$

$$V_{out}(-12V) = -V_{ref}(1+\frac{R_{54}}{R_{49}}) = -1.25(1+\frac{1000}{120}) = -11.67V$$

3. Проектиране на софтуера

3.1 Проектиране на FIR филтри в среда на MATLAB и практическото им реализиране върху системата Signal processing board v1.0

В тази глава на дипломната работа е изложена последователността при проектирането на **FIR** филтри и практическата им реализация чрез интегрирането им във вградени микроконтролерни системи, работещи в реално време.

Примерите и решенията, разгледани в тази глава, дават възможност да бъде разработено и реализирано практическо упражнение по дисциплината "**Цифрова обработка на сигнали**". Упражнението може да включва следните етапи:

- Проектиране на филтри в среда на Matlab
- Запис на коефициентите на филтъра в контролера
- Снемане на АЧХ на цифровия филтър с помощта на тестови сигнали.
- Дискретизация и филтрация на аналогови сигнали

В частта "Проектиране на цифрови **FIR** филтри" ще се разгледат и използват възможностите на **Matlab** и в частност графичната програмата **Filter Design & Analysis Tool**, която се намира в пакета **Filter Design toolbox[4].**

Във втория етап изчислените с **Matlab** **FIR** филтри ще бъдат реализирани върху хардуерната програмируема система **Signal processing board v1.0**, която е специално проектирана за целите на дипломната работа и е разгледана подробно в глава (2) . Подробно ще бъде описана и програмата реализираща **FIR** алгоритъма, както и управлението на хардуерните модули.

Ще се разгледат и използваните за контролера **PIC18F8520** компилатори и програматори..

В третия етап ще бъдат снети АЧХ на цифровите филтри и ще се разгледат някои от сигналите, които се използват за целта.

В четвъртия етап се изследва как цифровите филтри подтискат сигналите с честоти извън лентата им на пропускане.

3.2 Проектиране на FIR филтри в среда на Filter Design & Analysis Tool

Програмата има разнообразни възможности, които улесняватбързо проектиране на цифрови **FIR** и **IIR** филтри. Някои от тях са:

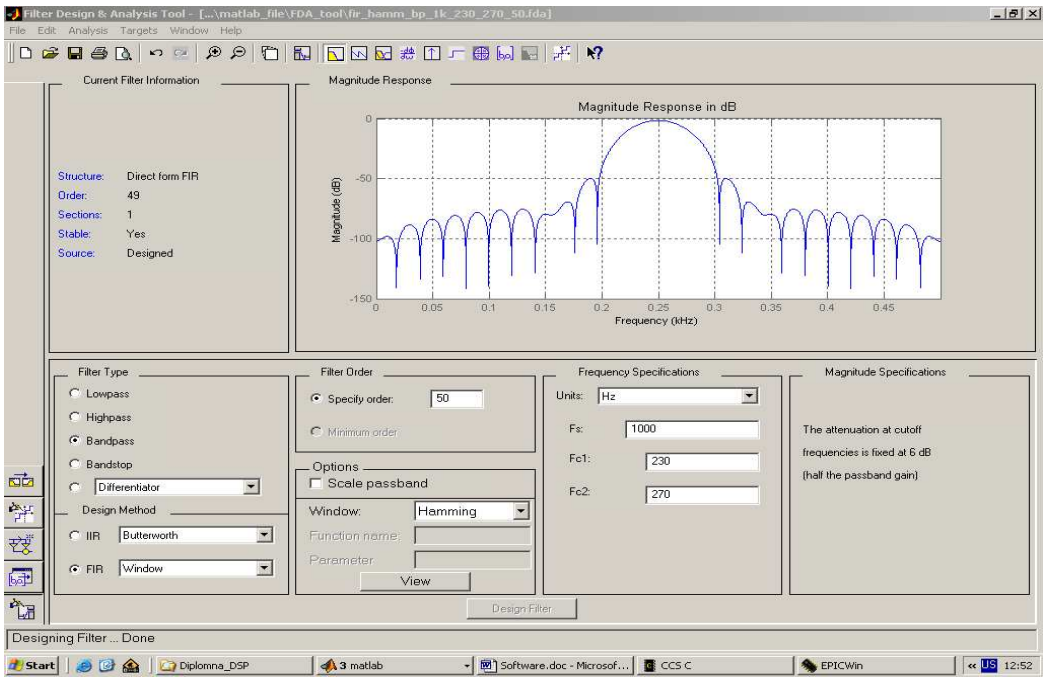
- конвертиране на структурата на цифровите филтри от един тип в друг (директна форма, звена от втори ред и др).
- филтрова трансформация на **НЧФ** прототип във всички останали типове филтри

- НЧФ → ВЧФ
- НЧФ → ЛФ
- НЧФ → ЗФ

3.2.1 Проектиране на лентов FIR филтър с прозорец на Hamming

Графичният прозорец на **Filter Design & Analysis Tool,** в който ще се извърши проектирането е показан на фиг(3.1). С отметки се отбелязват желаните характеристики на филтъра[4]:

- типа на филтъра **LPF,HPF,BPF,BSF** в случая **Band Pass Filter**
- тип на метода за проектиране на филтъра **FIR**
- тип на прозореца - **Hamming**
- ред на филтъра **50**
- честота на дискретизация **fs = 1000Hz**
- долна срязваща честота **fc1=230Hz**
- горна срязваща честота **fc2= 270Hz**



фиг3.1Графичен прозорец на Filter Design & Analysis Tool

За да може контролерът да работи в реално време е необходимо коефициентите, получени в **MATLAB** да се нормират и преобразуват в цели числа със знак (**8bit** или **16bit signed**). Нормирането на коефициентите може да измени поведението на филтрите чрез известно намаляване на способността им да подтискат сигналите с честоти извън лентата им на пропускане. Нормирането за привеждането на коефициентите във вид на 8 битово число със знак става на два етапа:

- умножават се действителните коефициенти на филтъра поелементно със 128: coef = h*128;
- получените междинни коефициенти се закръгляват към най-близкото цяло число: coef = round(coef);

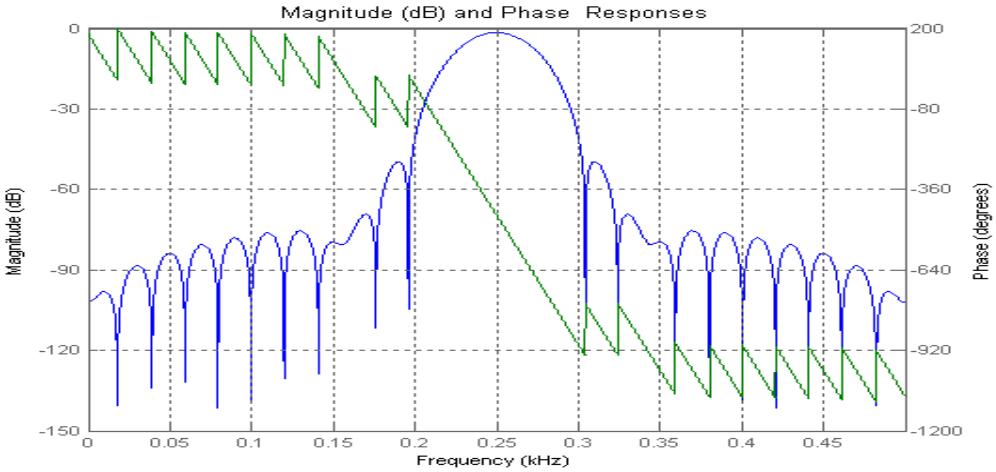
Програмата има възможност за автоматично нормиране на коефициентите на филтъра. Нормираните коефициенти на проектирания лентов филтър са:

```
coef [51] = {
0,   0,   0,   0,   0,   0,   0,  -1,   0,   2,   0,  -3,
0,   4,   0,  -5,   0,   7,   0,  -8,   0,   9,   0, -10,
0,  10,   0, -10,   0,   9,   0,  -8,   0,   7,   0,  -5,
0,   4,   0,  -3,   0,   2,   0,  -1,   0,   0,   0,   0,
0,   0,   0
};
```

Диференчното уравнение на FIR филтъра е:

$$y(k) = \sum_{m=0}^{M} h(m)x(k-m) = x(k)h(0) + x(k-1)h(1) + ... + x(k-m)h(m) \qquad (3.1)$$

На фиг.(3.2) са дадени АЧХ и ФЧХ на проектирания (ненормиран) филтър. Затихването на така проектирания филтър е средно –80dB в лентата на задържане.



фиг3.2 АЧХ и ФЧХ на проектирания (ненормиран) лентов филтър FIR(Hamming) BPF fs=1000Hz; fc1=230Hz; fc2=270Hz

3.2.2 Проектиране на НЧ FIR филтър с прозорец на Hamming

Заданието на филтъра е следното:

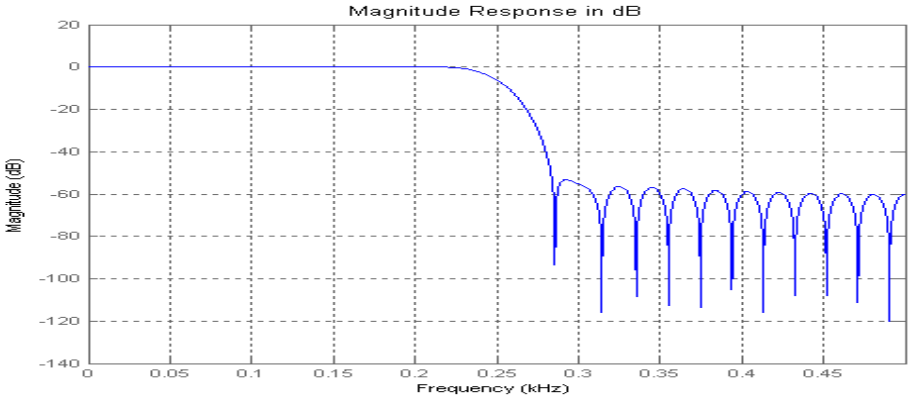
- тип на филтъра НЧФ - **LPF**
- тип на метода за проектиране на филтъра **FIR**
- тип на прозореца - **Hamming**
- ред на филтъра **50**
- честота на дискретизация **fs = 1000Hz**
- срязваща честота **fc=250Hz**

Проектирането се извършва в средата на **Filter Design & Analysis Tool** аналогично на това в предходната точка.

Нормираните коефициенти на НЧФ, с които ще работи микроконтролера са :

```
coef[51] = {
    0,   0,   0,   0,   0,   0,   0,   0,   1,   0,  -1,   0,
    2,   0,  -2,   0,   3,   0,  -5,   0,   7,   0, -13,   0,
   41,  64,  41,   0, -13,   0,   7,   0,  -5,   0,   3,   0,
   -2,   0,   2,   0,  -1,   0,   1,   0,   0,   0,   0,   0,
    0,   0,   0
};
```

АЧХ на ненормирания НЧФ е дадена на фиг(3.3). Затихването на така проектирания филтър е средно –60dB в лентата на задържане.



фиг3.3 АЧХ на FIR(Hamming) LPF ord=50; fs=1000Hz; fc=250Hz

3.2.3 Проектиране на ВЧ FIR филтър с прозорец на Kaiser

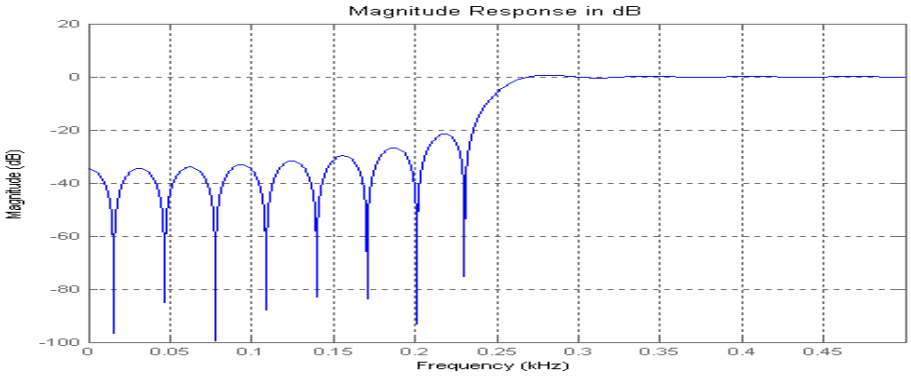
Заданието на филтъра е следното:

- тип на филтъра **ВЧФ - BPF**
- тип на метода за проектиране на филтъра **FIR**
- тип на прозореца – **Kaiser**
- **β=0.5**
- ред на филтъра **30**
- честота на дискретизация **fs = 1000Hz**
- срязваща честота **fc=250Hz**

Нормираните коефициенти на ВЧФ , които са изчислени във **Filter Design & Analysis Tool** са :

coef[31] = {
3, 0, -3, 0, 4, 0, -4, 0, 6, 0, -8, 0,
14, 0, -41, 64, -41, 0, 14, 0, -8, 0, 6, 0,
-4, 0, 4, 0, -3, 0, 3};

АЧХ на ненормирания ВЧФ е дадена на фиг(3.4):



фиг3.4 АЧХ на FIR(Kaiser β=0.5) HPF ord=30; fs=1000Hz; fc=250Hz

3.2.4 Проектиране на заграждащ FIR филтър с прозорец на Hamming

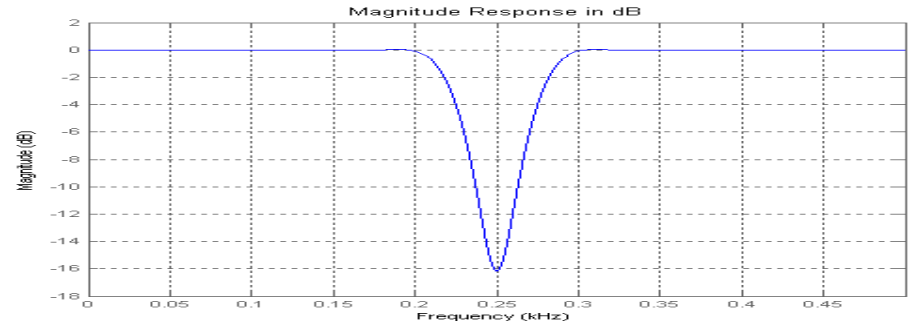
Заданието на филтъра е следното:

- тип на филтъра ЗФ - **BSF**
- тип на метода за проектиране на филтъра **FIR**
- тип на прозореца - **Hamming**
- ред на филтъра **50**
- честота на дискретизация **fs = 1000Hz**
- честота **fc1=230Hz**
- честота **fc2=270Hz**

Нормираните коефициенти на ЗФ са:

coef[51] = {
0, 0, 0, 0, 0, 0, 0, 1, 0, -2, 0, 3,
0, -4, 0, 5, 0, -7, 0, 8, 0, -9, 0, 10,
0, 118, 0, 10, 0, -9, 0, 8, 0, -7, 0, 5,
0, -4, 0, 3, 0, -2, 0, 1, 0, 0, 0, 0,
0, 0, 0};

АЧХ на ненормирания ЗФ е дадена на фиг(3.5):



фиг3.5 АЧХ FIR(Hamming) BSF ord=50; fs=1000Hz; fc1=230Hz; fc2=270

3.3 Среда за проектиране и компилиране на програми разработени за програмируемия микроконтролер PIC18F8520 .

За разработката на програмите написани за микроконтролера **PIC18F8520** е използван компилатора **CCS C Compiler[3]**. Това е C компилатор разработен за серията микроконтролери на фирмата **Microchip**. Той предлага широк набор от вградени (за управление на хардуера) и стандартни **C** функции, както и "**header"** файлове за всички микроконтролери от сериите на **Microchip.** Тези негови качества правят разработката на програми за серията микроконтролери лесна и бърза, в сравнение например с програмите написани на **Асемблер** за същите контролери. Написаните програми задължително трябва да са с разширение *.c. При компилирането на програмата, компилаторът прави проверка за грешки. Ако има такива на монитора се изписва съобщение за грешка. Съобщението съдържа информация за мястото и вида на грешката. Ако програмата е вярна компилаторът извежда съобщение за големината на използваната **RAM** и **ROM** памет, и генерира **HEX** файл, който по нататък се използва от програмиращия софтуер.

Компилаторът има възможност да генерира и други изходни файлове (**LST**,**ERR**,**STA**), които са помощни, но дават ценна информация относно това как би работила програмата. Например **LST** файла дава съответсвието между програмата, която е написана на "**C**" и съответстващия и асемблерен код. Това е много полезно, защото може да се сравняват различните версии на програмите, които са написани по отношение на тяхното бързодействие (брой на инструкциите в асемблер кода). Изходните файлове за съответната програма обикновенно се записват в директорията, където се намира и сорс файла.

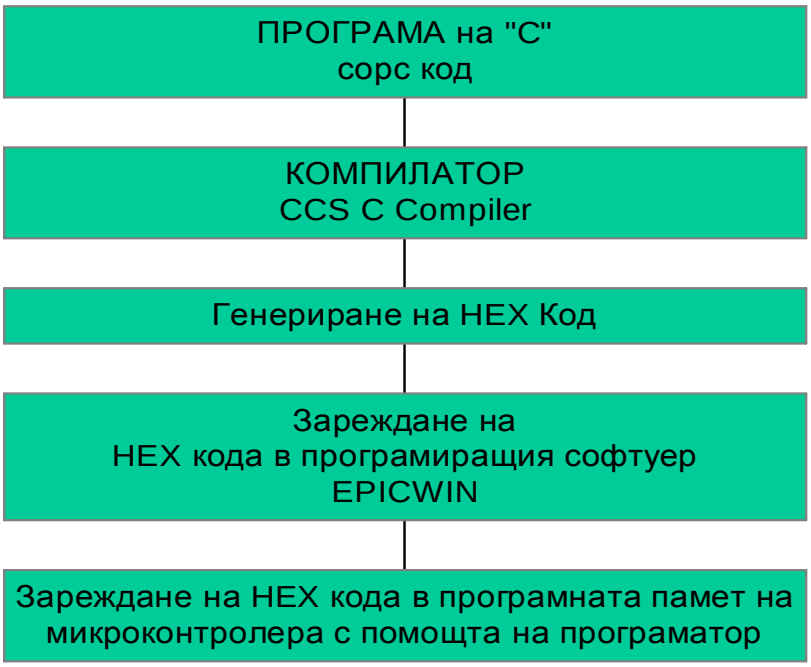
Програмиращият софтуер се използва, за да се зареди генерирания **HEX код** в програмната памет на контролера. Това се прави посредством хардуерен програматор през паралелния порт на персоналния компютър.

Използваният в случая софтуер е програмата **EPICWIN,** която е стандартна **Windows** програма. Може се използва и друг сходен продукт.

EPICWIN има следните възможности:

- широк набор от **PIC** контролери, с които е съвместим
- зарежда **HEX файла** от директорията, в която се намира
- изтриване на паметта на контролера
- програмиране и верификация на записаните данни
- четене на програмната (**FLASH**) памет на контролера
- четене на паметта за данни (**EEPROM**) на контролера

На фиг.(3.6) са показани етапите, през които трябва да се премине при проектирането на софтуер за системи, в които са използвани **PIC** микроконтролери .

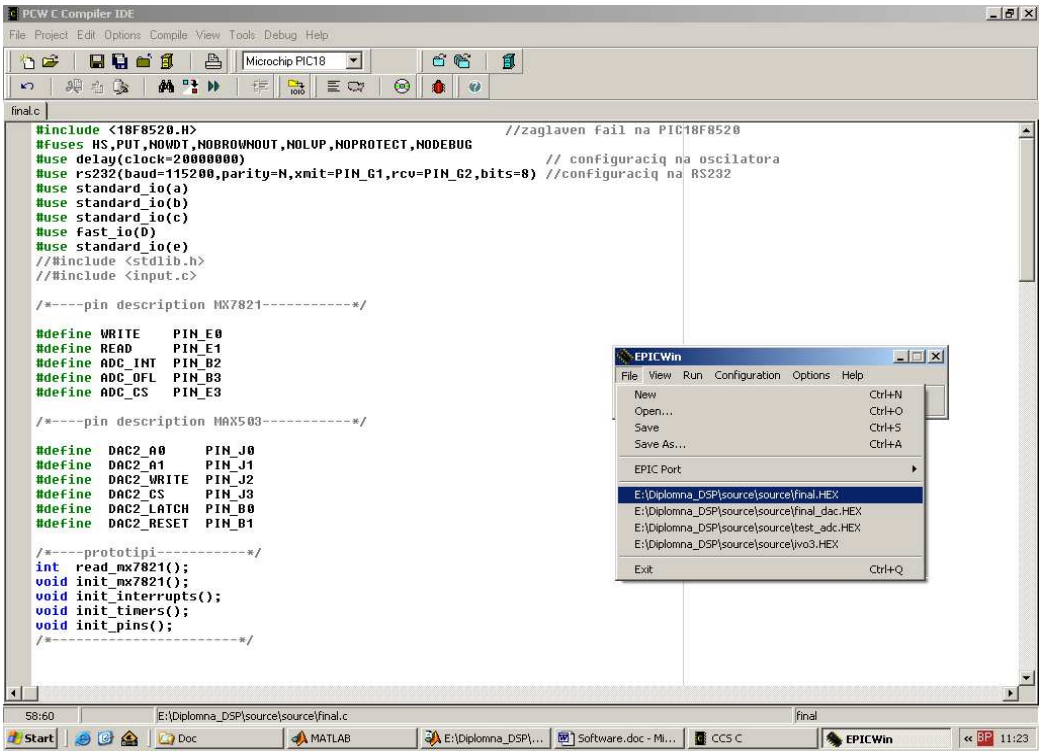


фиг.3.6

Графичната среда на компилатора **CCS C** и програмата **EPICWIN** са показани на фиг(3.7).

Компилаторът **CCS C** е продукт на **Monash University,** а програмата **EPICWIN** е разработена от **microEngineering Labs, Inc** .

За написване на програмата може да бъдат използвани и други текстови редактори, но трябва да им се посочи пътя до компилиращата програма. Един пример за такъв текстови редактор е програмата “**Ultra Edit**” .



фиг. 3.7 Графична среда на компилатора CCS C

3.3.1.Разработване на програма реализираща цифрови FIR филтри за системата Signal processing board v1.0

В тази част на дипломната работа ще се разгледа подробно действието на програмата, реализираща цифрови **FIR** филтри.

Програмата трябва има следните възможности:

- да управлява и настройва серийната комуникация на **Signal processing board v1.0** с **PC** посредством интерфейса **RS232**
- да управлява и комуникира с **АЦП MX7821,** който дискретизира постъпващия аналогов сигнал
- да използва хардуерните таймери на контролера за точното определяне на такта на дискретизация
- да реализира програмно алгоритъма за изчисление на цифров **FIR** филтър
- да предава входните и изходните за цифровия филтър сигнали към **PC(през RS232)** при снемане на АЧХ и само на изходните сигнали при нормална работа

Разработена е програма, написана на С за компилатора **CCS C Compiler,** която е предназначена за хардуерната система **Signal processing board v1.0** :

Програма реализираща цифров FIR филтър

```
#include <18F8520.H>                                  //хедър файл на PIC18F8520
#fuses HS,PUT,NOWDT,NOBROWNOUT,NOLVP,NOPROTECT,NODEBUG
#use delay(clock=20000000)                            // конфигурация на осцилатора 20 MHz
#use rs232(baud=115200,parity=N,xmit=PIN_G1,rcv=PIN_G2,bits=8) //конфигурация  RS232
#use standard_io(a)
#use standard_io(b)
#use standard_io(c)
#use fast_io(D)
#use standard_io(e)
/*-----дефиниране на изводите на контролера, които ще се използват за---------*/
/*----------------------управление на АЦП MX7821-----------------------------------------*/
#define WRITE       PIN_E0
#define READ        PIN_E1
#define ADC_INT   PIN_B2
#define ADC_OFL  PIN_B3
#define ADC_CS     PIN_E3
/*----pin description MAX503----------*/
#define  DAC2_A0     PIN_J0
#define  DAC2_A1     PIN_J1
#define  DAC2_WRITE  PIN_J2
#define  DAC2_CS     PIN_J3
#define  DAC2_LATCH  PIN_B0
#define  DAC2_RESET  PIN_B1
/*-----------------прототипи на използваните в програмата функции---------------*/
int  read_mx7821();       //програма за управление и четене на АЦП mx7821
void init_mx7821();       // инициализация на началното състояние на изводите на  mx7821
void init_interrupts();   // инициализация на прекъсванията
void init_timers();       // инициализация на използваните таймери
void init_pins();         // инициализация на направлението на I/O изводи
/*-----------------------------------------*/
int ord = 51;             //задаване на реда на филтъра
int conv=0;               //флаг за стартиране на преобразуването
signed int16 sum=0;         //клетка използвана за натрупване във FIR алгоритъма
signed int16 term=0;       //помощтна клетка за междинни р-ти във FIR алгоритъма
int8 x[51]={""}, *px; // буфер в който ще постъпват текущите и минали стойности на
                      //входния сигнал
signed int8 coef[51] = {                          // нормирани коефициенти на FIR филтъра
    0,   0,   0,   0,   0,   0,   0,  -1,   0,   2,   0,  -3,     //BPF fs=1000Hz;fc1=230Hz;fc2=270Hz
    0,   4,   0,  -5,   0,   7,   0,  -8,   0,   9,   0, -10,
    0,  10,   0, -10,   0,   9,   0,  -8,   0,   7,   0,  -5,
    0,   4,   0,  -3,   0,   2,   0,  -1,   0,   0,   0,   0,
    0,   0,   0
};
#int_TIMER2                //прекъсване генерирано при препълване на таймер 2
TIMER2_isr(){              //програма за обслужване на прекъсването на всеки   992us
          conv = 1;        // установяване на флага за старт на преобразуването
}
int  read_mx7821(){             //програма за управление и четене на АЦП mx7821. Връща int8
  output_low(ADC_CS);       //активиране на MX7821
  output_low(WRITE);        // команда за начало на преобразуването
  delay_cycles (2);         //TWR=400 ns закъснение. Два цикъла по 200ns
  output_high(WRITE);
  while(ADC_INT == 1);     //изчакване за приключване на преобразуването
  output_low(READ);
  output_high(READ);       //команда за стартиране на прочитането на резултата
  return input_d();         //прочитане и  връщане на резултата от преобразуването
}
```

```
void init_mx7821(){          // инициализация на началното състояние на изводите на  mx7821
   output_high(ADC_CS);
   output_high(WRITE);
   output_high(READ);
}

void init_pins(){                  // I/O конфигурация на изводите
   set_tris_a (0xff);
   set_tris_b (0xff);
   set_tris_d (0xff);              //порт D 0-7 са входове. Data bus
   set_tris_e (0xff);
}
void init_timers() {                              // инициализация на използваните таймери
   setup_wdt(WDT_OFF);                           //watch dog timer изключен
   //setup_timer_2(T2_DIV_BY_1,250,5);     //250us
   //setup_timer_2(T2_DIV_BY_1,208,12);   //500us
   //setup_timer_2(T2_DIV_BY_4,250,5);     //1000us
   //setup_timer_2(T2_DIV_BY_4,177,7);     //991us
   //setup_timer_2(T2_DIV_BY_4,83,15);     //996us
   setup_timer_2(T2_DIV_BY_4,124,10);     //прекъсване на  992us
}
void init_interrupts(){                          // инициализация на прекъсванията
    enable_interrupts(INT_TIMER2);        // разрешение за прекъсвания от таймер 2
    enable_interrupts(GLOBAL);
}

void Main() {
    int i;
    printf("system ready \r\n");   //готовност на системата. Показава че контролера работи
    init_pins();
    init_mx7821();
    init_timers();
    init_interrupts();
    px = &x[ord-1];     //указателя приема адреса на последната клетка на входния буфер
          while (1) {
      if(conv == 1){      //ако conv = 1 стартирай преобразуването
        conv=0;                              //нулиране на флага
        *px = read_mx7821();               //запис на резултата на адрес в кръговия буфер
        output_high(ADC_CS);              //изключва АЦП mx7821
        printf(" %u \r\n", *px);           //изпраща резултата от преобразуването към PC
                                           //!!!! Само в режим на снемане на АЧХ
        for (i =0 ; i < ord;i++ ) {         //реализация на FIR алгоритъм чрез кръгов буфер
              if (px < &x[0]) px = &x[ord-1]; //следи за правилното движение на указателя
              term = *(px--) ;            //зареждане на вх. Сигнал чрез указател в term.
              term*= coef[i];             //умножител x(i-m)*coef(i)
              sum += term;                //натрупващ суматор
        }
             sum/=128;                        //денормиране на изхода
             printf("%ld \r\n ",sum);      //изпраща филтрирания сигнал към PC
             term=0;                          // нулиране на  term
             sum=0;                           //нулиране на  sum
             ++px;                            //увеличаване на адреса на указателя px
      }
    }
}          //край на Main
```

В разработената програма филтрирането на сигнала се извършва чрез директно изчисляване на конволюцията във времевата област.
Изчисляването на изходния сигнал чрез диференчното уравнение

$$y(k) = x(k)h(0) + x(k-1)h(1) + ... + x(k-m)h(m) \qquad (3.2)$$

се извършва с помощта на цикъл **FOR ,** умножител и натрупващ суматор. Текущата и миналите стойности на входния сигнал се съхраняват в кръгов буфер (масива **x[])**, като големината му се определя от реда на филтъра. Коефициентите на **FIR** филтъра се съхраняват в едномерния масив **coef[](**във формулата отбелязан с h**).**
За реализирането на **FIR** филтъра е използван следния алгоритъм:

1. Указателят **px** се зарежда с адреса на последната клетка на буфера **x**
2. Стартира се АЦП (на всяка 1 ms при fs=1000Hz)
3. Посредством указателя **px** на адреса, към който той сочи в момента се записва резултата от преобразуването(най-новия дискрет).
4. Чрез натрупване в цикли **FOR** се изчислява изхода:

$$sum(k) = sum(k) + x(k-m)coef(m)$$

за m=0,1,2…M; , където M е реда на филтъра, а k - номера на изходния дискрет,който ще бъде изчислен.

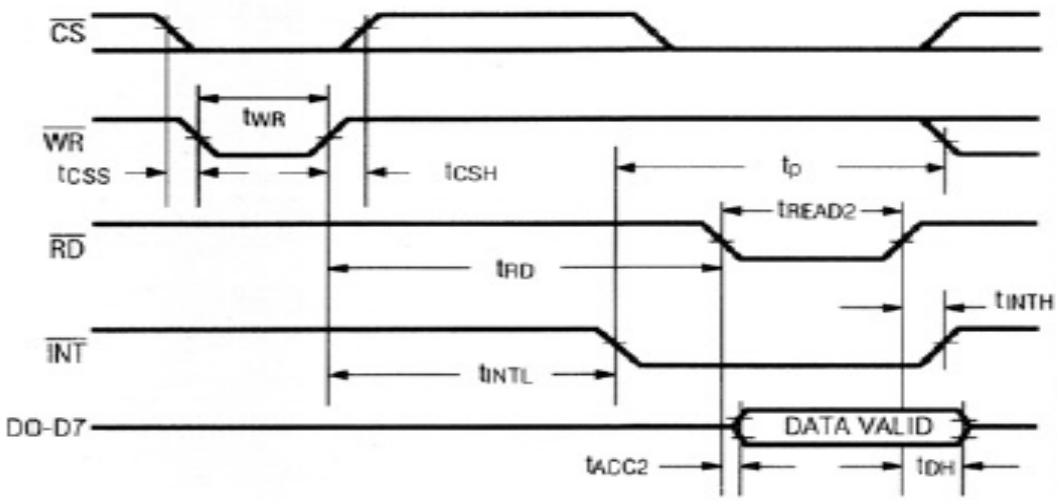
За получаване на адресите на елементите на **x[],** съдържащи стойностите на входния сигнал, се използва указателя **px ,** като адреса му се намалява по модул с 1 във всеки цикъл. По този начин той ще обходи последователно всички записани във входния буфер елементи. Последователността на обхождане е: x(k), x(k-1),x(k-2),……,x(k-m) .

Във всеки цикъл елементите на масива coef[] , съдържащи коефициентите на филтъра се умножават в необходимия ред с елементите на масива **x[]**. Коефициентите на филтъра се извикват чрез директна индексация от масива coef[i].

След извъртане на цикъла указателят **px** сочи отново към адреса, който съдържа най-новия дискрет на входния сигнал x(k) .

5. Изчисленият изход се изпраща към **PC**
6. Нулира се натрупващия суматор (в програмата е означен със **sum**)
7. Увеличава се адреса на указателя **px** по модул с 1. Така следващият най-нов дискрет ще се запише върху най-старата запазена досега стойност на входния сигнал. Това означава, че алгоритъмът записва най-новите стойности на сигнала върху най-старите, като по този начин ги изтрива (откъдето идва термина кръгов буфер).
8. Преминава се към точка **2** от алгоритъма .

Управлението на АЦП трябва да става на базата на времедиаграмите за контрол, дадени от фирмата производител. Контролерът трябва да се програмира много точно, така че да се спазят последователността и продължителността на управляващите сигнали. Времедиаграмата показваща акуратното управление на АЦП MX7821 е показана на фиг.(3.8):



фиг.3.8 Времедиаграма за управлението на АЦП (виж в прог. функция read_mx7821())

АЦП се управлява посредством три сигнала за контрол ADC_CS, WRITE, READ и два сигнала за статус ADC_INT, ADC_OFL. Данните се предават паралелно към контролера, като за това се използва **PORT D** на микроконтролера.

В програмата реализираща FIR филтъра, управлението на АЦП MX7821 се изпълнява от подпрограмата **read_mx7821()**. Нейното действие е в съответствие с времедиаграмата на фигура(3.8).

Подпрограмата "**read_mx7821()"** има за цел да активира чипа MX7821, да стартира преобразуването, да прочете и върне резултата от него. В програмата с коментари е отбелязано предназначението на всеки сигнал и закъснение, така че тук няма да се спрем на това .

Друга важна настройка на програмата е точното изчисляване на времето на дискретизация. Филтърът е предвиден да работи с честота на дискретизация fs =1000Hz или време на дискретизация $t_0 = 1ms$. За точното определяне на времето, през което трябва да постъпват новите дискрети, се използва вграденият в контролера **PIC18F452** хардуерен модул **TIMER 2.** Настройката на **TIMER 2** трябва да се извърши по такъв начин, че при препълването му да се генерира прекъсване през време малко по-малко от 1ms. При генерирането на това прекъсване ще се вдигне флага за старт на преобразуването, отбелязан в програмата като "**conv" ,** който малко след това ще доведе до стартиране на АЦП и

преизчисляване на филтрирания сигнал (нов изходен дискрет). Времето за генериране на прекъсване се изчислява по следния начин:

$$t_0 = \frac{4}{f_{osc}} k_{DIV}\, n\, C \qquad (3.3),$$

където

- f_{osc} - честота на осцилатора
- k_{DIV} - коефициент на делене на осцилатора 1,2,4,8,16
- n - число в границите 0-255 показващо стойността, при която таймер 2 се препълва
- C – брой на препълванията на таймер 2 до генериране на прекъсване

Ако заместим в уравнение (3.3), следните стойности:

- fosc=20MHz
- kdiv=4
- n=124
- C=10

ще получим:

$$t_0 = \frac{4}{20000000} * 4 * 124 * 10 = 992\mu s$$

Конфигурирането на тези изчисления за програмируемия контролер става с командата :

setup_timer_2(T2_DIV_BY_4,124,10); //992us

За да може контролерът да генерира прекъсвания, те трябва задължително да му бъдат разрешени предварително. Това се прави в подпрограмата **init_interrupts(),** където се разрешават прекъсвания от **таймер 2**.

Конфигурирането на серийния порт се извършва с командата:

#use rs232(baud=115200,parity=N,xmit=PIN_G1,rcv=PIN_G2,bits=8)

Тя настройва следните параметри:

- **Baud rate** = 115200 bps скорост на обмен на данните
- **Parity = N** без проверка
- **xmit=PIN_G1** хардуерен извод на контролера, който ще се използва за предаване на данни
- **rcv=PIN_G2** хардуерен извод на контролера, който ще се използва за приемане на данни
- **bits=8** брой на предаваните битове в протокола

3.3.2 Снемане на АЧХ на лентов филтър. Сравнение между проектираната и експерименталната АЧХ

След като беше разгледан принципът на действие на програмата реализираща **FIR** филтри, следва микроконтролерът да бъде програмиран и тестван за изчисления в **т. 3.2.1** лентов филтър.

Задачата, която е поставена в тази част на дипломната работа, е снемането на АЧХ на програмирания в контролера цифров филтър. За получаването на АЧХ на филтъра е необходимо да се изчисли отношението на честотните спектри на изходния към входния сигнал, подаван на системата за всички честоти в диапазона 0-fs/2.

$$A_{filter}(f) = \frac{A_{out}(f)}{A_{in}(f)} \qquad (3.4)$$

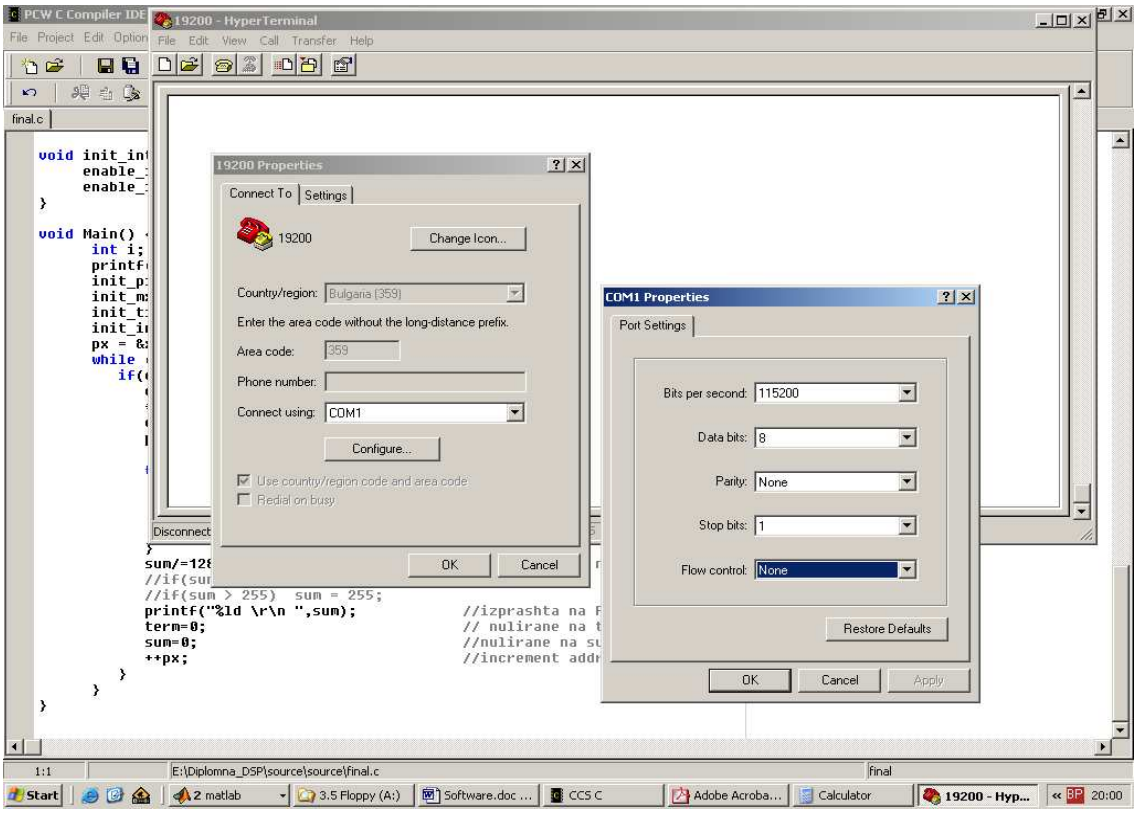
Сигналите използвани за оценяване на АЧХ на филтрите трябва да съдържат всички честоти в посочения по-горе интервал, така че да се възбудят всички собствени честоти на изследваната цифрова система.

Тестовите сигнали, които най-често се използват за изследване на честотните свойства на ситемите са бял шум или сигнал с линейно изменяща се във времето честота(вобел генератори). В случая за тестовете е избран да се използва сигнал от вобел генератор. Тестовия сигнал се генерира в среда на **MATLAB** с помощта на софтуерен генератор и се извежда през звуковата карта на **PC**, като аналогов напрежителен сигнал.

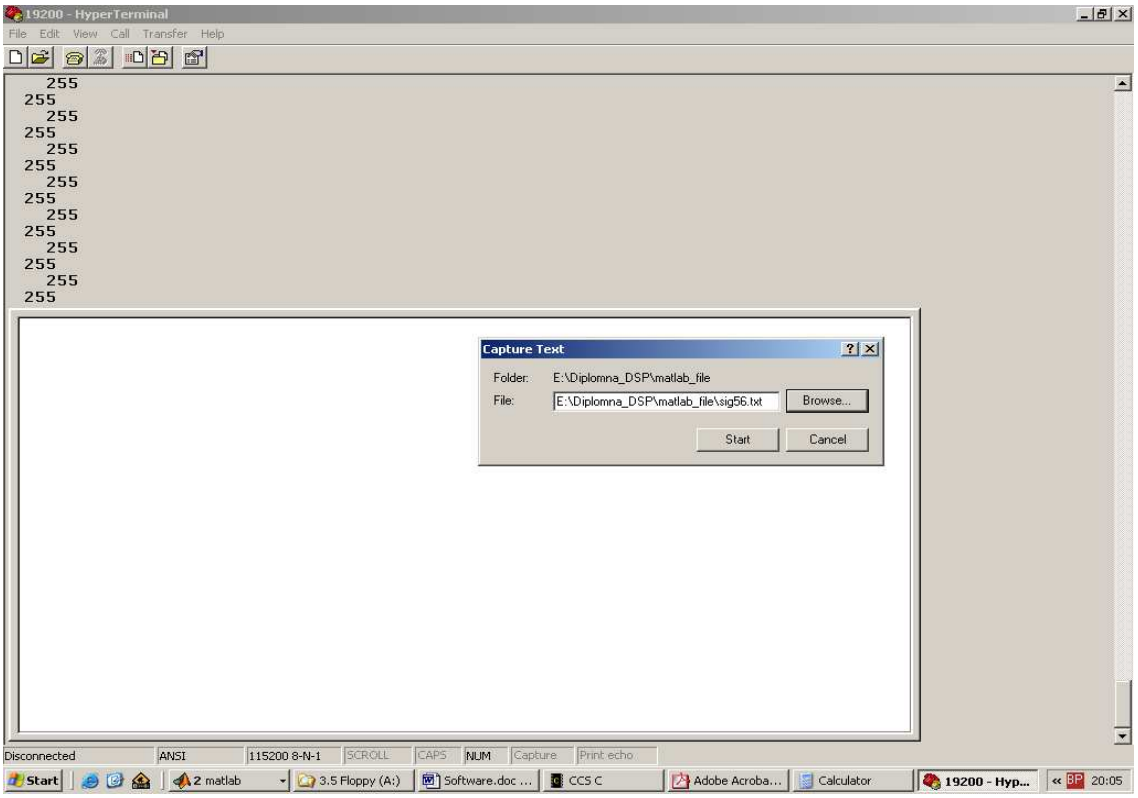
Като параметри за настройка на вобел генераторите се задават началната и крайната честота на генератора, както и амплитудата и времето за достигане от началната до крайната честота.

Програмата която се използва за приемане на данните от платката е универсален хипертерминал показан на фиг.3.9. Неговите настройки, трябва да бъдат в съответствие с настройките, заложени в програмата управляваща процесора на системата.Това означава,че скоростите, броя на предаваните битове и всичко останало трябва да бъдат еднакви както от страната на PC, така и от страната на **Signal processing board v1.0 .** На фиг.3.9 са показани настройките на хипертерминала за конкретния случай.

В режима, при който се тестват честотните свойства на филтрите **Signal processing board v1.0** изпраща по серийния порт към компютъра последователно стойностите на входния и на изчисления изходен сигнал. Данните се записват в текстов файл. Задаването на името на файла и директорията, в която да се записват експерименталните данни е показано на фиг.3.10. Експерименталните данни, снети от лентовия филтър са записани във текстов файл с име **sig54.txt**



фиг.3.9 Настройка на хипер терминала



фиг.3.10 Указване на пътя и файла в който ще се записват данните

След записването на данните е необходимо те да бъдат разделени поради редуващите се във файла стойности на входа и изхода. За тази цел в дипломната работа е разработен скрипт файл **(analize_signal.m**), който зарежда експерименталните данни във вектор-ред и след това ги разделя по четност – четните в един, а нечетните в друг масив. Освен това скрипт файла изчислява спектрите на разделените вече входни и изходни сигнали чрез 2048 точково БФП(Бързо Фурие Преобразувание) и визуализира на една графика техните спектри.

!!! ВНИМАНИЕ След като се визуализират спектрите на двата сигнала, потребителят трябва да определи кой от тях е входен и кой изходен. Това не представлява особена трудност поради факта, че спектърът на сигнала от вобел генератора има относително постоянна амплитуда за целия честотен диапазон, в който работи. В спектъра на филтрирания изходен сигнал пък липсват сигналите с честоти извън лентата му на пропускане

След като са изчислени вече спектрите на сигнала и е решено кой от тях е входен и кой изходен, не остава нищо друго освен да бъде пресметната и АЧХ на изследвания филтър на базата на зависимостта (3.5).

$$A_{filter}(f) = \frac{A_{out}(f)}{A_{in}(f)} \qquad (3.5)$$

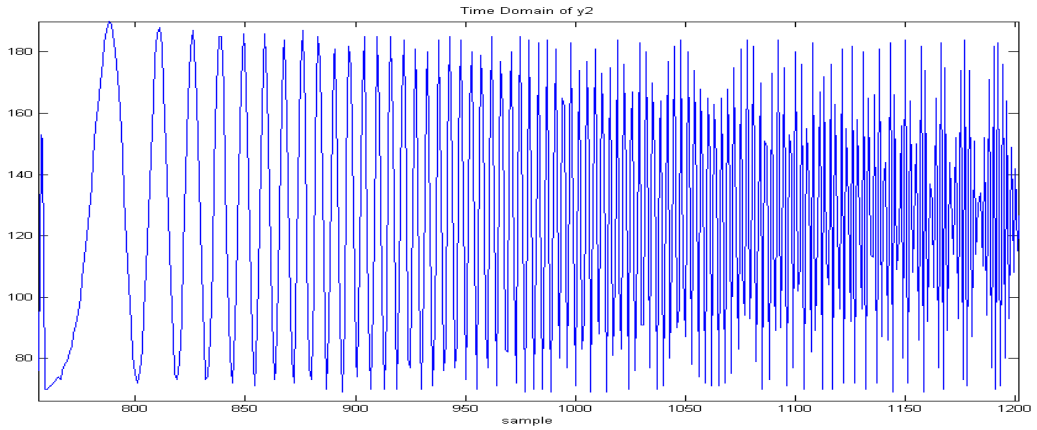
Скрипт файла **analize_signal.m** изчислява и горното отношение чрез поелементно разделяне на двата спектъра и визуализира експериментално снетата АЧХ в диапазона 0-500 Hz. Ординатата на графиката е представена като 20*log10(Aout/Ain) в dB и отразява коефициента на предаване на филтъра за различните честоти.

Скрипт файла е даден по-долу:

```
fs=1000;                %честота на дискретизация
N=2048;                 %брой на точките за които ще бъде изчислен спектъра
a=1;
y = load('sig35.txt');
for i=1:2:size(y)       %отделяне на първия сигнал от файла
    y1(a)=y(i);
    a=a+1;
end;
a=1;
for i=2:2:size(y)       %отделяне на втория сигнал от файла
    y2(a)=y(i);
    a=a+1;
end;
figure(1);plot(y1);             %изчертаване на първия сигнал във времевата област
title('Time Domain of y1')
xlabel('sample')
figure(2);plot(y2);             %изчертаване на втория сигнал във времевата област
title('Time Domain of y2')
xlabel('sample')
Y1 = fft(y1,2048);              %2048 точково Фурие на първия сигнал
```

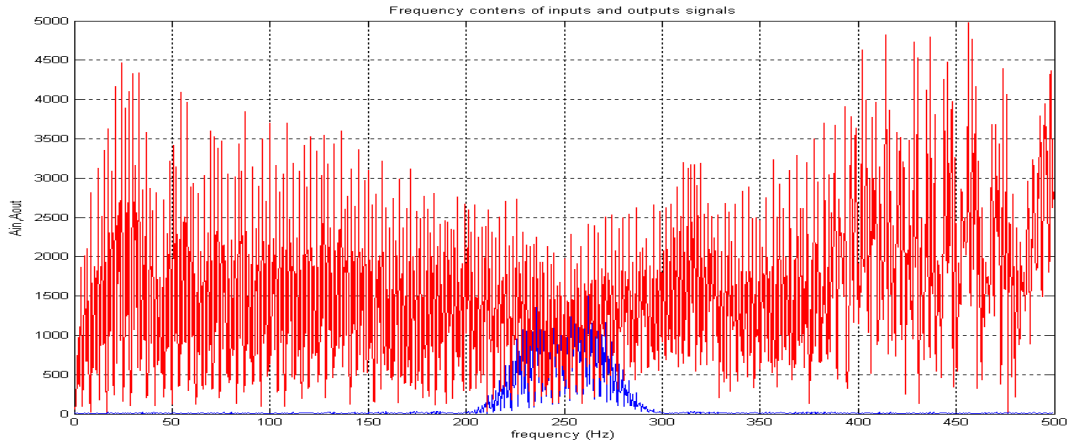
```
Y2 = fft(y2,2048);                    %2048 точково Фурие на първия сигнал
f = fs*(0:1023)/N;                    %формиране на честотен вектор
figure(3);plot(f(2:1024),abs(Y1(2:1024)),'b', f(2:1024),abs(Y2(2:1024))','r'),grid
title('Frequency contens of inputs and outputs signals')
xlabel('frequency (Hz)')
ylabel('Ain,Aout')
Y=abs(Y1)./abs(Y2);                   %изчисляване на АЧХ на цифровия филтър
figure(4);plot(f(2:1024),20*log10(abs(Y(2:1024)')));grid  %изчертавана на АЧХ на цифровия филлтър
title('Frequency response of digital filter')
xlabel('frequency (Hz)')
ylabel('20*log10(Aout/Ain)')
```

Тестовият сигнал от вобел генератора, разгледан във времевата област е даден на фиг.3.11. (експериментални данни от изхода на 8 битовия АЦП)



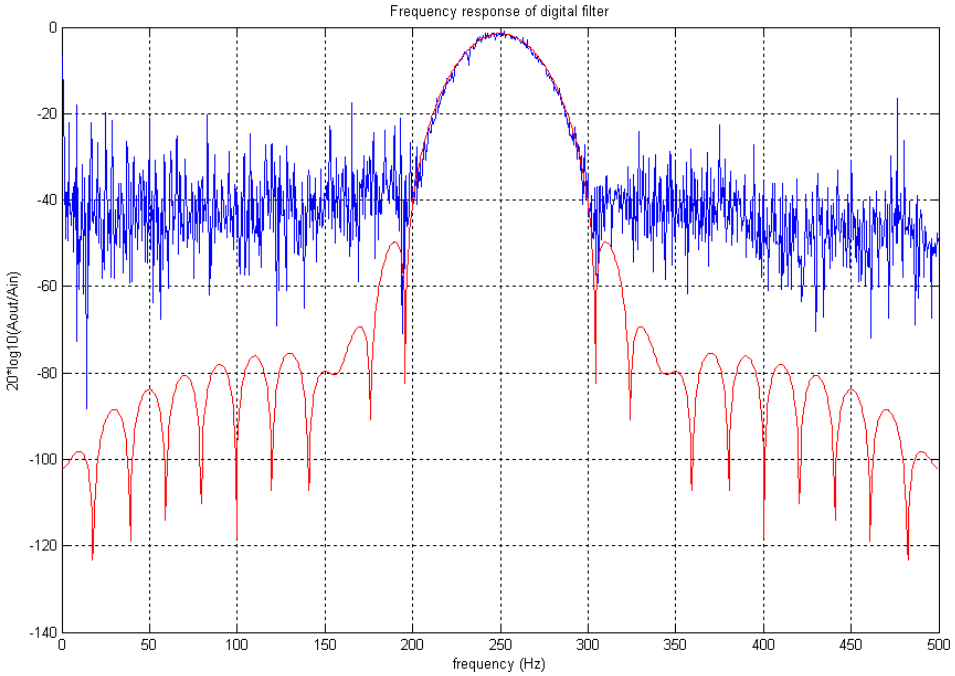
фиг.3.11 Запис на част от тестовия сигнал постъпващ на входа на цифровия филтър.Данни са взети от изхода на АЦП MX7821

На фиг.3.11 са дадени спектралните характеристики на входния и изходния сигнал, отношението на които се използва за получаване на АЧХ на цифровия филтър.



фиг3.11 Честотен спектър на сигналите на входа и изхода на цифровия филтър

На фиг.3.12 върху една графика са показани експериментално снетата АЧХ на цифровия лентов филтър и “проектираната” такава, изчислена в т.3.1.



фиг3.12 АЧХ на теоритичния и реализирания със Signal processing board v1.0 филтър FIR(Hamming) ord=50; fs=1000Hz; fc1=230Hz; fc2=270Hz

Сравнението между двете характеристики показва, че нормирането и закръгляването на коефициентите на цифровия филтър, както и главно използването на 8 битов АЦП изменя неминуемо АЧХ на реалния филтър.

Тестваният в примера филтър запазва свойствата си да подтиска сигналите с честоти извън лентата му на пропускане, макар и в по-малка степен, отколкото реално проектирания (средно –45dB за практическия срещу около 90dB за теоретичния). По-добри резултати могат да бъдат получени, ако се използват АЦП с по-висока разрядност, както и ако коефициентите бъдат нормирани и закръглени към 16 битови цели числа със знак (в момента са към 8bit signed). Това обаче би увеличило значително времето за изчисление (над 4-6 пъти) и би натоварило допълнително конторлера **PIC18F8520**. По тази причина тестовете са направени с коефициенти на филтъра тип 8bit signed и 8bit АЦП.

Следващият пример показва с колко dB най-много може да затихне входния сигнал при положение, че се използва 8 битов АЦП.

Ако приемем, че филтърът има коефициент на усилване в лентата си на пропускане равен на единица, то входният и изходният сигнал се променят в границите между (0 - 255 през 1). Тогава за максимално възможното затихване на сигнала можем да запишем:

$$Atenuation_{\max}(dB) = 20 * \log_{10}(1/255) \approx -48\,dB \qquad (3.6)$$

Изводът от този пример е, че с **8bit АЦП** могат да бъдат апроксимирани сравнително добре цифрови филтри, подтискащи сигналите в лентата си на задържане с не повече от **–48dB**.

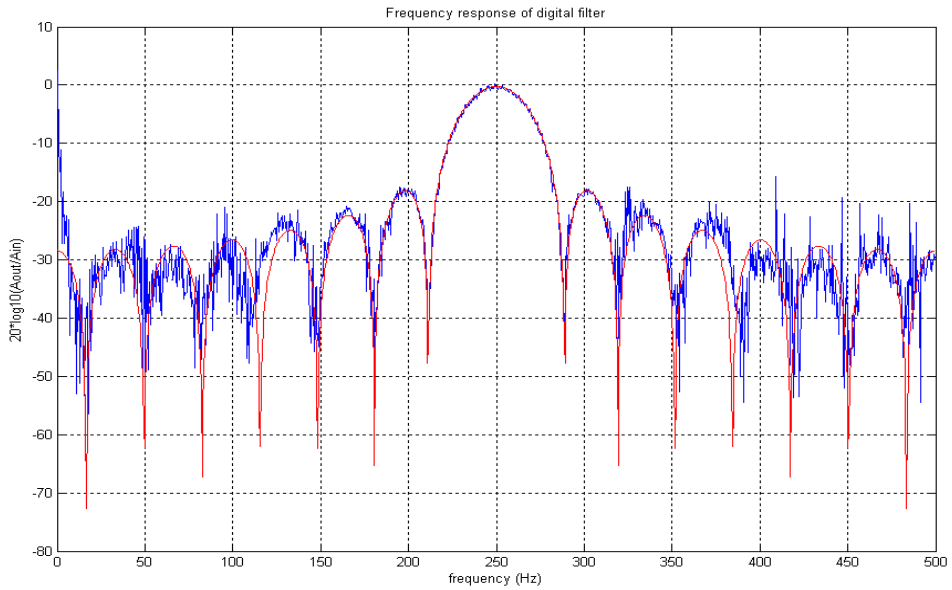
Едно визуално потвърждение на това твърдение ще се даде, като се проектира и тества нов филтър със задание:

- Лентов FIR филтър проектиран с прозорец на Каизер (β=0.5)
- Ред 30-ти
- Fs=1000Hz
- Fc1=230Hz
- Fc2=270Hz

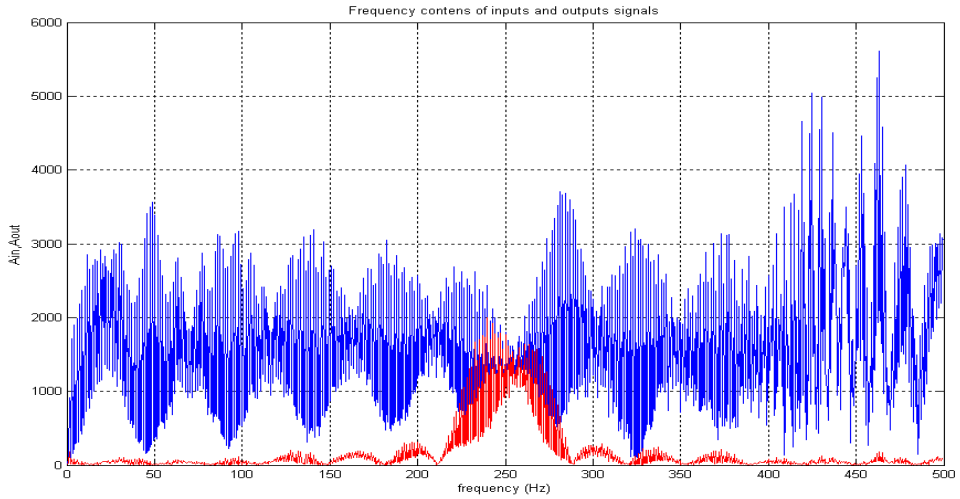
Този филтър има затихване в лентата на задържане приблизително -30dB, фиг. 3.13. Коефициентите на нормирания филтър се изчисляват в средата на **Filter Design Analysis Tools** и са:

```
coef[31] = {
    0,  -5,   0,   7,   0,  -8,   0,   8,   0,  -9,   0,  10,
    0, -10,   0,  10,   0, -10,   0,  10,   0,  -9,   0,   8,
    0,  -8,   0,   7,   0,  -5,   0
};
```

Експерименталните данни са записани във файл с име **sig57.txt**. Проектираната и експериментално снетата АЧХ са дадени на фиг3.13. Тук вече ясно се вижда, че практическият цифров филтър има АЧХ много близка до тази на проектирания.



фиг.3.13 ЧХ на проектирания и реализирания със Signal processing board v1.0 филтърFIR(Kaiser) ord=30; fs=1000Hz; fc1=230Hz; fc2=270Hz



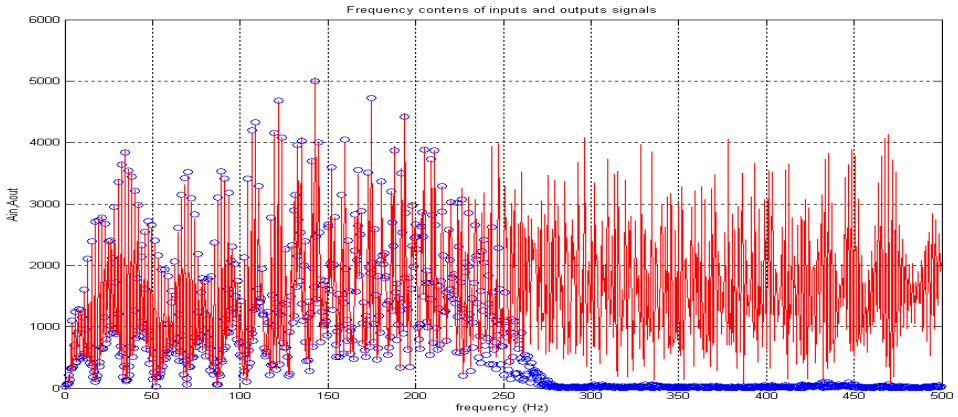
фиг3.14Спектри на входния и изходния сигнал получени при тестовете заFIR(Kaiser) ord=30; fs=1000Hz; fc1=230Hz; fc2=270Hz

Спектрите на входния и получения изходен сигнал са дадени на фиг.3.14.

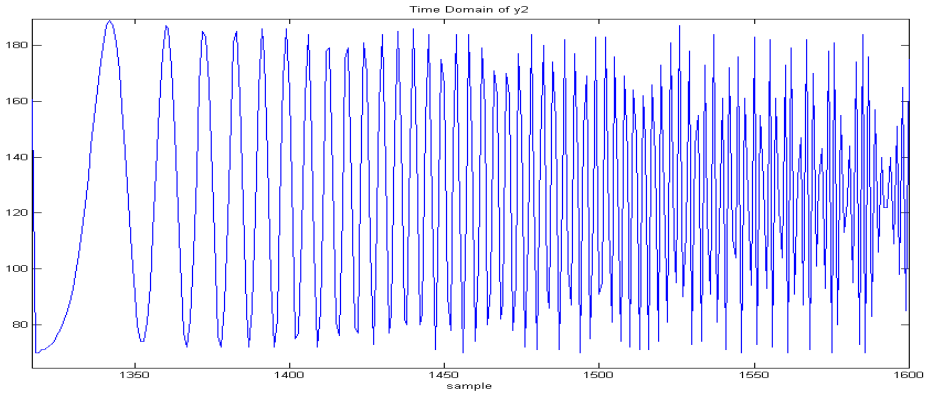
3.3.3 Снемане на АЧХ на НЧФ. Сравнение между проектираната и експерименталната АЧХ

Процедурата по снемане на АЧХ на останалите типове филтри реализирани на базата на системата Signal processing board v1.0 е абсолютно аналогична на тази при снемането на лентовия филтър. Единствената разлика е, че програмата реализираща FIR филтъра трябва да бъде компилирана с новите филтрови коефициенти и реда на новия филтър(масива coef[] и константата ord в програмата). След това контролера се препрограмира и системата става готова за тестване с вградените нови цифрови филтри.

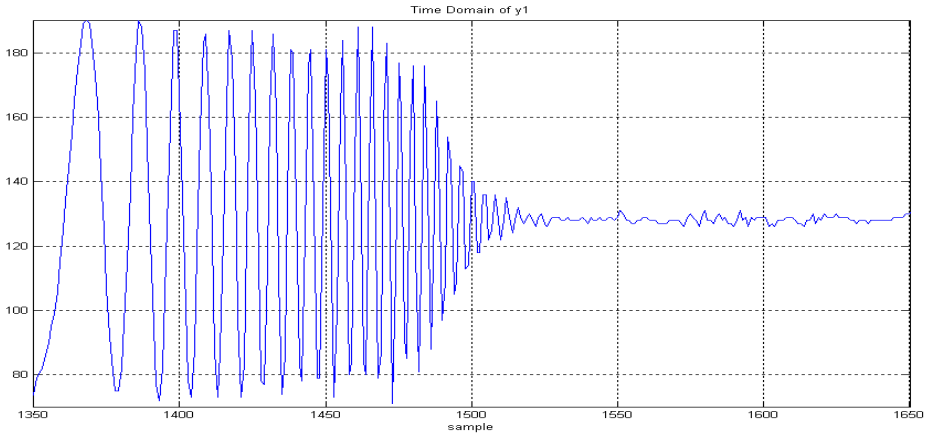
Микроконтролерът се програмира и тества за изчисления в **т. 3.2.2** НЧФ. Тестовете за снемане на АЧХ на НЧФ са проведени при същите условия (вх. сигнал , 2048 БФП) както примера разгледан в т.3.3.2 .
На фиг.3.15 са показани спектрите на сигнала постъпващ на входа на филтъра, както и този получен на изхода му.



фиг.3.15 Спектри на входния и изходния сигнал получени при тестовете на FIR(Hamming) LPF ord=50; fs=1000Hz; fc=250Hz



фиг3.16 Отрязък от сигнала постъпващ на входа на цифровия филтър даден във времевата област. Данните са взети от изхода на АЦП MX7821

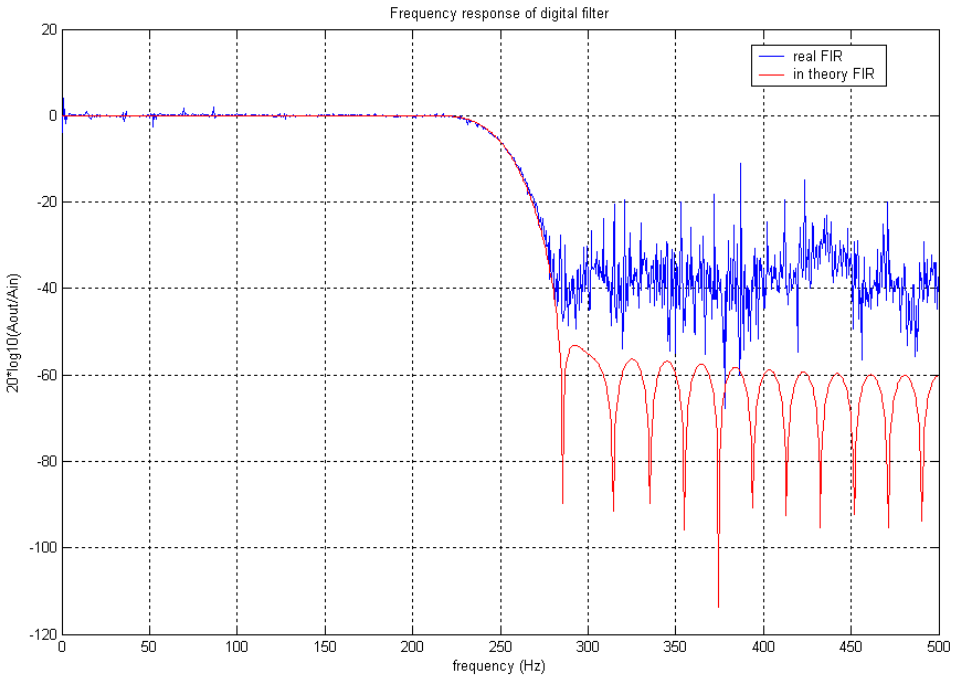


фиг.3.17 Филтриран сигнал отговарящ на входното въздействие дадено на горната фигура. Данните са получен след 50 такта закъснение внесени от FIR филтъра

На фиг3.16 и 3.17 са дадени съответно сигналът подаден на входа и сигнала получен на изхода на изследвания цифров НЧФ. Тези две фигури, много добре илюстрират действието на филтъра, разгледано във времевата област. Подаденият входен сигнал има линейно изменяща се във времето честота, което ни позволява да видим момента от време,в който цифровият филтър ще започне да подтиска входния сигнал. Много ясно се вижда преходната област(дискрети от 1475-1525), и зоната на задържане (дискрети 1525-1650).

Може да се види и времезакъснението от 50 такта, което филтърът внася в сигнала. В този случай при честота на дискретизация fs=1000Hz (to=1ms) закъснението, което ще внесе FIR филтъра в сигнала ще е **delay=50*to=50ms;**

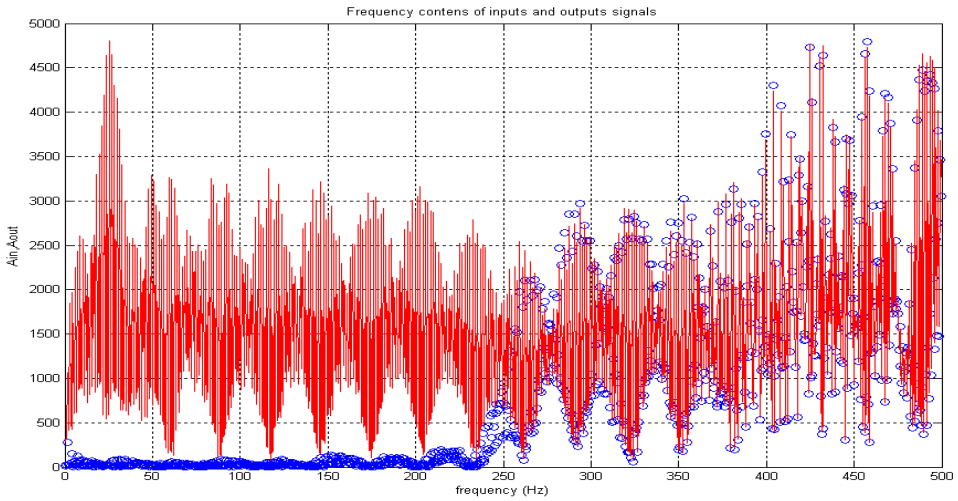
На фиг.3.18 на една графика са показани проектираната в т.3.1.2 и експерометалната АЧХ на цифровия НЧФ. Експерименталната АЧХ е получена чрез разделяне на спектрите на изходния към входния сигнал. Тук също се наблюдава явлението описано в предната точка, отнасящо се до това, че реалният филтър не може да достигне зададеното в лентата на задържане затихване. Както беше отбелязано, това се дължи на използването на 8 битов АЦП.



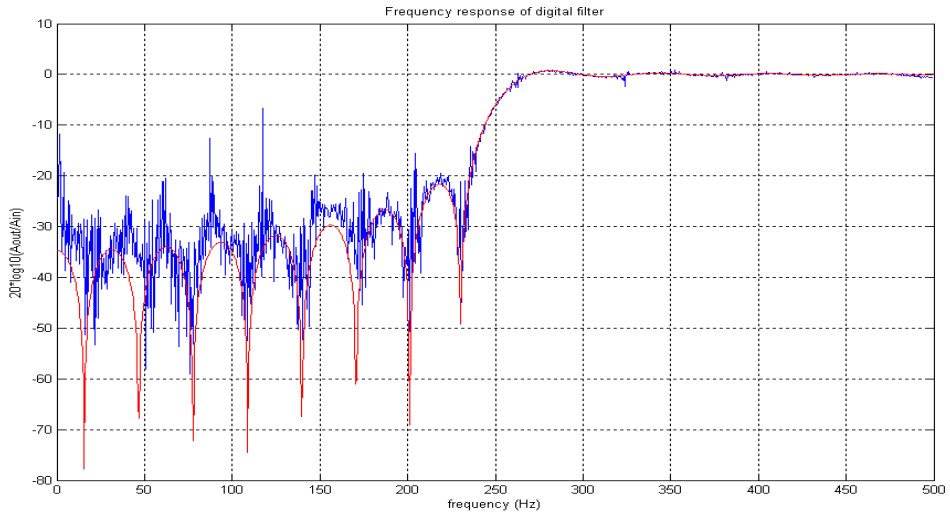
фиг3.18 АЧХ на теоритичния и реализирания със Signal processing board v1.0 филтър FIR(Hamming) LPF ord=50; fs=1000Hz; fc=250Hz

3.3.4 Снемане на АЧХ на ВЧФ и ЗФ.
Сравнение между проектираната и експерименталната АЧХ

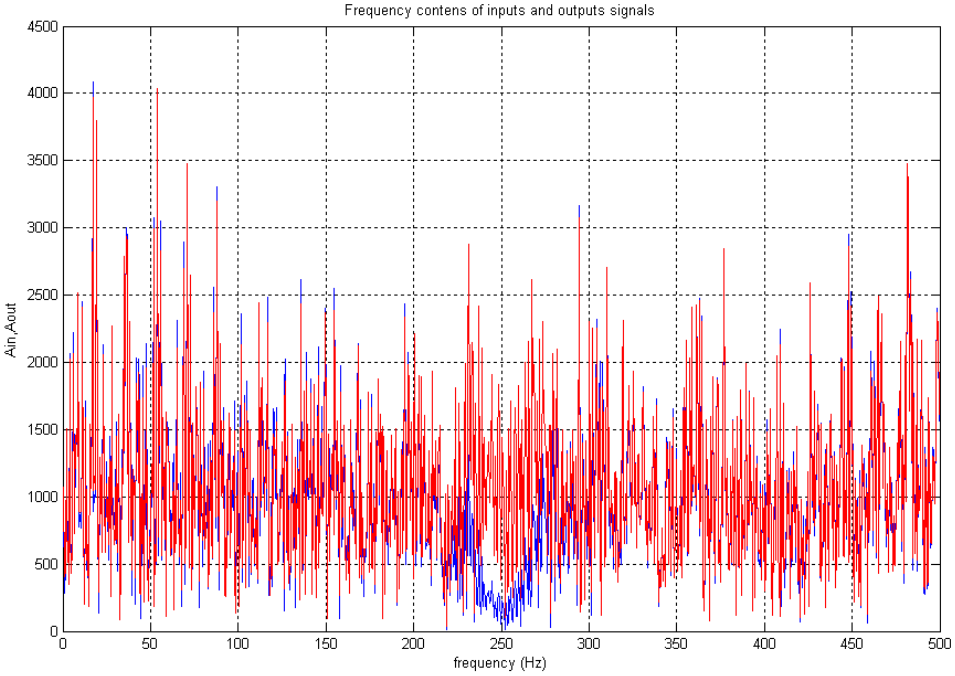
За да се избегнат повторенията, в тази точка са дадени само спектрите на сигналите и експериментално снетите **АЧХ** характеристики на цифровите **ВЧФ** и **ЗФ**. Характеристиките са снети аналогично на разгледаните в предходните точки и се отнасят единствено за различни типове филтри . Микроконтролерът се програмира и тества за изчислените в **т. 3.2.3 т. 3.2.4** ВЧФ ЗФ



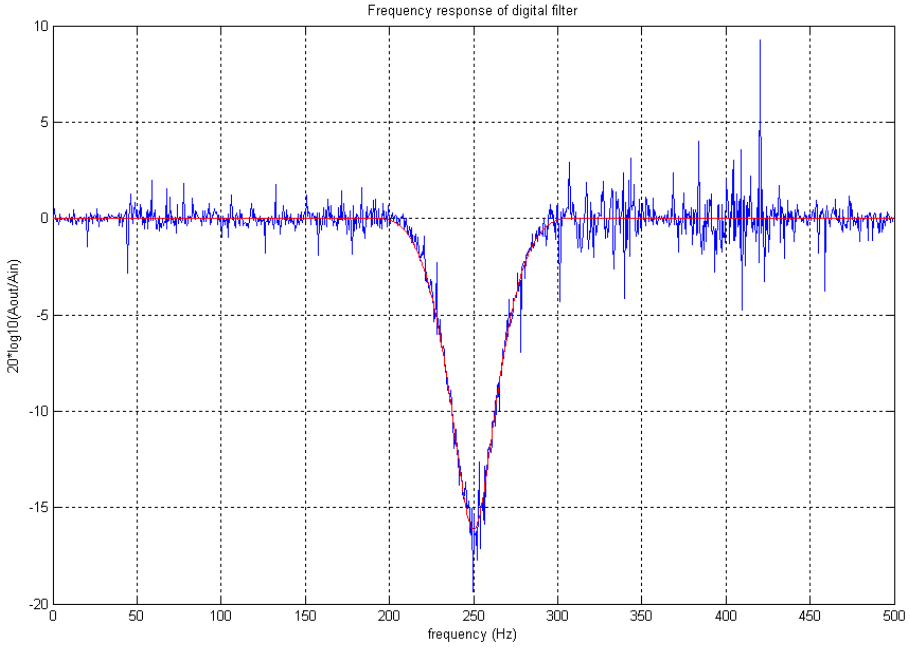
фиг3.19 Спектри на входния и изходния сигнал получени при тестовете на FIR(Kaiser β=0.5) HPF ord=30; fs=1000Hz; fc=250Hz



фиг.3.20 АЧХ на проектирания и реализирания със Signal processing board v1.0 филтър FIR(Kaiser β=0.5) HPF ord=30; fs=1000Hz; fc=250Hz



фиг. 3.21Спектри на входния и изходния сигнал получени при тестовете на FIR(Hamming) BSF ord=50; fs=1000Hz; fc1=230Hz; fc2=270



фиг.3.22 АЧХ на проектирания и реализирания със Signal processing board v1.0 филтър FIR(Hamming) BSF ord=50; fs=1000Hz; fc1=230Hz; fc2=270

Заключение

В настоящата дипломна работа са разгледани основните аспекти при проектирането и реализирането на цифрови филтри.

За целите на дипломната работа е разработена електронна система **Signal processing board v1**.0, базирана на програмируемия микроконтролер **PIC18F8520 .** Електрическата схема и печатната платка на устройството са проектирани с програмния пакет **ORCAD 10.0 .**

Практическо приложение хардуерната система може да намери както в системи реализиращи цифрова филтрация на сигнали в реално време, така и в реализирането на цифрови контролери (PID,PI,FUZZY,PLC) за целите на управлението. Конкретното приложение зависи единствено от програмата, която ще бъде заложена в микроконтролера.

Разучен е **Filter Design toolbox** на **MATLAB** и в частност софтуера за проектиране на цифрови филтри **Filter Design &Analysis Tool.** Проектирани са всички основни филтри (НЧФ,ВЧФ,ЗФ,ЛФ) от тип **FIR** .

Създадена е програма реализираща **FIR** филтър за програмируемия микроконтролер **PIC18F8520** и е тествана върху хардуерната система **Signal processing board v1**.0 . Програмата е написана на език **C** и е предназначена за компилатора **CCS C Compiler .** Тя събира данните идващи от **АЦП,** обработва ги чрез заложения в нея алгоритъм реализиращ **FIR** филтър, и ги изпраща към персонален компютър през серийния му порт.

С реализираните върху **Signal processing board v1**.0 цифрови филтри са проведени тестове за снемането на амплитудно-честотните им характеристики чрез подаването на специален аналогов тестов сигнал.
За снемането на АЧХ е написан допълнителен скрипт файл обработващ данните пристигащи от платката. Сравнени са АЧХ на проектираните в **MATLAB** филтри с експериментално снетите такива. Даден е пример за максималното затихване, което може да бъде постигнато с използването на 8 битови АЦП .

Данните от експериментите проведени за НЧФ, ВЧФ, ЗФ, ЛФ показват добро съвпадение на **АЧХ** на реализираните със **Signal processing board v1**.0 филтри и проектираните при зададено затихване в лентата на задържане до –48dB.

На база на разработения в дипломната работа хардуер и софтуер може да бъде разработено упражнение за студенти по дисциплината “Цифрова обработка на сигнали”.

Литература

1. Г. Мошиц, П.Хорн — Проектирование активных фильтров Москва “Мир” 1984

2.З.Каракехайов, К.Кристенсен, Винтер — Проектиране на вградени микрокомпютърни системи с микроконтролери Pensoft, София-Москва , 2000

3.Х.Шилдт — Практически самоучител С Софтпрес

4.В.Дьяконов,И.Абраменкова — MATLAB Обработка сигналов и изображений Питер, Санкт-Петербург, 2002

5.К.Велев — Теория на автоматичното управление Мартилен, София, 1993

6.Б.Боянов — Цифрова обработка на сигнали Бряг Принт – АД,Варна, 2003

7. Н.Кенаров — PIC микроконтролери Млад конструктор, Варна, 2003

8. Е.Шойкова и колектив — Методология за проектиране на електронни схеми с Pspice Технически университет, София,2000

9. www.microchip.com
10. www.maxim-ic.com
11. Е.Шойкова — Синтез на активни филтри Технически университет, София,2000

12. www.orcadpcb.com

ПРИЛОЖЕНИЕ 1

ПРИЛОЖЕНИЕ 2

ПРИЛОЖЕНИЕ 3