1 IMPLEMENTÁCIA IIR FILTRA POMOCOU PROCESORA ADSP218X

1.1 ÚVOD

IIR filter je, vzhľadom na prítomnosť spätnej väzby, **zložitejšia štruktúra** ako FIR filter. Z predchádzajúcich cvičení vieme, že je výhodné prenosovú funkciu IIR filtra **rozložiť** na jednoduchšie sekcie druhého rádu - **bikvady**. Všeobecný bikvad je charakterizovaný pomocou **5 koeficientov** b_0, b_1, b_2, a_1, a_2. Prakticky všetky dostupné signálové procesory sú optimalizované aj pre implementáciu IIR bikvadu a procesory Analog Devices ADSP218x nie sú v tomto smere výnimkou. Programový kód pre IIR filter umožňuje demonštrovať vyžitie posúvača (Barrel Shifter), ktorý umožňuje využitie mierkových koeficientov bikvadov uložených v zlomkovom formáte 1.15. Podobne ako kód pre implementáciu FIR filtra aj kód pre IIR filter efektívne využíva základné vlastnosti duálnej Harvardskej architektúry, modulo adresovanie a využitie MAC inštrukcie.

Opísaný zdrojový kód zatiaľ neumožní prácu pomocou reálnych technických prostriedkov (vývojovou doskou EZ-KIT2181 Lite) a jeho funkčnosť bude overovaná pomocou simulátora.

1.2 IIR FILTER

Existuje niekoľko verzií implementácie IIR filtrov, ktoré sa líšia štruktúrou zapojenia a počtom koeficientov. V praktických implementáciách IIR filtrov sa najčastejšie využíva realizácia IIR filtra pomocou kaskádného zapojenia sekcií 2 rádu – bikvadov. Prenosové funkcie bikvadov získame z prenosovej funkcie IIR filtra rozkladom do tvaru

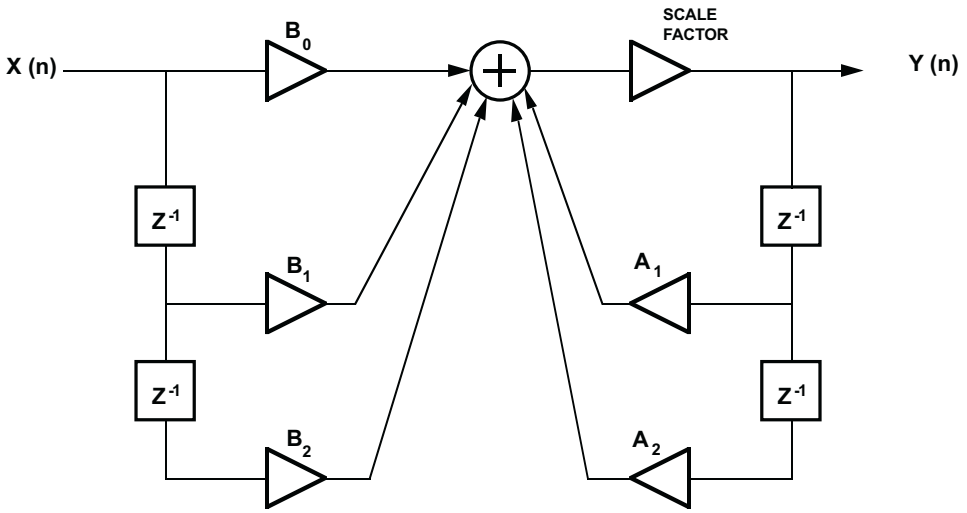
$$H_{IIR}(z) = \prod_{k=1}^{N/2} \left(\frac{b_{0k} + b_{1k} z^{-1} + b_{2k} z^{-2}}{1 + a_{1k} z^{-1} + a_{2k} z^{-2}} \right) = \prod_{k=1}^{N/2} H_k(z) \tag{1.1}$$

pričom na implementáciu prenosovej funkcie $H_{IIR}(z)$ využijeme kaskádne zapojenie $N/2$ bikvadov. Z predchádzajúcich cvičení [1] vieme, že prakticky využívané koeficienty bikvadu a_{jk} môžu byť z intervalu

$$|a_{jk}| < 2.0 \qquad j = 1,2 \quad k = 1,2,\ldots,N/2 \tag{1.2}$$

a tak tieto koeficienty musia byť v prípade DSP s pevnou rádovou čiarkou (pretože interval čísel je u typických DSP obmedzený na $\langle -1,1)$) uložené so zmenenou mierkou. V prípade bikvadov stačí všetky koeficienty a_{jk} vydeliť mierkovou konštantou 2 .

Vieme tiež, že pri technickej realizácii sú často využívané **kanonické formy**, ktoré využívajú minimálny počet stavebných prvkov (predovšetkým oneskorovacích členov). V prípade procesora ADSP218x je však výhodnejšie využiť **priamu formu I**, ktorá využíva 4 oneskorovacie členy/na bikvad a je nekanonickou realizáciou[1]. Hlavý dôvod na využitie nekanonickej formy je práve jednoduchá možnosť pracovať s koeficientmi, ktoré boli vydelené mierkovou konštantou. Štruktúra implementovaného bikvadu je zobrazená na obrázku 1.



Obr.1 Nekanonická realizácia bikvadu – základného bloku IIR filtra

Preberaný podprogram IIR filtra, ktorý využíva uvedenú realizáciu bikvadu patrí do knižnice voľne dostupného programového vybavenia dodávaného firmou Analog Devices a je detailne opísaný v knihe [2] na str. 77-80. Bikvad realizuje výpočet diferenčnej rovnice

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + a_1 y(n-1) + a_2 y(n-2) \tag{1.3}$$

pričom

$$B_0 = \frac{b_0}{SCALE}, B_1 = \frac{b_1}{SCALE}, B_2 = \frac{b_2}{SCALE}, A_1 = \frac{a_1}{SCALE}, A_2 = \frac{a_2}{SCALE} \tag{1.4}$$

a v praxi je najčastejšie používaná hodnota $SCALE$ v tvare 2^s, $s = 0,1,\ldots$

Uvedený podprogram je uložený v súbore **bikvad.asm**. Program, ktorý inicializuje všetky potrebné registre a zabezpečí načítanie vstupných vzoriek je uložený v súbore **IIR_test.asm**. Tieto súbory boli upravené tak, aby ich bolo možné spracovať v prostredí VisualDSP. Súbor **IIR_test.zip [3]** obsahuje všetky súbory projektu v prostredí

[1] Keďže však 2 oneskorovacie členy môžu byť zdieľané s nasledujúcim bikvadom, je celkový počet oneskorovacích členov rovný $N+2$ čo je len o 2 viac ako v prípade kanonickej realizácie. Pre väčšie hodnoty N je tento rozdiel prakticky zanedbateľný.

VisualDSP ako aj ďalšie testovacie súbory potrebné na overenie správnej činnosti kompletného IIR filtra. Súbor bikvad.asm upravený pre prostredie VisualDSP vyzerá takto:

```
/*****************************************************************

Nazov suboru:          BIKVAD.asm

Datum modifikacie:     17-03-2002 MD

Opis:                        Podprogram realizuje kaskadne zapojenie bikvad sekcii IIR filtra

Parametre:             SR1 = vstupna vzorka X(n)
                       I0 --> ukazuje na oneskorovaciu linku X(n-2), X(n-1), Y(n-2), Y(n-1)
                       L0 = 0
                       I1 --> ukazuje na skalovacie faktory pre kazdu bikvad sekciu
                       L1 = 0 (v pripade jedneho bikvadu)
                       L1 = pocet bikvad sekcii (v pripade viacerych bikvadov)
                       I4 --> skalovane koeficienty B2,B1,B0,A2,A1,  B2,B1,B0,A2,A1, ...
                       L4 = 5x pocet bikvadov
                       M0,M4 = 1
                       M1 = -3
                       M2 = 1  (v pripade viacerych bikvadov)
                       M2 = 0  (v pripade jedneho bikvadu)
                       M3 = (1 - dlzka oneskorovacej linky)
                       CNTR = pocet bikvad sekcii

Predpoklady:           Oneskorovacia linka musi mat dlzku N+2
                       Oneskorovacia linka je umiestnena v bloku DM.
                       Instrukcie a koeficienty su ulozene v bloku PM.

Navratove hodnoty:SR1 = vystupna vzorka
                       I0 --> do oneskorovacej linky
                       I1 --> na zaciatok skalovcich faktorov
                       I4 --> zaciatok tabulky koeficientov filtra (v PM)

Zmenene registre:   MX0,MX1,MY0,MR,SE,SR

Pocet cyklov:          8*N/2 + 4 cyklov

Vyuzitie pamate:       Pocet datovych slov (16-bitovych):
                       5N  - pocet koeficientov (24-bitovych)
                       N+2  - pocet slov v oneskorovacej linke (16-bitovych)
                       N  - mierkove (skalovacie) factory

Poznamky:              Vsetky koeficienty bikvadov  a vzorky su v zlomkovom formate 1.15
                       Mierkove (skalovacie) koeficienty su vo formate 16.0

*****************************************************************/

.GLOBAL bikvad;

.section/pm program;

bikvad: DO sections UNTIL CE;
        SE=DM(I1,M2);
        MX0=DM(I0,M0), MY0=PM(I4,M4);                          /* nacita x(n-2), B2 */
        MR=MX0*MY0(SS), MX1=DM(I0,M0), MY0=PM(I4,M4);          /* nacita x(n-1), B1 */
        MR=MR+MX1*MY0(SS), MY0=PM(I4,M4);                      /* nacita B0 */
        MR=MR+SR1*MY0(SS), MX0=DM(I0,M0), MY0=PM(I4,M4);       /* nacita y(n-2), A2 */
        MR=MR+MX0*MY0(SS), MX0=DM(I0,M1), MY0=PM(I4,M4);       /* nacita y(n-1), A1 */
        DM(I0,M0)=MX1, MR=MR+MX0*MY0(RND);                     /* ulozi x(n-1) ako novu x(n-2) */
sections:  DM(I0,M0)=SR1, SR=ASHIFT MR1 (HI);                  /* ulozi x(n) ako novu x(n-1)} */
      DM(I0,M0)=MX0;
      DM(I0,M3)=SR1;
      RTS;
```

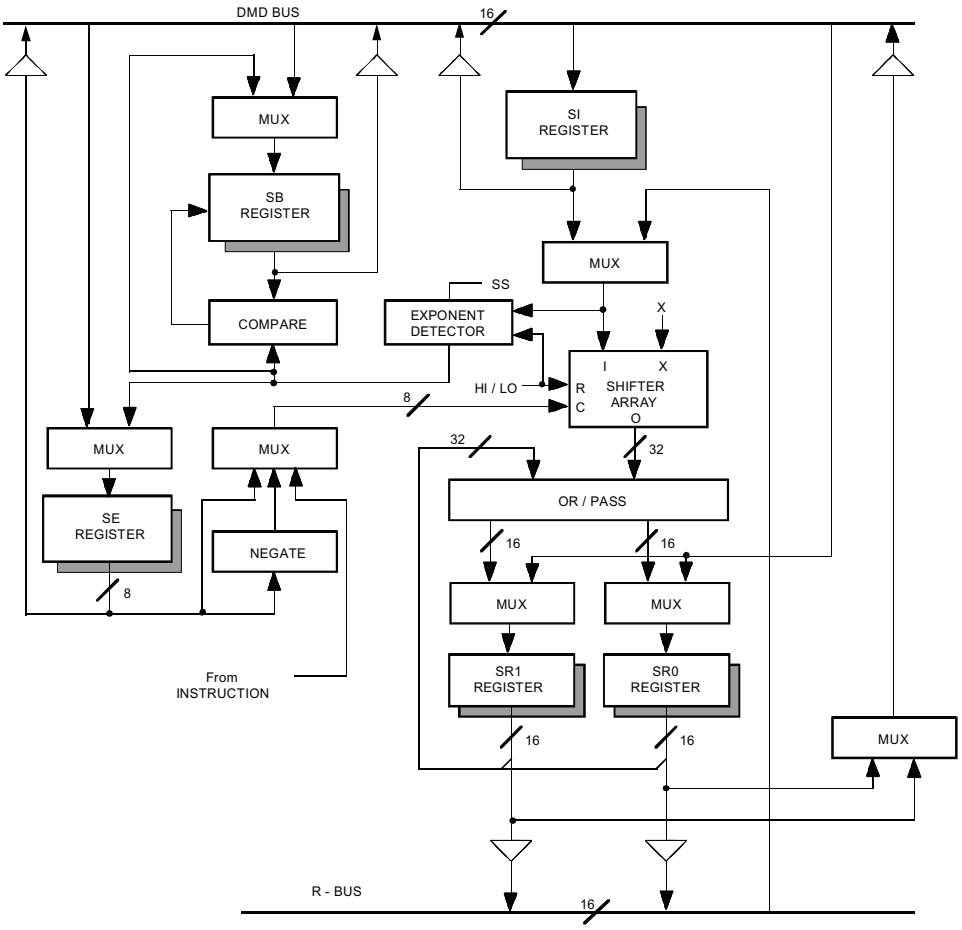
1.2.1 PODPROGRAM BIKVAD.ASM

Pre opis funkcie bikvad.asm je dôležité predovšetkým nastavenie adresovania pre registre **i0, i1, i4,** hodnoty **m1**, **m2**, **m3**, **m4** a inicializácia modulo adresovania, ktoré zabezpečuje hlavný program iir_test.asm. V programovej pamäti (PM) sú uložené koeficienty IIR filtra v poradí $B2, B1, B0, A2, A1$ pre jednotlivé sekcie bikvadov a v oneskorovacej linke s veľkosťou $N+2$ sú uložené vzorky

$$x_1(n-2), x_1(n-1), y_1(n-2), y_1(n-1) \ldots y_{N/2}(n-2), y_{N/2}(n-1) \qquad (1.5)$$

pričom výstupné vzorky bikvadu sú zároveň vstupnými vzorkami ďalšieho bikvadu a teda vzorky v oneskorovacej linke môžu byť čiastočne zdieľané.

Z pohľadu inštrukčnej sady a registrov je v podprograme bikvad použitý posúvač a jeho registre SE (Shifter Exponent) a SR (Shifter Result), ktorý sa skladá z dvoch 16-bitových registrov SR1 a SR0. Umiestnenie výsledku (t.j. výber cieľového registra SR1 alebo SR0) je realizovaný prepínačom HI/LO priamo v inštrukcii. Inštrukcie a registre, ktoré súvisia s posúvačom sú v kóde bikvad.asm označené zvýrazneným písmom. Blokový diagram posúvača je zobrazený na obrázku 2.



Obr.2 Blokový diagram posúvača (Barrel Shifter)

Na obrázku 3 sú zobrazené možné výsledky posunu (ktoré sú v našom prípade definované obsahom registra SE) v závislosti na hodnote prepínača HI/LO.

Control Code		Shifter Array Output			
HI Reference	LO Reference				
+16 to +127	+32 to +127	00000000	00000000	00000000	00000000
+15	+31	R0000000	00000000	00000000	00000000
+14	+30	PR000000	00000000	00000000	00000000
+13	+29	NPR00000	00000000	00000000	00000000
+12	+28	MNPR0000	00000000	00000000	00000000
+11	+27	LMNPR000	00000000	00000000	00000000
+10	+26	KLMNPR00	00000000	00000000	00000000
+9	+25	JKLMNPR0	00000000	00000000	00000000
+8	+24	IJKLMNPR	00000000	00000000	00000000
+7	+23	HIJKLMNP	R0000000	00000000	00000000
+6	+22	GHIJKLMN	PR000000	00000000	00000000
+5	+21	FGHIJKLM	NPR00000	00000000	00000000
+4	+20	EFGHIJKL	MNPR0000	00000000	00000000
+3	+19	DEFGHIJK	LMNPR000	00000000	00000000
+2	+18	CDEFGHIJ	KLMNPR00	00000000	00000000
+1	+17	BCDEFGHI	JKLMNPR0	00000000	00000000
0	+16	ABCDEFGH	IJKLMNPR	00000000	00000000
-1	+15	XABCDEFG	HIJKLMNP	R0000000	00000000
-2	+14	XXABCDEF	GHIJKLMN	PR000000	00000000
-3	+13	XXXABCDE	FGHIJKLM	NPR00000	00000000
-4	+12	XXXXABCD	EFGHIJKL	MNPR0000	00000000
-5	+11	XXXXXABC	DEFGHIJK	LMNPR000	00000000
-6	+10	XXXXXXAB	CDEFGHIJ	KLMNPR00	00000000
-7	+9	XXXXXXXA	BCDEFGHI	JKLMNPR0	00000000
-8	+8	XXXXXXXX	ABCDEFGH	IJKLMNPR	00000000
-9	+7	XXXXXXXX	XABCDEFG	HIJKLMNP	R0000000
-10	+6	XXXXXXXX	XXABCDEF	GHIJKLMN	PR000000
-11	+5	XXXXXXXX	XXXABCDE	FGHIJKLM	NPR00000
-12	+4	XXXXXXXX	XXXXABCD	EFGHIJKL	MNPR0000
-13	+3	XXXXXXXX	XXXXXABC	DEFGHIJK	LMNPR000
-14	+2	XXXXXXXX	XXXXXXAB	CDEFGHIJ	KLMNPR00
-15	+1	XXXXXXXX	XXXXXXXA	BCDEFGHI	JKLMNPR0
-16	0	XXXXXXXX	XXXXXXXX	ABCDEFGH	IJKLMNPR
-17	-1	XXXXXXXX	XXXXXXXX	XABCDEFG	HIJKLMNP
-18	-2	XXXXXXXX	XXXXXXXX	XXABCDEF	GHIJKLMN
-19	-3	XXXXXXXX	XXXXXXXX	XXXABCDE	FGHIJKLM
-20	-4	XXXXXXXX	XXXXXXXX	XXXXABCD	EFGHIJKL
-21	-5	XXXXXXXX	XXXXXXXX	XXXXXABC	DEFGHIJK
-22	-6	XXXXXXXX	XXXXXXXX	XXXXXXAB	CDEFGHIJ
-23	-7	XXXXXXXX	XXXXXXXX	XXXXXXXA	BCDEFGHI
-24	-8	XXXXXXXX	XXXXXXXX	XXXXXXXX	ABCDEFGH
-25	-9	XXXXXXXX	XXXXXXXX	XXXXXXXX	XABCDEFG
-26	-10	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXABCDEF
-27	-11	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXABCDE
-28	-12	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXABCD
-29	-13	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXABC
-30	-14	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXAB
-31	-15	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXA
-32 to -128	-16 to -128	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX

LEGEND:
ABCDEFGHIJKLMNPR represents the 16-bit input pattern
X stands for the extension bit

Obr. 3 Možné výstupy posúvača

Činnosť celého podprogramu BIKVAD bude podrobne vysvetlená v rámci cvičenia, doteraz prebrané informácie o procesore ADSP218x (predchádzajúce prednášky a cvičenia) sú však dostatočné na jeho kompletnú analýzu.

1.2.2 PROGRAM IIR_TEST.ASM

Program **iir_test.asm** využíva predchádzajúci podprogram bikvad na realizáciu celého IIR filtra s konkrétnymi koeficientmi a umožňuje overiť činnosť IIR filtra pre konkrétne vstupné hodnoty.

Implementovaný IIR filter využíva koeficienty navrhnuté pomocou toolboxu **Filter Design & Analysis Tools** programu Matlab. Pomocou tohto programu je možné **navrhnúť** a **optimalizovať** koeficienty filtra pre rôzne štruktúry a typy aritmetiky

použité pri realizácií IIR filtra. Po konverzii koeficientov IIR filtra pre formát 1.15 pre priamu formu I program poskytuje koeficienty[2] vyjadrené v zlomkovom formáte:

```
Quantized Direct form I filter
------- Section 1 -------
Numerator
    QuantizedCoefficients{1}{1}    ReferenceCoefficients{1}{1}
 (1)        0.493560791015625  0.493571239820478850
 (2)       -0.926971435546875 -0.926968313906811560
 (3)        0.493560791015625  0.493571239820473460
Denominator
    QuantizedCoefficients{1}{2}    ReferenceCoefficients{1}{2}
 (1)        0.500000000000000  0.500000000000000000
 (2)       -0.837280273437500 -0.837268927234705250
 (3)        0.475219726562500  0.475230500598364140
------- Section 2 -------
Numerator
    QuantizedCoefficients{2}{1}    ReferenceCoefficients{2}{1}
 (1)        0.501647949218750  0.501638082829171330
 (2)       -0.135986328125000 -0.135986116086180940
 (3)        0.501647949218750  0.501638082829174990
Denominator
    QuantizedCoefficients{2}{2}    ReferenceCoefficients{2}{2}
 (1)        0.500000000000000  0.500000000000000000
 (2)       -0.500427246093750 -0.500439834250219430
 (3)        0.459442138671875  0.459451804001463460
------- Section 3 -------
Numerator
    QuantizedCoefficients{3}{1}    ReferenceCoefficients{3}{1}
 (1)        0.529113769531250  0.529107525230331440
0 (2)       0.000000000000000 -0.000000000000003290
 (3)       -0.529113769531250 -0.529107525230337990
Denominator
    QuantizedCoefficients{3}{2}    ReferenceCoefficients{3}{2}
 (1)        0.500000000000000  0.500000000000000000
 (2)       -0.673706054687500 -0.673717728365859970
 (3)        0.420318603515625  0.420315531927695880

  FilterStructure = df1
     ScaleValues = [0.25  0.25  0.25  1]
 NumberOfSections = 3
 StatesPerSection = [4  4  4]
```

Keďže použitá implementácia (bikvad.asm) neumožňuje zmenšenie signálov pred vstupom do každej sekcie, sú **ScaleValues = [0.25 0.25 0.25]** zahrnuté do príslušných konštánt $B0, B1, B2$, t.j. ich hodnoty sú vynásobené hodnotami $0,25$. Po vynásobení

[2] V čitateľoch sú uvádzané koeficienty $B0, B1, B2$ pre jednotlivé sekcie, v menovateľoch sú uvádzané hodnoty $A0, A1, A2$, pričom platí $SCALE = 1/A0$. Pre všetky sekcie je použitá rovnaká hodnota $SCALE = 2^1$ (t.j. $A0 = 0,5$ a sú označené zvýrazneným písmom). Samotné vyjadrenie koeficientov vo formáte 1.15 však ešte nezaručuje správnu implementáciu. Problém môže spôsobovať pretečenie medzivýslekov pri výpočtoch (1.3) mimo interval $<-1,1)$ a teda pretečenie medzivýsledkov pri výpočte jednotlivých bikvadov. Tieto pretečenia je možné eliminovať zmenšením amplitúdy vstupného signálu (ktorý je v našom prípade tiež vo formáte 1.15, t.j. z intervalu $<-1,1)$) pred vstupom do jednotlivých bikvadov. Programy, ktoré navrhujú (a sú pre tento účel **optimalizované** – ako napr. použitý toolbox) koeficienty IIR pre DSP s aritmetikou v pevnej rádovej čiarke určujú ďalšie mierkové konštanty, ktoré zmenšujú vstupný signál pred vstupom do jednotlivých bikvadov. Tieto mierkové konštanty boli určené s využitím tzv. L2 normy a majú hodnotu $0,25$ pre všetky vstupy bikvadov. Tieto hodnoty sú opäť zapísané zvýrazneným písmom. Posledná hodnota 1 určuje, že výstup z posledného bikvadu nie je potrebné meniť. Algoritmy a metódy návrhu (napr. význam L2 normy, ...) týchto mierkových konštánt však presahuje rámec tohto predmetu a je preberaný v iných špecializovaných predmetoch (napr. v predmete Digital Filters). Po zahrnutí týchto opatrení umožní implementovaný IIR filter spracovanie vstupných signálov z intervalu $<-1,1)$ s minimálnou pravdepodobnosťou pretečenia medzivýsledkov v jednotlivých sekciách bikvadov. Na druhej strane uvedená zmena mierky sa snaží čo najlepšie využiť dostupný 16-bitový dynamický rozsah procesora ADSP218x.

hodnotou 2^{15} a zmenení znamienok[3] pri hodnotách $A1, A2$ dostávame obsah súboru **coef.dat**, ktorý obsahuje koeficienty vo formáte 16.0 zoradené v poradí, ktoré vyžaduje podprogram bikvad.asm:

```
4043,          <- B2        1. sekcia
-7594,         <- B1
4043,          <- B0
-15572,        <- A2
27436,         <- A1
4110,          <- B2        2. sekcia
-1114,         <- B1
4110,          <- B0
-15055         <- A2
16398,         <- A1
-4335,         <- B2        3. sekcia
0,             <- B1
4335,          <- B0
-13773,        <- A2
22076,         <- A1
```

Hodnoty v súbore **scal.dat** vyjadrujú hodnoty $s = 1,1,1$ pre všetky bikvady, t.j. určujú hodnotu $SCALE = 2^s = 2$. Program **iir_test.asm** používa na načítanie vstupných a vysielanie výstupných hodnôt sériový port SPORT1 podobne ako program fir_test.asm [4] a vyzerá takto:

```
/*****************************************************************************************************************
Nazov suboru:          IIR_test.asm

Datum modifikacie:     17-03-2002 MD

Opis:          Demonstruje inicializaciu a cinnost IIR filtra implementovaneho v procesore
               ADSP2181. Priklad vyuziva kaskadne zapojenie BIKVADOV s jednoduchou
               presnostou. Vtupne data a koeficienty su v zlomkovom formate 1.15, mierkove
               faktory vo formate 16.0.
               Vstupne vzorky su nacitavane z prijimacieho registra RX1 serioveho rozhrania
               SPORT1 a filtrovane udaje su zapisovane do vysielacieho registra TX1 portu SPORT1.
*****************************************************************************************************************/
#define    N              3          /* pocet bikvad sekcii */
#define    Ncoef          5*N        /* pocet koeficientov (5 koef/bikvad) */
#define    Nline          2*N+2      /* 4 prvky na bikvad, 2 zdielane s nasledujucim */
#define    Fvz            256        /* deliaci pomer 256 */

.EXTERN bikvad;

/* datova pamat - DM data */
.section/data data1;
.VAR    Delay_Line[2*N+2];                  /* oneskorovacia linka */
.VAR/circ    Scale_List[N] = "scal.dat";    /* mierkove (skalovacie) faktory pre bikvady */

/* programova pamat - PM data */
.section/pm data2;
.VAR/circ   COEFF[Ncoef] = "coef.dat";      /* skalovane koeficienty bikvadov v tvare
                                               B2, B1, B0, A2, A1, ... B2, B1, B0, A2, A1 */
```

[3] Toolbox počíta koeficienty pre diferenčnú rovnicu $y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) - a_1 y(n-1) - a_2 y(n-2)$, t.j. používa opačné znamienka pri koeficientoch a_1, a_2 než aké používa funkcia bikvad.asm. Je preto **vždy potrebné porovnať** diferenčnú rovnicu ktorá je implementovaná a rovnicu, ktorú používa návrhový softvér. Ak sa použijú nesprávne znamienka, zvyčajne dostávame IIR filter, ktorý je nestabilný a na výstupe generuje kvázichaotický priebeh.

```
.section/pm  interrupts;
                                        /* --tabulka interuptovych vektorov-- */
__reset: JUMP start; NOP; NOP; NOP;     /* resetovaci vektor */
      RTI; NOP; NOP; NOP;               /* IRQ2 */
      RTI; NOP; NOP; NOP;               /* IRQL1 */
      RTI; NOP; NOP; NOP;               /* IRQL2 */
      RTI; NOP; NOP; NOP;               /* SPORT0 vysielanie*/
      RTI; NOP; NOP; NOP;               /* SPORT0 prijem */
      RTI; NOP; NOP; NOP;               /* IRQE */
      RTI; NOP; NOP; NOP;               /* BDMA */
      RTI; NOP; NOP; NOP;               /* SPORT1 vysielanie */
      JUMP sample; NOP; NOP; NOP;       /* SPORT1 prijem */
      RTI; NOP; NOP; NOP;               /* casovac */
      RTI; NOP; NOP; NOP;               /* znizeny prikon (Power down) */

.section/pm  program;
                        /* -----inicializacia----- */
/***Inicializacia serioveho portu a zakladnych riadiacich registrov***/
/* aj ked nasledujuci kod este nie je urceny pre konkretne technicke
   prostriedky, jeho filozofia sa priblizuje praktickemu vyuzitiu.
   Vstupne vzorky sa nacitavaju z prijimacieho registra SPORT1 (RX1)
   a po filtracii, ktora je realizovana v prijimacom interrupte SPORT1,
   su filtrovane vzorky zapisovane do vysielacieho registra SPORT1 (TX1)
*/
start:
      AX0=0x0000;
      DM(0x3FFE)=AX0;                   /* vsetky DM pouzivaju 0 cakacich stavov */
      DM(0x3FFD)=AX0;                   /* casovac je nepouzity */
      DM(0x3FFC)=AX0;                   /* nulovanie registrov */
      DM(0x3FFB)=AX0;
      DM(0x3FFA)=AX0;                   /* viackanalove prijimanie */
      DM(0x3FF9)=AX0;                   /* zakazane */
      DM(0x3FF8)=AX0;                   /* viackanalove prijimanie */
      DM(0x3FF7)=AX0;                   /* zakazane */
      DM(0x3FF6)=AX0;                   /* riadenie SPORT0 nepouzite */
      DM(0x3FF5)=AX0;                   /* casovanie SPORT0 nepouzite */
      DM(0x3FF4)=AX0;                   /* casovanie SPORT0 nepouzite */
      DM(0x3FF3)=AX0;                   /* autobufer SPORT0 nepouzity */
/* konfiguracia SPORT1 */
      AX0=0x6B1F;                       /* interne seriove hodiny */
      DM(0x3FF2)=AX0;                   /* RFS reqd, normalny ramec,*/
                                        /* TFS reqd, normalny ramec,*/
                                        /* interne RFS, TFS, */
                                        /* bez kompresie, 16-bitove slova */
      AX0=0x0002;                       /* generuje 2.048 MHz SCLK1*/
      DM(0x3FF1)=AX0;                   /* z 12.288 MHz CLKIN */
      AX0=Fvz-1;                        /* deli SCLK1 256 pre 8 kHz */
      DM(0x3FF0)=AX0;                   /* vzorkovaciu frekvenciu */
      AX0=0x0000;
      DM(0x3FEF)=AX0;                   /* autobufer SPORT1 nepouzity */

      I0=Delay_Line; M0=1; L0=0;        /* incializacia smernika na oneskorovaciu linku */
      I1=Scale_List;                    /* incializacia smernika na skalovacie faktory */
      M1=-3;
      L1=LENGTH (Scale_List);           /* inicializacia modulo adresovania */
      I4=COEFF; M4=1;                   /* incializacia smernika na koeficienty */
      L4=length(COEFF);                 /* inicializacia modulo adresovania */
      M3=1-Nline;
      M2=1;
      CNTR=LENGTH(Delay_Line);          /* nulovanie oneskorovacej linky */
      DO zero UNTIL CE;
zero:  dm(I0,M2)=0;                     /* nulovanie oneskorovacej linky */
      M2=1;                             /* !!!viac ako jedna sekcia */
      ICNTL=0x07;                       /* povolenie (edge sensitive) IRQs*/
      IMASK=0x02;                       /* povolenie prijimacieho int. od SPORT1 */

      AX0=0x0C00;                       /* SPORT1 povoleny, cakacie stavy */
      DM(0x3FFF)=AX0;                   /* PM=0, boot cakacie stavy=0, */
                                        /* zavadzacia stranka (boot page) 0 */
      DIS M_MODE;                       /* zlomkovy mod MAC jednotky */

/*------ Cakanaie na vzorku ----------*/
receive:IDLE;                           /* cakanie na prerusenie od prijimaca */
      Jump receive;
```

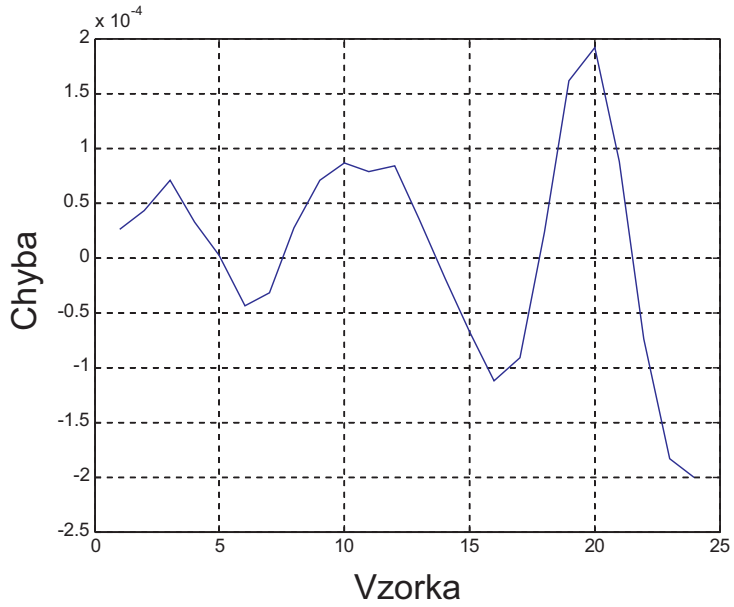
```
/*------ Spracovanie vzorky ------------*/
sample: SR1=RX1;                    /* zapis prijatej vzorky do SR1 */
        CNTR=N;                     /* inicializacia pocitadla bikvadov */
        I0=Delay_Line;
        call bikvad;                /* volanie podprogramu IIR filtra */

        TX1=SR1;                    /* vyselanie filtrovanej vzorky (v SR1) */
        RTI;                        /* navrat z prerusenia */
```

a je až na malé odlišnosti zhodný s programom iir_test.asm. Činnosť programu iir_test.asm bude podrobne analyzovaná počas cvičenia.

1.3 LADENIE IIR FILTRA V PROSTREDÍ VISUALDSP

Počas cvičenia bude vykonaná kompletná simulácia programu iir_test.asm so vstupnými vzorkami zo súboru **x.dat**, ktoré sú súčasťou projektu [3]. V projekte je aj program test.m v Matlabe, ktorý umožňuje porovnať výsledky zo simulácie ADSP2181 s referenčným výpočtom FIR filtra v Matlabe, ktorý umožňuje vypočítať a zobraziť chybu výpočtu IIR filtra pomocou ADSP218x, ktorá je zobrazená na obrázku 4. Z obrázku je zrejmé, že maximálna chyba je podstatne väčšia ako 1 LSB $\doteq 2^{-15} = 3.1*10^{-5}$, čo dokumentuje vplyv spätnej väzby v štruktúre IIR filtra.



Obr.4 Chyba výpočtu IIR filtra v procesore ADSP218x

1.4 ZÁVER

Implementovaný IIR filter vyžaduje na realizáciu IIR filtra rádu N $4N+4$ inštrukčných cyklov. Signálové procesory rodiny ADSP218x realizujú od 33 do 80 MIPS a tieto hodnoty určujú maximálne frekvencie vzorkovania F_{vz} resp. pri danej frekvencii vzorkovania maximálny rád IIR filtra N, ktorý je možné implementovať. Napr. pre IIR filter rádu $N=20$ (čo už je pomerne zložitý IIR filter) je možné s procesorom, ktorý realizuje 50 MIPS pracovať až do

$$F_{vz} \approx \frac{50\times10^6}{4N+4} = \frac{50\times10^6}{84} = 595\left[kHz\right] \tag{1.6}$$

čo dokumentuje efektívnosť architektúry ADSP218x pri realizácii aj tohto základného algoritmu ČSS.

LITERATÚRA

[1] Číslicové filtre a FFT (opakovanie). (dostupné v elektronickej forme **\SPvT\Cvicenia\spvt_2cv.pdf**)

[2] Mar, A.: *Digital Signal Processing Applications using the ADSP-2100 Family, Volume 1*. Prentice Hall, Englewood Cliffs, 1992. (dostupné aj v elektronickej forme **\SPvT\Knihy\DSP_Books\Using_ADSP-2100\…**)

[3] (dostupné v elektronickej forme **\SPvT\Cvicenia\IIR_test.zip**)

[4] Implementácia FIR filtra pomocou procesora ADSP218x. (dostupné v elektronickej forme **\SPvT\Cvicenia\spvt_3cv.pdf**)