

Úloha č.10: referát: NE

Návrh a implementace filtru IIR na mikroprocesoru Motorola 68HC16

Úkol cvičení:

Pomocí programu FDAS navrhnete filtr IIR a implementujete jej na výukové desce EVB HC16 firmy Motorola..

Parametry filtru ověříte měřeními na osciloskopu a spektrálním analyzátoru.

Parametry filtru

Metodou bilineární transformace navrhnete z analogového filtru **AF1 typu** číslicový filtr **ČF1 typu** pro vzorkovací kmitočty **VK1**.

K návrhu filtru použijte program FDAS.

Legenda:

AF1 :

1. Butterworth
2. Tchebyshev
3. Bessel

ČF1 :

1. dolní propust
 - zlomový kmitočet v propustném pásmu 100 Hz
 - zvlnění v propustném pásmu 0.1 dB
 - zlomový kmitočet v nepropustném pásmu 150 Hz
 - útlum v nepropustném pásmu 30 dB
2. horní propust
 - zlomový kmitočet v propustném pásmu 200 Hz
 - zvlnění v propustném pásmu 0.1 dB
 - zlomový kmitočet v nepropustném pásmu 250 Hz
 - útlum v nepropustném pásmu 30 dB

VK1 :

1. 2016 Hz
2. 4096 Hz

Pozn.

Každý si vybere určitý typ návrhu.

Teorie

Přenosová funkce číslicových filtrů s nekonečnou impulsovou odezvou je dána vztahem:

$$H(z) = \sum_{n=0}^{\infty} h(n)z^{-n} = \frac{\sum_{i=0}^r L_i z^{-i}}{\sum_{i=0}^m K_i z^{-i}}$$

Při řešení aproximačních úloh je třeba určit koeficienty L_i a K_i tak, aby přenosová funkce splňovala požadavky kladené na příslušný filtr buď v časové nebo frekvenční oblasti.

Z metod pro převod přenosové funkce do roviny z má nejvýhodnější vlastnosti (z hlediska zobrazení, jednoduchosti provedení atd.) tzv. bilineární transformace. Ta transformuje analogovou přenosovou funkci $H(s)$ pomocí substituce

$$s = \frac{2}{T} * \frac{z-1}{z+1}$$

v $H(s)$ k dosažení $H_d(z)$. Ačkoliv tato transformace posouvá frekvenční osy, specifikace filtru je dodržena.

Aproximace

Při aproximaci přenosové funkce analogového filtru vycházíme z aproximace ideální normované dolní propusti.

Tento způsob výchozí aproximace využíváme za předpokladu, že dokážeme převést určitými transformačními vztahy přenosovou funkci normované dolní propusti na přenosovou funkci filtru požadovaného typu a vlastností.

Ve většině případů se aproximace provádí tzv. **aproximačním polynomem** ve spojitě oblasti. Typ polynomu pak určuje hlavní charakteristiky výsledného filtru, zpravidla také nese jméno autora (Butterworth, Bessel, Tchebyshev ap.)

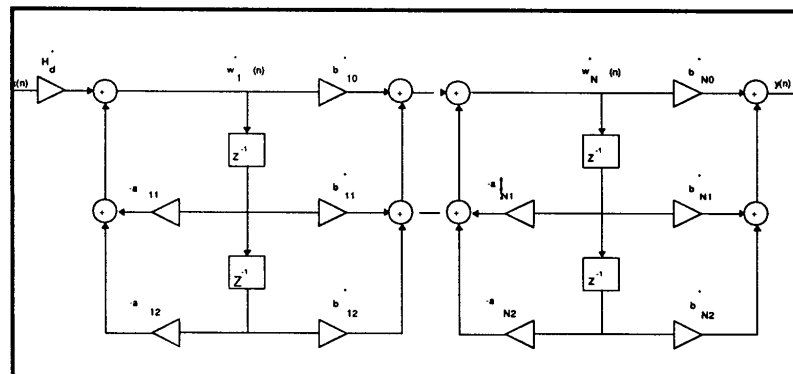
Ve verzi realizovatelné na počítači výstupní hodnota filtru IIR **n** v časové oblasti vypadá takto:

$$y(n) = \sum_{i=0}^M a_i x(n-i) + \sum_{i=1}^N b_i y(n-i)$$

kde a_i jsou koeficienty nerekurzivní části filtru a b_i jsou koeficienty rekurzivní části filtru. Hodnoty $x(n-i)$ jsou vstupní vzorky v čase $n, n-1, \dots$, hodnoty $y(n-i)$ jsou výstupní hodnoty filtru v čase $n-1, n-2, \dots$, $y(n)$ je aktuální nově vypočtená hodnota filtru.

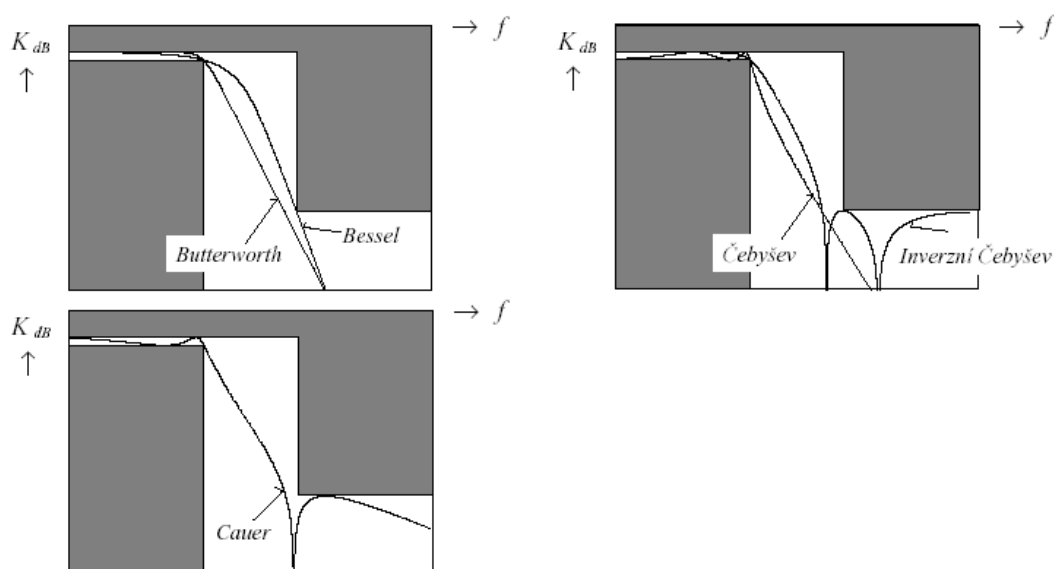
Realizace IIR filtrů na počítači může být několikerým způsobem. Většinou (kvůli zachování jednoduchosti návrhu, stability filtru ap.) se filtr n -tého řádu sestavuje ze sekcí druhého řádu (second order sections). Jejich výpočet a řazení určují pak druh realizace, v případě použití programu FDAS je možno volit mezi tzv. **kaskádní transponovanou** a **kaskádní kanonickou formou** řazení sekcí druhého řádu.

Fig. 2-14 Cascade Form 2



Program FDAS podporuje generování assemblerovského kódu Motoroly pro typ kanonický. (Cascaded Canonic Second Order Sections)

Příklady jednotlivých analogových filtrů typu dolní propust – pro názornost u všech použito stejné toleranční schéma:



Pozn. : Toto toleranční schéma splňují pro představu filtry typu: Bessel 7.řádu, Butterworth 5.řádu, Čebyšev 4.řádu, Inv. Čebyšev 4.řádu a Cauer (elipt.) 3.řádu.

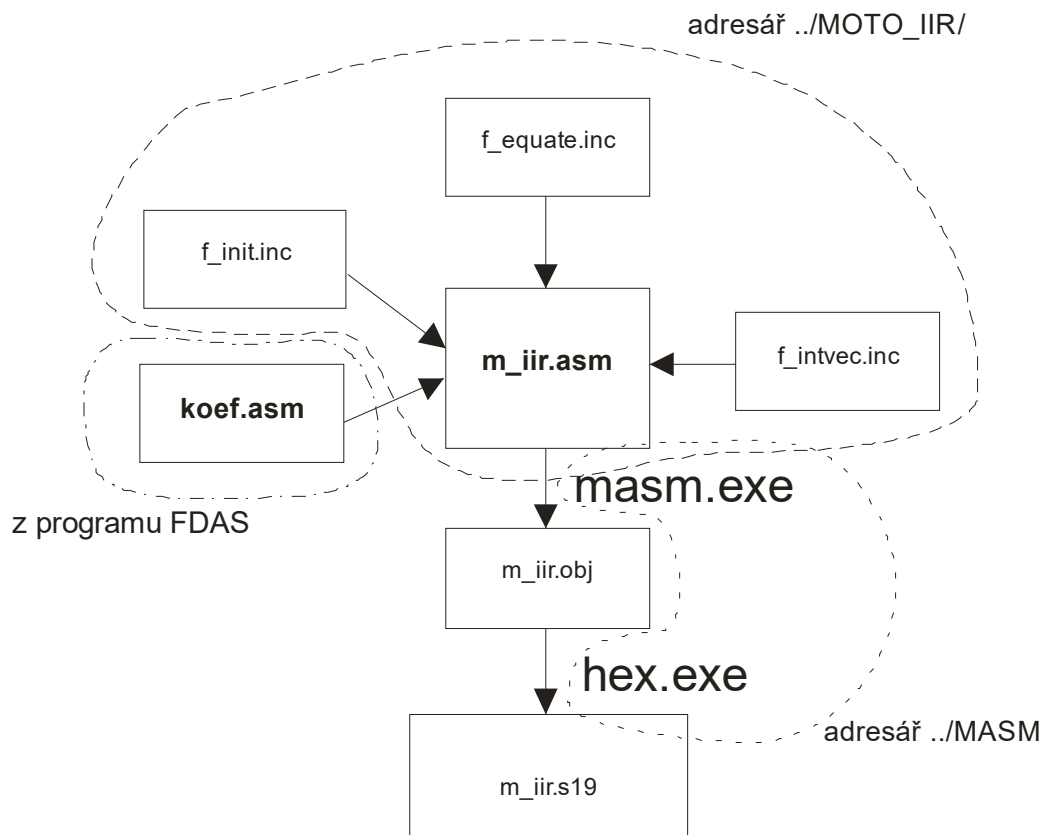
Popis vygenerování zdrojového kódu (včetně koeficientů) z programu FDAS Návrh koeficientů

1. Volit vzorkovací frekvence 1008, 2016, 4096 Hz.
2. Bilineární transformace
3. Kaskádní kanonická forma
4. Použít Code generator umístěný v hlavním menu k vygenerování kódu.

Úpravy na vygenerovaném souboru koeficientů:

1. První řádek „**\$include casd1mds.asm**“ vymazat.
2. Vymazat také celý podprogram *spec*.
3. U všech instrukcí **mac** dopsat čárku mezi čísla.
4. **ORG \$2000** přepsat na **ORG \$4400**.
5. Všechny **dw 0xxxx** přepsat na **dc.w \$0xxxx**

Struktura programu



Výpis programu IIR.asm

; *** hlavní soubor IIR filtru ***

; v programu jsou linkovány tyto soubory :

; `f_equate.inc` - přiřazení symbolických jmen řídicím registrům

; `f_intvec.inc` - definice reset vektoru a prerusovacich vektoru

; `f_init.inc` - inicializační rutiny

; `isb6.asm` - koeficienty filtru a zdrojový kód filtru vygenerované z FDASu

; - koeficienty musejí začínat na adrese \$4400 !

```

;          - zmenit org uvnitr
;
;
; v celem programu jsou pouzity nasledujici rozsirujici adresni registry:
;
; PK = $0 - rozsireni pro PC
; SK = $0 - rozsireni pro SP
; EK = $F - rozsireni pro adresovani registers of modules of M68hc16
; XK = $0 - pro adresaci pameti s pohyblivym indexem
; YK = $0 - pro adresaci pameti

;***** INICIALIZACE POLI *****

SAMPLE_TIME EQU $0022      ;pricitani TOC2 (Hz)
                ; $0022 - 2016 Hz
                ; $0042 - 1008 Hz
                ; $0005 - 13107 Hz
                ; $0010 - 4096 Hz

ORG $4000      ;vynuluje 512 bytu od adr.$4000
BSZ 256        ;pro data

ORG $4400
BSZ 256        ;alokuje pole pro koeficienty

;*****

include f_equate.inc ;definice jmen registru
include f_intvec.inc ;definice prerusovacich vektoru
org CODEADR         ;zde bude program

;*****
;*****

START:
include f_init.inc  ;inicializace

BSR INIT_UNITS     ;nastavi prostredi- ADC,GPT,PWM

LDX #0
LDD #$1006
STD TMSK1_,X      ;povoleni preruseni
LOOP:
BRA LOOP          ;jdi dokola

BDM:
BGND              ;background mod

```

,***** PODPROGRAMY *****

INIT_UNITS:

 ORP #\$0010 ;nastavi saturacni bit v CCR

;nastaveni MAC jednotky

 ldx #\$4400 ;set address for F page addressing
 ldy #\$4000 ;set starting address for intermediate signals
 sty \$0f0,x ;the starting address for signal is stored at 020F0

 LDD #\$ffff ;nastaveni modulo 256
 TDMSK ;nastaveni masek

;nastaveni modulu ADC,GPT,PWM

 LDD #0
 STD ADCMCR ;MCR reg
 LDD #\$0080
 STD ADCTL0 ;ADCTL0 reg
 LDD #\$0001
 STD ADCTL1 ;ADCTL1 reg

; LDD #\$1000
; STD PDDR ;
 LDD #0
 STD TCTL1 ;TCTL1 reg
 STD TOC2 ;TOC2 reg
 STD PWMC ;PWMC reg

 STAB PFPAR ;pouze pro kontrolu
 LDAB #\$FF ;vzork. kmitoctu
 STAB DDRF ;v int.rutine blika s PortemF

 LDAB #0
 TBEK ;shodi EK

 RTS

;-----

;prerusovaci rutina casovace - pocita FIR a posila to ven pres PWMA

TIMERINT:

 LDX #0
 LDD TOC2_,X ;vezmi posledni hodnotu citace
 ADDD #SAMPLE_TIME ;prijeti delku vzorku
 STD TOC2_,X ;programuj komparator
 LDAB #0
 STAB TMSK1_,X ;zakaze preruseni

 LDAB #\$FF
 EORB PORTF0_,X
 STAB PORTF0_,X ;kmita s portem F

```

WHILE_: LDD    ADSTAT_,X    ;cte stav ADC a ceka na ukoncení
        LSRD                ;prevodu
        BCC    WHILE_

        LDD    LJSRR0_,X    ;D=obsah prevodu
        std    0,y    ;store the signal D in IY at IY4000

        LDD    #$1
        STD    ADCTL1_,X    ;spusteni noveho prevodu

        ; vlastni vypocet -----

        clrm                ;clear AM
        ldx    #$4400 ;set starting address for coefficients
        ldhi                ;load HR and IR

; zde skok do kodu generovaneho programem FDAS
        jsr    iir    ;jump to iir filter
; -----

        ldx    #$4400 ;set starting address for coefficients
        ldd    $0f0,x    ;load the previous starting address of signal at 020F0
        addb    #02    ;B = B+2
        std    $0f0,x    ;save the current starting address of signal at 020F0
        xgdy                ;D <=> IY, set starting address for signals

        tmet                ;load the signal to E

        ADDE    #$8000                ;prevod Signed int na Unsigned int
        LDX    #0
        STE    PWMA_,X    ;vystup E na PWMA (pouze horni byte)

; -----

        LDAB    TFLG1_,X    ;cte int. flagy (nutne pro shození !)
        ANDB    #$EF                ;maska
        STAB    TFLG1_,X    ;shození flagu

        LDAB    #$10                ;
        STAB    TMSK1_,X    ;povoleni preruseni
        RTI

,***** KOEFICIENTY FILTRU *****
        include im1132.asm    ;nahraje koeficienty do pameti
                                ;obsahuje ukladaci adresu

,*****

```