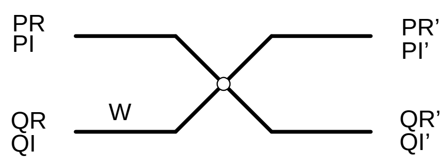
12.11 Fast Fourier Transforms

Fourier transforms are an important tool often used in digital signal processing systems. The transform converts information from the time domain to frequency. The inverse Fourier transform converts information back to time from frequency. Implementations of Fourier transforms that perform computations efficiently are known as fast Fourier transforms (FFT).

Certain ’C54x features are particularly well suited for FFTs. The high speed of the device (20-ns cycle time) makes the implementation of real-time algorithms easier. The powerful indexing scheme in indirect addressing facilitates the access of FFT butterfly legs with different spans. The repeat basic (RPTB) instruction is a scheme that reduces the loop overhead of algorithms that are heavily dependent on loops (such as the FFTs). This scheme has the efficiency of in-line coding but has the form of a loop. Instructions with a parallel store and the CPU architecture, which includes two accumulators, also minimizes the FFT-butterfly code. The FFT butterfly shown in Figure 12–12 can be performed in nine machine cycles. The ‘C54x code is shown in Example 12–25.

Figure 12–12. Radix-2 DIT FFT Butterfly



PR’ = {PR+(QR*WR+QI*WI)}/2
PI’ = {PI+(QI*WR–QR*WI)}/2

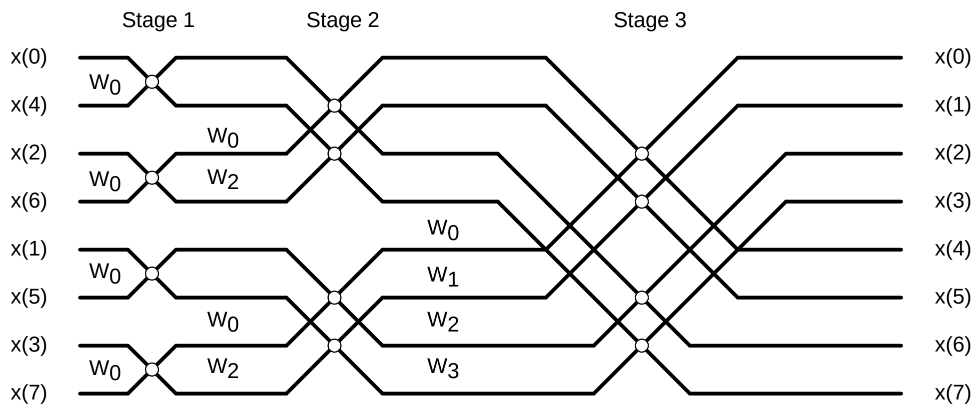
QR’ = {PR–(QR*WR+QI*WI)}/2
QI’ = {PI–(QI*WR–QR*WI)}/2

Notes: 1) WR = cos W

2) WI = sin W

Legend:
P Input
Q Input
P Output
Q’ Output
R Real Part
I Imaginary Part

Figure 12–13. An In-Place DIT FFT With In-Order Outputs and Bit-Reversed Inputs



As illustrated in Figure 12–13, the FFT input is in bit-reversed order when the output is in regular order. In Example 12–26, the input data is stored in data memory in bit-reversed order using the PORTR instruction. Bit-reversed addressing mode is used. It does not require extra cycles to arrange the data. Bit-reversed addressing mode is described in Section 7.6.

The macros used in Example 12–25 are:

- ❑ COMBO, which is used to calculate butterflies for stage 1 and stage 2 of the FFT signal flow graph.
- ❑ ZERO, which is used for a butterfly with a twiddle factor of W0 in a 16-point radix-2 FFT.
- ❑ PBY4, which is used for a butterfly with a twiddle factor of W2 in a 16-point radix-2 FFT.
- ❑ PBY2, which is used for a butterfly with a twiddle factor of W4 in a 16-point radix-2 FFT.
- ❑ PBY4, which is used for a butterfly with a twiddle factor of W6 in a 16-point radix-2 FFT.
- ❑ BUTTERFLY, which is used for a butterfly with a twiddle factor of W1, W3, W5, or W7 in a 16-point radix-2 FFT.

Example 12–26 shows the code to implement a 16-point radix-2 complex FFT.

Example 12–25. Macros for 16-Point DIT FFT

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; FILE : C54xrad2.mac –> macro file for radix 2 FFTs
; based on 320c5x
;
; COPYRIGHT TEXAS INSTRUMENTS INC. 1994
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; MACRO 'COMBO' FOR THE COMPLEX, RADIX-2 DIT FFT
;
; ORGANIZATION OF INPUT DATA MEMORY:
; R1,I1,R2,I2,R3,I3,R4,I4
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; THE MACRO 'COMBO' PERFORMS the FOLLOWING CALCULATIONS:
;
; R1  := [(R1+R2)+(R3+R4)]  INPUT         OUTPUT
; R2  := [(R1–R2)+(I3–I4)]  ------------  -------------
; R3  := [(R1+R2)–(R3+R4)]  AR0 =  7
; R4  := [(R1–R2)–(I3–I4)]  AR2 –> R1,I1  AR2 –> R1,I2
; I1  := [(I1+I2)+(I3+I4)]  AR3 –> R2,I2  AR3 –> R2,I2
; I2  := [(I1–I2)–(R3–R4)]  AR4 –> R3,I3  AR4 –> R3,I3
; I3  := [(I1+I2)–(I3+I4)]  AR5 –> R4,I4  AR5 –> R4,I
; I4  := [(I1–I2)+(R3–R4)]
;
; For a 16-point radix-2 complex FFT the macro COMBO54x
; has to be repeated N/4 times (4 times for a 16
; point FFT).
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;

COMBO    .MACRO NUM
         STM    #:NUM:–1,BRC

         RPTB   COMB_END–1

         ADD    *AR2,*AR3,A      ;R1+R2 –> A
         SUB    *AR2,*AR3,B      ;R1–R2 –> B
         ST     A,*AR2           ;(R1+R2)/2 –> R1
         ||LD   *AR5,A           ;R4 –> A
         ST     B,*AR3+          ;(R1–R2)/2 –> R2
         ||ADD  *AR4,B           ;R3+R4 –> B
         ST     B,*AR4           ;(R3+R4)/2 –> R3
         ||SUB  *AR4,B           ;R3–R4 –> B
         ST     B,*AR5+          ;(R3–R4)/2 –> R4
         ||LD   *AR4,A           ;(R3+R4)/2 –> A
         ADD    *AR2,16,A        ;(R1+R2+R3+R4)/2 –> A
         ST     A,*AR2+          ;(R1+R2+R3+R4)/4 –> R1
         ||SUB  *AR2+,B          ;{R1+R2–(R3+R4)}/2 –> A
```

Example 12–25. Macros for 16-Point DIT FFT (Continued)

```
          ST     A,*AR4+          ;{R1+R2–(R3+R4)}/4 –> R3
          ||LD   *AR3,B           ;I2 –> B
          ADD    *AR2,16,B,A      ;I1+I2 –> A
          ST     A,*AR2           ;(I1+I2)/2 –> I1
          ||SUB  *AR2,A           ;(I1–I2) –> A
          LD     *AR4,15,B        ;I3/2 –> B
          ADD    *AR5,15,B        ;(I3+I4)/2 –> B
          ST     A,*AR3–          ;(I1–I2)/2 –> I2
          ||ADD  *AR2,A           ;(I1+I2+I3+I4)/2 –> A
          ST     A,*AR2           ;(I1+I2+I3+I4)/4 –> I1
          ||SUB  *AR2,A           ;{(I1+I2–(I3+I4)}/2 –> A
          LD     *AR4,15,B        ;I3/2 –> B
          SUB    *AR5–,15,B       ;(I3–I4)/2 –> B
          ST     A,*AR4           ;{I1+I2–(I3+I4)}/4 –> I3
          ||ADD  *AR3,A           ;(R1–R2+I3–I4)/2 –> A
          ST     A,*AR3+          ;(R1–R2+I3–I4)/4 –> R2
          ||SUB  *AR3+,A          ;{(R1–R2)–(I3–I4)}/2 –> A
          LD     *AR5,16,B        ;(R3–R4)/2 –> B
          ST     A,*AR5+          ;{(R1–R2)–(I3–I4)}/4 –> R4
          ||ADD  *AR3,A           ;{(I1–I2)+(R3–R4)}/2 –>A
          ST     A,*AR5           ;{(I1–I2)+(R3–R4)}/4 –>I4
          ||SUB  *AR3,A           ;{(I1–I2)–(R3–R4)}/2 –>A
          STH    A,ASM,*AR3+0     ;{(I1–I2)–(R3–R4)}/4 –>I2
          MAR    *AR2+0
          MAR    *AR4+0
          MAR    *AR5+0

COMB_END
       .ENDM

;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;       MACRO  ’ ZERO ’    number of words : 6
;
;     AR3 –> PR,PI
;     AR4 –> QR,QI
;
;     CALCULATE Re[P+Q] AND Re[P–Q]
;     QR=(PR–QR)/2
;     PR=(PR+QR)/2
;     PI=(PI+QI)/2
;     QI=(PI–QI)/2
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

Example 12–25. Macros for 16-Point DIT FFT (Continued)

```
ZERO      .MACRO
          ADD    *AR3,*AR4,A  ;PR+QR –> A
          SUB    *AR3,*AR4+,B ;PR–QR –> B
          ST     A,*AR3+      ;(PR+QR)/2 –> PR
          ||LD   *AR4–,A      ;QI –> A
          ST     B,*AR4+      ;(PR–QR)/2 –> QR
          ||ADD  *AR3,B       ;PI+QI –> B
          ST     B,*AR3       ;(PI+QI)/2 –> PI
          ||SUB  *AR3+,B      ;PI–QI –> B
          STH    B,ASM,*AR4+  ;(PI–QI)/2 –> QI
          .ENDM

;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;      MACRO  ’PBY4’    number of words: 9
;
;      T=SIN(45)=COS(45)=W45
;
;      PR’= PR+(QI+QR)*W  (<– AR3)
;      QR’= PR–(QI+QR)*W  (<– AR4)
;      PI’= PI+(QI–QR)*W  (<– AR3)
;      QI’= PI–(QI–QR)*W  (<– AR4)
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

PBY4      .MACRO
          MPY    *AR4+,B      ;QR*W –> B
          MAC    *AR4,B       ;(QI+QR)*W –> B
          ADD    *AR3,16,B,A  ;PR+(QR+QI)*W –> A
          ST     A,*AR3       ;{PR+(QR+QI)*W}/2 –> PR
          ||SUB  *AR3+,A      ;PR–(QR+QI)*W –> A
          MPY    *AR4–,B      ;QI*W –> B
          MAS    *AR4,B       ;(QI–QR)*W –> B
          ST     A,*AR4+      ;{PR–(QR+QI)*W}/2 –> QR
          ||ADD  *AR3,A       ;PI+(QI–QR)*W –> A
          ST     A,*AR3       ;{PI+(QI–QR)*W}/2 –>PI
          ||SUB  *AR3+,A      ;PI–(QI–QR)*W –> A
          STH    A,*AR4+      ;{PI–(QI–QR)*W}/2 –> QI

          .ENDM

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;      MACRO ’PBY2’    number of words: 8
;
;      PR’ = (PR+QI)/2       PI’ = (PI–QR)/2
;      QR’ = (PR–QI)/2       QI’ = (PI+QR)/2
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

Example 12–25. Macros for 16-Point DIT FFT (Continued)

```
PBY2     .MACRO
         MAR    *AR4+
         LD     *AR4–,16,A   ;QI –> A
         ADD    *AR3,16,A,B  ;PR+QI –> B
         ST     B,*AR3       ;(PR+QI)/2 –> PR
         ||SUB  *AR3+,B      ;PR–QI –> B
         ST     B,*AR4       ;(PR–QI)/2 –> QR
         ||LD   *AR4+,A      ;QR –> A
         ADD    *AR3,16,A,B  ;PI+QR –> B
         ST     B,*AR4+      ;(PI+QR)/2 –> QI
         ||SUB  *AR3,A       ;PI–QR –> A
         STH    A,*AR3+      ;(PI–QR)/2 –> PI
         .ENDM


;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;      MACRO ’P3BY4’      number of words: 9
;
;      MACRO ENTRANCE: AR3–>PR,PI
;              AR4–>QR,QI
;              TREG=W=COS(45)=SIN(45)
;
;     PR’ = PR+(QI–QR)*W    (<–AR3)
;     QR’ = PR–(QI–QR)*W    (<–AR4)
;     PI’ = PI–(QI+QR)*W    (<–AR3)
;     QI’ = PI+(QI+QR)*W    (<–AR4)
;
;      MACRO EXIT: AR3–>PR,PI
;            AR4–>QR,QI
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

P3BY4    .MACRO
         MPY    *+AR4(1),B   ;QI*W –> B
         MAS    *+AR4(–1),B  ;(QI–QR)*W –> B
         ADD    *AR3,16,B,A  ;PR+(QI–QR)*W –> A
         ST     A,*AR3       ;{PR+(QI–QR)*W}/2 –> PR
         ||SUB  *AR3+,A      ;PR–(QI–QR)*W –> A
         MPY    *AR4+,B      ;QR*W –> B
         MAC    *AR4–,B      ;(QI+QR)*W –> B
         ST     A,*AR4+      ;{PR–(QR–QI)*W}/2 –> QR
         ||ADD  *AR3,A       ;PI+(QI+QR)*W –> A
         ST     A,*AR4+      ;{PI+(QI+QR)*W}/2 –>QI
         ||SUB  *AR3,A       ;PI–(QI+QR)*W –> A
         STH    A,*AR3+      ;{PI–(QI+QR)*W}/2 –> PI
         .ENDM
```

Example 12–25. Macros for 16-Point DIT FFT (Continued)

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;  MACRO:  'BUTTERFLY'    number of words: 9
;
;     MACRO ENTRANCE:  AR3->PR,PI
;             AR4 -> QR,QI
;             AR5 -> WI=SIN(W)
;             AR2 -> WR=COS(W)
;
;      PR' = {PR+(QR*WR+QI*WI)}/2    (<-AR3)
;      PI' = {PI+(QI*WR-QR*WI)}/2    (<-AR3)
;      QR' = {PR-(QR*WR+QI*WI)}/2    (<-AR4)
;      QI' = {PI-(QI*WR-QR*WI)}/2    (<-AR4)
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
BUTTERFLY .MACRO
         RPTB   BUT_END-1

         MPY    *AR4+,*AR2,A       ;WR*QR -> A
         MAC    *AR4,*AR5,A        ;A = (WR*QR+WI*QI)*2^16
         ADD    *AR3,16,A,B        ;{PR+(WR*QR+WI*QI)}
                                   ;*2^16 -> B
         ST     B,*AR3+            ;{PR+(QR*WR+WI*QI)}
                                   ;/2 -> PR
         ||SUB  *AR3+,B            ;{PR-(QR*WR+WI*QI)}
                                   ;*2^16 -> B
         MPY    *AR4-,*AR2+,A      ;A = WR*QI
         MASR   *AR4,*AR5+,A       ;A = (WR*QI-WI*QR)*2^16
         ST     B,*AR4+            ;{PR-(QR*WR+WI*QI)}
                                   ;/2 -> QR
         ||ADD  *AR3,B             ;{PI+(WR*QIWI*QR)}
                                   ;*2^16 -> B
         ST     B,*AR3+            ;{PI+(WR*QI-WI*QR)}
                                   ;/2 -> PI
         ||SUB  *AR3+,B            ;{PI-(WR*QI-WI*QR)}
                                   ;*2^16 -> A
         STH    B,ASM,*AR4+        ;{PI-(WR*QI-WI*QR)}
                                   ;/2 -> QI

BUT_END
         .ENDM
```

Example 12–25. Macros for 16-Point DIT FFT (Continued)

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;    MACRO 'STAGE 3'    number of words 37
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
STAGE3   .MACRO NUM
         STM    #:NUM:–1,BRC
         RPTB   STAGE3_END–1
         STM    PI_4,T
         ZERO
         PBY4
         PBY2
         P3BY4
         MAR    *+AR3(8)
         MAR    *+AR4(8)

STAGE3_END
         .ENDM

do_btfly .MACRO    NUM
         CALLD     MACROS
         STM       #:NUM:–1,BRC ;execute ZERO + NUM–1 times
                                ;BUTRFLY
         .ENDM

do_loops .MACRO    NUM

         STM       #:NUM:–1,BRC ;execute ZERO + NUM–1 times
                                ;BUTRFLY
         .ENDM

ex_btfly .MACRO
MACROS:  ZERO                   ;execute MACRO ZERO
         BUTRFLY
         RETD
         MAR       *+AR3(N/2)
         MAR       *+AR4(N/2)
            .ENDM
```

Example 12–26. 16-Point Radix-2 Complex FFT

```
          .title    "C54x 16-point FFT"
          .mmregs
          .global   RESET
          .global   PI_4
          .include  FFT.MAC
          .bss      SP_INIT,1
          .bss      INPUT,32
          .bss      WR,7
          .bss      WI,7
N         .set      16
PI_4      .set      23170
          .data
COSINE    .word     30274
          .word     23170
          .word     12540
          .word     00000
          .word     –12540
          .word     –23170
          .word     –30274
SINE      .word     12540
          .word     23170
          .word     30274
          .word     32768
          .word     30274
          .word     23170
          .word     12540

          .sect     "vector"
RESET     BD        INIT
          STM       #SP_INIT,SP
          .space    124*16

          .text
INIT:     LD     #1,DP
          LD     #–1,ASM
          SSBX   FRCT               ;set FRCT = 1
          STM    #N,AR0             ;half FFT size
          STM    INPUT,AR5          ;reverse input
          RPT    #N–1               ;read input data from
                                    ;ext I/O
          PORTR  1000h,*AR5+0B      ;into data memory

          STM#   WR,AR2             ;WR = cosine
          RPT    #6                 ;input cosine
          MVPD   COSINE,*AR2+
          STM    #WI,AR5            ;WI = sine
          RPT    #6                 ;input sine
          MVPD   SINE,*AR5+

          STM    #7,AR0             ;INDEX REGISTER
          STM    #INPUT,AR2         ;POINT TO R1
          STM    #INPUT+2,AR3       ;POINT TO R2
          STM    #INPUT+4,AR4       ;POINT TO R3
          STM    #INPUT+6,AR5       ;POINT TO R4
```

Example 12–26. 16-Point Radix-2 Complex FFT (Continued)

```
                    COMBO  4
                    STM    #3,AR0          ;INDEX REGISTER
                    STM    #INPUT,AR3      ;POINT TO R1
                    STM    #INPUT+8,AR4    ;POINT TO R2

                    STAGE3 2
                    STM    #INPUT,AR3      ;POINT TO PR
                    STM    #INPUT+16,AR4   ;POINT TO QI
                    STM    #WR,AR2         ;POINT TO COS
                    STM    #WI,AR5         ;POINT TO SIN

                    do_loops  7
                    ex_btfly

          NEXT      B        NEXT
```