APPENDIX B

Real-Valued Input FFT

B.1 Faster real FFT for the DSP96002

```
   page 132,60,1,1
   opt mex
;****************************************
;Motorola Austin DSP Operation  20 August 1992
;****************************************
;Test program for DSP96002 rfft96.asm
;*************************************************************************
;       1024 real-valued inputs
;   Maximum sample rate:  0.58 ms at 40.0 MHz
;   Memory Size:   Prog:141 + 32 words ;
;                  Data:2*1024 words(idata+odata) + 256 words (twiddle factor)
;   Number of clock cycles: 23200 (11600 instruction cycles)
;   Clock Frequency: 40.0MHz
;   Instruction cycle time: 50.ns
;*************************************************************************
;
;  Real-Valued Input Radix 2 Cooley-Tukey Decimation in Time FFT
;
;
;      normally ordered input data
;      normally ordered output data
;

;*************************************************************************
; Equates Section
;*************************************************************************

   RESET     equ      $00000000          ; reset isr
   MAIN      equ      $00000100          ; main routine

   points    equ      512                ;points=real data number /2
   passes    equ      9                  ;log2(points)=passes
   idata     equ      $0
   odata     equ      $1000
   coef      equ      $800
```

Figure B-1 Faster real FFT for the DSP96002 (sheet 1 of 4)

```
    BCRA      equ      $FFFFFFFE          ; port a bus control reg
    BCRB      equ      $FFFFFFFD          ; port b bus control reg
    PSR       equ      $FFFFFFFC          ; port select reg

    include   'sincosf.asm'              ;using external cos and sin table,
                                         ;if use internal ROM, delete this line
    include   'gen96.asm'
    include   'cfft96.asm'
    include   'split96.asm'


;*************************************************************************
;*************************************************************************
    sincosf   points,coef   ;twiddle factor for split is a full cycle sin and cos
    gen96     points,idata

    org   p:MAIN
    movep #$0,x:BCRA             ; no wait states for portb P,X,Y,I/O
    movep #$0,x:BCRB             ; ...don't care about page fault
    movep #$00FF00FF,x:PSR       ; external X:memory on Port-B
                                 ;          Y:memory on Port-A
    bclr  #$3,omr                ; disable the internal data ROMs


    CFFT96  points,passes,idata,coef,odata
    SPLIT96 points,coef,odata
    nop
    nop
    jmp   *

    END


;
; Sine-Cosine Table Generator for rfft96.asm
;
; Last Update 5 August 92
;
sincosf  macro   points,coef
sincosf  ident   1,2
;
;       sincosf-   macro to generate sine and cosine coefficient
;                  lookup tables for Decimation in Time FFT
;                  twiddle factors.
;
;       points -   number of points (2 - 32768, power of 2)
;       coef   -   base address of sine/cosine table
;                  positive cosine value in X memory
;                  positive sine value in Y memory
;
;    8/12/92

pi        equ      3.141592654
freq      equ      2.0*pi/@cvf(points*2)

          org      x:coef-points/2
```

Figure B-1 Faster real FFT for the DSP96002 (sheet 2 of 4)

```
count    set      0
         dup      points/2
         dc       @cos(@cvf(count)*freq)
count    set      count+1
         endm

         org      y:coef-points/2
count    set      0
         dup      points/2
         dc       -@sin(@cvf(count)*freq)
count    set      count+1
         endm

freq1    equ      2.0*pi/@cvf(points)

       org   x:coef
count  set   0
       dup   points/2
       dc    -@cos(@cvf(count)*freq1)
count  set   count+1
       endm

       org   y:coef
count  set   0
       dup   points/2
       dc    -@sin(@cvf(count)*freq1)
count  set   count+1
       endm

       endm                                  ;end of sincosf  macro

;************************************************************************
;
; Split N/2 Complex FFT(Hn) for N real FFT(Fn)
;
SPLIT96 macro points,coef,odata
SPLIT96 ident 1,2
;
; Fi=0.5(Hi+Hn/2-i*)-0.5j(Hi-Hn/2-i*)W i=0,1,,,N-1
;
; points is real data /2
;
    move  #points-1,n0                     ;number of complex FFT input data
    move  #points/2-1,n4                   ;loop counter
    move  #odata,r0                        ;r0 ptr to A=Hi
    move  r0,r4                            ;r4 ptr to A'
    move  #-1,m6                           ;linear address
    move  m6,m5    move     m6,m4
    move  #coef-points/2+1,r6              ;twiddle factor start location
    lea   (r0)+n0,r1                       ;r1 ptr to B= Hn/2-i
    move  r1,r5                            ;r5 ptr to B'
    move  x:(r0)+,d0.s y:,d1.s             ;DC=Ar0+Ai0
    faddsub.s  d0,d1    x:(r0)+,d2.s y:,d3.s   ;d0=Niquest=Ar0-Ai0,
                                           ;d1=DC,d2=Ar,d3=Ai
    move  d1.s,x:(r4)+ d0.s,y:             ;save DC and Niq
```

Figure B-1 Faster real FFT for the DSP96002 (sheet 3 of 4)

```
    move  x:(r1)-,d7.s y:,d1.s                  ;d7=Br,d1=Bi
    faddsub.s d7,d2     x:(r6)+,d8.s y:,d9.s    ;d7=Br-Ar=-H2i,
                                                ;d2=Ar+Br=H1r,d8=Wr,d9=Wi<0
    fmpy  d9,d7,d0 faddsub.s d3,d1     d2.s,d5.s;d0=Wi*H2i,d3=Ai-Bi=H1i,
                                                ;d1=Ai+Bi=H2r,d5=H1r
    fmpy.sd8,d1,d1 d1.s,d6.s                    ;d1=Wr*H2r,d6=H2r
    move  #0.5,d4.s                             ;d4=0.5
;-------------------------------------------------------------------------
;  H1r=Ar+Br, H1i=Ai-Bi, H2r=Ai+Bi, H2i=Ar-Br
;  Ar'=(H1r+Wr*H2r-Wi*H2i)/2
;  Br'=(H1r-(Wr*H2r-Wi*H2i))/2
;  Ai'=(Wi*H2r-Wr*H2i+H1i)/2
;  Bi'=((Wi*H2r-Wr*H2i)-H1i)/2
;
;-------------------------------------------------------------------------

    do  n4,_end_split                           ;do points/2-1
    fmpy  d8,d7,d2  fsub.s   d0,d1     x:(r1),d7.s   ;d2=-Wr*H2i, d1=Wr*H2r-
                                                ;Wi*H2i, d7=nBr
    fmpy  d9,d6,d0 faddsub.s d5,d1     x:(r6)+,d8.s y:,d9.s
                                ;d0=Wi*H2r, d1=2*Ar',d5=2*Br',d8=nWr,d9=nWi
    fmpy  d4,d5,d2 fadd.s    d2,d0     x:(r0),d1.s  d1.s,d6.s
                                ;d2=Br',d0=Wi*H2r-Wr*H2i, d6=2*Ar',d1=nAr
    fmpy  d4,d6,d2 faddsub.s d0,d3     d2.s,x:(r5)   ;d2=Ar', d3=2*Ai',
                                                ;d0=2*Bi',save Br'
    fmpy.sd4,d3,d3 d2.s,x:(r4)   y:(r1)-,d2.s       ;d3=Ai',save Ar',d2=nBi
    fmpy  d4,d0,d0 faddsub.s     d7,d1  d3.s,d6.s    y:(r0)+,d3.s
;d0=Bi',d1=nAr+nBr,d7=nBr-nAr=nH2i,d3=nAi
    fmpy  d9,d7,d0 faddsub.s d3,d2 d1.s,d5.sd0.s,y:(r5)
;d0=nWi*nH2i,d2=nAi+nBi=nH2r,d3=nAi-nBi,save Bi'
    fmpy.s  d8,d2,d1    d2.s,d6.s d6.s,y:(r4)+  ;d1=nWr*nH2r,d6=nH2r,save Ai'
_end_split
    move      y:(r4),d0.s                       ;conjugate of last Ai element
    fneg.s    d0
    move      d0.s,y:(r4)

    endm                                        ;split
```

Figure B-1 Faster real FFT for the DSP96002 (sheet 4 of 4)

B.2 Real FFT for DSP56001/2

```
;
; This program originally available on the Motorola DSP bulletin board.
; It is provided under a DISCLAMER OF WARRANTY available from
; Motorola DSP Operation, 6501 Wm. Cannon Drive W., Austin, Tx., 78735.
;
; 1024-Point Real Input Non-In-Place FFT. (test program)
; 34886 clock cycles. Sampling period can be 0.87215ms @ 40 Mhz clock rate
; Use 292 program words,4*512 words for data and 2*(128+256) words for twiddle
; factor
;
; Store EVEN index input data to X memory and ODD index input data to Y.
; Assume scaling down at input, i.e. all input data are divided by 1024 before FFT.
; The outputs of this real input FFT are twice larger than true values. If the
; original FFT values are desired, scaling up factor should be 512.
;
; 'sincosr' generates twiddle factor for FFT.
; 'bitrevtwd56' sorts the twiddle factor in bit-reverse order.
; The generation and reordering of twiddle factors can be done off-line.
; 'gen56' generates input test signals, delete it if you provide input.
; 'CFFT56' does 512 points FFT.
; 'SPLIT56' extractes 512-point complex values for real input FFT.
; Only DC to Niquest frequency are calculated by this program.
; Input data always starts at location IDATA=0, a 512-complex buffer starts at any
; external memory location, ODATA, is required to hold 256-point output data
; groups.
;
; The output of the FFT replace the inputs started at IDATA.
; X:IDATA contains DC*2 and Y:IDATA contains Niquest*2.
;
;
RFFT56T ident  1,0
        page   132,60
        opt    nomd,nomex,loc,nocex,mu

        include    'sincosr'
        include    'bitrevtwd56'
        include    'gen56'
        include    'cfft56'
        include    'split56'
;
;
; Latest revision - Nov. 11 92

reset   equ        0
start   equ        $40
POINTS  equ        512
IDATA   equ        $00
ODATA   equ        $1000
COEF    equ        $800

        sincosr    POINTS,COEF
        gen56      POINTS,IDATA
```

Figure B-2 Real FFT for DSP56001/2 (sheet 1 of 5)

```
        opt         mex
        org         p:reset
        jmp         start

        org    p:start
        movep  #0,x:$fffe                       ;0 wait states
        bitrevtwd56     POINTS,COEF
        CFFT56          IDATA,COEF,POINTS,ODATA
        SPLIT56         IDATA,COEF,POINTS,ODATA

         end


;
; Sine-Cosine Table Generator for rfft56.asm
;
; Last Update 11/11/92
;
sincosr  macro   points,coef
sincosr  ident   1,2
;
;       sincosr-   macro to generate sine and cosine coefficient
;                  lookup tables for Decimation in Time real FFT
;                  twiddle factors. Only points/4 coefficients
;                  are generted. For real FFT another points/4
;                  coefficients with higher freq. are created.
;
;       points -   number of points (2 - 32768, power of 2)
;       coef   -   base address of sine/cosine table
;                  positive cosine value in X memory
;                  positive sine value in Y memory
;
;    8/12/92

pi      equ    3.141592654
freq    equ    2.0*pi/@cvf(points*2)

        org    x:coef-points/2
count   set    0
        dup    points/2
        dc     @cos(@cvf(count)*freq)
count   set    count+1
        endm

        org    y:coef-points/2
count   set    0
        dup    points/2
        dc     -@sin(@cvf(count)*freq)
count   set    count+1
        endm

freq1   equ    2.0*pi/@cvf(points)

        org    x:coef
count   set    0
```

Figure B-2 Real FFT for DSP56001/2 (sheet 2 of 5)

```
        dup    points/4
        dc     @cos(@cvf(count)*freq1)
count   set    count+1
        endm

        org    y:coef
count   set    0
        dup    points/4
        dc     @sin(@cvf(count)*freq1)
count   set    count+1
        endm

        endm                                        ;end of sincosr  macro


bitrevtwd56  macro   POINTS,COEF
bitrevtwd56  ident   1,2
;
;       bitrevtwd  -    macro to sort sine and cosine coefficient
;                       lookup tables in bit reverse order for 56156
;
;       POINTS     -    number of points (2 - 32768, power of 2)
;       COEF       -    base address of sine/cosine table
;                       negative cosine (Wr) and negative sine (Wi) in X memory
;
; Wei Chen
; July-28, 1992
;
        move   #COEF,r1               ;twiddle factor start address
        move   #0,m0                  ;bit reverse address
        move   #POINTS/8,n0           ;sincosr use N/4 points data,
                                      ;offset for bit rev. is N/8
        move   #POINTS/4-1,n2
        move   r1,r0                  ;r1 ptr to normal order data
        move   (r1)+                  ;no swap on 1st data
        move   (r0)+n0                ;r0 ptr to bitrev
        do     n2,_end_bit            ;does N/4-1 points swap
        move   r1,x0
        move   r0,b
        cmp    x0,b
        jgt    _swap
        move   (r1)+                  ;no swap but update points
        move   (r0)+n0
        jmp    _nothing
_swap
        move   r1,r5
        move   r0,r4
        move   x:(r1),x0 y:(r5),y0
        move   x:(r0),a  y:(r4),b
        move   x0,x:(r0)+n0 y0,y:(r4)
        move   a,x:(r1)+  b,y:(r5)
_nothing
        nop
_end_bit
        endm                          ;end of bitrevtwd macro
```

Figure B-2 Real FFT for DSP56001/2 (sheet 3 of 5)

```
;*************************************************************************
;
;
; Split N/2 Complex FFT(Hn) for N real FFT(Fn)
;
SPLIT56 macro IDATA,COEF,POINTS,ODATA
SPLIT56 ident    1,0
;
;
; Fi=0.5(Hi+Hn/2-i*)-0.5j(Hi-Hn/2-i*)W i=0,1,,,N-1
;
; Bit reverse input, Normal order output
; This macro amplifies coefficients of FFT by 2.
; If absolute values of spectrum are desired, then scaling up factor is 2^(N-1),
; assuming inputs are scaled by 2^N before complex FFT.
; POINTS is the number of real data /2
; COEF is twiddle factor location other than TF used in complex FFT (see sincosr)
;
;

    move      #POINTS-1,n0                ;number of complex FFT input data -1
    move      #POINTS/2-1,n2              ;loop counter
    move      #ODATA,r0                   ;r0 ptr to Ar=Hi
    move      #COEF-POINTS/2+1,r2         ;twiddle factor start location
    move      r2,r6                       ;r6 -> Wi
    lea       (r0)+n0,r5                  ;r5 ptr to Br & Bi
    move      #IDATA,r3                   ;r3 pointer for A'
    move      r3,r4
    move      n0,r1                       ;r1 ptr for B', r1=B
    move      #POINTS/2,n0
    move  n0,n5
    move  m5,m3                                ;m3 and m1 linear address
    move  m5,m1
    move  #0,m0                                ;bit reverse address
    move  m0,m5                                ;bit reverse address
    move           x:(r0),b                    ;b=Ar0
    move           x:(r5),x1     y:(r0),a      ;a=Ai0,x1=Br
    add   a,b      x:(r1)+,x0                  ;b=Ar0+Ai0=DC, for ptr reason inc r1
    subl  b,a      r3,r4                       ;r4 ptr to temp location
    asl   b                      y:(r1),a      ;a=something
    asl   a        b,x:(r3)+     y:(r5),b      ;a=Niquist=Ar0-Ai0, save DC,b=Bi
    move                         a,y:(r0)+n0   ;save Niq in y:ODATA temp,
    move                         y:(r0),y0     ;y0=Ai

    do    n2,_end_split
    add   y0,b     y0,a     a,y:(r1)-          ;b=Ai+Bi=H2r,a=Ai, save prev. Bi'
    subl    b,a    x:(r0),b b,y1               ;a=Ai-Bi=H1i, b=Ar, y1=H2r
    sub   x1,b     x:(r0)+n0,a a,y:(r4)        ;b=Ar-Br=H2i,a=Ar again,
                                               ;save H1i temp,r0->nA
    subl  b,a      x:(r2)+,x1    y:(r6)+,y0    ;a=Ar+Br=H1r,x1=Wr,y0=Wi
    mac   x1,y1,a  b,x0          a,y:(r5) ;a=H1r+Wr*H2r,x0=H2i,save H1r temp
    macr  y0,x0,a                y:(r5)-n5,b   ;a=H1r+Wr*H2r-Wi*H2i=Ar', b=H1r
    subl  a,b      a,x:(r3) y:(r4),a           ;b=H1r-(Wr*H2r-Wi*H2i)=Br',
                                               ;a=H1i,save Ar'
    mac   -x1,x0,a b,x:(r1)      y:(r5),b      ;a=H1i-Wr*H2i,save Br',b=nBi
```

Figure B-2 Real FFT for DSP56001/2 (sheet 4 of 5)

```
    macr  y1,y0,a                 y:(r4),y0     ;a=Wi*H2r-Wr*H2i+H1i=Ai',y0=H1i
again
    sub   y0,a      x:(r5),x1     a,y:(r3)+     ;a=Wi*H2r-Wr*H2i, x1=nBr,save
Ai'
    sub   y0,a                    y:(r0),y0     ;a=Wi*H2r-Wr*H2i-H1i=Bi',y0=nAi
_end_split
    move  y0,a      a,y:(r1)                    ;save last Bi',conjugate last Ai
    neg   a         #ODATA,r5
    move            #IDATA,r0
    move                          a,y:(r4)
    move                          y:(r5),a
    move                          a,y:(r0)      ;move Niq. back

    endm                                        ;split56
```

Figure B-2 Real FFT for DSP56001/2 (sheet 5 of 5)