

Úloha č. 9: referát: ANO

Návrh NRDF (FIR) filtru na procesoru Motorola MC68HC16

1/ Navrhněte metodou Fourierova rozvoje a Kaiserova okna pomocí programu FDAS

dolnopropustný filtr FIR :

Vzorkovací kmitočet	1000 Hz
Zlomový kmitočet v propustném pásmu (passband cutoff frequency)	100 Hz
Zvlnění v propustném pásmu (passband ripple)	0.1 dB
Zlomový kmitočet v nepropustném pásmu (stopband frequency)	150 Hz
Útlum v nepropustném pásmu (stopband attenuation)	30 dB

2/ Filtr realizujte na signálovém procesoru Motorola 68HC16

3/ Pomocí spektrálního analyzátoru odměřte frekvenční charakteristiky filtru.

4/ Pozorujte vliv (periodičnost v kmitočtové oblasti) diskrétního zpracování signálu na výstupu číslicového filtru, není - li použit analogový antialiasing filtr (v rozsahu alespoň $<0..3f_s>$)

Úkoly ke splnění:

- 1) nakreslit vámi navržený filtr (toleranční schéma)
- 2) vykreslit impulsovou odezvu navrženého filtru (koeficienty z FDASu)
- 3) popis zobrazení koeficientů - tj. ukázka převodu dvou čísel (kladné a záporné) z dekadické soustavy do hexadecimálního a zpět (koeficienty získané z FDASu)
- 4) stručný popis funkce programu (vývojový diagram)
- 5) popis pozorování aliasingu, proměření a vykreslení amplit.frekv.charakteristiky v pásmu $<0..3f_s>$

Teorie:

Systémová funkce $H(z)$ kauzálního NRDF řádu M je dána:

$$H(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M} = \sum_{n=0}^M b_n z^{-n}$$

Impulsní odezva: (využití pravidla IZT $z^{-M} = \delta(n - M)$)

$$h(n) = b_0 \delta(n) + b_1 \delta(n - 1) + b_2 \delta(n - 2) + \dots + b_M \delta(n - M)$$

Délka filtru:

Shodná s délkou odezvy filtru, tedy $(M+1)$

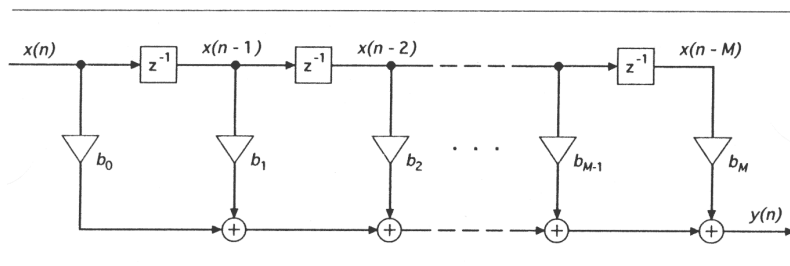
Diferenční rovnice:

$$y(n) = b_0 x(n) + b_1 x(n - 1) + \dots + b_M x(n - M)$$

Struktura filtru:

Transverzální struktura: Zpožďovací členy obsahují pouze vzorky vstupního signálu.

Diskrétní realizace zpožďovací linky



Formát generovaných koeficientů:

Dvojkový doplněk

Popis procesoru HC16

Instrukční cyklus:

Max. rychlost ADC: $ADCCLK = SYSCLK / 4$ (ovšem max 2.1 MHz) ; $SYSCLK = 16.78MHz$ max

Max. vzorkovací rychlost: $16 * ADCCLK$; 15 μs / 8bitů; 16 μs / 10bitů

Módy: vzorkování jednoho / více kanálů

Odebíraná data: Unsigned Right-Justified Format

Signed Left-Justified Format

Unsigned Left-Justified Format

Limity použití mikroprocesoru MC68HC16Z1:

Výpočet FIR filtrů do vzorkovací frekvence cca 10kHz

Výpočet IIR filtrů do vzorkovací frekvence cca 32kHz

Program využívá MAC a RMAC instrukce.

MAC instrukce

(HR) * (IR) => E : D

(AM) + (E:D) => AM

((IX) • (/XMASK) + ((IX) + xo) • XMASK) => IX

((IY) • (/YMASK) + ((IY) + yo) • YMASK) => IY

(HR) => IZ

(M : M+1)x => HR

(M : M+1)y => IR

Legenda:

- Bitové násobení
- * násobení
- + sčítání

RMAC instrukce

Repeat:

(HR) * (IR) => E : D

(AM) + (E:D) => AM

((IX) • (/XMASK) + ((IX) + xo) • XMASK) => IX

((IY) • (/YMASK) + ((IY) + yo) • YMASK) => IY

(HR) => IZ

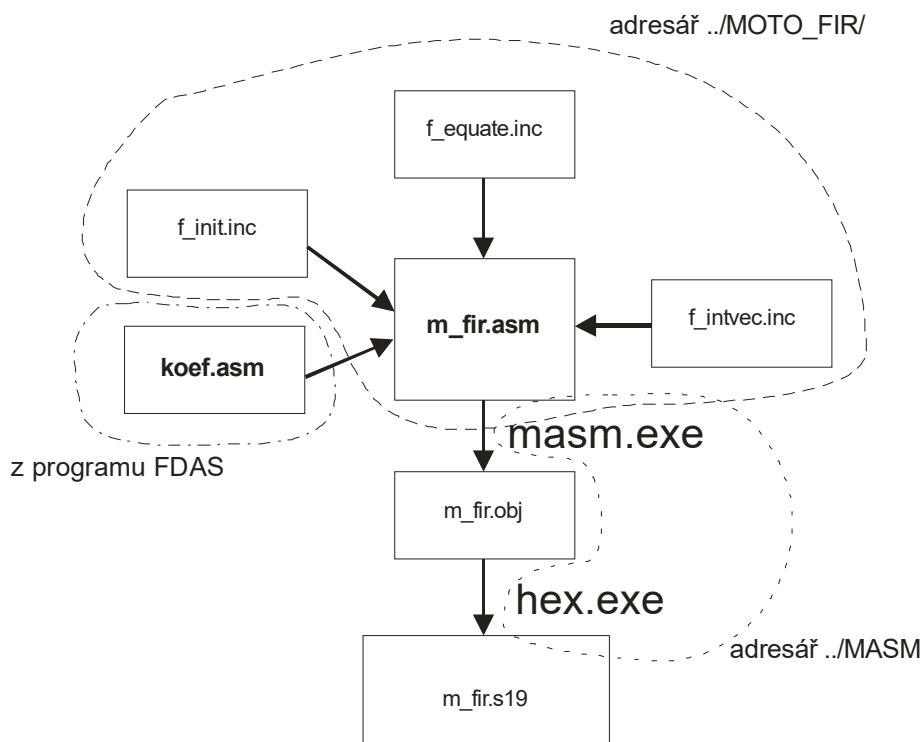
(M : M+1)x => HR

(M : M+1)y => IR

(E) - \$0001 => E

Until (E) < \$0000

Struktura programu NRDF (FIR) filtru implementovaného na procesoru Motorola MC68HC16



VSTUPNÍ SOUBORY

- m_fir.asm:** vlastní hlavní program výpočtu filtru (kam se také zahrnují soubory *.inc)
- + *f_equate.inc*: definice registrů procesoru
 - + *f_init.inc*: HW konfigurace procesoru (systémové hodiny SYSCLK, konfigur. paměti, ...)
 - + *f_intvec.inc*: definice přerušovacích vektorů, včetně námi použitého přerušení od GPT (TIMERINT) + inicializace systémových registrů
 - + *koef.asm*: koeficienty filtru získané z programu FDAS, příp. navržené v MATLABu

VÝSTUPNÍ SOUBORY

- m_fir.s19:** S19 formát Motoroly, sloužící k nahrání kódu do procesoru přes ladící prostředí (debugger) start_evb16.exe

PŘEKLADAČ

- masm.exe:** makroassembler pro procesory řady HC16
- hex.exe:** překladač .obj kódu do .s19 formátu pro procesory řady HC16
- pf4l.bat:** dávkový soubor, který volá masm.exe a hex.exe a vytvoří *.s19 formát

- **NUTNÁ úprava souboru koeficientů: koef.asm:**
 1. vymazat celý první řádek `#include...`
 2. vymazat řádky `„ldaa...”` a `„ldab...”` v návěští `spec`:
 3. v téže návěští do údaje o počtu koeficientů, např. `„lde #xxx”` přidat `$`, na `„lde #$xxx”`
 4. `ORG $2000 -> ORG $4000`
 5. všechny `„dw 0xxxx”` předělat na `„dc.w $0xxxx”`
- **NUTNÁ úprava hlavního souboru m_fir.asm:**
 - Při použití jiné `fs` než 1kHz, nutno upravit konstantu `„SAMPLE TIME EQU $0042”`
 - Na konci programu změnit soubor koeficientů `„include xx.asm” -> „include <vase_koef> .asm`
- **NUTNÁ úprava dávkového souboru pf4l.bat na svoje názvy souborů**

Výpis zdrojového kódu hlavního souboru m_fir.asm:

```
; *** hlavni soubor FIR filtru ***
;
;
; jsou includovany tyto soubory:
;
; equates.inc - prirazeni symbolickych jmen ridicim registrum
; intvec.inc - definice reset vektoru a prerusovacich vektoru
; init.inc - inicializacni rutiny
; fir64_3.asm - koeficienty filtru - museji zacinat na adrese $4000 !
; - 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 registru
; XK = $0 - pro adresaci pameti s pohyblivym indexem
; YK = $0 - pro adresaci pameti
; ZK = $0 - rozsireni pro vystup na DAC

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

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

DATAPTR:        ORG    $4400
                DS.W    256                ;alokuje pole pro data z ADC

N_TAPS:         DS.W    1                  ;pocet odbocek
BUFFER          DS.W    1                  ;velikost kruhoveho bufferu

.*****
;

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

SAMPLE_TIME EQU    $0042                  ;pricitani TOC2 (1kHz)
                                                ;$0021 - 2kHz, atd. cca do 4kHz

.*****
;
START:
                include f_init.inc;inicializace

                LBSR    spec                ;ulozi pocet odbocek filtru do E
                BSR     INIT_N_TAPS         ;zaokrouhli n_taps na nejblizsi vyssi
                                                ;a ulozi do N_TAPS
                BSR     INIT_BUFFER         ;nastavi kruhovy buffer
                BSR     INIT_UNITS          ;nastavi prostredi- ADC,GPT,PWM
                LDD      #$1006
                STD      TMSK1_,X           ;povoleni preruseni
LOOP:           BRA     LOOP                ;jdi dokola

BDM:
                BGND                        ;background mod
```

***** PODPROGRAMY *****

INIT_N_TAPS: ;vezme z akumulatoru E pocet odbocek navrzeneho
;filtru z FDASu a zaokrouhli to na najblizsi vyssi
;hodnotu mocninne rady 2^n .

LDX #0
LDY #0
LDD #0

 XGDE ;vymena horniho a dolniho
 XGAB ;bytu E pres D,
 XGDE ;pro ucely dalsiho zpracovani
 LDD #0 ;vynulovani pocitadla
pocet: ASLE ;shift E doleva - inf. o vel. bufferu
 INCB ;pocitadlo shiftu
 BCC pocet ;

 DECB ;korekce pocitadla
 XGDE ;pocitadlo do E
 LDD #\$00FF ;kruh.buffer max FF
uprav: LSRD ;zmensovani kruh.bufferu
 ADDE #-1 ;v zavislosti na pocitadle
 BNE uprav ;

LDX #\$0
STD N_TAPS,X ;delka kruh.bufferu do BUFFER

RTS

INIT_BUFFER: ;upravi delku kruhoveho bufferu v zavislosti na N_TAPS
;musi zaokrouhlit na mocniny 2^n ; n=1..8

LDX #0
LDY #0

 LDE N_TAPS,X ;N_TAPSnutno doplnit na delku modulo !
 ASLE ;N_TAPS*2
 XGDE ;vymena horniho a dolniho
 XGAB ;bytu E pres D,
 XGDE ;pro ucely dalsiho zpracovani

 LDD #0 ;vynulovani pocitadla
pocet_: ASLE ;shift E doleva - inf. o vel. bufferu
 INCB ;pocitadlo shiftu
 BCC pocet_ ;

 DECB ;korekce pocitadla
 XGDE ;pocitadlo do E
 LDD #\$00FF ;kruh.buffer max FF
uprav_: LSRD ;zmensovani kruh.bufferu
 ADDE #-1 ;v zavislosti na pocitadle
 BNE uprav_ ;

LDX #\$0
STAB BUFFER,X ;delka kruh.bufferu do BUFFER

RTS

```

INIT_UNITS:
    ORP    #$0010        ;nastavi saturacni bit v CCR

;nastaveni MAC jednotky
    LDX    #coef          ;adresa pole koeficientu do IX
    LDY    #DATAPTR       ;adresa pole dat do IY
    LDHI                   ;kopirovani obsahu IX a IY do MAC

    LDX    #0
    LDD    BUFFER,X       ;
    TBA                      ;rozsireni pro XMSK a YMSK

    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
    TBK                    ;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    ;pricti delku vzorku
    STD    TOC2_,X         ;programuj komparator
    LDAB   #0
    STAB   TMSK1_,X        ;zakaze preruseni
;
;
;
    EORB   PORTF0_,X
    STAB   PORTF0_,X       ;kmita s portem F

    LDE    N_TAPS,X        ;E=N_TAPS
    adde   #1
    LDX    #coef           ;vezmi ukazatel do IX
    LDHI                   ;IX a IY do HI a IR

    CLRM
    RMAC   2,2

```

```

    TMER                                ;AM do E - zaokrouhlene
    ADDE    #$8000                      ;prevod Signed int na Unsigned int

    LDX     #0
    STE     PWMA_,X                    ;vystup E na PWMA (pouze horni byte)

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

        LDD     LJSRR0_,X              ;D=obsah prevodu
        STD     0,Y                    ;obsah prevodu do pole dat
        LDD     #$1
        STD     ADCTL1_,X              ;spusteni noveho prevodu

        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 aa.asm    ;nahraje koeficienty do pameti
                      ;obsahuje ukladaci adresu

```

```

,*****
,

```