```
/*************************************
Testprogramm für ein Unterprogramm
IIR-Filter mit Parallelarbeit

Dem Unterprogramm iir werden übergeben:

B0 - Adresse für den x-Vektor
M0 - Modifier
L0 - Länge des x-Vektors

B1 - Adresse für den y-Vektor
M1 - Modifier
L1 - Länge des y-Vektors

B8 - Adresse für den Koeffizientenvektor bk
M8 - Modifier
L8 - Länge des Koeffizientenvektors bk

B9 - Adresse für den Koeffizientenvektor ak
M9 - Modifier
L9 - Länge des Koeffizientenvektors ak

F0 der neue x-Wert

*************************************/
#define ordnung 4
.SECTION /PM seg pmda;
.VAR    akoeforg[]="akorg.dat";
.VAR    bkoeforg[]="bkorg.dat";
.VAR    bkoef[ordnung];
.VAR    akoef[ordnung];

.SECTION /DM seg dmda;
.VAR    xvek[ordnung];
.VAR    yvek[ordnung];

.VAR    eingangssignal[]="xein.dat";
.VAR    ausgangssignal[210];

.SECTION /PM seg_rth;

            NOP;
            NOP;
            NOP;
            NOP;
            NOP;
            JUMP start;

.SECTION /PM seg_pmco;

start:      B0=xvek;    /* x-Vektor adressieren */
            M0=1;
            L0=@bkoef;
            B1=yvek;    /* y-Vektor adressieren */
            M1=1;
            L1=@akoef;
            B8=bkoef;   /* bk Koeffizientenvektor adressieren */
            M8=1;
            L8=@bkoef;
            B9=akoef;   /* ak Koeffizientenvektor adressieren */
            M9=1;
            L9=@akoef;
            CALL umspeich;
            B3=eingangssignal;  /* Eingangsvektor adressieren */
            M3=1;
            L3=@eingangssignal;
            B4=ausgangssignal;  /* Ausgangsvektor adressieren */
            M4=1;
            L4=@ausgangssignal;
/* y-Vektor und x-Vektor löschen */
            lcntr = L0, do loeschen UNTIL LCE;
            DM(I1,M1)=0.0;
loeschen:   DM(I0,M0)=0.0;
```

```
            F1=0.0;
/* Augangswert durch Filter berechnen */
            lcntr = L3, do filtern UNTIL LCE;
            F0=DM(I3,M3);   /*neuen Eingabewert lesen */
            R15=ordnung-1;  /* Filterordnung übergeben */
            CALL iir;       /* IIR Unterprogramm */
            nop;
            nop;
    /* neuen y - Wert in Ausgabe speichern */
filtern:    DM(I4,M4)=F1;   /* neuen Ausgabewert speichern */
            nop;
            nop;

ende:       idle;
/*  Unterprogramm IIR-Filter
    belegte Register:
            F12 Ergebnis der Teilprodukte bk*xk
            F0  Wert aus Vektor xk
                Rückgabe neuer y - Wert
            F4  Wert aus Vektor bk
            F8  Wert für Summe der Teilprodukte bk*xk
            F13 Ergebnis der Teilprodukte ak*yk
            F1  Wert aus Vektor yk
            F5  Wert aus Vektor ak
            F9  Wert für Summe der Teilprodukte ak*yk
*/
/* IIR Filter   */

        /*  neuen Wert in x-Vektor schreiben
            Arbeitsregister loeschen                */
iir:
            R12=R12 xor R12, DM(I0,M0)=F0;
        /*  1. Wert aus dem x-Vektor lesen
            1. Wert aus dem bk Vektor lesen         */
            R8=R8 xor R8, F0=DM(I0,M0), F4=PM(I8,M8);
        /*  1. Wert aus dem y-Vektor lesen
            1. Wert aus dem ak Vektor lesen         */
            R13=R13 xor R13, F1=DM(I1,M1), F5=PM(I9,M9);
            R9=R9 xor R9;
        /*  Filterschleife                          */
            lcntr = R15, do rechnung until LCE;
        /*  Multiplikation x*bk
            Lesen nächsten Wert
            aus x-Vektor und bk-Vektopr lesen       */
            F12=F0*F4, F0=DM(I0,M0), F4=PM(I8,M8);
        /*  Multiplikation y*ak
            Lesen nächsten Wert
            aus y-Vektor und ak-Vektopr lesen
            Aufaddieren des neuen Produktes x*bk    */
            F13=F1*F5, F8=F8+F12, F1=DM(I1,M1),F5=PM(I9,M9);
        /*  Aufaddieren des neuen Produktes a*ak    */
rechnung:   F9=F9+F13;
        /*  letzte Multiplikation x*bk              */
            F12=F0*F4;
        /*  letzte Multiplikation y*ak
            Aufaddieren des letzten Produktes x*bk  */
            F13=F1*F5, F8=F8+F12;
        /*  Aufaddieren des letzten Produktes y*ak  */
            F9=F13+F9;
        /*  Gesamtsumme                             */
            F1=F8+F9;
            DM(I1,M1)=F1; /* neuen y Wert in y-Vektor speichern */
            rts;

/* Umspeichern der Koeffizienten aus MATLAB-Files */
umspeich:   B10=bkoeforg;   /* bk Koeffizientenvektor adressieren */
            M10=-1;
            L10=@bkoeforg;
            F0=PM(I10,M10);
            lcntr = L8, do umordbk until LCE;
            F0=PM(I10,M10);
umordbk:    PM(I8,M8)=F0;
            B11=akoeforg;   /* bk Koeffizientenvektor adressieren */
```

```
            M11=-1;
            L11=@akoeforg;
            PM(I9,M9)=0.0;
            R2=@akoeforg-1;
            F1=-1.0;
            F0=PM(I11,M11);
            lcntr = R2, do umordak until LCE;
            F0=PM(I11,M11);
            F0=F0*F1;
umordak:    PM(i9,M9)=F0;
            RTS;
```