

## Návrh a implementace filtru IIR

### Úkol cvičení:

Pomocí programu MATLAB navrhnete filtr IIR a implementujte jej na výukové desce DSK Plus firmy Texas Instruments.

Parametry filtru ověřte měřením na osciloskopu a spektrálním analyzátoru.

### Parametry filtru

Dolní propust -  $f_{\text{vst}}$  do 7 kHz, zlomový kmitočet  $f_{\text{zlom}} = 2$  kHz

Horní propust -  $f_{\text{vst}}$  do 7 kHz, zlomový kmitočet  $f_{\text{zlom}} = 3,4$  kHz

Pásmová propust -  $f_{\text{vst}}$  do 10 kHz, pásmo propustnosti  $f_{\text{propust}} = 300 - 3400$  Hz

Pásmová zádrž -  $f_{\text{vst}}$  do 10 kHz, zádržné pásmo  $f_{\text{zádrž}} = 3$  kHz - 4 kHz

$N = 4$

Řád filtru  $N$  je z důvodu jednoduchosti předem stanoven - volba jiného řádu by vedla k mnohem složitějším úpravám hlavního programu, než tomu bylo u filtrů FIR.

### Teorie

Amplituda výstupního signálu IIR filtru v čase  $n$  je určena nejen předchozími hodnotami vstupních vzorků, ale i předchozími hodnotami výstupu.

$$y(n) = b(0) \cdot x(n) + b(1) \cdot x(n-1) + b(2) \cdot x(n-2) - a(1) \cdot y(n-1) - a(2) \cdot y(n-2) ,$$

Tento vztah popisuje poměry u základního stavebního prvku - IIR filtru 2. řádu.

Vlastnosti filtru tedy závisí pouze na koeficientech, které se použijí při výpočtu, a program zůstává stejný pro všechny druhy filtrů.

Tyto koeficienty vám vygeneruje program **iir**, který spustíte v prostředí MATLAB pod MS Windows.

### Funkce programu iir.m:

1. provede výpočet koeficientů filtru ze zadaných amplitudových a frekvenčních charakteristik a řádu filtru

#### Vstupy:

**druh filtru, zlomové frekvence, útlumy, řád filtru** (dán)

2. vizualizace výsledků - zobrazí amplitudovou a útlumovou frekvenční charakteristiku filtru
3. vytvoří soubor **coeffs.asm**, který uloží na disk do adresáře .. MATLAB/BIN/

**Opět je třeba dle zadání správně nastavit vzorkovací frekvenci převodníku:**

Hodnoty v tabulce jsou v dekadické soustavě.

registr A	FCLK [kHz]	f(LP) [kHz]	registr B	Fs [kHz]	f(HP) [Hz]
36	138,89	3,47	20	6,94	34,72
			18	7,72	38,58
			15	9,26	46,30
			10	13,89	69,44
32	156,25	3,91	20	7,81	39,06
			18	8,68	43,40
			15	10,42	52,08
			10	15,63	78,13
28	178,57	4,46	20	8,93	44,64
			18	9,92	49,60
			15	11,90	59,52
			10	17,86	89,29
24	208,33	5,21	20	10,42	52,08
			18	11,57	57,87
			15	13,89	69,44
			10	20,83	104,17
20	250,00	6,25	20	12,50	62,50
			18	13,89	69,44
			15	16,67	83,33
			10	25,00	125,00
16	312,50	7,81	20	15,63	78,13
			18	17,36	86,81
			15	20,83	104,17
			10	31,25	156,25
12	416,67	10,42	20	20,83	104,17
			18	23,15	115,74
			15	27,78	138,89
			10	41,67	208,33

**Postup při cvičení:**

Zkopírujte celý adresář G:\TEXAS\IIR\ do svého domovského adresáře.

Pro daný frekvenční rozsah vstup. signálu vyčtěte z tabulky hodnoty v registrech **A a B**.

Převeďte je do hexadecimální soustavy.

V adresáři s FIR filtrem otevřete soubor **ac01init.asm** a přepište 2 pravé číslice v hexadecimálním zápisu hodnoty v registrech REG1 a REG2 (měním jen dolních 8 bitů registrů A a B).

Př.

```
REG1      .set    124h
```

```
REG2      .set    20fh      ; nastaví A na 36 a B na 15
```

Uložte soubor a spusťte program **iir** v MATLABu pod MS Windows. Po zapsání koeficientů na disk zkopírujte z adresáře C:\windows\matlab\bin\ soubor coeffs.asm do vašeho domovského adresáře s programy IIR a ukončete práci Windows.

Pod DOSem přeložte ve vašem domovském adresáři pomocí DSKPLASM.EXE zdrojového program IIR.ASM (DSKPLASM.EXE IIR.ASM), který vytvoří soubor IIR.OBJ.

Spusťte program Code Explorer pod MS Windows.

Pomocí menu FILE >> LOAD PROGRAM nahrajte program IIR.OBJ do signálového procesoru a klepnutím na tlačítko RUN jej spust'ete.  
Ověřte parametry filtru na osciloskopu a spektrálním analyzátoru.

\*\*\*\*\*

```

.title      "Low pass filter"

.mmmregs    ;umoznuje pouzivat jmena pametove mapovanych registru
.width      80      ;nastaveni pro listing
.length     55      ;      -- ||--

.setsect ".text",0x1800,0      ;nastavi jednotlivé oblasti pameti
.setsect ".data",0x0200,1      ;spustitelny kod, data, vektory
.setsect "vectors",0x0180,0    ;preruseni

.sect "vectors"                ; tabulka vektoru preruseni ulozena
.copy "vectors.asm"            ; na adrese 0x0180

.sect ".data"                  ; delayed data stored at 0x0200
.set 2                          ; počet sekci 2. radu
OUTPUT.word 0
d.word 0,0,0,0,0,0            ; delay buffer in 2nd order

.sect ".text"
goto(start)                    ; jdi na spustitelny kod
.copy coeffs.asm
.copy ac01init.asm

start:  intm = 1                ; zakaz interrupty
        dcall AC01INIT         ; inicializuj prevodnik
        DP = #0
        nop
        pmst = #01a0h          ; Interrupt pointer maps vectors to address 180h
        OVM = 1
        FRCT = 1               ; pocitam ve fraktalnim tvaru, tj. v rozsahu 0 .. 1
        sp = #0ffah            ; stack pointer located in Communications Kernal
        imr = #240h            ; unmask TDM RINT and HPIINT (host port
interface)
        intm = 0               ; enable all interrupts
        ARO=#2

wait:   NOP                    ;hlavni program - nekonecna smycka
        goto(wait)

receive:                ;Zacatek interrupt service rutiny
        DP=#OUTPUT            ;která se spusti při prichodu cisla z A/D pr.
        AR3=#(d+5)            ;AR3:d(n),d(n-1),d(n-2)
        AR4=#coeffs           ;AR4:nejprve ukazují na A2, dále A1,B2,B1,B0
        A=TRCV                ;do akumulátoru dam hodnotu z převodníku

        B = A & #8000h         ;Zde zacina ponekud slozity postup, který
        if(BEQ) goto(plus);ma za úkol spravne interpretovat znamenko
minus:  B = #0FFFFh            ;cisla z převodníku (16bitové), které se nyní

```

```

        B = B << 15          ;nachazi ve 40tibitovem akumulatoru
        A = A | B            ;zde konci znamenkova extenze

plus:   A = A << 15
        BRC=#(N-1)           ;nastav počet opakovani bloku na počet sekci
        nop                   ;2. radu
        blockrepeat(ENDLOOP) ;konec bloku je navesti ENDLOOP

* mezi LOOP a ENDLOOP se pocitaji tyto 2 diferencni rovnice:
*d(n) = input + d(n-2)*A2 + d(n-1)*A1
*y(n) = d(n-2)*B2 + d(n-1)*B1 + d(n)*B0

LOOP:
* Feedback path
    A+= (*AR4+)*(*AR3-)          ;input+d(n-2)*A2
    A+= (*AR4+)*(*AR3-)
    *AR3=hi(A)                   ;d(n) = input+d(n-2)*A2+d(n-1)*A1
* Forward path
    A = (*AR4+)*(*AR3-)          ;d(n-2)*B2
    A+= (*AR4+)*(*AR3-)          ;d(n-2)*B2+d(n-1)*B1
    DELAY(*AR3-)                 ;d(n-2)=d(n-1)
    A+= (*AR4+)*(*AR3-)          ;d(n-2)*B2+d(n-1)*B1+d(n)*B0
    DELAY(*AR3-)                 ;d(n-1)=d(n)
ENDLOOP

@OUTPUT = hi(A)                 ;Ze 40tibitoveho akumulatoru vezmi
A = @OUTPUT                     ;16 nejdulezitejsich bitu
A = #0FFFCh & A                 ;Dva LSB musí byt 0 (pro prevodnik)
TDXR = A                        ;Chci hornich 16 bitu akumulatoru
return_enable                    ;Konec ISR a navrat do nekonecne smycky
.end

```

V souboru **coeffs.asm** jsou jen koeficienty filtru:

```

coeffs:
.word -12016                    ;A2 z 1. sekce
.word 32767                     ;A1 z 1. sekce
.word 2214                      ;B2 z 1. sekce
.word 3291                      ;B1 z 1. sekce
.word 2214                      ;B0 z 1. sekce
.word -25355                    ;A2 z 2. sekce
.word 32767                     ;A1 z 2. sekce
.word 10507                     ;B2 z 2. sekce
.word 2565                      ;B1 z 2. sekce
.word 10507                     ;B0 z 2. sekce

```